

## HOVEDOPPGAVE

---

**Kandidatenes navn:** Håvard Alstad og Espen Gjøs

**Fag:** Datateknikk

**Oppgavens tittel (norsk):** Verktøystøtte for distribuert KJ

**Oppgavens tittel (engelsk):** Tool support for distributed KJ

**Oppgavens tekst:**

KJ er et hjelpemiddel for kreativ problemløsning. I oppgaven ønsker vi å studere hvordan KJ kan benyttes i distribuerte omgivelser ved hjelp av samarbeidsteknologi. Første steg på veien er å utvikle en applikasjon for KJ som kan benyttes i distribuerte omgivelser.

Deretter ønsker vi å sammenligne distribuert KJ med KJ gjennomført av personer lokalisert i samme rom. Sammenligningen skal skje ved hjelp av et eksperiment. På bakgrunn av eksperimentet håper vi å kunne trekke noen konklusjoner om denne formen for distribuert samarbeid. Vi ønsker å finne ut om distribuert KJ gir andre resultater enn KJ gjennomført med deltakerne i samme rom. Det vil også være interessant finne ut hva deltakerne synes om å samarbeide distribuert. Eksperimentet vil kunne gi oss generell kunnskap om distribuert arbeid og dets fordeler og ulemper.

---

Oppgaven gitt:	20. januar 2002
Besvarelsen leveres innen:	20. juni 2002
Besvarelsen levert:	14. juni 2002
Utført ved:	Institutt for datateknikk og informasjonsvitenskap
Veileder:	Tor Stålhaneå

Trondheim, 14. juni 2002

Tor Stålhane  
Faglærer



# Sammendrag

Temaet for oppgaven er *Distribuert idegenerering*. KJ er en strukturert metode for ide-myldring som benyttes flere steder i industrien. Med oppgaven ønsker vi å se på hvordan metoden kan benyttes i distribuerte omgivelser og hvordan dette påvirker prosessen og resultatene. For å gjennomføre KJ i distribuerte omgivelser er man avhengig av verktøystøtte. Vi har ikke funnet noe fullgodt verktøy på markedet, og har derfor utviklet prototypen KjApp for å kunne eksperimentere med distribuert KJ. Oppgaven består derfor av to deler; utvikling av verktøystøtte for KJ og eksperimentering med distribuert KJ.

Rapportens del I beskriver eksperimentet som er gjennomført. Eksperimentet sammenligner distribuert og lokal KJ. Lokal KJ gjennomføres ved hjelp av post-it lapper og tavle med deltakerne i samme rom, mens distribuert KJ gjennomføres ved hjelp av KjApp, NetMeeting og telefonkonferanse med deltakerne på forskjellige rom. Fem grupper á tre personer har gjennomført en distribuert og en lokal KJ sesjon. Hensikten med eksperimentet er å se om distribuert KJ kan fungere like godt som lokal. *KJ gjennomført med deltakerne i samme rom fungerer bedre enn KJ gjennomført i distribuerte omgivelser* er valgt som nullhypotese. Bakgrunnen for hypotesen er at det er lettere å samarbeide i en lokal prosess blant annet fordi øyekontakt og kroppsspråk bidrar til å synkronisere prosessen. Vi har imidlertid kommet frem til at nullhypotesen under visse forutsetninger kan forkastes til fordel for alternativ hypotesen som sier at distribuert og lokal KJ kan fungere like bra. Ved å analysere utvalgte kvalitetsattributter har vi kommet frem til at faktoren distribuert eller lokal gjennomføring er mindre betyngningsfull for resultatet enn først antatt. Distribuert KJ tar nesten dobbelt så lang tid som lokal gjennomføring, noe som i stor grad skyldes at KjApp ikke er en distribuert applikasjon og mangler et lokalt datalager hvor ideer kan forberedes. Dette påvirket også deltakernes opplevelse av flyt og *trøkk*. Lokal gjennomføring hadde signifikant høyere *trøkk* enn distribuert. Dersom man ser bort fra *trøkk* og tidsforbruk viser analysen at distribuert og lokal gjennomføring ikke har nevneverdige forskjeller. Resultatene viser derimot at deltakernes kjennskap til hverandre og engasjement i diskutert problem har større betydning for om en KJ sesjon er vellykket eller ikke. Det poenget er vist ved å gjennomføre statistiske analyser hvor deltakerne deles i to grupper; en gruppen med *profesjonelle* deltakere fra bedrifter og en annen gruppen bestående av studenter.

Rapportens del II beskriver utviklingsarbeidet med prototypen KjApp. Et viktig poeng for oss har vært å gjøre distribuert KJ mest mulig lik KJ gjennomført med deltakerne i samme rom. KjApp har derfor tilstrekkelig funksjonalitet for å støtte KJ metoden, men ingen unødvendig *nice to have* funksjonalitet. Brukergrensesnittet er enkelt og logisk oppbygd slik at deltakerne i eksperimentet lett kan ta i bruk prototypen. KjApp er utviklet som en frittstående applikasjon og distribueres ved hjelp av NetMeeting. Kom-

mentarer fra deltakerne i eksperimentet viser at KjApp har fungert godt og vi mener derfor at målsetningen med prototypen er innfridd. Dette er viktig for eksperimentet. Dersom KjApp ikke hadde fungert tilfredsstillende ville resultatene fra eksperimentet hatt redusert verdi. Erfaringer fra eksperimentet viser imidlertid at distribuering ved hjelp av NetMeeting ikke er optimalt. Utvikling av en distribuert versjon av KjApp er derfor en naturlig forlengelse av oppgaven og vil gjøre presentasjon av ideer raskere og føre til økt samtidighet.

# Forord

Oppgaven er utført ved institutt for datateknikk og informasjonsvitenskap (IDI) ved Norges teknisk -naturvitenskapelige universitet. Forfatterne er to femteårsstudenter som går linje for systemutvikling (SU) ved IDI.

Vi ønsker å takke alle som har deltatt i eksperimentet for å ta seg tid til å være med og for god innsats. Videre vil vi takke Svein Løvland for gode innspill i forbindelse med programmering i Java. Vi vil også takke Torgeir Dingsøyv ved SINTEF for gode kommentarer på rapporten og Monica Divitini ved IDI for innspill på resultatene fra eksperimentet. Sist men ikke minst ønsker vi å takke vår veileder Tor Stålhane for god oppfølging, fine kommentarer og for å ha fått oss tilbake på rett spor når dette har vært nødvendig.

Trondheim 14. juni 2002

---

Håvard Alstad og Espen Gjøs



# Innhold

<b>Sammendrag</b>	<b>iii</b>
<b>Forord</b>	<b>v</b>
<b>1 Innledning</b>	<b>1</b>
1.1 Bakgrunn . . . . .	1
1.2 Problemstilling . . . . .	1
1.3 Beskrivelse av arbeidet . . . . .	2
<b>2 Idegenerering</b>	<b>3</b>
2.1 Affinitetsdiagrammer . . . . .	3
2.2 Beskrivelse av KJ metoden . . . . .	4
2.3 Eksisterende verktøy for idegenerering . . . . .	5
<b>I Eksperimentet</b>	<b>7</b>
<b>3 Metode</b>	<b>9</b>
3.1 Empiriske metoder . . . . .	10
3.2 Hypotese . . . . .	11
3.2.1 Nullhypotese . . . . .	11
3.2.2 Alternativ hypotese . . . . .	12
3.2.3 Faktorer ved hypotesetesting . . . . .	12
3.3 Validitet . . . . .	12
3.3.1 Konklusjonsvaliditet . . . . .	13
3.3.2 Intern validitet . . . . .	13
3.3.3 Konstruksjonsvaliditet . . . . .	13
3.3.4 Ekstern validitet . . . . .	14
3.4 Goal/Question/Metric . . . . .	14
3.4.1 Spørsmål for fokus . . . . .	15
3.4.2 Spørsmål for omgivelser . . . . .	17
3.5 Analyseteknikker . . . . .	18
3.5.1 ANOVA . . . . .	18
3.5.2 Kruskal-Wallis . . . . .	19
3.6 Omgivelser for eksperimentet . . . . .	19
3.6.1 Deling av applikasjoner i NetMeeting . . . . .	20
3.6.2 Telefonkonferanse . . . . .	21
3.7 Oppsett av eksperimentet . . . . .	21

3.7.1	Kjøreplan . . . . .	22
3.7.2	Problemstilling . . . . .	23
3.7.3	Teknisk oppsett . . . . .	23
3.7.4	Datainnsamling . . . . .	23
<b>4</b>	<b>Resultater fra eksperimentet</b>	<b>25</b>
4.1	Gjennomføring . . . . .	25
4.1.1	Giant Leap Technologies . . . . .	25
4.1.2	SINTEF . . . . .	27
4.1.3	Studentgruppe 1 . . . . .	28
4.1.4	Diplomstudenter . . . . .	29
4.1.5	Studentgruppe 2 . . . . .	29
4.2	Observasjoner . . . . .	30
4.3	Beskrivende analyse . . . . .	32
4.3.1	Personlige opplysninger . . . . .	32
4.3.2	Distribuert og lokal sesjon . . . . .	32
4.3.3	Tidsbruk, antall ideer og grupper . . . . .	34
4.3.4	Oppfølgingsspørsmål . . . . .	37
<b>5</b>	<b>Statistisk analyse</b>	<b>43</b>
5.1	Hypotesetesting . . . . .	43
5.1.1	ANOVA av kvalitative resultater . . . . .	43
5.1.2	Analyse av kvantitative resultater . . . . .	45
5.1.3	ANOVA splittet på deltakernes profesjon . . . . .	46
5.2	Oppsummering av statistisk analyse . . . . .	48
<b>6</b>	<b>Videre arbeid</b>	<b>49</b>
6.1	Deltakersammensetning . . . . .	49
6.2	Kreativt utfordrende problemstillinger . . . . .	50
6.3	Varighet og fysisk kontakt . . . . .	50
<b>7</b>	<b>Diskusjon eksperimentet</b>	<b>51</b>
<b>8</b>	<b>Konklusjon eksperimentet</b>	<b>53</b>
<b>II</b>	<b>Prototypen</b>	<b>55</b>
<b>9</b>	<b>Utviklingsmetode og -miljø</b>	<b>57</b>
9.1	Utviklingsmetode . . . . .	58
9.2	Utviklingsmiljø . . . . .	58
9.2.1	Java . . . . .	58
9.2.2	Støtteverktøy . . . . .	58
9.2.3	Støttelitteratur . . . . .	59
<b>10</b>	<b>Kravspesifikasjon</b>	<b>61</b>
10.1	Funksjonelle krav . . . . .	61
10.1.1	Applikasjonsnivå . . . . .	62
10.1.2	Tavlenivå . . . . .	63
10.1.3	Gruppenivå . . . . .	64
10.1.4	Post-it lapp nivå . . . . .	65

10.2	Ikke-funksjonelle krav . . . . .	66
10.3	Skjermbilder . . . . .	68
<b>11</b>	<b>Design</b>	<b>73</b>
11.1	Model-View-Controller . . . . .	73
11.2	Composite . . . . .	75
11.3	Klassediagram for KjApp . . . . .	76
11.3.1	Model . . . . .	76
11.3.2	View . . . . .	78
11.3.3	Controller . . . . .	78
11.4	Sekvensdiagrammer . . . . .	81
11.4.1	Ny tavle . . . . .	81
11.4.2	Tegn Tavle . . . . .	82
11.4.3	Ny post-it . . . . .	83
11.4.4	Flytt tavleelement . . . . .	84
11.5	Algoritmer . . . . .	86
11.5.1	Algoritme for å finne tavle elementer . . . . .	86
11.5.2	Zooming av tavle . . . . .	88
11.5.3	Algoritme for å tegne tavle . . . . .	89
<b>12</b>	<b>Implementering</b>	<b>91</b>
12.1	Model . . . . .	91
12.2	View . . . . .	92
12.2.1	Komponenter benyttet i View . . . . .	93
12.2.2	Muselyttere i GrafiskeTavleElementer . . . . .	93
12.2.3	Knappelyttere i GrafiskeTavleElementer . . . . .	96
12.3	Controller . . . . .	98
<b>13</b>	<b>Test</b>	<b>99</b>
13.1	Systemtest . . . . .	99
13.2	Brukbarhetstest . . . . .	100
13.3	Distribuert test . . . . .	101
<b>14</b>	<b>Installasjon og brukerveiledning</b>	<b>103</b>
14.1	Installasjon av KjApp . . . . .	103
14.2	Brukerveiledning . . . . .	104
<b>15</b>	<b>Videre arbeid</b>	<b>113</b>
15.1	Forbedringer av KjApp . . . . .	113
15.2	Utvidelser av KjApp . . . . .	114
15.2.1	Distribuert versjon . . . . .	114
<b>16</b>	<b>Diskusjon KjApp</b>	<b>117</b>
<b>17</b>	<b>Konklusjon KjApp</b>	<b>119</b>
<b>A</b>	<b>Spørreskjema</b>	<b>121</b>
A.1	Spørreskjema til eksperimentet med KJ metoden . . . . .	121
A.2	Oppfølgingsspørsmål . . . . .	126

<b>B</b>	<b>Svar på spørsmål - rådata</b>	<b>131</b>
B.1	Svar på spørreundersøkelse . . . . .	131
B.1.1	Sesjonsopplysninger . . . . .	131
B.1.2	Distribuert sesjon . . . . .	132
B.1.3	Lokal sesjon . . . . .	132
B.2	Svar på oppfølgingsspørsmål . . . . .	133
B.2.1	Idegenerering . . . . .	133
B.2.2	Flyt og trøkk . . . . .	134
B.2.3	Fremtidig bruk og funksjonalitet . . . . .	135
<b>C</b>	<b>Grafisk fremstilling av resultater</b>	<b>137</b>
C.1	Spørreskjema fra eksperimentet . . . . .	137
C.2	Tidsbruk, antall post-it og grupper generert . . . . .	144
C.3	Fordeling av svar på oppfølgingsspørsmål . . . . .	146
<b>D</b>	<b>Analyse av resultater</b>	<b>157</b>
D.1	ANOVA . . . . .	157
D.1.1	One-way ANOVA: Distribuert mot lokal sesjon, alle data . . . . .	157
D.1.2	One-way ANOVA: GLT og SINTEF mot de andre gruppene . . . . .	160
D.2	Kruskal-Wallis . . . . .	165
<b>E</b>	<b>Test</b>	<b>167</b>
E.1	Systemtest . . . . .	167
E.2	Brukarhetstest . . . . .	173
<b>F</b>	<b>Program listing</b>	<b>175</b>
F.1	Package . . . . .	175
F.1.1	Brukergrensesnitt.java . . . . .	175
F.1.2	GrafiskeTavleElementer.java . . . . .	178
F.1.3	Gruppe.java . . . . .	184
F.1.4	IOtavle.java . . . . .	189
F.1.5	KjApp.java . . . . .	189
F.1.6	KjAppDialog.java . . . . .	190
F.1.7	Kontroller.java . . . . .	193
F.1.8	Menyer.java . . . . .	206
F.1.9	Posisjon.java . . . . .	213
F.1.10	PostIt.java . . . . .	213
F.1.11	Relasjon.java . . . . .	215
F.1.12	Tavle.java . . . . .	216
F.1.13	TavleElement.java . . . . .	220
F.1.14	TavleHandler.java . . . . .	222
F.1.15	XMLFilter.java . . . . .	225
F.1.16	TXTFilter.java . . . . .	225
<b>G</b>	<b>Installsjons CD</b>	<b>227</b>

# Figurer

2.1	En deltaker i en KJ sesjon presenterer en av sine ideer . . . . .	4
2.2	Skjerm bilde fra Mind Mapper . . . . .	5
2.3	Skjerm bilde fra Inspiration . . . . .	6
3.1	Dataflyt i forbindelse med deling av applikasjon i NetMeeting . . . . .	21
4.1	Lokal sesjon med GLT . . . . .	26
4.2	Lokal sesjon med SINTEF . . . . .	27
4.3	Distribuert sesjon med datateknikk studenter . . . . .	28
4.4	Distribuert sesjon med diplomstudenter . . . . .	29
4.5	Distribuert sesjon med datateknikk og industriell økonomi studenter . . . . .	30
4.6	Gjennomsnittsverdier for distribuerte og lokale sesjoner . . . . .	35
4.7	Post-it lapper generert i sesjonene . . . . .	36
4.8	Antall ideer generert i sesjonene . . . . .	37
10.1	Nivåinndeling i applikasjonen . . . . .	62
10.2	Use case diagram av funksjonelle krav på applikasjonsnivå . . . . .	62
10.3	Use case diagram av funksjonelle krav på tavlenivå . . . . .	63
10.4	Use case diagram av funksjonelle krav på gruppenivå . . . . .	64
10.5	Use case diagram av funksjonelle krav på post-it lapp nivå . . . . .	65
10.6	Layout i tavle for tavleelementer . . . . .	67
10.7	Filmeny . . . . .	68
10.8	Opprettelse av ny tavle . . . . .	69
10.9	Ny tavle . . . . .	69
10.10	Post-it lapper fra Hærens forsyningskommando . . . . .	70
10.11	Gruppering av post-it lapper . . . . .	70
10.12	Meny for redigering . . . . .	71
10.13	Meny for post-it . . . . .	71
10.14	Meny for gruppe . . . . .	72
10.15	Høyreklikkmeny for gruppe . . . . .	72
11.1	Konseptuelt diagram av MVC . . . . .	74
11.2	Designmønsteret <i>Composite</i> . . . . .	75
11.3	Eksempel på en komposisjon av bildeobjekter . . . . .	75
11.4	Overordnet klassediagram for kjApp . . . . .	76
11.5	Klassediagram, <i>Model</i> . . . . .	77
11.6	Klassediagram, <i>View</i> . . . . .	79
11.7	Klassediagram, <i>Controller</i> . . . . .	80
11.8	Sekvensdiagram for å lage en ny tavle . . . . .	81

11.9	Sekvensdiagram for å tegne en tavle . . . . .	82
11.10	Sekvensdiagram for å lage en ny post-it . . . . .	83
11.11	Sekvensdiagram for å flytte et TavleElement . . . . .	84
11.12	Layout i tavle for tavleelementer . . . . .	86
11.13	Trestruktur for tavleelementer . . . . .	86
11.14	Sekvensdiagram for å finne et tavle element . . . . .	88
11.15	Zoomfaktor . . . . .	88
12.1	Flytdiagram for mousePressed . . . . .	94
12.2	Flytdiagram for mouseDragged . . . . .	95
12.3	Flytdiagram for mouseReleased . . . . .	97
13.1	Knapp for å velge distribuert eller lokal modus for KjApp . . . . .	102
14.1	Snarvei til KjApp . . . . .	103
14.2	Opprette ny tavle . . . . .	104
14.3	Angi tema for ny tavle . . . . .	104
14.4	Opprett ny post-it . . . . .	105
14.5	Angi egenskaper for ny post-it . . . . .	105
14.6	Post it slik den blir opprettet på tavlen . . . . .	106
14.7	Post it etter størrelsen er tilpasset . . . . .	106
14.8	Opprett ny gruppe . . . . .	107
14.9	Ny gruppe er opprettet på tavlen <i>Gjennstående arbeid</i> . . . . .	107
14.10	Flytting av post-it . . . . .	108
14.11	Merking av et område . . . . .	109
14.12	Opprett gruppe etter område er valgt . . . . .	109
14.13	Kopier merkede elementer . . . . .	110
14.14	Lim inn merkede elementer . . . . .	110
14.15	Opprett ny relasjon . . . . .	111
14.16	Sletting av relasjon . . . . .	111
14.17	Lagring av tavle . . . . .	112
14.18	Angi navn og lagringsformat for tavle . . . . .	112
15.1	Vanlig markør . . . . .	114
15.2	Markør for relasjon . . . . .	114
15.3	Distribuert versjon av KjApp . . . . .	116
C.1	Fortrolighet med deltakerne i gruppen . . . . .	137
C.2	Fordeling av svar på spørsmålet <i>Var det morsomt</i> . . . . .	138
C.3	Fordeling av svar på spørsmålet <i>Følte du at du bidro</i> . . . . .	138
C.4	Fordeling av svar på spørsmålet <i>Hvordan fungerte gruppen</i> . . . . .	139
C.5	Fordeling av svar på spørsmålet <i>Hadde sesjonen trøkk</i> . . . . .	139
C.6	Fordeling av svar på spørsmålet <i>Kan du benytte resultatet</i> . . . . .	140
C.7	Fordeling av svar på spørsmålet <i>Kom det frem nye momenter</i> . . . . .	140
C.8	Fordeling av svar på spørsmålet <i>Kunne du kommet frem til resultatet på egenhånd</i> . . . . .	141
C.9	Fordeling av svar på spørsmålet <i>Foretrekker du å være anonym</i> . . . . .	141
C.10	Fordeling av svar på spørsmålet <i>Hvordan fungerte KjApp og NetMeeting</i> . . . . .	142
C.11	Gjennomsnittsverdier for spørsmålene fra distribuerte og lokale sesjoner . . . . .	143
C.12	Tidsbruk for alle sesjoner . . . . .	144

C.13 Post-it lapper generert for alle sesjoner . . . . .	144
C.14 Antall grupper generert for alle sesjoner . . . . .	145
C.15 Fordeling av svar på spørsmålet <i>Vurderte du ideene på nytt under distribuert sesjon</i> . . . . .	146
C.16 Fordeling av svar på spørsmålet <i>Forkastet du noen ideer under distribuert sesjon</i> . . . . .	146
C.17 Fordeling av svar på spørsmålet <i>Var du mer opptatt av at lappene under distribuert sesjon skulle være selvforklarende</i> . . . . .	147
C.18 Fordeling av svar på spørsmålet <i>I hvilken sesjon benyttet du flest forkortelser</i> . . . . .	147
C.19 Fordeling av svar på spørsmålet <i>Hvilken sesjon genererte de beste ideer</i>	148
C.20 Fordeling av svar på spørsmålet <i>Hvilken sesjon ga best oversikt</i> . . . .	148
C.21 Fordeling av svar på spørsmålet <i>Hvordan ble resultatet påvirket av varigheten på distribuert sesjon</i> . . . . .	149
C.22 Fordeling av svar på spørsmålet <i>Hvordan ble resultatet påvirket av at distribuert KJ var en ny erfaring for alle</i> . . . . .	149
C.23 Fordeling av svar på spørsmålet om <i>Mer erfaring med distribuert KJ og KjApp</i> . . . . .	150
C.24 Fordeling av svar på spørsmålet <i>Var telefonkonferansen et hinder</i> . . .	150
C.25 Fordeling av svar på spørsmålet <i>Var KjApp et hinder</i> . . . . .	151
C.26 Fordeling av svar på spørsmålet <i>Var NetMeeting et hinder</i> . . . . .	151
C.27 Fordeling av svar på spørsmålet <i>Hvor viktig mener du at stemmekontakt er i en slik sesjon</i> . . . . .	152
C.28 Fordeling av svar på spørsmålet om <i>Fortrolighet</i> . . . . .	152
C.29 Fordeling av svar på spørsmålet om <i>Sleivkommentarer</i> . . . . .	153
C.30 Fordeling av svar på spørsmålet om <i>Tidsstyrt sesjon</i> . . . . .	153
C.31 Fordeling av svar på spørsmålet <i>Vite hvem som har satt opp lapper</i> . .	154
C.32 Fordeling av svar på spørsmålet om <i>Lokalt datalager i KjApp</i> . . . . .	154
C.33 Fordeling av svar på spørsmålet om <i>Teknisk diskusjon Trondheim-Paris-London</i> . . . . .	155



# Tabeller

3.1	Oversikt over skala typer . . . . .	19
3.2	Oppsett av eksperimentet . . . . .	22
3.3	Tidsplan for gjennomføring av eksperimentet . . . . .	22
5.1	ANOVA av svar på spørsmål om deltakers opplevelse av sesjonen . . .	44
5.2	ANOVA av svar på spørsmål om deltakers tilfredshet med resultatene	45
5.3	Kvantitative resultater fra eksperimentet . . . . .	46
5.4	ANOVA på deltakernes opplevelse av sesjonen, splittet på GLT og SIN-TEF mot <i>studentgrupper</i> . . . . .	47
5.5	ANOVA på deltakernes tilfredshet med resultatene, splittet på GLT og SINTEF mot <i>studentgrupper</i> . . . . .	47
12.1	Klasser i Model . . . . .	91
12.2	Klasser i View . . . . .	92
12.3	Klasser i Controller . . . . .	98
E.1	Oversikt over resultater fra systemtest . . . . .	167
E.2	Oversikt over resultater fra systemtest . . . . .	168
E.3	Oversikt over resultater fra systemtest på gruppenivå . . . . .	170
E.4	Oversikt over resultater fra systemtest . . . . .	172



# Kapittel 1

## Innledning

### Innhold

---

1.1	Bakgrunn . . . . .	1
1.2	Problemstilling . . . . .	1
1.3	Beskrivelse av arbeidet . . . . .	2

---

### 1.1 Bakgrunn

Verdensøkonomien er for tiden inne i en tøff periode og næringslivet i Norge merker presset. Oppsigelser og konkurser er noen av resultatene. Presset i verdensøkonomien fører til at mange bedrifter strammer inn på reisebudsjetter og øker kravene til produktivitet. Katastrofen i USA 11. september 2001 har i tillegg ført til økt frykt for å reise, og da spesielt med fly. Dette er momenter som tvinger næringslivet til å tenke nytt og til å ta i bruk nye og billigere løsninger for sin virksomhet. Ved å la de ansatte arbeide distribuert, blant annet ved hjelp av hjemmekontorer og videokonferanseutstyr, kan bedrifter redusere sine reisebudsjetter.

I dagens bedrifter ligger store deler av verdiene i de ansattes kunnskaper. For å forvalte disse verdiene er det viktig å kunne hente ut og spre kunnskapen i bedriften. Idemyldring ved hjelp av KJ metoden er et hjelpemiddel som kan benyttes. KJ metoden er en strukturert metode for å hente ut og organisere informasjon om et tema. Metoden har fått sitt navn etter japaneren Jiro Kawakita. Metoden fungerer som et friskt innslag i de ansattes hverdag og stimulerer til økt innsats og kreativitet da den er forskjellig fra tradisjonell møtevirksomhet.

Distribuert arbeid har både fordeler og ulemper. I denne oppgaven ønsker vi å studere hvilken effekt distribuert arbeid har på ideskaping og problemløsning.

### 1.2 Problemstilling

I oppgaven ønsker vi å studere hvordan KJ kan benyttes i distribuerte omgivelser ved hjelp av samarbeidsteknologi. Første steg på veien er å utvikle en prototyp for KJ som

kan benyttes i distribuerte omgivelser.

Deretter ønsker vi å sammenligne distribuert KJ med KJ gjennomført av personer lokalisert i samme rom. Sammenligningen skal skje ved hjelp av et eksperiment. På bakgrunn av eksperimentet håper vi å kunne trekke noen konklusjoner om denne formen for distribuert samarbeid. Vi ønsker å finne ut om distribuert KJ gir andre resultater enn KJ gjennomført med deltakerne i samme rom. Det er også interessant finne ut hva deltakerne synes om å samarbeide distribuert. Vår antagelse er at lokal idegenerering fungerer bedre enn distribuert. Dette fordi deltakerne kommuniserer både muntlig og ved hjelp av kroppsspråk og synkronisering av idegenereringsprosess blir dermed enklere. Men er det virkelig slik? Og hvilke faktorer er viktige for at idegenereringen skal føre til et godt resultat?

### 1.3 Beskrivelse av arbeidet

Arbeidet består av utvikling av verktøystøtte for KJ metoden og eksperimentering med distribuert KJ.

#### Utvikling av verktøystøtte for KJ metoden

Utviklingen gjøres ved hjelp av programmeringsspråket Java. Java er valgt fordi begge studentene kjenner språket fra tidligere prosjekter. Utviklingsarbeidet skal resultere i en prototyp som støtter følgende elementer i KJ metoden:

- Generere og ta vare på ideer
- Vise genererte ideer
- Gruppere ideer i områder med beskrivende navn
- Tegne opp ferdige affinitetsdiagram

Vi har valgt å kalle applikasjonen som skal utvikles KjApp. Omfanget på utviklingsoppgaven skal være moderat for å få god tid til å eksperimentere med det utviklede verktøyet. Målet er å utvikle en prototyp med et enkelt og logisk oppbygd brukergrensesnitt og tilstrekkelig funksjonalitet til å støtte KJ metoden.

#### Eksperimentering med distribuert arbeid

I eksperimentet ønsker vi å sammenligne distribuert KJ med KJ gjennomført med deltakerne i samme rom. Vi vil benytte forsøkspersoner fra SINTEF, Giant Leap Technologies (GLT) og studenter fra NTNU. Data samles inn ved hjelp av spørreskjema og ved å ha uformelle samtaler med deltakerne etter hver gjennomføring. Vi vil i tillegg registrere kvantitative resultater som tid og antall genererte ideer.

# Kapittel 2

## Idegenerering

### Innhold

---

2.1	Affinitetsdiagrammer . . . . .	3
2.2	Beskrivelse av KJ metoden . . . . .	4
2.3	Eksisterende verktøy for idegenerering . . . . .	5

---

Kapittelet presenterer idegenereringsmetoden oppgaven tar for seg. Først beskrives KJ affinitetsdiagrammer generelt. Deretter beskrives vår fortolkning av hvordan KJ metoden skal gjennomføres basert på egne erfaringer og samtaler med ansatte ved SINTEF som er erfarne brukere av metoden. Til slutt beskrives eksisterende verktøystøtte for idegenerering.

### 2.1 Affinitetsdiagrammer

KJ affinitetsdiagrammer er en strukturert metode for idemyldring som har fått sitt navn etter Jiro Kawakita [?]. Metoden brukes mye i industrien, blant annet av Nokia og SINTEF. KJ benyttes for å generere ideer for et gitt problem. Relaterte ideer grupperes og gruppene gis et beskrivende navn som sammenfatter ideer i gruppen. Affinitetsdiagrammer er nyttige når:

- oversikt over eksisterende ideer mangler
- problemet er *ullent*
- det er behov for banebrytende tenkning

Generering av KJ affinitetsdiagrammer er mer en kreativ enn en logisk prosess. Fordi alle ideer inkluderes vil metoden oppmuntre til aktiv deltagelse. Affinitetsdiagrammer organiserer informasjon ved å ta utgangspunkt i de enkelte ideer, for så å gruppere relaterte ideer sammen med en beskrivende overskrift. Ved behov kan flere grupper grupperes i *supergrupper*. Metoden benytter en bunn-opp tilnærming for å bygge et hierarki av ideer.

## 2.2 Beskrivelse av KJ metoden

Avsnittet beskriver KJ metoden slik den gjennomføres ved SINTEF. I tillegg har vi studert metoden i [?], [?] og [?]. Beskrivelsen og egne erfaringer med bruk av metoden ved Hærens forsyningskommando [?] danner grunnlaget for funksjonelle krav til prototypen som skal utvikles.

Deltakerne i en KJ sesjon bør ha kompetanse som gjør at flere aspekter ved temaet som behandles kommer frem. Alle personer som kan ha noe å bidra med bør inkluderes. Gruppen bør bestå av tre til seks personer. Temaet for sesjonen formuleres åpent. Dette sikrer at flest mulig ideer kommer frem. Så snart deltakerne er enige om temaet, plasseres dette godt synlig på tavlen. KJ metoden benytter tradisjonelle retingslinjer for idemyldring. Dette innebærer at man ikke kritiserer andres ideer og generer mange ideer på kort tid. Som forberedelse til sesjonen gis deltakerne fem minutter til å skrive egne ideer på post-it lapper. Hver ide bør være så konkret som mulig og beskrives med fem til syv ord. Tvetydighet forhindres ved at ideen inneholder et verb og et substantiv. Ideen skrives tydelig og så stort som mulig. Presentasjonen av ideer foregår ved at deltakerne plasserer en ide på tavlen og forteller kort hva ideen beskriver. Presentasjonen går på rundgang til alle ideer er presentert. Figur 2.1 viser en person i en KJ sesjon som presenterer en av sine ideer på tavlen. Etter at ideene er presentert grup-



Figur 2.1: En deltaker i en KJ sesjon presenterer en av sine ideer

perer deltakerne ideene i fellesskap. Deltakerne oppfordres til å reagere dersom de er uenige i plasseringen av en ide fremfor å godta det de ser. Stor hastighet og høyt aktivitetsnivå er viktigere enn nøye overveielse og ettertenksomhet. Uoverenstemmelser om plassering av en ide løses udiplomatisk. Dette innebærer at deltakere flytter ideer som de ikke liker plasseringen til uten å ta hensyn til hva andre deltakere mener. Gruppen som helhet vil etterhvert komme til enighet. Dersom gruppen ikke kommer til enighet om plasseringen av en ide, kan dette løses ved å kopiere ideen og plassere den i alle aktuelle grupper. Det er viktig at deltakerne er åpne for ny gruppering av ideer. Deltakerne må unngå å plassere ideer i sikre, kjente kategorier siden dette kan forhindre gjennombrudd i tenkningen.

Etter at ideene er gruppert, skal det lages overskrifter til idegruppene. Overskriften sammenfatter det essensielle i idegruppen og skrives på en egen lapp. Overskriften plasseres over idegruppen og en utenforstående skal kunne forstå essensen av ideene i

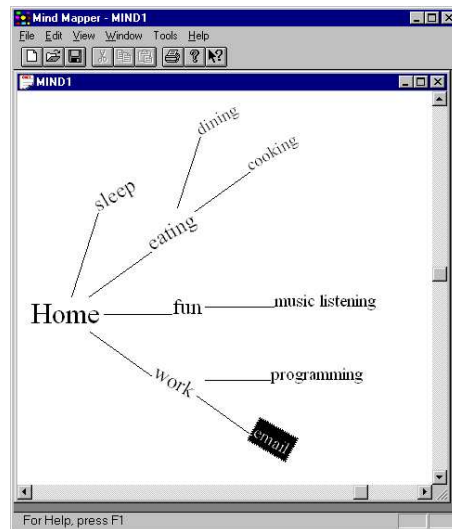
gruppen bare ved å lese overskriften. Relaterte grupper plasseres i nærheten av hverandre og bindes sammen med linjer.

## 2.3 Eksisterende verktøy for idegenerering

Det finnes flere verktøy på markedet som støtter idegenerering. Verktøyene er laget for å stimulere til kreativ tenking alene eller sammen med andre. Vårt inntrykk er at verktøyene er laget generelle for å støtte flest mulig tilfeller hvor det er behov for idegenerering. Verktøyene har ofte mye funksjonalitet som brukeren må sette seg inn i og som han kanskje aldri får bruk for. Vi har ikke funnet noen verktøy som gjenskaper tavlemiljøet som benyttes i KJ metoden. Etter vårt syn er det viktig at deltakerne i en KJ sesjon har minst mulig fokus på verktøyet som benyttes. Verktøyet skal støtte prosessen, men ikke ta fokus vekk fra problemet som skal løses ved hjelp av idegenerering. For bedrifter som er vant med KJ metoden mener vi det er nyttig med et verktøy som i størst mulig grad gjenspeiler miljøet de normalt jobber i. Eksempler på eksisterende verktøy for idegenerering presenteres i de følgende avsnitt.

### Mind Mapper

Mind Mapper er et shareware program for Windows plattformen. Programmet visualiserer ideer ved hjelp av hjernekart som vist i figur 2.2. Underideer presenteres som barnenoder til andre ideer. Denne måten å presentere ideer på, mener vi ikke godt nok gjenspeiler lokal gjennomføring av en KJ sesjon.

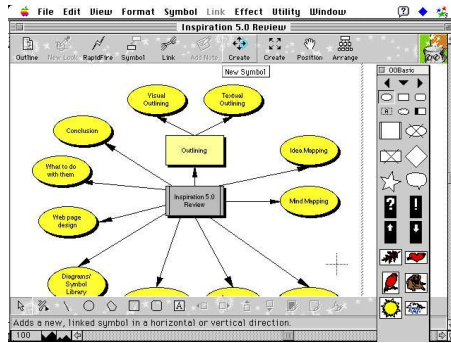


Figur 2.2: Skjermbilde fra Mind Mapper

### Inspiration

Inspiration er et program for å visualisere diagrammer. Programmet gir brukeren mulighet til å presentere ideer som kan kobles sammen ved hjelp av linjer og piler. Brukergrensesnittet gir tilgang til en rekke forskjellige typer bokser, piler og symboler.

Vi mener forøvrig at heller ikke dette programmet presenterer ideer å en måte som er forenelig med KJ metoden. Et skjermbilde fra programmet er vist i figur 2.3



Figur 2.3: Skjermbilde fra Inspiration

### Microsoft Visio

Microsoft Visio gir brukeren muligheten til å definere egne figurgrupper som kan benyttes for å visualisere elementene i en KJ sesjon. Visio kan deles ved hjelp av Microsoft NetMeeting og benyttes i en distribuert KJ sesjon. Verktøyet har imidlertid mye funksjonalitet som ikke er nødvendig for å visualisere post-it lapper og ideer. Vi mener derfor at en løsning med Visio vil kunne ta fokus bort fra idegenereringen.

**Del I**

**Eksperimentet**



# Kapittel 3

## Metode

### Innhold

---

3.1	Empiriske metoder . . . . .	10
3.2	Hypotese . . . . .	11
3.3	Validitet . . . . .	12
3.4	Goal/Question/Metric . . . . .	14
3.5	Analyseteknikker . . . . .	18
3.6	Omgivelser for eksperimentet . . . . .	19
3.7	Oppsett av eksperimentet . . . . .	21

---

Ved hjelp av empiriske studier ønsker vi å se på bruk av KJ-metoden i forskjellige omgivelser, nærmere bestemt distribuert og lokalt. Idegenerering er mer en kreativ enn en logisk prosess og vi ønsker å se på hvilke forskjeller som oppstår når idegenerering gjennomføres i ulike omgivelser. Vår erfaring tilsier at idegenerering fungerer best med deltakerne i samme rom. Deltakerne har da øyekontakt og kommuniserer både muntlig og ved hjelp av kroppsspråk. I distribuerte omgivelser går kroppsspråket tapt og dette kan føre til tregere kommunikasjon. Men er det virkelig slik at idegenerering fungerer best med deltakerne i samme rom? Og hvilke faktorer er det som avgjør hvor godt idegenereringen fungerer?

Dette er noen av spørsmålene vi ønsker å få svar på ved hjelp av empiriske studier. I dette kapittelet diskuteres og defineres eksperimentet som skal gjennomføres. Arbeidet er basert på *Experimentation in Software Engineering - An introduction* [?]. Kapittelet starter med å beskrive ulike typer empiriske studier i avsnitt 3.1. Videre defineres hypotesene som skal undersøkes ved hjelp av eksperimentet i avsnitt 3.2, mens trusler mot eksperimentets validitet diskuteres i avsnitt 3.3. Avsnitt 3.4 beskriver Goal/Question/Metric som er en analyse for å definere målet med eksperimentet og hvilke målinger som skal gjøres. Resultatene fra eksperimentet skal analyseres ved hjelp av statistiske metoder som beskrives i avsnitt 3.5. Eksperimentets omgivelser presenteres i avsnitt 3.6. Kapittelet avsluttes med å beskrive oppsettet til eksperimentet i avsnitt 3.7.

### 3.1 Empiriske metoder

I forbindelse med empiriske studier er det hovedsakelig tre forskjellige typer som kan gjennomføres; inspeksjon (survey), casestudie (tilfellestudie) og eksperiment [?].

#### Inspeksjon

Inspeksjon gjennomføres i etterkant av at en teknikk eller et verktøy har blitt benyttet en stund. Ved å gjennomføre intervjuer og sende ut spørreskjema kan kvalitative og kvantitative data samles inn. Det er viktig at deltakerne i en inspeksjon er representative for populasjonen som studeres. Dersom studien ser på effekten av en ny metoden for testing, vil man få forskjellige svar ved å stille de samme spørsmål til en avdelingsleder og en testansvarlig. Avdelingslederen vil kanskje fokusere på effekter som kundetilfredshet og mer stabile produkter som følge av færre feil i utviklet produkt, mens en testansvarlig vil kanskje fokusere på endring i egne arbeidsrutiner. Intervjuobjektene må derfor velges med tanke på hvilket perspektiv studien ønsker å fokusere på. Ved å analysere resultatene fra inspeksjonen kan man utarbeide beskrivende og forklarende konklusjoner om teknikken eller verktøyet som har blitt benyttet.

#### Casestudie

Casestudier gjennomføres ved å overvåke pågående prosjekter eller aktiviteter og samle inn relevant informasjon underveis. Casestudier er egnet for industriell evaluering av metoder eller verktøy og benyttes for å studere effekter i en karakteristisk situasjon. Ulempen er at det er vanskelig å generalisere resultatene fra et casestudie. Casestudie kan for eksempel benyttes for å sammenligne en ny inspeksjonsmetode mot en eksisterende. Dersom en bedrift erfarer at en inspeksjonsteknikk fungerer godt, så er det ikke sikkert at en annen bedrift vil trekke de samme konklusjoner. Faktorer som de ansattes erfaring og kompetanse er vanskelig å kontrollere og kan være avgjørende for resultatet. Et problem med casestudie er derfor manglende kontroll over omgivelser. For eksempel er det vanskelig å si om suksess med et nytt verktøy kommer av at de ansatte er entusiastiske over å få prøve noe nytt eller om det nye verktøyet er vesentlig bedre enn det gamle. Økt kontroll over situasjonen oppnås ved å gjennomføre eksperimenter.

#### Eksperimenter

Eksperimenter gjennomføres normalt i laboratoriemiljøer hvor deltakerne i eksperimentet *utsettes* for forskjellige behandlinger. Hensikten er å manipulere enkelte variable og holde de andre variable konstante. Med resultatene fra eksperimenter kan statistiske analyser gjennomføres og man kan finne ut hvilken innvirkning de ulike behandlingene har. Et ekte eksperiment krever at deltakerne plukkes ut tilfeldig og at behandlingen de utsettes for velges tilfeldig fra en mengde behandlinger som man ønsker å studere effekten av. Det er imidlertid vanskelig å oppnå fullstendig tilfeldig utplukking av deltakere. Deltakerne må være villige til å stillet opp i eksperimentet og i tillegg må de være innenfor målgruppen som studeres. Mangel på tilfeldig utvelgelse fører til at eksperimentet blir et kvasi-eksperiment.

#### Valg av empirisk metode

Vi har valgt å benytte eksperiment, nærmere bestemt kvasi-eksperiment, som metode for de empiriske studiene av idegenerering. Det er flere grunner til dette. Vi kjenner ikke

til noen bedrifter som benytter KJ metoden i distribuerte omgivelser og det er derfor ikke mulig å gjennomføre inspeksjon for å hente ut erfaringer om denne typen arbeid. Casestudie hos bedrifter kunne vært en mulig angrepsvinkel på de empiriske studiene. For å kunne gjøre det må vi finne en bedrift som regelmessig benytter idegenerering og som kan ha en gevinst av å gjennomføre dette i distribuerte omgivelser. Bedrifter med geografisk spredning er aktuelle kandidater. I tillegg må bedriften være villig til å utforske bruken av en arbeidsform som ikke kan dokumenteres som effektiv. Vi tror det kan være vanskelig å finne en bedrift som er villig til å prøve ut dette i utstrakt bruk. Vi mener derfor at et eksperiment er den beste måten å studere effektene av idegenerering i forskjellige omgivelser da det gir mulighet for å kontrollere omgivelsene. Det er viktig å merke seg at eksperimentet i stor grad påvirkes av menneskelige faktorer. Det er mennesker som skal gjennomføre sesjonene og resultatet er avhengig av deltakernes bakgrunn og erfaring. I tillegg er området som studeres *ullent*. Eksperimentet vil antagelig ikke gi oss et klart svar på hvilken type sesjon som fungerer best. Det vi kan vente oss er indikatorer på hva som fungerer og ikke fungerer under de ulike sesjonene. Eksperimentet vil i tillegg gi viktig informasjon i form av observasjon av deltakerne. Observasjonene av sesjonene vil gi oss en pekepinne på hvilke faktorer som er viktige ved gjennomføring av en KJ sesjon.

## 3.2 Hypotese

Deltakerne i eksperimentet sørger for at relevante data fra lokale og distribuerte gjennomføringer produseres. Svar på spørreskjema og kvantitative resultater fra eksperimentet danner datagrunnlaget som benyttes for å teste hypoteser. Før eksperimentet gjennomføres er det imidlertid viktig at hypotesene defineres. I de følgende avsnitt presenteres nullhypotesen og alternativ hypotese. Til slutt i avsnittet presenteres faktorene vi ønsker å benytte ved hypotesetesting.

### 3.2.1 Nullhypotese

Tradisjonelt gjennomføres KJ med deltakerne i samme rom. Effekten av å gjennomføre KJ i samme rom, er at deltakerne motiveres av hverandres engasjement i saken og dermed produserer flere og bedre ideer. Det å samle deltakerne i samme rom virker positivt på kommunikasjon. Deltakerne ser og hører hverandre, og kroppsspråket kan bidra til effektiv kommunikasjon. KJ gjennomført i distribuerte omgivelser møter utfordringer i forhold til dette. Deltakerne kan ikke se hverandre og kommunikasjon via kroppsspråk går tapt. Det kan også være vanskelig å prate åpent og naturlig ved hjelp av telefonkonferanse. I tillegg skal deltakerne i distribuert KJ benytte samarbeidsteknologi som NetMeeting og verktøyet KJApp for idegenerering. Disse faktorene kan bidra til at resultatene fra distribuert KJ blir dårligere enn fra en lokal gjennomføring. Vår nullhypotese er derfor:

*KJ gjennomført med deltakerne i samme rom fungerer bedre enn KJ gjennomført i distribuerte omgivelser*

Gjennom eksperimentet ønsker vi å se om dette virkelig er sant. Distribuert KJ har en rekke fordeler i forhold til lokal KJ. Dette gjelder blant annet reduserte reisekostnader og tidsbesparing ved at deltakerne kan gjennomføre KJ sesjonen fra eget kontor.

### 3.2.2 Alternativ hypotese

Alternativ hypotese følger av nullhypotesen og er:

*KJ gjennomført distribuert fungerer like bra som eller bedre enn KJ gjennomført med deltakerne i samme rom*

Vi mener det er visse forutsetninger som må være på plass for at dette skal være tilfelle. For det første bør deltakerne i en distribuert sesjon være vant med å benytte samarbeidsteknologi. Vi tror også det er en fordel om deltakerne kjenner hverandre fra før.

### 3.2.3 Faktorer ved hypotesetesting

Hypotesene er på mange måter vagt formulert og sier ingen ting om hva som gjør en sesjon bedre eller dårligere enn en annen. For å kunne foreta en hypotesetest må dette være på plass. Vi mener at svar på følgende spørsmål indikerer om en sesjon er bedre enn en annen:

- Følte du at du bidro?
- Hvordan fungerte gruppen?
- Følte du at sesjonen hadde trøkk?
- Var det morsomt?

Videre er det interessant å spørre deltakerne om hvordan de bedømmer kvaliteten på resultatet. Dette kan gjøres med spørsmålene:

- I hvor stor grad tror du at du kan benytte resultatet?
- Kom det frem nye momenter/ideer?
- Kunne du kommet frem til resultatet alene?

Spørsmålene tar for seg kvalitative sider ved en sesjon og gir et svar på hva deltakerne mener om de ulike gjennomføringene. En høy verdi på svaret er bedre enn en lav verdi. I tillegg må kvantitative faktorer som tid og antall ideer generert vurderes. Det er imidlertid vanskelig å gi et klart svar på om et kvantitativt resultat indikerer at sesjonen er bra eller dårlig. I utgangspunktet kan man anta at mange ideer er bedre enn få ideer. Men i enkelte tilfeller kan det være motsatt, at få og gode ideer er bedre enn mange dårligere. Vi mener derfor at kvantitative faktorer må ses i sammenheng med kvalitative faktorer. Goal/Question/Metric analysen i avsnitt 3.4 beskriver dette nærmere.

Under hypotesetesting velger vi å benytte et signifikansnivå på 0.1. Dette vil si at vi forkaster nullhypotesen dersom vi med nitti prosent sannsynlighet kan si at distribuert KJ fungerer like bra som eller bedre enn lokal. Dersom vi ikke kan gjøre det, beholdes nullhypotesen.

## 3.3 Validitet

Det er flere faktorer som truer validiteten til et eksperiment. I dette avsnittet diskuteres de forskjellige truslene mot validitet i vårt eksperimentet og hva vi gjør for å redusere risikoen.

### 3.3.1 Konklusjonsvaliditet

Konklusjonsvaliditet handler om evnen til å trekke de rette konklusjoner av hvilken effekt behandlingen har på resultatet. Med eksperimentet ønsker vi å finne ut om det er et statistisk signifikant forhold mellom behandlingen vi utsetter deltakerne for og resultatet. Deltakerne i eksperimentet utsettes for to behandlinger; idegenerering under distribuerte og lokale forhold. Med eksperimentet ønsker vi å finne ut om lokal gjennomføring fungerer bedre enn distribuert (nullhypotesen fra avsnitt 3.2) eller om de kan fungere like godt (alternativ hypotese fra avsnitt 3.2). I forbindelse med konklusjonsvaliditet er det en fare for at vi forsøker å *fiske* etter et bestemt resultat, nemlig at distribuert sesjon kan fungere like godt som lokal. Vi innser at dette er en trussel for gyldigheten av eksperimentet, men mener at vi skal klare å beholde et objektivt syn på resultatene fra eksperimentet. For oss går det ingen prestisje i å oppnå et bestemt resultat, noe som kunne vært en fare dersom eksperimentresultatene skulle benyttes for å forsvare en teori vi har utarbeidet. En annen trussel i denne sammenheng er påliteligheten til målingene som gjøres. Ideelt sett skal et eksperiment som gjennomføres to ganger, gi de samme måleresultatene. I vårt eksperiment er det deltakerne som produserer måleresultatene i form av svar på spørreskjema og genererte ideer. Det er derfor lite trolig at to etterfølgende sesjoner med de samme deltakerne vil gi eksakt samme resultat. Omgivelsene til eksperimentet vil også kunne være en trussel mot konklusjonsvaliditet. Elementer utenfor eksperimentet kan virke forstyrrende på deltakerne og påvirke resultatet i en eller annen retning. Dersom uforutsette hendelser inntreffer er det viktig å notere disse, slik at resultatene kan tolkes på bakgrunn av dette.

### 3.3.2 Intern validitet

Tilfeldigheter kan påvirke uavhengige variable i en eller annen retning. Dette kan igjen påvirke konklusjonen med tanke på sannsynlig sammenheng mellom behandling og resultat. Uavhengige variable er de variable som manipuleres og kontrolleres i eksperimentet og i vårt eksperiment er faktoren lokal eller distribuert gjennomføring en slik variabel. Det finnes en rekke trusler mot intern validitet. Et eksempel er at ulik bakgrunn hos deltakerne i eksperimentet påvirker resultatet i en eller annen retning. I vårt eksperiment vil for eksempel deltakere med erfaring innen idegenerering eller distribuert arbeid antagelig takle omgivelsene bedre enn deltakere uten erfaring. Et annet eksempel er naturlig variasjon innen deltakernes yteevne, noe som helt klart påvirker resultatet. Ved å benytte flere grupper og et balansert design mener vi at trusselen mot intern validitet reduseres. En annen trussel mot intern validitet er at deltakernes ytelse kan påvirkes av behandling de utsettes for. I vårt eksperiment vil for eksempel en deltaker som fortrekker å gjennomføre idegenerering lokalt kunne produsere et dårligere resultat distribuert. Dette kan deltakeren gjøre fordi han misliker å arbeide distribuert og ikke fordi det er vanskeligere å arbeide distribuert.

### 3.3.3 Konstruksjonsvaliditet

Konstruksjonsvaliditet omhandler muligheten for generalisere resultatene til konseptene eller teorien som danner grunnlaget for eksperimentet. Truslene mot konstruksjonsvaliditet relateres til eksperimentets design og til sosiale faktorer. I vårt eksperiment er det en trussel mot konstruksjonsvaliditet at vi ikke presist nok har klart å definere *fungere bedre* og *fungere like godt* fra hypotesene i avsnitt 3.2. En annen trussel mot konstruksjonsvaliditet er at deltakerne forsøker å gjette hva eksperimentet skal studere

og dermed påvirker innsats eller svarene i en eller annen retning. Deltakere som blir evaluert kan også finne på å definere seg selv som flinkere enn det de er. For å minske risikoen benytter vi ulike typer målinger, nemlig spørreskjema, intervju, observasjon og kvantitative målinger (for eksempel tid og antall ideer). På denne måten kan de ulike resultatene kobles opp mot hverandre og styrke resultatet i en eller annen retning.

### 3.3.4 Ekstern validitet

Trusler mot ekstern validitet er vilkår som gjør at resultatene fra eksperimentet ikke kan generaliseres til *allmenn gyldighet*. Resultatene fra vårt eksperiment er vanskelige å generalisere fordi deltakerne kommer fra et teknologiintensivt miljø og antagelig har bedre forutsetninger for å beherske distribuerte omgivelser enn deltakere fra andre miljøer. Likevel vil eksperimentet kunne gi indikasjoner på faktorer som må være til stede for at idegenerering i distribuerte omgivelser skal fungere.

## 3.4 Goal/Question/Metric

Goal/Question/Metric (GQM) baserer seg på antagelsen om at dersom en organisasjon skal måle noe på en meningsfull måte må den gjøre tre ting. For det første må målene for organisasjonen og dets prosjektene spesifiseres. Videre må målene følges til dataene som definerer målene. Til slutt må et rammeverk være på plass for å tolke dataene i forhold til målene [?] [?].

GQM opererer på tre nivå:

1. Konseptuelt nivå (Goal). Et mål defineres for et objekt. Objekter kan være produkter, prosesser eller ressuserer.
2. Operasjonelt nivå (Question). En mengde spørsmål definerer hvordan evalueringen av et bestemt mål skal gjøres. Spørsmålene karakteriserer objektet som studeres med hensyn på kvalitetsattributter.
3. Kvantitativt nivå (Metrics). En mengde data assosieres med hvert spørsmål og skal på en kvantitativ måte, enten subjektivt eller objektivt, besvare spørsmålet.

Som en forberedelse til eksperimentet gjennomførte vi en GQM analyse. Eksperimentet skal studere forholdet mellom KJ gjennomført lokalt (deltakerne i samme rom) og KJ gjennomført distribuert (deltakerne i forskjellige rom). Målet for eksperimentet er:

*Analysér KJ prosessen for å se på forskjeller mellom lokal og distribuert KJ sett fra deltakerne i diplomeksperimentets synspunkt.*

Neste skritt i GQM analysen er å definere spørsmål. GQM skiller mellom spørsmål for fokus og spørsmål for omgivelser. Spørsmål for fokus omhandler primæreffekten som studeres i eksperimentet. Dette kan for eksempel være effektivitet, kostnad eller pålitelighet. Våre spørsmål for fokus beskrives i avsnitt 3.4.1. Spørsmål for omgivelser beskriver omgivelsene til eksperimentet. Omgivelser definerer deltakerne i eksperimentet, for eksempel med tanke på erfaring og bakgrunn. I tillegg defineres eksperimentets objekter, for eksempel egenskaper ved verktøy som er benyttet. Våre spørsmål for omgivelser beskrives i avsnitt 3.4.2.

### 3.4.1 Spørsmål for fokus

Eksperimentet som skal kjøres har som mål å studere effekter av KJ gjennomført i distribuerte og lokale omgivelser. Vi ønsker å studere både kvantitative og kvalitative effekter av KJ prosessen. Følgende spørsmål er definert for å se på kvantitative effekter av KJ:

- Q1: Hvor mange post-it lapper ble generert
- Q2: Hvor mange ideer ble generert
- Q3: Hvor mange idegrupper ble generert
- Q4: Hvor lang tid brukte gruppen

De kvantitative effektene er enkle å måle, men gir begrenset informasjon om de ulike gjennomføringene. Følgende spørsmål ble definert for å se på kvalitative effekter av KJ:

- Q5: Hvordan opplevde deltakerne KJ prosessen
- Q6: Hvordan bedømmer deltakerne kvaliteten på løsningen
- Q7: Følte du at sesjonen hadde trøkk

De følgende avsnitt utdyper spørsmålene og angir metrikker. Svar på spørsmålene utgjør metrikken. Metrikker for spørsmålene for fokus er tall. Vi har valgt å benytte en fempunkts bipolar Likert skala for de kvalitative spørsmålene Q5, Q6 og Q7. Dette er en ordinal skala som gir en del fordeler med tanke på statistisk analyse av resultatene. Dette er beskrevet i avsnitt 3.5 om analyseteknikker.

#### **Q1: Hvor mange post-it lapper ble generert**

Hensikten med spørsmålet er å finne ut om antallet lapper som genereres under lokale og distribuerte sesjoner er forskjellig. Tidligere erfaring med KJ metoden har vist at hver deltaker genererer omkring fem ideer under en KJ sesjon og vi forventer at resultatet på Q1 er antall deltakere multiplisert med fem. Antallet post-it lapper telles etter endt sesjon og lagres i metrikk M1.

#### **Q2: Hvor mange ideer ble generert**

Q2 skal i forhold til Q1 kvantifisere hvor mange frittstående ideer som genereres i sesjonen. Dette gjøres ved å spørre deltakerne om hvor mange ideer de mener er generert. Post-it lapper som beskriver like eller nesten like ideer telles en gang. Det er opp til den enkelte deltaker å avgjøre om ideer er like eller forskjellige. Metrikk M2 tar vare på resultatet for Q2 og forventes å være litt i underkant av M1.

#### **Q3: Hvor mange grupper ble generert**

En gruppe inneholder en eller flere post-it lapper. Q3 kvantifiserer antallet grupper generert i sesjonen. I analysen kan det være interessant å se på forholdet mellom antall ideer og antall idegrupper, men også forholdet mellom antall idegrupper for lokale og distribuerte sesjoner. En gruppe som inneholder to undergrupper vil i denne sammenheng regnes som en gruppe i analysen. Metrikk M3 forventes å være lik en tredjedel av M1.

**Q4: Hvor lang tid brukte gruppen**

Spørsmålet Q4 fanger opp varigheten til sesjonen. Under eksperimentet noterer moderator start- og sluttidspunkt. Moderator merker seg om han avbrøt sesjonen på grunn av at tidsgrensen på førtifem minutter var oversteget. Metrikk M4 lagrer varigheten på sesjonene og forteller hvilken type sesjon som tok lengst tid. I tillegg vil M4 sammen med verdier for M1 til M3 si hvor mange ideer og grupper som genereres per tidsenhet. M4 forventes å ha verdien tretti minutter. Metrikk M5 vil i tillegg fortelle oss om sesjonen ble avbrutt av moderator eller ikke. M5 lik (0) indikerer at sesjonen ikke er avbrutt, M5 lik (1) indikerer at den er avbrutt.

**Q5: Hvordan opplevde deltakerne sesjonen**

Q5 skal fange opp de kvalitative opplevelsene hos deltakerne. For å nyansere spørsmålet, stilles følgende underspørsmål med metrikk M5.1 til M5.3:

- Q5.1: Var det morsomt
- Q5.2: Følte du at du bidro
- Q5.3: Hvordan fungerte gruppen

Deltakerne besvarer spørsmålene på en skala fra (1) til (5). Alternativ (1) er negativt og indikerer at det ikke var morsomt, og at gruppen ikke fungerte. Alternativ (5) er positivt. Spørsmålet gjør det mulig å danne seg et bilde av hva deltakerne mener fungerer best. Sammenligning av lokale og distribuerte sesjoner er interessant, både for de enkelte gruppene og på tvers av grupper, for eksempel ved hjelp av gjennomsnittsverdier. Fordi vi tror på metoden, forventer vi at svarene på M5.1, M5.2 og M5.3 ligger mellom (4) og (5).

**Q6: Hvordan var kvaliteten på løsningen**

Q6 indikerer kvaliteten på resultatet. Deltakerne foretar en subjektiv vurdering av kvaliteten på løsningen. Spørsmålet nyanseres ved hjelp av underspørsmål med metrikk M6.1 til M6.3:

- Q6.1: I hvor stor grad tror du at du kan benytte resultatet
- Q6.2: Kom det frem nye momenter
- Q6.3: Hadde du kommet frem til resultatet alene

Deltakerne besvarer spørsmålet på en skala fra (1) til (5). Alternativ (1) er mest negativt og alternativ (5) mest positivt. For M6.1 indikere alternativ (1) at resultatet er ubrukelig, mens alternativ (5) indikerer at hele resultatet kan brukes. For M6.2 indikerer alternativ (1) at det ikke kom frem noen nye momenter, mens alternativ (5) sier at det i stor grad kom frem nye momenter. For M6.3 indikerer alternativ (1) at deltakeren mener at det er åpenbart han ville kommet frem til resultatet alene. Dette er negativt for KJ metoden. Alternativ (5) indikerer at han neppe ville kommet frem til resultatet alene, noe som er positivt for KJ metoden. Spørsmålet benyttes for å analysere resultatets kvalitet i de forskjellige sesjonene og for å se på eventuelle tendenser. Metrikkene M6.1 til M6.3 kan besvare om lokal sesjon gir bedre resultater enn distribuert eller om det er omvendt.

**Q7: Følte du at sesjonen hadde trøkk**

Deltakerne gir en subjektiv vurdering av hvor god flyt det var under sesjonen. Kommer ideene på løpende bånd sier vi at det var mye trøkk. Dersom ideene satt langt inne var det lite trøkk. Deltakerne svarer på en skala fra (1) til (5) og resultatet lagres i metrikk M7. Vi forventer at lokal sesjon har større trøkk enn distribuert sesjon fordi deltakere i samme rom lettere kan stimulere hverandre til økt innsats. M7 forventes å ha (4) som verdi for lokal sesjon og (3) for distribuert sesjon.

**3.4.2 Spørsmål for omgivelser**

Spørsmålene i dette avsnittet skal hjelpe oss med å karakterisere omgivelsene til eksperimentet. Vi har laget spørsmål både for å definere deltakerne i eksperimentet (QB og QC) og verktøyet som benyttes (QE). I tillegg finnes to spørsmål som tar for seg om gjennomført sesjon var lokal eller distribuert og rekkefølgen på sesjonene (QA og QD). For spørsmålene QB, QC og QE benyttes igjen en fempunkts bipolar Likert skala.

- QA: Var sesjonen distribuert eller lokal
- QB: Setter deltakeren pris på å være anonym i prosessen
- QC: Hvor fortrolig er du med de andre deltakerne i prosessen
- QD: Startet gruppen med lokal eller distribuert KJ
- QE: Fungerte applikasjonen KjApp tilfredsstillende

**QA: Var sesjonen distribuert eller lokal**

QA benyttes kun for å skille mellom lokal og distribuert gjennomføring. Svaret lagres i metrikken M8 som tar verdiene (0) for distribuert og (1) for lokal.

**QB: Setter deltakeren pris på å være anonym i prosessen**

QB benyttes for å angi deltakernes ønske om anonymitet i en KJ sesjon. Med anonymitet mener vi her at deltakerne ikke har øyekontakt. QB kan for eksempel benyttes for å se om deltakerne som foretrekker å være anonyme synes at KJ distribuert fungerer best. Svaret for QB lagres i metrikken M9 og deltakerne svarer på en skala fra (1) til (5). (1) indikerer at deltakeren ikke setter pris på å være anonym og (5) at han setter veldig pris på å være anonym.

**QC: Hvor fortrolig er du med de andre deltakerne i prosessen**

QC kan til en viss grad relateres til QB. Metrikk M10 angir hvor fortrolige deltakerne er med hverandre. Deltakerne svarer på en skala fra (1) til (5), hvor (1) indikerer lite fortrolig og (5) svært fortrolig. Fortrolige deltakere trives i hverandres selskap og vil kanskje derfor foretrekke KJ gjennomført lokalt.

**QD: Startet gruppen med lokal eller distribuert KJ**

Metrikk M11 angir om deltakeren startet med lokal eller distribuert sesjon. For eksempel benyttes M11 for å finne ut om første gjennomføring gir flere ideer uansett om den gjennomføres lokal eller distribuert. Kanskje andre gjennomføring gir færre ideer fordi deltakerne er slitne?

### QE: Fungerte applikasjonen KjApp og NetMeeting tilfredsstillende

Deltakernes vurdering av hvor godt verktøyene fungerer lagres i metrikk M12. Deltakerne svarer på en skala fra (1) til (5), hvor (1) indikerer at verktøyene fungerte dårlig. Dersom distribuert gjennomføring med verktøyet får lave verdier på resultat kvalitet, kan M12 si noe om det er verktøyet KjApp eller NetMeeting som er årsaken eller ikke.

## 3.5 Analyseteknikker

Resultatene lagret i GQM metrikkene utgjør datamaterialet som skal analyseres og det finnes flere forskjellige statistiske metoder for gjøre dette. Parametriske tester kan benyttes på datamateriale som har en bestemt fordeling, som regel normalfordelt. Vårt datamateriale består av femten sampler og erfaring tilsier derfor at datamaterialet er normalfordelt. Parametriske tester krever i tillegg at det minimum benyttes intervallskala. Tradisjonelt skal man derfor ikke benytte parametriske tester på datasett som er hentet inn ved hjelp av en ordinal skala. Imidlertid kan skillet mellom ordinal- og intervallskala være flytende når ordinalskalaen som benyttes har monotont stigende verdier. I GQM analysen har vi bestemt oss for å benytte en fempunkts bipolar Likert skala. Dette er en ordinal skala hvor (3) utgjør *middelverdien* og det er like lang avstand til alternativ (1) og som til alternativ (5). Empiriske studier har vist at en slik fempunkts bipolar skala behandles som en intervallskala av studiesubjekter og dermed kan resultatene behandles som om de var produsert ved hjelp av en intervallskala. Dette er uavhengig av hvilke svaralternativer som benyttes. Et eksempel på en fempunkts bipolar Likert skala benyttet for spørsmålet *Hvordan fungerte gruppen* er:

1. Ikke i det hele tatt
2. Under middels
3. Middels
4. Over middels
5. Veldig bra

Videre har det vist seg at parametriske tester kan benyttes på datamateriale fra en skala som ikke strengt tatt er en intervallskala. En serie eksperimenter gjennomført av Baroudi og Orlikowski i 1989 [?] viser at dette ikke fører til feilaktige svar, annet enn i ekstreme tilfeller. For å forsikre oss om at datamaterialet vårt ikke utgjør et ekstremtilfelle, har vi foretatt både parametriske og ikke-parametriske analyse for deler av datasettet. Tillegg D.2 og D.1 viser resultatene for henholdsvis ikke-parametriske Kruskal-Wallis test (beskrevet i avsnitt 3.5.2) og parametriske ANOVA test (beskrevet i avsnitt 3.5.1). Testene gir samme konklusjon, og viser at parametriske tester kan benyttes. Vi har derfor valgt å benytte ANOVA for å analysere datamaterialet, da dette er en statistisk sterkere test i forhold til det ikke-parametriske alternativet Kruskal-Wallis. For en mer utdypende gjennomgang av temaet og flere referanser til kilder som underbygger vårt valg av analysemetode, se [?]. Forskjellige skalatyper er presentert i tabell 3.1.

### 3.5.1 ANOVA

ANOVA (ANalysis Of VAriance) [?] [?] er en parametriske test som benyttes for å teste sannsynligheten for at populasjoner har lik middelvei og kan benyttes på mange

Tabell 3.1: Oversikt over skala typer

Skala	Vanlige operasjoner	Tillatte statistiske operasjoner
Nominal	Likhet / Ulikhet	Antall tilfeller, mode
Ordinal	Større enn / Mindre enn	Median, persentil, størrelsesorden
Interval	Likhet i intervaller	Aritmetisk middelvei, varians
Ratio	Likhet i forhold	Geometrisk middelvei

eksperimentelle design. ANOVA kan benyttes mot flere enn to populasjoner. I testen bruker ANOVA aggregerte middelveier og kvadratsummer. Ideen bak ANOVA er å studere *variasjon innenfor* en populasjon og *variasjon mellom* populasjoner. Variasjon innenfor et utvalg oppstår av forskjellige årsaker og vi sier at variasjonen er tilfeldig. Variasjon mellom populasjoner sier vi er systematisk og oppstår på grunn av faktoren som splitter populasjonene. Hensikten med ANOVA er å teste om variasjonen mellom utvalg oppstår på grunn tilfeldig eller systematisk variasjon. Variansen deles dermed inn i komponentene:

- Variasjon mellom utvalg
- Variasjon innenfor utvalg

Dersom det er en signifikant forskjell mellom middelveien til to populasjoner og testen viser at forskjellen kommer av systematisk variasjon, tyder dette på at faktoren som splitter populasjonene er den avgjørende. Men det kan også være andre faktorer som ikke er vurdert som skiller populasjonene og som er den egentlige årsaken til forskjellen i middelvei. Et *banalt* eksempel er to populasjoner splittet på bakgrunn av geografisk lokasjon og der eksperimentet finner ut at den ene populasjonen har bedre skriveferdigheter enn den andre. En rask konklusjon kan da være at geografisk lokasjon er avgjørende for hvor flink man er til å skrive. Imidlertid kan det være en annen faktor som skiller populasjonene; utdanningsnivå. Og antagelig er det denne faktoren som avgjør skriveferdighetene.

### 3.5.2 Kruskal-Wallis

Kruskal-Wallis [?] er et ikke-parametrisk alternativ til ANOVA. Testen kan benyttes for å teste sannsynligheten for at medianer i et utvalg er lik. Testen gjennomføres ved å rangere dataene i en serie. Testen bør alltid benyttes istedenfor ANOVA dersom det er uklart om kravene for ANOVA tilfredsstilles. Detaljert beskrivelse finnes i [?] og [?].

## 3.6 Omgivelser for eksperimentet

Tradisjonelt gjennomføres KJ med alle deltakerne i samme rom. Rommet som benyttes bør ha en tavle som er godt synlig og lett tilgjengelig for deltakerne i sesjonen. For en distribuert gjennomføring blir omgivelsene anderledes. Deltakerne sitter på hvert sitt kontor og gjennomfører KJ ved hjelp av samarbeidsteknologi. Vi har valgt å benytte Microsoft NetMeeting for å distribuere KJApp og telefonkonferanse for å overføre lyd mellom deltakerne.

### 3.6.1 Deling av applikasjoner i NetMeeting

NetMeeting er standard programvare på Windows plattformen og benyttes for oppkobling av konferanser via Internett. Konferansene kan inneholde lyd, bilde (video) og tekst (chat). NetMeeting støtter i utgangspunktet bare lyd mellom to brukere. For å koble opp en flerveis konferanse med lyd behøves en Multipoint Control Unit (MCU). NTNU har ingen egne MCUer og *åpne* MCUer gir ingen garanti for kvalitet. Av denne grunn har vi valgt å benytte telefonkonferanse for lydoverføring. NetMeeting gir også støtte for deling av applikasjoner. Applikasjondeling innebærer at en deltaker starter en applikasjon og gjør den tilgjengelig for de andre deltakerne i konferansen. Den delte applikasjonen kjøres kun hos deltakeren som foretok utdelingen. Denne deltakeren kalles eier av applikasjonen og fungerer som moderator i en KJ sesjon. Hos de andre deltakerne i konferansen vises en kopi av skjermbildet til den delte applikasjonen. Oppdateringer i applikasjonen reflekteres hos alle deltakerne.

Deltakere kan be eieren om tillatelse til å overta kontrollen over applikasjonen. Dersom eieren gir tillatelse, kan deltakeren som får kontrollen kjøre kommandoer og foreta oppdateringer på den delte applikasjonen. Deltakeren som har kontrollen kalles *aktiv deltaker*. Fortsatt reflekteres oppdateringene hos alle deltakerne i konferansen. Oppdatering skjer ved at tastatur- og musehendelser registreres hos aktiv deltaker, for deretter å sendes over til den kjørende instansen hos eieren av applikasjonen. Hos eieren blir applikasjonen oppdatert i henhold til kommandoene fra aktiv deltaker og det oppdaterte skjermbildet sendes ut til alle. På denne måten får aktiv deltaker tilgang til all funksjonalitet i applikasjonen, selv om den kjøres på en annen PC. I figur 3.1 illustreres hvilke data som sendes når aktiv deltaker har kontrollen over applikasjonen. I figuren deler eieren ut et støtteverktøy for KJ metoden og gir kontrollen til aktiv deltaker.

#### Eier

Eier starter applikasjonen og deler den ut til de andre deltakerne i konferansen. Applikasjonen kjøres på eierens PC. Eier gir kontrollen til aktiv deltaker. Eiers NetMeeting foretar følgende:

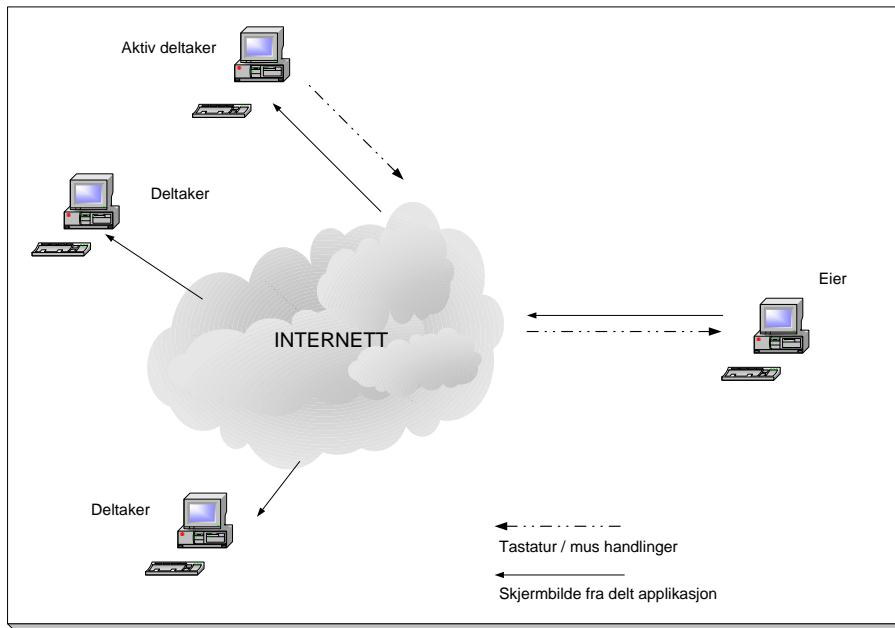
- Sender skjermbilde av delt applikasjon til alle deltakere
- Mottar tastatur- og musehendelser fra aktiv deltaker
- Oppdaterer delt applikasjon på bakgrunn av aktiv deltakers tastatur- og musehendelser
- Sender oppdatert skjermbilde av delt applikasjon til alle deltakere

#### Aktiv deltaker

- Mottar skjermbilde fra delt applikasjon fra eieren
- Sender tastatur- og musehendelser for å kontrollere delt applikasjon
- Mottar oppdatert skjermbilde av delt applikasjon fra eieren

### Andre deltakere

- Mottar skjermbilde av delt applikasjon fra eieren
- Mottar oppdatert skjermbilde av delt applikasjon fra eieren etter at aktiv deltaker har oppdatert denne



Figur 3.1: Dataflyt i forbindelse med deling av applikasjon i NetMeeting

### 3.6.2 Telefonkonferanse

For å overføre lyd under distribuerte KJ sesjoner benyttes telefonkonferanse. Telefonkonferanse er valgt fordi det gir pålitelig overføring av lyd med minimal tidsforsinkelse. Dagens telefoner kobles opp i konferanse enten via lokal sentral eller via sentral til tjenesteleverandøren, noe som i de fleste tilfeller betyr Telenor. Ved NTNU og SINTEF beskrives konferanseoppkobling i lokal telefonkatalog. For telefonkonferanser med deltakere tilknyttet forskjellige sentraler, må tjenesten settes opp via tjenesteleverandør. Telefonkatalogen beskriver prosedyre for oppkobling hos Telenor.

## 3.7 Oppsett av eksperimentet

Eksperimentet gjennomføres ved Institutt for datateknikk og informasjonsvitenskap (IDI), med unntak av sesjonen med SINTEF som kjøres i deres lokaler. Tremanns grupper benyttes og deltakerne skal ved hjelp av KJ diskutere felles problemstillinger. Alle grupper skal gjennomføre to sesjoner med idegenerering, en distribuert og en lokal. Rekkefølgen defineres slik at vi får et balansert design. Første gjennomføring bestemmes ved loddrekning. Rekkefølgen for Giant Leap Technologies ble trukket og som

tabell 3.2 viser, starter GLT med lokal sesjon (1) og gjennomfører deretter distribuert sesjon (2).

Tabell 3.2: Oppsett av eksperimentet

Gruppe	Distribuert	Lokal
GLT	2	1
SINTEF	1	2
Studenter	2	1
Diplomstudenter	1	2
Studenter	2	1

### 3.7.1 Kjøreplan

Deltakerne samles først til en introduksjon av eksperimentet. Her presenteres eksperimentet og deltakerne får vite hva de skal være med på. KJ metoden presenteres raskt og deltakerne får deretter prøve verktøyet KjApp som skal benyttes under distribuert sesjon. Førstehåndserfaring med KjApp tror vi vil redusere problemer under distribuert sesjon. Deltakerne lærer å bruke KjApp og hva de kan forvente av prototypen. Uten introduksjonen vet ikke deltakerne hva de kan forvente av verktøyet og frustrasjon over KjApp kan føre til at idegenereringen vil lide. Før selve eksperimentet starter, blir deltakerne enige om problemstillinger som skal diskuteres under distribuert og lokal sesjon. Både distribuert og lokal sesjon styres av en moderator. Moderator passer på at alle deltar aktivt og at tidsplanen følges. Moderator skal også få deltakerne *på gli* dersom sesjonen flyter dårlig. Tabell 3.3 viser tentativ tidsplan for eksperimentet.

Tabell 3.3: Tidsplan for gjennomføring av eksperimentet

15 minutter	Introduksjon til eksperimentet
15 minutter	Demonstrasjon av støtteverktøy KjApp
30 minutter	Gjennomføring av første KJ sesjon
15 minutter	Besvarelse av spørreskjema og pause
30 minutter	Gjennomføring av andre KJ sesjon
15 minutter	Besvarelse av spørreskjema
15 minutter	Debrief

Til distribuerte sesjoner stilte IDI med fire kontorer. Deltakerne plasseres på hvert sitt kontor og idegenerering skjer ved hjelp av KjApp, NetMeeting og telefonkonferanse. Deltakerne får først fem minutter til å forberede egne ideer. Forberedelsen skjer ved at deltakerne skriver ideer ned på eget ark eller ved at de tenker gjennom problemstillingen. Deretter presenteres ideer ved hjelp av KjApp og muntlig forklaring gis via

telefonkonferanse. Når alle ideer er presentert, grupperes relaterte ideer og eventuelle relasjoner mellom grupper opprettes.

Under lokal sesjon samles deltakerne i et møterom med en tavle. Også her får deltakerne fem minutter til å forberede egne ideer og skriver disse på post-it lapper. Deretter presenteres ideer ved at lapper henges opp på tavlen og en muntlig forklaring gis til de andre deltakerne. Når alle ideer er presentert skal relaterte ideer grupperes og eventuelle relasjoner mellom grupper opprettes.

### 3.7.2 Problemstilling

Deltakerne i eksperimentet skal ha et felles ståsted. Med dette mener vi at de enten jobber med samme oppgave eller relaterte oppgaver. Hensikten er at det skal diskuteres problemstillinger under KJ sesjonen som gir noe tilbake til deltakerne. Deltakergruppene er satt sammen slik at de har et felles ståsted. Den første gruppen består av ansatte i Giant Leap Technologies (GLT) som arbeider med utvikling av programvare for håndholdte datamaskiner. Gruppen fra SINTEF jobber med prosessforbedring i industrien. Vi har to grupper med studenter som jobber med et fellesprosjekt. Til slutt har vi en gruppe med diplomstudenter som arbeider med oppgaver innen Extreme Programming (XP).

Deltakergruppene velger selv problemstillinger de ønsker å diskutere. Vi mener dette fører til større engasjement og en mer livlig diskusjon enn dersom de skulle diskutere problemer som vi presenterer. Ulempen med å la gruppene selv velge problemstilling er at det er vanskeligere å sammenligne resultater fra forskjellige grupper. Deltakerne velger to problemstillinger, en for distribuert og en for lokal sesjon.

### 3.7.3 Teknisk oppsett

Før eksperimentet starter forberedes kontorer og møterom. På kontorene logger vi på datamaskinen med eget brukernavn og stater deretter NetMeeting. Konferansen i NetMeeting kobles opp fra moderators datamaskin mot de andre deltakerne og oppkoblingen skjer ved hjelp av IP adresser. KjApp deles deretter ved hjelp av NetMeeting og det kontrolleres at alle datamaskiner i sesjonen har et godt bilde av verktøyet. Tilslutt kobles det opp telefonkonferanse mellom kontorene og lydforbindelsen mellom deltakerne kontrolleres. Kontorene må utstyres med skrivesaker slik at deltakerne kan forberede egne ideer. Møterommet for lokale sesjoner forberedes ved at bord og stoler grupperes slik at deltakerne kan se hverandre og tavlen. Det er viktig med god plass foran tavlen i forbindelse med presentasjon og gruppering av ideer. I tillegg må skrivesaker være tilgjengelig. Pennene skal kunne benyttes på tavlen.

### 3.7.4 Datainnsamling

Moderator noterer starttidspunktet når deltakerne er på plass og klare til å forberede ideer. Sluttidspunktet noteres når deltakerne sier seg ferdig med å generere og gruppere ideer eller når maksimaltiden på førtifem minutter er overskredet. For distribuert sesjon skal moderator med jevne mellomrom lagre tavlen som deltakerne produserer. Ideer som produseres skrives ned på eget ark for å sikre at ingen data går tapt ved en eventuell systemkræsj. Antall post-it lapper og grupper generert telles opp etter endt sesjon.

**Spørreskjema**

Etter distribuert og lokal sesjon fyller deltakerne ut hvert sitt spørreskjema. Tavlen med ideer er tilgjengelig under utfyllingen da enkelte spørsmål er relatert til denne. Spørreskjemaet kobles til deltaker med et identitetsnummer. På denne måten kan svarene fra distribuert og lokal sesjon sammenlignes og eventuelle avvik analyseres.

**Debrief**

Eksperimentet avsluttes med en uformell samtale med deltakerne. Samtalen har som mål å få frem deltakernes opplevelse av eksperimentet ut over det spørreskjemaet fanger opp. Deltakerne står fritt til å kommentere. Vi har også muligheten til å spørre deltakerne om momenter som vi har notert oss under gjennomføringen.

# Kapittel 4

## Resultater fra eksperimentet

### Innhold

---

4.1 Gjennomføring . . . . .	25
4.2 Observasjoner . . . . .	30
4.3 Beskrivende analyse . . . . .	32

---

Kapittelet beskriver datamaterialet som deltakerne i eksperimentet har produsert. Avsnitt 4.1 presenterer de ulike gjennomføringene av eksperimentet og deltakernes kommentarer til disse. I avsnitt 4.2 kommer våre egne kommentarer til gjennomføringen av eksperimentet, og til slutt i avsnitt 4.3 presenteres en beskrivelse av deltakernes svar på spørreskjemaene. Fullstendig datamateriale finnes i tillegg B.

### 4.1 Gjennomføring

Avsnittet beskriver hver enkelt gjennomføring av eksperimentet. Gruppen i hver gjennomføring beskrives først sammen med problemstillingene som skal diskuteres. Deretter presenteres kommentarene fra deltakerne.

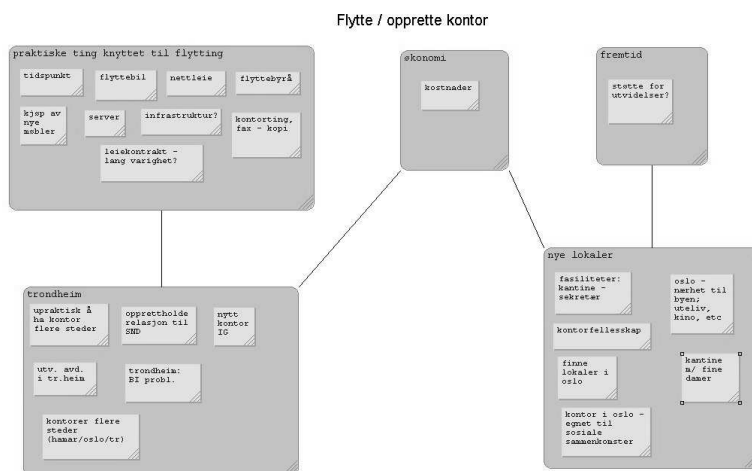
#### 4.1.1 Giant Leap Technologies

GLT består av diplomstudenter fra Institutt for Datateknikk og Informasjonsvitenskap (IDI) og Institutt for Industriell Økonomi og Teknologiledelse (IØT). GLT utvikler programvare for håndholdte datamaskiner. GLT deltok med tre personer i eksperimentet.

Deltakerne mente at KjApp burde gi støtte for et lokalt datalager for egne ideer. I datalageret kunne egne ideer plasseres før de presenteres for de andre i sesjonen. Ideer i datalagert skal bare være synlig for eieren. Fordi et slik datalager manglet, forberedte deltakerne seg ved å skrive ned ideene et annet sted, for eksempel på egne ark. Under presentasjon ble ideene skrevet inn på nytt i KjApp. Dette kan være noe årsaken til at deltakerne brukte lenger tid på distribuert enn på lokal gjennomføring.

En av deltakerne påpekte at han fikk bedre tid til å tenke på nye ideer under distribuert gjennomføring, og anså dette som positivt. Ved lokal gjennomføring var alle ideer

skrevet på forhånd før de skulle presenteres. Dette førte til at deltakere med like eller nesten like ideer allikevel limte ideen på tavlen med en kommentar som *jeg hadde også tenkt på dette momentet*. Lokal presentasjon foregikk i høyt tempo, og det var liten tid til refleksjon før man skulle presentere en ny ide. Den distribuerte sesjonen fungerte anderledes. Ideer ble skrevet inn *på direkten* og presentert. Dersom to deltakere hadde samme ide, ble ideen kun skrevet en gang. De andre deltakerne kom med muntlige kommentarer og egne tilleggsopplysninger. Dette kan være en årsak til at det ble generert færre post-it lapper under distribuert sesjon. Deltakerne fortalte i tillegg at de var



Figur 4.1: Lokal sesjon med GLT

mer kritiske til formulering av egne ideer under distribuert sesjon. De forsøkte å gjøre lappene selvforklarende, brukte flere ord og færre forkortelser.

Stemmekontakt var viktig for deltakerne. Uten dette ville distribuert gjennomføring blitt adskillig mindre effektiv. Telefonkonferanse er et hjelpemiddel å benytte når ideer skal diskuteres med personer lokalisert forskjellige steder. GLT mente at telefonkonferanser vil bli adskillig mer produktive og effektive ved bruk av spesialtilpasset verktøy som KjApp.

Deltakerne fra GLT mente at det var en fordel å se hvem som har satt opp de forskjellige lappene. Dette gjør det enklere å tolke og forstå betydningen av lapper. I den lokale sesjonen hadde deltakerne post-it lapper med forskjellige farger og det var derfor lett å bestemme hvem som hadde satt opp hvilke. Deltakerne var svært fortrolige med hverandre og dette var en stor fordel både for lokal og distribuert gjennomføring.

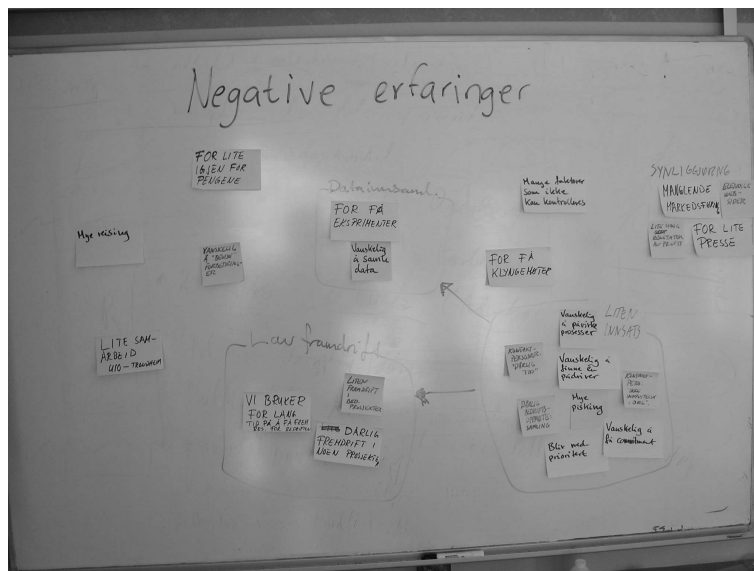
Deltakerne fra GLT mente at KJ metoden fungerer best med alle i samme rom. Faktorer som reisekostnader og tid kan allikevel gjøre lokal sesjon til et dårligere alternativ enn distribuert sesjon. Figur 4.1 viser tavlen fra lokal sesjon.

### 4.1.2 SINTEF

SINTEF deltok med tre ansatte som blant annet arbeider med forbedringsprosjekter i industrien. Deltakerne har stor erfaring med KJ og benytter metoden regelmessig. Eksperimentet ble gjennomført ved SINTEF og deltakerne benyttet egne PCer og kontorer til distribuert sesjon.

Et moment fra deltakerne var tidsforbruket under sesjonene. Distribuert sesjon tok igjen lenger tid enn lokal og deltakerne hadde flere kommentarer til dette. Måten ideer legges inn distribuert var antagelig hovedårsaken. Dersom KjApp hadde gitt støtte for et lokalt datalager hvor den enkelte kan forberede egne ideer, for så å dra disse over til fellesområde ved presentasjon, ville sesjonen gått raskere. Videre var det et irritasjonsmoment at KjApp manglet støttet for enkelte tastekombinasjoner som deltakerne kjente fra PowerPoint (som for eksempel shift-museklikk for å markere flere post-it lapper).

Det ble i tillegg påpekt at det kom flere *sløivkommentarer* under distribuert gjennomføring. En mulig årsaken kan være at det var irriterende å vente på at de andre for eksempel retter opp skrivefeil under innlegging av ideer. En annen årsak kan være at deltakerne var *tøffere i kjeften* når de snakket i telefon. I tillegg kom det frem at deltakerne gjorde andre ting under ventetiden, for eksempel lesing av e-post og toalettbesøk. Deltakerne brukte flere ord under distribuert sesjon, og dette gjorde det vanskeligere å få oversikt over alle ideer. De fleste deltakerne benyttet headset under telefonkonferansen, noe som var et godt hjelpemiddel. Under både distribuert og lokal gjennomføring *trigget* deltakere på andres ideer og kom på nye ideer. En av deltakerne ytret et ønske



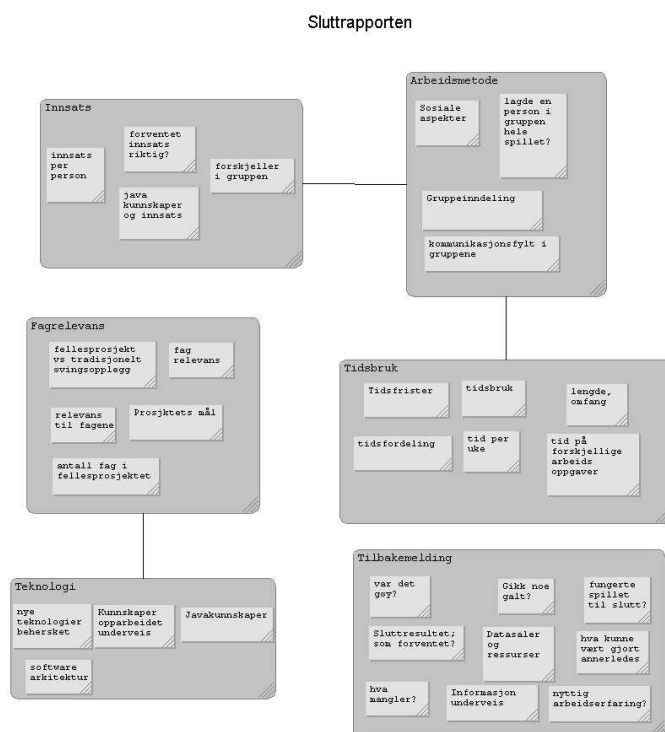
Figur 4.2: Lokal sesjon med SINTEF

om at KjApp i større grad skulle styre tidsforløpet i sesjonen. Dette kan løses ved at man definerer et sesjonsoppsett før start. For eksempel kan fem minutter settes av til forberedelse av ideer, tyve minutter til presentasjon av ideer og ti minutter til gruppering av ideer. KjApp går da automatisk over til ny modus etterhvert som tiden skrider

frem. På denne måten sikres det at sesjonen gjennomføres i sin helhet. Under sesjoner som deltakerne har vært med på i industrien har det hendt at personer må gå før man er ferdige på grunn av overskridelse av tidsskjemaet. Resultatet fra lokal sesjon er vist i figur 4.2.

### 4.1.3 Studentgruppe 1

Studentene arbeider med fellesprosjektet for fagene SIE 5003 (Kommunikasjon, tjenester og nett), SIF 8018 (Systemutvikling), SIF 8020 (Datamodellering og databasesystemer) og SIF 8040 (Menneske-maskin, interaksjon og grafikk). Fellesprosjektet består i utvikling av spillet MUGGE (MultiUser Gløshaugen Game Environment). Deltakerne hadde ingen erfaring med KJ og hverandre, men skulle løse en felles problemstilling. Deltakerne hadde forskjellige oppfatninger av sesjonene. En deltaker kommenterte at



Figur 4.3: Distribuert sesjon med datateknikk studenter

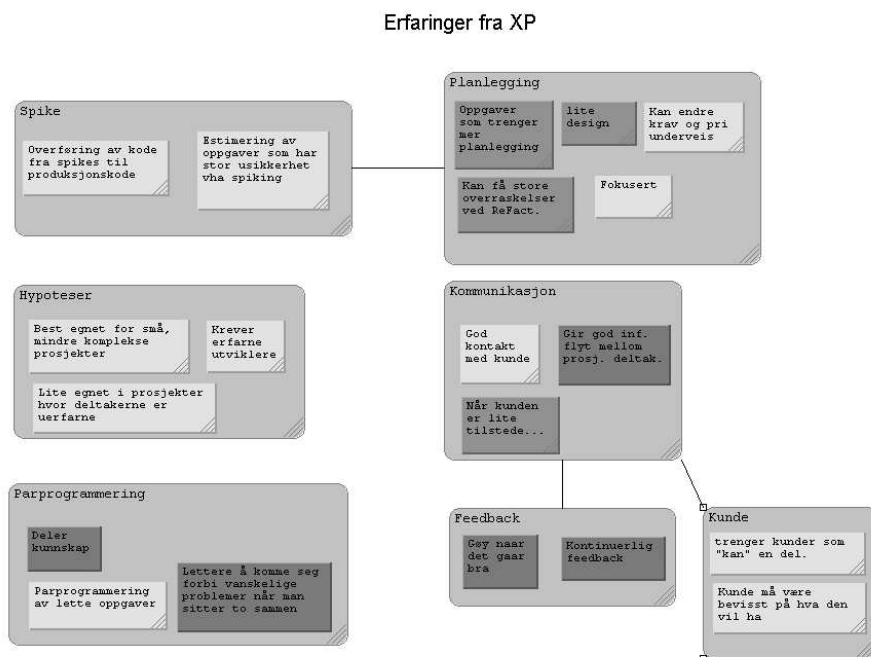
den distribuerte sesjonen fungerte på en uheldig måte. Ideer ble som forberedelse skrevet ned på et ark, og ved presentasjon ble de skrevet inn på nytt i KjApp. Problemet var at deltakeren vurderte kvaliteten på ideer når disse ble skrevet inn på nytt i KjApp. Deltakeren mente dette reduserte det kreative aspektet ved idemyldringen. Videre mente deltakeren at gruppen hadde dårlig muntlig kommunikasjonsflyt under den distribuert sesjonen og at resultatet ble best når alle befant seg i samme rom.

Manglende implementering av hurtigtaster var et irritasjonsmoment for flere deltakere. Hurtigtaster for å lage ny post-it/gruppe og for å merke flere tavleelementer var et

ønske. Deltakerne mente også måten kontroll til applikasjonen overføres fra person til person var et hinder for en effektiv prosess. Et ønske var å la kontrollen gå på rundgang automatisk. Resultatene fra distribuert sesjon vises i figuren 4.3.

#### 4.1.4 Diplomstudenter

Studentene arbeider med diplomoppgaver innen temaet Extreme Programming (XP). To av deltakerne arbeider på en felles oppgave. Deltakerne valgte å diskutere *Erferinger fra XP* og *Spiking*. Kommentarene fra diplomstudentene sammenfaller i stor grad med tidligere kommentarer. Deltakerne påpekte at distribuert sesjon hadde lavere trøkk enn lokal sesjon, og enkelte syntes at det var uvant med idemyldring via telefonkonferanse. Under distribuert sesjon var det få muntlige innspill og presentasjonen av ideer gikk tregt. Deltakerne savnet et lokalt datalager i KJApp hvor ideer kan forberedes før presentasjon, og mente at mangelen reduserte *trøkket*. Under lokal sesjon var det høyt tempo under presentasjonen av ideer og mange muntlige innspill fra de andre deltakerne. Tavlen fra distribuert sesjon vises i figur 4.4.

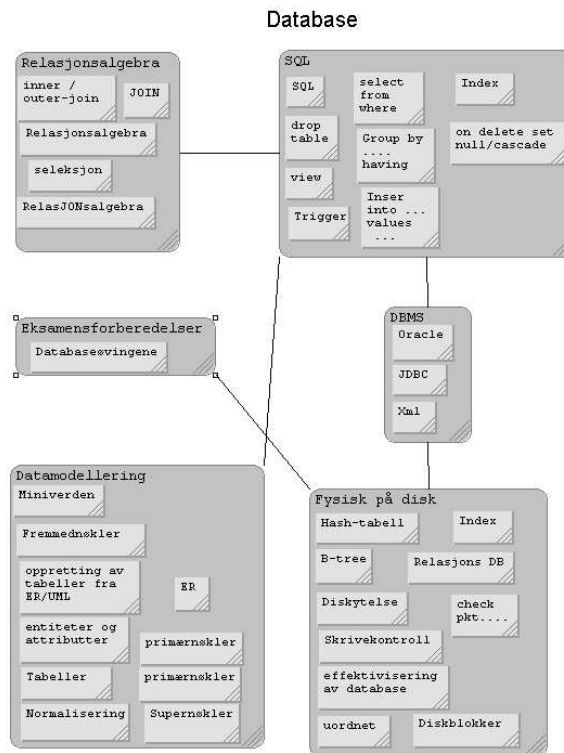


Figur 4.4: Distribuert sesjon med diplomstudenter

#### 4.1.5 Studentgruppe 2

Studentene er ferdige med fellesprosjektet for fagene SIF 8018 (Systemutvikling) og SIF 8020 (Datamodellering og databasesystemer). Deltakerne hadde ingen erfaring med KJ og har ikke jobbet sammen som gruppe tidligere.

En deltaker mente at det var lettere å komme med nye ideer under distribuert sesjon dersom man *trigges* av de andre deltakernes ideer. Under lokal sesjon var det et tiltak å sette seg ned å skrive en lapp dersom man kom på en ny ide. Deltakerne syntes det var spennende å være med i eksperimentet og bli kjent med KJ metoden. De mente også at bruken av verktøyet vil bli adskillig bedre med litt trening. Dette gjelder også for Net-Meeting. Deltakerne mente at KjApp var bra for å strukturere ideer, men at det til tider kunne gå tregt. Deltakerne savnet *samtidig arbeid* som under lokal sesjon. Et positivt moment var at resultatet på distribuert sesjon var *ferdig* når sesjonen var over. For lokal sesjon kreves det mer etterarbeid for å kunne nyttiggjøre seg resultatene. Deltakerne mente også at det var greit å få tenkt seg litt om mens de andre presenterte ideer under distribuert sesjon. tavlen fra den distribuerte sesjonen er vist i figur 4.5.



Figur 4.5: Distribuert sesjon med datateknikk og industriell økonomi studenter

## 4.2 Observasjoner

Observasjon av deltakerne under eksperimentet ga nyttig tilleggsinformasjon, informasjon som hverken ble fanget opp av spørreskjema eller debrief. Ulike sesjoner og grupper fungerte på svært forskjellige måter og vi fikk etterhvert ideer om hvorfor forskjellene oppsto og hvilken innvirkning dette har på resultatene fra KJ sesjonene. Ideene kan under statistisk analyse fungere som rettesnor for hva vi kan se nærmere på. Det er viktig å merke seg at observasjonene ikke baserer seg på svarene fra

spørreskjemaene. Observasjonene som presenteres i dette avsnittet gir mer et bilde av *magefølelsen* vi hadde under gjennomføringen.

De to første gruppene som deltok i eksperimentet var GLT og SINTEF. Gruppene skilte seg ut fra de andre gruppene, *studentgruppene*, på flere måter. Deltakerne hadde profesjonell innstilling til eksperimentet og viste stort engasjement. Problemstillingene som ble diskutert var aktuelle og relevante i forhold til deltakernes daglige gjøremål. Gruppene som helhet var tjent med at alle bidro best mulig og at man kom frem til en god løsning. Både lokal og distribuert sesjon hadde fin flyt og deltakerne taklet KjApp, Net-Meeting og telefonkonferanse på en god måte. Det virket også som at GLT og SINTEF var mer fornøyd med resultatene enn de andre gruppene. *Studentgruppene* var mer eller mindre tilfelling sammensatt for anledningen. Etter vår mening bar gruppene preg av at de ikke jobbet mot et felles mål og var mindre engasjert enn GLT og SINTEF. Det virket som at det spilte mindre rolle for dem om resultatet ble godt eller dårlig. I forhold til GLT og SINTEF diskuterte gruppene *lekeproblemer* og dette kan være noe av årsaken til mindre engasjement.

Under distribuert gjennomføring følte vi at deltakerne var usikre i begynnelsen. De pratet mindre fritt enn under lokal sesjon, og det ble ikke gitt like mange kommentarer og innspill på ideer som ble presentert. For GLT og SINTEF tok det imidlertid ikke lang tid før deltakerne var fortrolige med distribuerte omgivelser. På slutten av sesjon fløt samtalen fint og deltakerne kom med kommentarer og ros til presenterte ideer. Deltakerne fra SINTEF benytter KJ metoden regelmessig og var svært drevne. SINTEF gruppen klaget imidlertid over at det tar ekstra tid å overføring kontroll over KjApp og at KjApp ikke gir støtte for flere samtidige brukere. Allikevel virket det på oss som at dette ikke skapte store problemer og at resultatene fra sesjonen var gode. Hos *studentgruppene* forsvant ikke usikkerheten og famlingen som hos GLT og SINTEF, men vedvarte gjennom hele den distribuerte sesjonen. Samtalen fløt dårlig og det virket som at deltakerne hadde mer fokus på de distribuerte omgivelsene fremfor å diskutere problemet. Selv om *studentgruppene* på debriefingen sa at de var fornøyd med distribuert sesjon, virket det som at samarbeidsteknologiene var et større hinder her enn for GLT og SINTEF. Mindre erfaring med denne typen arbeid hos deltakerne i *studentgruppene* kan være en årsak til dette.

Et gjennomgående trekk for gruppene var at det ble benyttet flere ord og færre forkortelser på post-it lappene under distribuert sesjon. På oss virket det som at deltakerne forsøkte å gjøre post-it lappene mer selvforklarende enn for lokal sesjon. Kanskje var deltakerne usikre på hvor godt de ville få formidlet ideen muntlig ved hjelp av telefonkonferanse? Under lokal sesjon tror vi deltaker baserte seg på å forklare ideen muntlig og mange nøyde seg med å beskrive ideen ved hjelp av forkortelser. En av deltakerne fra SINTEF mente at den distribuerte sesjonen ga dårligere oversikt over ideene enn lokal. Dette må den automatiske orddelingen i KjApp ta noe av skylden for, da lange ord deles med bindestrek. Oversikten i KjApp kan imidlertid bedres ved å endre størrelsen på post-it lappene og dermed slippe orddeling.

Den ene gruppen med andreklasser studenter var et spesielt tilfelle for eksperimentet. Under lokal sesjon fikk vi inntrykk av at *en* deltaker dominerte gruppen. Den *dominerende* deltakeren hadde et stort antall og mangfold av ideer og tok på mange måter kontrollen over sesjonen. Deltakeren mente at lokal sesjon hadde *trøkk* og vurderte eget bidrag som veldig bra. En annen deltaker hadde få ideer i forhold til den *dominerende*

og følte at eget bidrag ikke var så bra. Under distribuerte sesjonen kom denne deltakeren bedre frem og bidro på lik linje med de andre. Her mislikte den *dominerende* at hun så sjelden hadde kontroll over KjApp og hun hadde heller ikke noe ønske om å være anonym. Deltakeren som ble *overkjørt* sa at han fortrakk en viss anonymitet. Etter vår mening viser dette at distribuert sesjon oppleves forskjellig ut i fra hva slags personlighet deltakeren har. Distribuerte omgivelser kan dermed være et virkemiddel for å fjerne avstanden mellom *dominerende* og *undertrykte* deltakere.

Deltakerne fra SINTEF var de eneste som benyttet egne kontorer under distribuert sesjon. I den forbindelse oppdaget vi et faremoment som vi ikke hadde vurdert tidligere. Dette besto i at deltakerne benyttet *ventetiden* til blant annet å lese e-post. Faren med dette er at fokus tas vekk fra problemet som diskuteres, noe som ikke oppstår på samme måte under lokal sesjon.

### 4.3 Beskrivende analyse

Som en forberedelse til statistisk analyse, beskrives resultatene tekstlig og grafisk. Hensikten er å få oversikt over datamaterialet og avdekke eventuelle trender. Datamaterialet består av besvarelser på spørreskjema gitt til deltakerne under eksperimentet. Spørreskjemaene er utarbeidet i henhold til GQM beskrevet i avsnitt 3.4 for distribuert og lokal sesjon. I tillegg har deltakerne svart på et skjema med *personlige* opplysninger. Under eksperimentet registrerte vi hvor lang tid sesjonene tok og hvor mange ideer og grupper som ble generert.

Gjennomføringen av eksperimentene avdekket momenter vi ikke tidligere hadde vurdert. Vi valgte derfor å lage oppfølgingsspørsmål, og disse ble besvart via et web-grensesnitt. Spørreskjemaer som er benyttet i eksperimentet finnes i tillegg A. Grafisk fremstilling av svarfordeling på alle spørsmål er vist i tillegg C.

#### 4.3.1 Personlige opplysninger

Fem grupper á tre personer deltok i eksperimentet og deltakerne hadde en gjennomsnittsalder på tjuetvå år. Tjue prosent av deltakerne var kvinner. Av femten deltakere var det seks andreårsstudenter og seks diplomstudenter ved sivilingeniørstudiet NTNU og tre yrkesaktive fra SINTEF. Det er viktig å merke seg at den ene gruppen med diplomstudenter utgjør GLT. Problemene som gruppen diskuterte var relatert til GLTs virksomhet og ikke diplomoppgaver.

Vi mener fortrolighet er en viktig faktor i forbindelse med idegenerering og spurte derfor deltakerne om hvor fortrolige de var med de andre i gruppen. Svarene viser at deltakerne er fra middels til veldig fortrolige med hverandre. Svarene på dette spørsmålet gjør det mulig å finne sammenhenger til andre variabler i eksperimentet. Kanskje finnes det en sammenheng mellom hvor fortrolige deltakerne er og hvor godt resultatet ble?

#### 4.3.2 Distribuert og lokal sesjon

Spørreskjemaene for distribuert og lokal sesjon opererer med en skala fra en til fem. For alle spørsmål gjelder det at alternativ en er mest negativt og alternativ fem mest positivt. Spørreskjemaet ble besvart rett etter at sesjonen var ferdig og mens deltakerne

hadde inntrykkene friskt i minnet. Svarfordelingene gir en indikasjon på hvor godt eller dårlig sesjonene fungerte. Grafisk fremstilling av svarfordelingen finnes i tillegg C.1.

#### Var det morsomt

Svarfordelingen for distribuert og lokal sesjon er ganske lik og dette tyder på at deltakerne opplever sesjonene på omtrent samme måte med tanke på hvor morsom den er. Tyngdepunktet ligger for begge sesjoner på *Morsomt* (3), med mellom seksti og sytti prosent av svarene. Et poeng å merke seg, er at for distribuert sesjon svarte en deltaker *Ikke morsom* (1). Dette alternativet er ikke benyttet for lokal sesjon. Lik fordelingen for sesjonene tyder på at det ikke er faktoren distribuert eller lokal gjennomføring som avgjør om det er morsomt eller ikke.

#### Følte du at du bidro

Deltakerne følte i større grad at de bidro under lokal sesjon enn under distribuert. For lokal sesjon svarte de fleste at de *Bidro mye* (4), mens tyngdepunktet for distribuert sesjon var (3). Dette tyder på at det er litt vanskeligere å komme til med bidrag under distribuert sesjon. Kanskje er distribuerte omgivelser som NetMeeting, KjApp og telefonkonferanse et hinder for kommunikasjonen?

#### Hvordan fungerte gruppen

Gruppene fungerte bra både under distribuert og lokal sesjon, med lokal sesjon et lite hakk bedre. For distribuert sesjon svarte over nitti prosent at gruppen fungerte fra *Middels* (3) til *Veldig bra* (5). Lokal sesjon har alle svarene i dette området. Svarfordelingen er ganske lik for begge sesjonen og kanskje betyr det at det ikke er distribuert eller lokal gjennomføring som avgjør hvor godt en gruppe fungerer.

#### Følte du at sesjonen hadde trøkk

På spørsmålet om *trøkk* har lokal sesjon klart de beste resultatene, med alle svarene i området *Middels* (3) til *Veldig bra trøkk* (5). For distribuert sesjon er spredningen større og svarene varierer mellom *Ingen trøkk* (1) til *Over middels trøkk* (4). Svarfordelingen er som forventet og vi mener at deltakere i samme rom lettere påvirker hverandre til økt innsats. I distribuerte omgivelser reduseres antagelig intensiteten på grunn av begrenset samtidighet og mangel på øyekontakt.

#### I hvor stor grad tror du at du kan benytte resultatet

Det var stor uenighet hos deltakerne om i hvor stor grad resultatet kan benyttes. Distribuerte sesjoner har et noe bedre resultat enn lokal, med tyngdepunktet på *Kan benytte det meste* (4). Svarene fra lokal sesjon er jevnt fordelt mellom *Kan benytte noen få deler* (2), *Kan benytte mange deler* (3) og (4). Stor spredning på spørsmålet mener vi er naturlig. Gruppene valgte selv hvilket problem som skulle diskuteres, men hvor mye av resultatet som er anvendbart vil variere blant deltakerne i gruppen. Dette fordi deltakerne har ulik erfaring og kompetanse innen problemet som diskuteres.

### **Kom det frem nye momenter/ideer**

Svarene viser stor spredning på hvor mange nye momenter/ideer deltakerne mente ble generert under sesjonene. Distribuert og lokal sesjon følger hverandre bra og begge har tyngdepunktet på *En del momenter/ideer* (3). En forklaring på spredning kan være at deltakerne har ulik kompetanse/erfaring innen diskutert problemområde. En deltaker som bidrar med mange ideer vil antagelig svare at det ikke kom frem så mange nye momenter i forhold til det deltakeren visste fra før.

### **Kunne du kommet frem til de samme resultatene på egenhånd**

Svarene viser at deltakerne er uenige om de kunne kommet frem til resultatet alene. Spredningen er stor for både distribuerte og lokale sesjoner. En forklaring kan være ulik kompetanse og innsikt i diskutert problem hos deltakerne. En person med god innsikt i problemet vil antagelig mene at han kunne kommet frem til resultatet på egenhånd. Distribuert og lokal sesjoner følger hverandre godt.

### **Setter du pris på å være *anonym* under sesjonen**

Spørsmålet ble stilt deltakerne i forbindelse med distribuert sesjon. Svarene er jevnt fordelt mellom *Ikke i det hele tatt* (1), *Under middels* (2) og *Middels* (3). Det vi kan merke oss er at ingen deltakere benytter alternativene (4) og (5). Deltakerne i eksperimentet har med andre ord ikke noe sterkt ønske om å være anonyme under en KJ sesjon.

### **Hvordan fungerte verktøyene KjApp og NetMeeting**

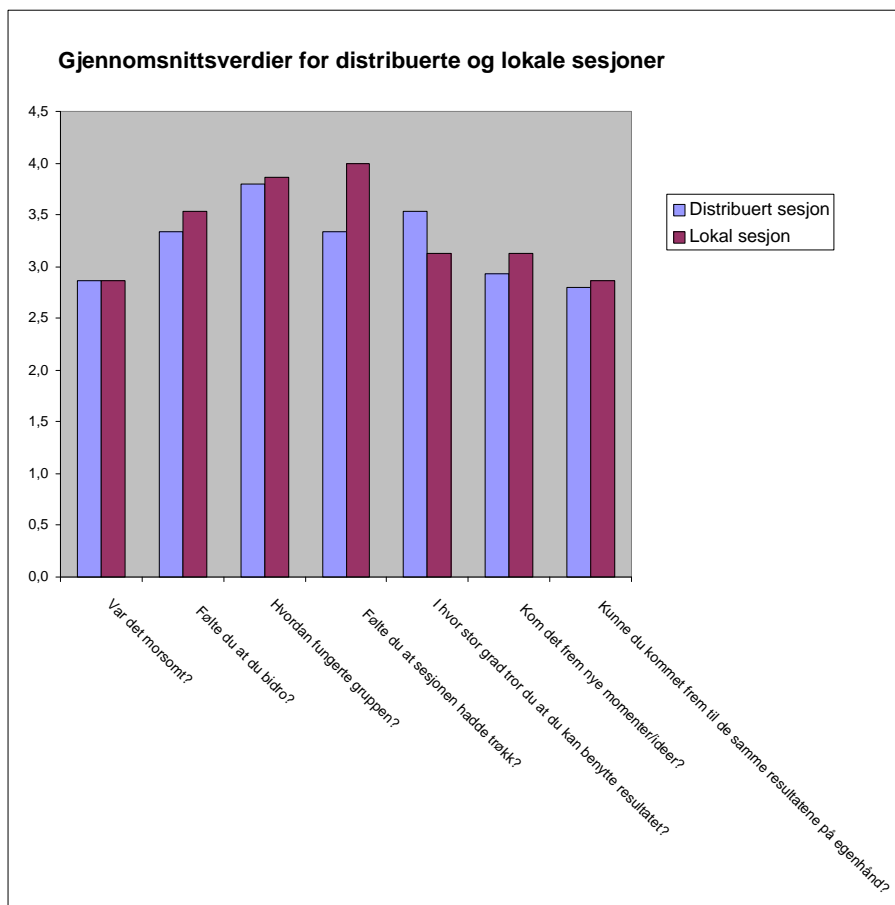
Svarene på spørsmålet viser at deltakerne syntes KjApp og NetMeeting fungerte fra *Middels* (3) til *Veldig bra* (5). Resultatet er positivt for eksperimentet og viser at støtteverktøyene har fungert tilfredsstillende. Spørsmålet er nyansert i oppfølgingsspørsmålene, ved at vi skiller KjApp og NetMeeting fra hverandre og i tillegg spør om hvordan telefonkonferansen fungerte.

### **Gjennomsnittsverdier for distribuert og lokal gjennomføring**

Figur 4.6 viser at gjennomsnittsverdiene for distribuert og lokal sesjon følger hverandre godt i de fleste tilfeller. Det største avviket finnes for spørsmålet om sesjonen hadde trøkk. Distribuert sesjon har et gjennomsnitt på 3,3, mens lokal sesjon har 4,0. Det er også en klar forskjell på spørsmålet om i hvor stor grad resultatet kan benyttes. Distribuert sesjon har her det beste resultatet med et snitt på 3,5, mens lokal sesjon ligger på 3,1.

### **4.3.3 Tidsbruk, antall ideer og grupper**

Under eksperimentet registrerte vi hvor lang tid sesjonene tok og hvor mange post-it lapper og grupper som ble generert. Resultatene presenteres i dette avsnittet, mens grafisk fremstilling av svarfordelingen kan studeres i tillegg C.2.



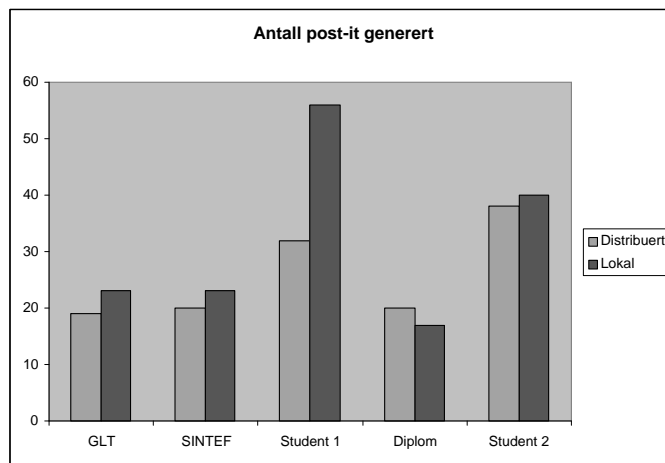
Figur 4.6: Gjennomsnittsverdier for distribuerte og lokale sesjoner

### Tidsbruk

Distribuerte sesjoner tok adskillig lenger tid enn lokale og forholdet er nesten 2:1. For distribuerte sesjoner registrerte vi et gjennomsnitt på trettini minutter, mens lokale sesjoner i gjennomsnitt varte i tjue minutter. Det er flere årsaker til dette. Deltakerne i eksperimentet hadde ingen erfaring med KjApp og enkelte ganger førte dette til noe tungvindt bruk. Videre mangler KjApp støtte for samtidighet og kun en deltaker kan jobbe med applikasjonen ad gangen. I tillegg tar distribuert sesjon lenger tid fordi ideer må skrives inn *på direkten*. Overføring av kontroll i NetMeeting var også et hinder for god flyt og førte til ekstra tidsbruk. En fullstendig distribuert løsning med et lokalt data-lager hvor ideer forberedes vil redusere tidsbruket for distribuerte sesjoner betraktelig. Og kanskje vil distribuerte og lokale sesjoner da ta like lang tid?

### Antall post-it lapper generert

Antall post-it lapper generert varierte noe for de ulike gjennomføringene. Figur 4.7 viser at de to første sesjoner var like, med noen flere post-it lapper generert lokalt. Tredje gjennomføring var et spesielt tilfelle lokalt med nesten seksti lapper generert. Her var det spesielt en deltaker som hadde mange ideer, og under distribuert sesjon ble tiden et problem. Deltakerne valgte derfor å forkaste mange ideer fordi det tok for å lang tid å presentere dem. Fjerde gjennomføring genererte flere lapper distribuert, noe som kun var tilfelle for denne sesjonen. Sesjon fem ligner på sesjon en og to, men med flere lapper generert. I gjennomsnitt ble det generert tjuetruer lapper distribuert og trettifire lapper lokalt. Vi mener at ulikt abstraksjonsnivå i forbindelse med generering

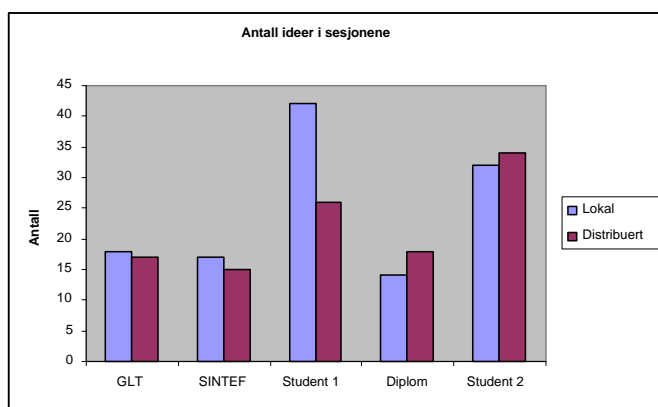


Figur 4.7: Post-it lapper generert i sesjonene

av ideer kan være noe av årsaken til at antallet varierer. Gjennomføring tre og fem besto av andreklassestudenter ved NTNU, og etter vår mening arbeidet de med ideer på et lavere abstraksjonsnivå i forhold til de andre gruppene. Ideene til disse to gruppene var mer detaljert og dette førte til at antallet økte.

### Antall ideer generert

Etter gjennomføringen av sesjonene spurte vi deltakerne om hvor mange enkeltstående ideer de mente ble generert under sesjonen. Hensikten med spørsmålet var å fjerne post-it lapper som beskrev like eller neste like ideer. Etter vår mening vil dette tallet gi et bedre svar på hvor god sesjonen var i forhold til antall post-it lapper generert. Imidlertid var det tydelig at deltakerne tolket spørsmålet forskjellig og resultatene varierte voldsomt. For samme sesjon mente en deltaker at det ble generert atten enkeltstående ideer, mens en annen deltaker mente at det ble generert tre. Vi valgte derfor å se bort fra svarene fra deltakerne og heller telle antall ideer selv. Dette gjorde vi i samarbeid med veilederen vår. Resultatene i figur 4.8 viser at det ikke er så stor forskjell mellom distribuerte og lokale sesjoner, bortsett fra tredje gjennomføring. Antall ideer er noe færre enn antall post-it lapper, men resultatene følger hverandre godt. Forklaringen gitt for antall post-it lapper generert mener vi også gjelder for antall ideer generert.



Figur 4.8: Antall ideer generert i sesjonene

### Antall grupper generert

Resultatene viser at det ble generert noen flere grupper for distribuerte enn for lokale sesjoner. Distribuerte sesjoner genererte mellom fem og seks grupper med et gjennomsnitt på 5,7. Under lokale sesjoner ble det generert mellom fire og syv grupper, med et gjennomsnitt på 4,3. Vårt inntrykk er at deltakerne var mer opptatt av å gruppere alle lappene under distribuert sesjon enn under lokal. Under distribuert sesjon var det sjelden at lapper ikke tilhørte noen gruppe. Dette skjedde oftere for lokale sesjoner. Vårt inntrykk er at deltakerne er mer formelle og *nøye* under distribuert sesjon og ønsker ikke at resultatet skal stå igjen halvferdig.

#### 4.3.4 Oppfølgingsspørsmål

Under gjennomføringen av eksperimentet dukket det opp momenter vi ikke hadde vurdert tidligere. Deltakerne kom med flere gode kommentarer til gjennomføring av sesjonene, og det var interessant å få bekreftet eller avkreftet om det var enighet om kommentarene. Kommentarene dreide seg om idegenerering, flyt og trøkk og ønsket

tilleggsfunksjonalitet i KjApp. Vi lagde derfor oppfølgingsspørsmål til deltakerne som ble besvart via et web-grensesnitt. Resultatene presenteres i dette avsnittet, mens grafisk fremstilling av svarfordelingen kan studeres i tillegg C.3.

### **Idegenerering**

Avsnittet tar for seg deltakernes kommentarer på hvordan idegenereringen fungerte for de ulike sesjonene og hvordan kvaliteten på resultatet var.

### **Vurderte du ideene dine på nytt når du skrev de inn i KjApp under distribuert sesjon**

En student mente at hun vurderte sine ideer på nytt ved innleggelse i KjApp og at dette hadde negativ innvirkning på idegenereringen. Vi spurte derfor de andre deltakerne om de følte det på samme måte. Deltakerne var uenige på dette punktet og svarene fordelte seg jevn mellom *Vurderte ingen ideer på nytt* (1) til *Vurderte alle ideene* (4). Tyngdepunktet lå på alternativ (2) og (3).

### **Hendte det at du under distribuert sesjon forkastet ideer som var forberedt når ideen skulle presenteres**

Under distribuert sesjon forberedte deltakerne ideer ved å skrive dem ned på et eget ark. Ved presentasjon ble ideene skrevet inn på nytt i KjApp. En kommentar vi fikk, gikk ut på at enkelte ideer ble forkastet ved presentasjon. Årsaken til at ideer ble forkastet kunne være at en lignende ide allerede var presentert eller at deltakeren mente at egen ide ikke var god nok. Vi stilte derfor deltakerne spørsmål om de forkastet ideer under distribuert sesjon. Svarene er fordelt på alle alternativene, men med tyngdepunktet på (2), *Ja, forkastet noen* (1-3), som er benyttet av litt over femti prosent av deltakerne. Tar vi i tillegg med (3), *Ja, forkastet mange* (4 eller flere), har omkring seksti prosent av deltakerne forkastet ideer. Alternativ (1), *Nei, forkastet ingen*, ble benyttet av omtrent tretti prosent av deltakerne.

Resultatene viser at flere deltakere har forkastet ideer som var forberedt. I forbindelse med idegenerering er det en gylden regel som sier at det ikke finnes dårlige ideer. Det er derfor negativt at ideer forkastes fordi de vurderes som for dårlige. Men dersom årsaken til at ideer forkastes er at ideen allerede er presentert, vil ikke sesjonen lide noe tap.

### **Var du mer opptatt av at lappene under distribuert sesjon skulle være selvforklarende**

Eksperimentet avdekket en forskjell i antall ord og bokstaver på lappene for distribuert og lokal sesjon. Distribuert sesjon hadde gjennomgående flere ord, flere bokstaver og færre forkortelser enn lokal sesjon. Vi ønsket å spørre deltakerne om dette var et bevisst forsøk på å gjøre lappene for distribuert sesjon selvforklarende. Resultatene viser at flertallet av deltakerne ikke mente det var noen forskjell på distribuerte og lokale sesjoner med tanke på dette. Omkring tretti prosent av deltakerne mente at det var en forskjell, men at den var liten. Bare ti prosent av deltakerne mente at forskjellen var stor.

Resultatet tyder på at det ikke har vært et bevisst valg hos deltakerne å gjøre lappene mer selvforklarende. En forklaring kan være at deltakerne blir mer formelle når de

skal uttrykke seg skriftlig. En annen forklaring kan være at det er lettere å forklare en ide muntlig under lokal sesjon og at man derfor ikke er så nøye på hvordan ideen uttrykkes skriftlig.

#### **Tror du at du benyttet flest forkortelser under distribuert eller lokal sesjon**

Som nevnt var det en betydelig forskjell i bruken av forkortelser, der distribuert sesjon hadde færre forkortelser enn lokal. Vi ville finne ut om dette var et bevisst valg, og ba deltakerne angi hvilken sesjon de hadde benyttet flest forkortelser. Svarene viser at flertallet av deltakerne mente at det var like mange forkortelser under distribuert som lokal sesjon. Bare tjue prosent støttet vår observasjon om at det ble benyttet flere forkortelser under lokal sesjon.

Resultatet tyder på at deltakerne ikke har noe bevisst forhold til bruken av forkortelser. Spørsmålet er en presisering av forrige spørsmål og vi mener at samme forklaring gjelder.

#### **Hvilken sesjon genererte de beste ideer**

Etter gjennomføring av distribuert og lokal sesjon spurte vi deltakerne om hvordan kvaliteten på ideene var. Som et oppfølgingsspørsmål ønsker vi å sammenligne distribuert og lokal sesjon, og finne ut hvilken gjennomføring som genererte de beste ideene. Svarene er nærmest normalfordelt, med i overkant av tjuenfem prosent på *Distribuert best* (1) og *Lokal best* (3) og resten på *Like gode* (2). Resultatet tyder på at det ikke er distribuert eller lokal gjennomføring som avgjør om resultatet blir godt eller ikke, men andre faktorer i forbindelse med idegenerering.

#### **Hvilken sesjon ga best oversikt**

Med hensyn på oversiktighet over genererte ideer har det kommet frem motstridende kommentarer. En deltaker mente det var vanskelig å få oversikt over ideer i KjApp på grunn av *dårlig* tekstbryting. Andre deltakere mente at det var lettere å få oversikt i KjApp fordi man slipper å tolke håndskrift. Vi spurte derfor deltakerne om hvilken sesjon som ga best oversikt.

Som kommentarene vi fikk under eksperimentet, viser også svarfordelingen at deltakerne er uenige om dette momentet. Tyngdepunktet lå på *Distribuert ga litt bedre oversikt* (2) og *Like god oversikt* (3), som hver ble benyttet av over tretti prosent av deltakerne. *Helt klart distribuert* (1) ble benyttet av nesten femten prosent, mens *Lokal ga litt bedre oversikt* (4) ble benyttet av nesten tjue prosent. Det vi kan merke oss er at alternativ (5), *Helt klart lokal*, ikke er benyttet. Resultatet tyder på at distribuert sesjon gir en litt bedre oversikt over ideene. Kanskje er det et ryddig grensesnitt i KjApp og det at deltakerne slipper å tolke håndskrift som er årsaken?

#### **Hvordan ble resultatet påvirket av varigheten på distribuert sesjon**

Distribuert sesjon tok betydelig lengre tid enn lokal sesjon, med et forhold på 2:1. Som et oppfølgingsspørsmål ønsket vi å få deltakernes mening om hvilken innvirkning dette har på idegenereringen. Svarfordelingen viser at tyngdepunktet ligger på *Like mye positivt som negativt* (3). En liten andel på fem prosent benyttet alternativ (1), *Utelukkende*

*negativt*. Fordelingen mellom *Mest negativt* (2) og *Mest positivt* var på henholdsvis tjuefem og femten prosent.

Resultatet viser at deltakerne har ulik oppfatning av hvordan varigheten på sesjonen virker inn på resultatet. Enkelte deltakere kommenterte at de ble irriterte av å vente på at de andre deltakerne skulle skrive inn ideene i KjApp. En annen kommentar vi fikk, var at ventetiden ble opplevd som positivt, fordi dette ga deltakeren muligheten til å fordøye innholdet i ideen og muligens komme på andre relaterte ideer selv.

### **Hvordan ble resultatet påvirket av at distribuert KJ var en ny erfaring for alle**

Distribuert KJ var en ny erfaring for alle deltakerne. Vi ønsket å høre deltakernes mening om hvordan dette påvirket resultatet fra distribuert sesjon. Svarfordelingen er noe overraskende for oss. Flertallet av deltakerne har benyttet *Like mye positivt som negativt* (3) og dette er som forventet. En andel på tjuefem prosent mener at resultatet påvirkes *Mest negativt* (2) og dette er heller ikke overraskende. Men at neste tju prosent mener at resultatet påvirkes *Mest positivt* (4) overrasker oss. En forklaring kan være at nye omgivelser, altså distribuerte omgivelser, kan virke stimulerende i forhold til idegenerering. Kanskje det er slik at deltakerne i *vante* omgivelser ikke yter sitt beste? Vi forventet at siden distribuert KJ er en ny erfaring for deltakerne, så ville det oppstå en del innkjøringsproblemer som påvirket resultatet i negativ retning. Svarene tyder på at det er delte meninger om dette.

### **Hvordan tror du resultatet ville blitt dersom du hadde mer erfaring med distribuert KJ og bruk av KjApp**

Som en oppfølging til forrige spørsmål, spurte vi deltakerne om hvordan de trodde resultatet ville blitt dersom de fikk mer erfaring med distribuert KJ og bruk av KjApp. Svarfordelingen er som forventet, med flertallet av svarene på *Noe bedre* (4). Nesten tretti prosent valgte *Like godt* (3), mens omkring fem prosent mente at resultatet ville blitt *Vesentlig bedre* (5) med mer erfaring med distribuert KJ. Resultatet sammenfaller med vår antagelse om mer erfaring fører til bedre resultater. Men som svarene fra forrige spørsmål viser, kan en ny opplevelse være stimulerende i forhold til idegenerering.

### **Flyt og trøkk**

Deltakernes opplevelse av flyt og trøkk i KJ sesjonene varierte. Vårt inntrykk er at distribuerte sesjoner hadde lavere trøkk og dårligere flyt enn lokale sesjoner. For å kartlegge hvilke momenter som påvirket flyt og trøkk stilte vi deltakerne noen oppfølgings-spørsmål om dette.

### **Mener du at telefonkonferansen var et hinder for brainstormingen**

Deltakerne i eksperimentet hadde ulik erfaring med bruk av telefonkonferanse. For å finne ut hvilken innvirkning telefonkonferanse hadde på idegenereringen, spurte vi om telefonkonferansen var et hinder. Resultatet viser at nesten seksti prosent av deltakerne valgte *Nei, ikke i det hele tatt* (1) på spørsmålet. En andel på tretti prosent valgte *Ja, til en viss grad* (2) og omkring fem prosent valgte alternativ (3), *Ja, i stor grad*. Resultatet viser at telefonkonferanse kan være et problem i forbindelse med distribuert KJ. Deltakere som ikke er vant med å benytte telefonkonferanse vil antagelig prate mindre og føle at de bidrar i mindre grad enn erfarne telefonkonferansedeltakere.

### Mener du at KjApp var et hinder for brainstormingen

Som for telefonkonferanse, ønsket vi å finne ut om KjApp var et hinder for idegenerering. Resultatene viser at nitti prosent valgte *Nei, ikke i det hele tatt* (1), noe som indikerer at prototypen har fungert etter intensjonen. Dermed mener vi at vi har nådd målet om å utvikle en prototyp med et logisk brukergrensesnitt og tilstrekkelig funksjonalitet for å gjennomføre en KJ sesjon.

### Mener du at NetMeeting var et hinder for brainstormingen

Deltakerne ble også spurt om NetMeeting var et hinder for idegenereringen. Resultatet viser at omkring sytti prosent valgte *Ja, til en viss grad* (2), noe som tyder på at bruk av NetMeeting ikke er optimalt. Kommentarer etter eksperimentet ga også indikasjoner på dette, da flere deltakere klaget over at det var tungvint å overføre kontroll over KjApp mellom deltakerne. En fullstendig distribuert løsning, som beskrevet i kapittel 15 vil løse dette problemet.

### Hvor viktig mener du stemmekontakt er i en slik sesjon

Flere av deltakerne påpekte under eksperimentet at stemmekontakt var helt avgjørende for en effektiv gjennomføring av distribuert sesjon. Svarene viser at deltakerne er enige på dette området, med i overkant av nitti prosent av svarene i området *Svært viktig* (4) til *Helt vesentlig* (5). Dette bekrefter våre antagelser om viktigheten av stemmekontakt.

### Hvor viktig mener du fortrolighet er for resultatet

Deltakerne ble under eksperimentet spurt om hvor fortrolige de var med de andre deltakerne. Oppfølgingsspørsmålet skal avdekke deltakernes syn på hvor viktig fortrolighet er for et godt resultat. Resultatet viser at deltakerne har forskjellig oppfatning om viktigheten av fortrolighet. Tyngdepunktet ligger dog på alternativet *Svært viktig* (4) med omkring seksti prosent av svarene. Ingen av deltakere har benyttet alternativet *Uvesentlig* (1). Resultatet tyder på at fortrolighet er en essensiell faktor for hvor godt resultatet blir ved idegenerering.

### Mener du at du er tøffere i kjeften på telefon enn ansikt til ansikt

En kommentar under eksperimentet var at det kom flere sleivkommentarer under distribuert sesjon enn under lokal. Deltakerne mente dette skyldes irritasjon ved venting på at de andre skal legge inn ideer i KjApp. En annen forklaring kan være at deltakerne er tøffere når de er *anonyme* som ved en telefonkonferanse. Resultatet viser at det er stor spredning på svarene. Tyngdepunktet ligger på *Like tøff* (3) med omkring førti prosent av svarene. Alternativ (1), *Nei, ikke i det hele tatt* er valgt av litt over tjuefem prosent. Dette tyder på at det er store individuelle forskjeller med tanke på *tøffhet*.

### Fremtidig bruk og funksjonalitet

Under eksperimentet fikk vi kommentarer fra deltakerne om ønsket funksjonalitet i KjApp. Med spørsmålene i dette avsnittet vil vi finne ut om ønskene er gjennomgående for deltakerne i eksperimentet. Vi har i tillegg stilt et spørsmål om fremtidig bruk av distribuert KJ.

### Tidsstyrt sesjon

I forbindelse med distribuert sesjon ble det ytret et ønske om at KjApp i større grad burde styre tiden i de forskjellige fasene av idegenereringen. For eksempel kunne applikasjonen være fem minutter i modus for ideforberedelse, deretter femten minutter i modus for idepresentasjon og tilslutt ti minutter i modus for gruppering av ideer. På denne måten sikrer man at sesjonen gjennomføres innenfor gitte tidsrammer. Vi stilte deltakerne spørsmål om hvordan de tror dette vil påvirke resultatet.

Svarene viser stor uenighet blant deltakerne. Alle svaralternativene er benyttet og flest deltakere har valgt alternativet *Mest negativt* (2). Motivasjonen for å styre en sesjon er å sikre at sesjonen gjennomføres under gitte tidsrammer og svarene fra deltakerne kan tyde på at mange mener at en slik sesjon vil produsere et dårligere resultat enn en sesjon som ikke er tidsstyrt.

### Hvor viktig mener du det er å vite hvem som har satt opp de forskjellige lappene

Under lokal sesjon kan deltakerne se hvem som har satt opp de forskjellige lappene på bakgrunn av deltakers håndskrift og eventuelt ved at deltakerne benytter lapper med forskjellige farger. Enkelte deltakere mente at dette var en fordel, da man lettere kunne tolke betydningen av en post-it lapp. Vi ønsket å se om flere deltakere delte dette synet og stilte derfor dette oppfølgingsspørsmålet. Et flertall, omkring seksti prosent, svarte *Uvesentlig* (1). En deltaker har valgt *Helt nødvendig* (5) og dette er noe som overrasker oss. En forklaring kan være at deltakeren tolker spørsmålet i den retning at det er vanskelig å forstå betydningen av en post-it lapp dersom du ikke selv har sett at lappen har blitt plassert på tavlen.

### Lokalt datalager i KjApp

Et gjennomgående ønske fra flere deltakere var et lokalt datalager i KjApp hvor egne ideer kan forberedes før de presenteres for de andre ved å dra ideen ut i et fellesområde. Deltakerne ble spurt om hvor viktig de syntes et slikt lokalt datalager er. Resultatet viser at deltakerne mener at dette er viktig. Svarene ligger i området *Middels* (3) til *Helt nødvendig* (5). Dette er viktig å merke seg i forbindelse med videre arbeid med en distribuert versjon av KjApp (se kapittel 15).

### Teknisk diskusjon Trondheim-Paris-London

Vårt siste oppfølgingsspørsmål tok for seg et tenkt tilfelle i fremtiden hvor et teknisk problem skal diskuteres med deltakere fra London, Paris og Trondheim. Deltakerne ble spurt om de kunne tenke seg å ta diskusjonen ved hjelp av telefonkonferanse, NetMeeting og KjApp. Åtti prosent deltakerne valgte *Ja, dersom jeg kjente kunden godt* (2), mens tjue prosent benyttet alternativ (3), *Ja, jeg ville uansett forsøke for å redusere reisekostnader*. Ingen av deltakerne har benyttet alternativ (1), *Nei, jeg ville reist uansett*. Svarene indikerer at deltakerne i eksperimentet finner distribuert KJ hensiktsmessig og at de er villig til å ta det i bruk ved en senere anledning.

# Kapittel 5

## Statistisk analyse

### Innhold

---

5.1 Hypotesetesting . . . . .	43
5.2 Oppsummering av statistisk analyse . . . . .	48

---

Kapittelet presenterer vår analyse av datamaterialet fra eksperimentet. Analysemetoden som benyttes er beskrevet i avsnitt 3.5. I avsnitt 5.1 presenteres en ANOVA av de kvalitative resultatene, analyse av de kvantitative resultatene og ANOVA splittet på deltakernes profesjon. Kapittelet oppsummeres i avsnitt 5.2.

### 5.1 Hypotesetesting

Resultatene fra eksperimentet benyttes i dette avsnitte for å teste om nullhypotesen skal forkastes eller ikke. Vi starter med å analysere de kvalitative spørsmålene fra eksperimentet. Vi velger å benytte p-verdi lik ti prosent ved testen. Dette innebærer at nullhypotesen kan forkastes med nitti prosent sannsynlighet. Fullstendige resultater av analysen er vist i tillegg D.1.

I tillegg D.2 finnes resultater fra Kruskal-Wallis analyse av svarene. Resultatene viser at ANOVA og Kruskal-Wallis gir samme konklusjon når de benyttes på faktorene for hypotesetesting definert i avsnitt 3.2.3. Dermed har våre analyseresultater vist at paramteriske tester gir samme konklusjon som ikke-parametriske tester. Vi velger derfor å benytte ANOVA, da denne er sterkere enn Kruskal-Wallis. Avsnitt 3.5 om analyse-teknikker underbygger dette valget.

#### 5.1.1 ANOVA av kvalitative resultater

Vi har benyttet ANOVA for å analysere svarene på de kvalitative spørsmålene fra eksperimentet. ANOVA benyttes på alle spørsmål som omhandler kvalitative aspekter ved sesjonene og skiller på faktoren *Distribuert* og *Lokal*. Med dette ønsker vi å finne ut om det er noen signifikant forskjell mellom resultatene fra distribuerte og lokale sesjoner. Resultatene fra arbeidet er samlet i tabellene 5.1 og 5.2.

Resultatene i tabell 5.1 viser at det ikke er noen signifikant forskjell på deltakernes opplevelse av sesjonene for de tre første spørsmålene, da p-verdiene ligger i området 0.5 til 1.0. For spørsmålet om *trøkk* viser analysen at lokal sesjon har signifikant større trøkk enn distribuert sesjon med en p-verdi på fire prosent. Resultatet var forventet, da

Tabell 5.1: ANOVA av svar på spørsmål om deltakers opplevelse av sesjonen

Spørsmål	Gj.snitt	Std.dev	p-verdi
<b>Var det morsomt</b>			<b>1,00</b>
Distribuert	2,87	0,74	
Lokalt	2,87	0,64	
<b>Følte du at du bidro</b>			<b>0,51</b>
Distribuert	3,33	0,82	
Lokalt	3,53	0,83	
<b>Hvordan fungerte gruppen</b>			<b>0,83</b>
Distribuert	3,80	0,94	
Lokalt	3,87	0,74	
<b>Følte du at sesjonen hadde <i>trøkk</i></b>			<b>0,04</b>
Distribuert	3,33	0,90	
Lokalt	4,00	0,76	

deltakere i samme rom har muligheten for å stimulere hverandre til økt innsats. Dersom man mener at *trøkk* er en avgjørende og kritisk faktor i en KJ sesjon, må nullhypotesen beholdes. Vi mener at *trøkk* er en viktig faktor i en KJ sesjon, men ikke for enhver pris. Årsaken til at deltakerne opplevde lavere *trøkk* for distribuerte sesjoner kommer av flere faktorer. KjApp distribueres ved hjelp av NetMeeting og dette setter begrensninger med hensyn på samtidighet. Det fører også til at ekstra tid brukes på å overføre kontroll mellom deltakerne. I tillegg må deltakerne skrive inn ideene *på direkten*. Disse faktorene fører til mer ventetid enn under lokal sesjon og oppleves av deltakerne som redusert *trøkk*. En fullstendig distribuert løsning med et lokalt datalager for å forberede egne ideer før de skal presenteres vil bedre dette og deltakerne vil antagelig oppleve at *trøkket* øker. Av den grunn mener vi at resultatet på dette spørsmålet ikke er tilstrekkelig for å beholde nullhypotesen.

Med p-verdier i området 0.3 til 0.86 viser tabell 5.2 at det ikke er noen signifikant forskjell på resultat kvalitet for lokale og distribuerte sesjoner. Dersom vi ser bort fra spørsmålet om *trøkk*, viser analysen av resultatene at det ikke er noen signifikant forskjell mellom distribuerte og lokale sesjoner. Dette indikerer at nullhypotesen kan forkastes. I det videre analysearbeidet velger vi å fjerne resultatet fra spørsmålet om *trøkk* og slår sammen resultatene fra lokale og distribuerte sesjoner. Dette kan vi gjøre da analysen har vist at det ikke er noen signifikant forskjell mellom resultatene fra distribuerte og lokale sesjoner.

Tabell 5.2: ANOVA av svar på spørsmål om deltakers tilfredshet med resultatene

Spørsmål	Gj.snitt	Std.dev	p-verdi
<b>I hvor stor grad tror du at du kan benytte resultatet</b>			<b>0,30</b>
Distribuert	3,53	1,06	
Lokalt	3,13	0,99	
<b>Kom det frem nye momenter eller ideer</b>			<b>0,53</b>
Distribuert	2,93	0,88	
Lokalt	3,13	0,83	
<b>Kunne du kommet frem til de samme resultatene på egenhånd</b>			<b>0,86</b>
Distribuert	3,20	0,94	
Lokalt	3,13	1,06	

### 5.1.2 Analyse av kvantitative resultater

Fra eksperimentet har vi kvantitative resultater for fem distribuerte og fem lokale sesjoner. Antallet er lite for statistisk analyse, men gir indikasjoner hvordan sesjonene fungerte. Vi tror at de kvantitative resultatene er mindre betydningsfulle enn de kvalitative. Som nevnt i avsnitt 3.2.3 er det vanskelig å tolke hva et kvantitativt resultat betyr for en sesjon. Betyr mange post-it lapper at resultatet fra sesjonen er godt? Er det positivt å bruke kort tid på en sesjon? Tilbakemelding fra deltakerne viser at det ikke er noe enkelt ja eller nei svar på disse spørsmålene.

De kvantitative resultatene i tabell 5.3 viser at distribuert sesjon tar adskillig lengre tid enn lokal gjennomføring. Forholdet mellom tid brukt distribuert og lokalt er nesten 2:1. Det er flere faktorer som påvirker resultatet. Som for spørsmålet om *trøkk* diskutert i avsnitt 5.1.1 er det et problem at KjApp mangler støtte for samtidig bruk og et lokalt datalager. Dette fører til at deltakerne bruker lengre tid på å presentere hver enkelt ide. I tillegg har deltakerne liten erfaring med bruken av KjApp, noe som fører til en noe tungvindt bruk. KjApp gir støtte for enkelte kontrolltaster (som for eksempel ctrl+c for å kopiere), men flere deltakere ønsket kontrolltaster for å opprette ny gruppe og post-it. En fullverdig distribuert applikasjon, mer erfaring med KjApp og flere kontrolltaster er faktorer som vil redusere tidsforbruket for distribuert sesjon. Selv om forskjellen i tidsforbruk er stor, må man huske på at distribuerte sesjoner gjennomføres fra eget kontor. Deltakerne slipper å benytte tid på reising og distribuert sesjon vil dermed totalt sett være mer tidseffektiv i forhold til lokal sesjon.

I tillegg kom det varierende tilbakemeldinger fra deltakerne om hvilken innvirkning tidsaspektet hadde på resultatene. Enkelte deltakere mente at det var frustrerende å bruke lang tid, mens andre syntes det var fint å kunne fordøye ideer som ble presentert. Vi stilte derfor deltakerne et oppfølgingsspørsmål om hvilken innvirkningen det hadde på resultatene at distribuert sesjon tok lengre tid enn lokal. Svarene varierte fra bare negativ innvirkning til mest positiv innvirkning, med tyngdepunktet på like mye positiv

som negativ innvirkning. Antall post-it lapper generert i de ulike sesjonene varierer.

Tabell 5.3: Kvantitative resultater fra eksperimentet

Sesjon	Distribuert			Lokalt		
	Tid	Post-it	Ideer	Tid	Post-it	Ideer
GLT	36	19	17	19	23	18
SINTEF	35	20	15	19	23	17
Stud. 1	45	32	26	22	56	42
Diplom	34	20	18	17	17	14
Stud. 2	40	38	34	21	40	32

Den første studentgruppen var et spesielt tilfelle med mange lapper generert under lokal sesjon. Særlig en deltaker var aktiv. Under distribuert sesjon ble tiden et problem, og den aktive deltakeren forkastet mange ideer. Årsaken til at antallet lapper varierer endel kan være at gruppene diskuterer problemet på ulikt abstraksjonsnivå. Vi mener at de to studentgruppene arbeidet på et lavere abstraksjonsnivå enn de andre tre gruppene. Årsaken til dette kan være problemstillingene som ble diskutert eller studentenes erfaring med denne typen arbeid. Studentene var mer detaljfiksert og dette førte til at flere lapper ble generert enn for de andre gruppene.

Deltakerne ble etter de ulike sesjonene bedt om å angi hvor mange enkeltstående ideer post-it lappene utgjorde. Tanken her var at like eller nesten like post-it lapper skulle telles som en ide. Resultatene fra deltakerne hadde veldig stor spredning og det ble klart for oss at deltakerne hadde tolket spørsmålet på forskjellig måte. Vi valgte derfor å sette oss ned sammen med veileder og selv telle antall ideer. På denne måten fjernet vi like eller nesten like ideer og sto igjen med et antall frittstående ideer. Dette antallet er presentert i tabell 5.3. Vi har samme problem med å tolke betydningen av resultatet som for antall post-it lapper, men kan igjen merke oss at de to studentgruppene har et høyere antall enn de andre gruppene. Igjen mener vi dette beror på hvilket abstraksjonsnivå som er valgt.

Under eksperimentet registrerte vi også hvor mange grupper som ble generert for de ulike sesjonene. I ettertid har vi funnet ut at dette resultatet ikke sier oss noe særlig og har valgt å fjerne det fra analysearbeidet. Antallet grupper lå mellom fire og syv for alle sesjonene.

### 5.1.3 ANOVA splittet på deltakernes profesjon

Gjennomføring av eksperimentet viste oss at de ulike sesjonene fungerte på forskjellige måter. Disse observasjonene er beskrevet i avsnitt 4.2. Det ble klart for oss at de profesjonelle gruppene GLT og SINTEF hadde et annet fokus og en annen driv under eksperimentet både for lokal og distribuert sesjon. Gruppene var samkjørte og det var en god tone i gruppene. I tillegg diskuterte gruppene problemstillinger som var relevante for daglige arbeidsoppgaver. Vi mener dette er faktorer som er viktige for et godt resultat. For å teste om så er tilfelle har vi gjennomført en ny ANOVA på de kvalitative

spørsmålene. Denne gangen har vi splittet GLT og SINTEF fra de andre gruppene, kalt *studentgrupper*. Resultatene presenteres i tabell 5.4 og 5.5.

Tabell 5.4: ANOVA på deltakernes opplevelse av sesjonen, splittet på GLT og SINTEF mot *studentgrupper*

Spørsmål	Gj.snitt	Std.dev	p-verdi
<b>Var det morsomt</b>			<b>0,16</b>
GLT/SINTEF	3,08	0,67	
Studentgrupper	2,72	0,67	
<b>Følte du at du bidro</b>			<b>0,00</b>
GLT/SINTEF	4,08	0,67	
Studentgrupper	3,00	0,59	
<b>Hvordan fungerte gruppen</b>			<b>0,00</b>
GLT/SINTEF	4,50	0,67	
Studentgrupper	3,39	0,61	

Tabell 5.4 viser en stor forskjell på resultatene når GLT og SINTEF splittes fra *studentgruppene*. For spørsmålene om deltakerne følte at de bidro og om hvordan gruppen fungerte var det en signifikant forskjell med p-verdier lik null. De profesjonelle deltakerne følte i større grad at de bidro og at gruppen fungerte godt. For spørsmålet *Var det morsomt* er forskjellen mindre og ikke signifikant. Også her er resultatet for de profesjonelle deltakerne bedre enn for deltakerne fra *studentgruppene*. Tabell 5.5 viser

Tabell 5.5: ANOVA på deltakernes tilfredshet med resultatene, splittet på GLT og SINTEF mot *studentgrupper*

Spørsmål	Gj.snitt	Std.dev	p-verdi
<b>I hvor stor grad tror du at du kan benytte resultatet</b>			<b>0,03</b>
GLT/SINTEF	3,83	0,83	
Studentgrupper	3,00	1,02	
<b>Kom det frem nye momenter eller ideer</b>			<b>0,12</b>
GLT/SINTEF	3,33	0,49	
Studentgrupper	2,83	0,99	
<b>Kunne du kommet frem til de samme resultatene på egenhånd</b>			<b>0,46</b>
GLT/SINTEF	3,33	0,65	
Studentgrupper	3,06	1,16	

resultatene for deltakernes tilfredshet med resultatet fra KJ sesjonene splittet på GLT og SINTEF mot *studentgruppene*. For spørsmålet om deltakerne tror at de kan benytte

resultatet er det en signifikant forskjell. En p-verdi på tre prosent viser at GLT og SINTEF har større tro på at resultatet kan benyttes. I tillegg har GLT og SINTEF høyere verdier for de andre spørsmålene, men forskjellen er ikke signifikant.

Denne analysen mener vi understøtter våre antagelser om at en samkjørt og profesjonell gruppe vil få bedre resultater enn en tilfeldig sammensatt gruppe med deltakerne som kjenner hverandre dårlig.

## 5.2 Oppsummering av statistisk analyse

Resultatene fra statistisk analyse indikerer at det ikke er store forskjeller på distribuert og lokal gjennomføring av KJ sesjoner. Deltakernes opplevelse av *trøkk* og varigheten på sesjonene er momenter hvor avviket mellom lokale og distribuerte sesjoner er størst. Vi mener likevel ikke dette er nok til å beholde nullhypotesen som sier at lokal KJ fungerer best. Forklaringen på avviket mener vi ligger i at KjApp ikke er en distribuert applikasjon. Dette fører til redusert samtidighet og deltakerne bruker derfor lengre tid på å presentere ideene. Deltakerne opplever dette som redusert flyt og *trøkk*.

Antall post-it, ideer og grupper generert under sesjonene har etter vårt syn begrenset betydning når de studeres alene. Dersom antallet ideer sees i sammenheng med hvor fornøyde deltakerne er med resultatet gir de større mening. Datamaterialet viser at det ikke er store forskjeller mellom distribuert og lokal sesjon for de fleste gruppene. Unntaket er første studentgruppe som produserte *ekstremt* mange post-it og ideer under lokal sesjon. Tiden ble et problem under distribuert sesjon og flere ideer ble derfor ikke presentert. Dersom det var et gjennomgående trekk at lokal sesjon genererte betydelig flere ideer enn distribuert samtidig som deltakerne var mer fornøyd med resultatet fra den lokale, kan dette indikere at antall ideer generert og et godt resultat henger sammen. Noen slik sammenheng finnes ikke i vårt datamateriale.

Vi har ikke foretatt statistisk analyse av oppfølgingsspørsmålene, men svarene har ikke vært bortkastet. Oppfølgingsspørsmålene belyser en del kommentarer som kom frem under eksperimentet og gir indikasjoner på hva som fungerer godt og hva som fungerer dårlig. Oppfølgingsspørsmålene har derfor hjulpet oss med å trekke det vi mener er riktig konklusjon; nemlig at distribuerte og lokale KJ sesjoner under visse forutsetninger fungerer like godt. Felles plattform og god kjennskap til hverandre er faktorer som er vesentlige for at distribuert KJ skal fungere. Dette viser analysen hvor deltakerne splittes i to grupper. De profesjonelle, samkjørte gruppene var mer fornøyd med resultatet enn de *tilfeldig* sammensatte studentgruppene. De følte i tillegg at de bidro mer og at gruppene fungerte bedre enn det studentgruppene gjorde. Dette indikerer at god kjennskap til hverandre er en viktig faktor i forbindelse med distribuert idegenerering.

# Kapittel 6

## Videre arbeid

### Innhold

---

6.1 Deltakersammensetning . . . . .	49
6.2 Kreativt utfordrende problemstillinger . . . . .	50
6.3 Varighet og fysisk kontakt . . . . .	50

---

Resultatene fra eksperimentet indikerer at det er liten forskjell på å resultatene fra distribuert og lokal gjennomføring av KJ sesjoner. Likevel er det vanskelig å trekke bastante konklusjoner på bakgrunn av eksperimentet. I dette kapittelet presenterer vi forslag til videre arbeid innen området. Etter vår mening bør videre eksperimentering foregå ved hjelp av en distribuert versjon av KjApp. Dette vil føre til økt samtidighet og deltakerne vil antagelig oppleve dette som at sesjonene har høyere *trøkk*. Forslagene til videre arbeid presenteres i kategoriene deltakersammensetning (6.1), kreativt utfordrende problemer (6.2) og lengre varighet og fysisk kontakt (6.3).

### 6.1 Deltakersammensetning

Deltakerne i eksperiment utgjør en forholdsvis homogen gruppe. De kommer fra et teknologiintensivt miljø og har i stor grad samme bakgrunn. Samarbeid mellom homogene deltakere er enklere, da diskusjonen bygger på en felles plattform. I næringslivet er det mer realistisk at deltakerne har ulik bakgrunn og kommer sammen for å løse et komplekst problem ved hjelp av en KJ sesjon. For eksempel kan det være problemområder som både må ta hensyn til mellommenneskelige og teknologiske faktorer. Dermed kreves det ekspertise fra ulike hold for å finne en løsning på problemet. I distribuerte omgivelser kan dette være en utfordring. Et eksperiment hvor deltakerne har forskjellig bakgrunn tror vi gir andre resultater enn våre. Idegenerering med en heterogen gruppe tror vi fungerer best med deltakerne i samme rom på grunn av bedre muligheter for kommunikasjon og synkronisering av prosessen. Et eksperiment vil kunne vise om dette stemmer.

Eksperimentet så på distribuert samarbeid med grupper på tre personer. Dersom antallet øker, tror vi det blir vanskeligere samarbeide under distribuerte forhold. Synkronisering av aktiviteter og kommunikasjon forverres når antallet deltakere øker. Det vil

derfor være interessant å gjennomføre eksperimenter for å undersøke dette forholdet. Det er også interessant å gjennomføre flere sesjoner enn det vi har gjort. Med et større antall deltakere vil resultatene fra statistisk analyse være mer pålitelige enn det våre er og det vil i større grad være mulig å gjennomføre statistisk analyse på kvantitative data som tid og antall ideer.

Våre resultater indikerer at samarbeid mellom profesjonelle deltakerne fungerer bedre enn mellom tilfeldig sammensatte studentgrupper. Et nytt eksperiment med flere profesjonelle deltakere vil kunne belyse flere sider ved dette. Resultatene fra et slikt eksperiment vil antagelig også være mer interessant, da de lettere kan generaliseres til næringslivet.

## 6.2 Kreativt utfordrende problemstillinger

Deltakerne i eksperimentet diskuterte problemstillinger ved å hente ut deltakernes erfaringer innen et område. Ingen problemstillinger var kreativt utfordrende. Vi tror kreativt utfordrende problemer er vanskeligere å diskutere under distribuerte forhold. Denne typen problemer er ofte ikke klart definert - *Vi vet vi har et problem, men vi vet ikke hva det er*. Med slike problemer er det behov for banebrytende tenkning, og dette tror vi er vanskeligere å få til under distribuerte omgivelser. Et eksperiment som ser på dette forholdet, for eksempel ved å la deltakerne løse ulike *gåter eller nøtter* sammen, kan belyse interessante aspekter i forhold til distribuert idegenerering.

## 6.3 Varighet og fysisk kontakt

Deltakerne i eksperimentet samarbeidet distribuert i en intensiv periode av kort varighet. Dersom sesjonene hadde hatt lenger varighet tror vi samarbeid under distribuerte forhold ville vært vanskeligere enn under lokale. Det kan være vanskeligere å holde motivasjonen og trykket oppe når deltakerne ikke fysisk er i kontakt med hverandre.

Et annet moment som kan studeres med videre eksperimentering, er hvilken innvirkning det har at deltakerne i eksperimentet traff hverandre før den distribuerte sesjonen skulle gjennomføres. Et eksperiment hvor deltakerne ikke møtes før gjennomføring av en distribuert KJ sesjon eller kanskje aldri har møtt hverandre kunne gi interessante resultater om distribuert samarbeid.

## Kapittel 7

# Diskusjon eksperimentet

Analyse av resultatene fra eksperimentet har etter vår mening vist at det liten forskjell på distribuert og lokal gjennomføring av KJ sesjoner. Dette tyder på at distribuert KJ kan være en god erstatning for lokal gjennomføring. For eksempel kan distribuert KJ benyttes dersom deltakerne må reise langt for å delta i en lokal sesjon. Spørsmålet er imidlertid om resultatene da blir like gode som ved lokal gjennomføring.

Eksperimentet tyder på at visse faktorer bør være tilstede for at resultatet ved distribuert KJ skal bli godt. Deltakerne bør kjenne hverandre og ha en felles plattform for diskusjonen. Vårt inntrykk er at en tilfeldig sammensatt gruppe vil ha større problemer med å samarbeide distribuert enn en samkjørt gruppe.

Det er flere sider ved eksperimentet som ikke er optimale for å trekke generelle konklusjoner. For det første er antallet deltakere lavt og dette er et risikomoment ved statistisk analyse. Det er derfor vanskelig å trekke bastante konklusjoner på bakgrunn av resultatet, men vi mener eksperimentet gir sterke indikasjoner i retning av at distribuert KJ kan fungere like godt som lokal. Et annet moment er at deltakerne kommer fra et teknologiintensivt miljø. Deltakerne er vant med å benytte PC og Windowsbasert programvare og fant seg raskt til rette i de distribuerte omgivelsene. Deltakere fra andre og mindre teknologiintensive miljøer vil muligens ikke mestre distribuert omgivelser like lett. Aldersfordelingen til deltakerne i eksperimentet er også en faktor som må vurderes. Deltakere var i aldersgruppen nitten til trettifire år og unge mennesker er kanskje mer åpne for å prøve nye løsninger enn den *eldre garde*. Dersom deltakerne i eksperimentet hadde hatt en annen aldersfordeling, ville kanskje konklusjonen blitt en annen.

Problemene som ble diskutert under sesjonene kan også utgjøre et faremoment i forhold til gyldigheten av eksperimentet. Deltakerne diskuterte i hovedsak teknologiske problemer og diskusjonen var i stor grad en oppsummering av deltakernes erfaringer innen problemområdet. Et annet og kreativt mer utfordrende problem vil kanskje være vanskeligere å diskutere under distribuerte omgivelser. Eksperimentet viser at distribuerte sesjoner har lavere *trøkk* enn lokale og *trøkk* kan være en viktig faktor når kreativt utfordrende problemer skal løses. Det er lettere å skape engasjement med deltakerne i samme rom enn når de spredt geografisk.

På slutten av vårt arbeid med oppgaven har vi fått tilgang til studier som langt på vei bekrefter resultatene vi har oppnådd. Distribuert KJ kan kategoriseres som en en-

kel samarbeidsform og en *protokol* styrer til en viss grad interaksjonen. Gruppene som samarbeider er relativt små og jobber sammen i en begrenset tidsperiode. I tillegg er gruppene i eksperimentet homogent sammensatt. Deltakerne kommer alle fra et teknologiintensivt miljø og har gode forutsetninger for å beherske distribuerte omgivelser. Studier av Bannon og Schmidt [?] og Ellis og Rein [?] viser at under disse forutsetninger vil distribuert samarbeid fungere godt ved hjelp av verktøy som for eksempel MicroSoft NetMeeting. Mer kompliserte former for samarbeid, for eksempel tillitskappende arbeid, kan imidlertid være mer problematisk i distribuerte miljøer.

## Kapittel 8

# Konklusjon eksperimentet

Før gjennomføringen av eksperimentet trodde vi at idegenerering med deltakerne i samme rom ville fungere langt bedre enn under distribuerte omgivelser. Men eksperimentet har vist at så ikke er tilfelle. Det er flere faktorer som skiller distribuerte og lokale sesjoner fra hverandre, men det virker som at disse ikke er avgjørende for hvor godt resultatet blir. Eksperimentet viser langt på vei at distribuert KJ fungerer like godt som lokal. Dette tyder på at distribuert KJ i mange tilfeller kan erstatte gjennomføringer med deltakerne i samme rom.

Den største forskjellen mellom distribuerte og lokale sesjoner var med tanke på deltakernes opplevelse av *trøkk* og sesjonsvarigheten. Lokal sesjon hadde langt bedre flyt enn distribuert og tok halvparten så lang tid. De distribuerte omgivelsene NetMeeting og telefonkonferanse må ta noe av skylden for dette forholdet. Likevel viser eksperimentet at deltakerne gjennomgående er mer fornøyde med resultatene fra distribuerte sesjoner.

Under analysen av resultatene fra eksperimentet delte vi deltakerne i to grupper; en profesjonell gruppe med GLT og SINTEF mot resten av deltakerne i en *studentgruppe*. Splittingen viser at den profesjonelle gruppen er langt mer fornøyd med resultatene sine enn deltakere i den mer tilfeldig sammensatte *studentgruppen*. Vårt inntrykk er at faktorer som god kjennskap til hverandre og det at deltakerne i gruppen har felles gevinst av en et godt resultatet, er mer avgjørende enn om sesjonen gjennomføres distribuert eller lokalt.



**Del II**

**Prototypen**



# Kapittel 9

## Utviklingsmetode og -miljø

### Innhold

---

9.1	Utviklingsmetode . . . . .	58
9.2	Utviklingsmiljø . . . . .	58

---

I denne delen beskrives arbeidet med å utvikle prototypen KjApp som skal støtte KJ metoden. Prototypen skal ha funksjonalitet for å generere og ta vare på ideer, vise genererte ideer, gruppere ideer i områder med beskrivende navn og vise det ferdige affinitetsdiagrammet. Videre skal prototypen ha et enkelt og logisk oppbygd brukergrensesnitt. Funksjonaliteten distribueres ved hjelp av Microsoft NetMeeting. Dette medfører at KjApp implementeres som en frittstående applikasjon og kun kjøres på en PC. Deltakerne i en distribuert KJ sesjon får tilgang til å styre applikasjonen ved hjelp av *applikasjonsdelingsfunksjonen* i NetMeeting.

I dette kapitlet beskrives utviklingsmetode og utviklingsmiljø som er benyttet i arbeidet. Videre følger kravspesifikasjonen i kapittel 10 som danner grunnlaget for applikasjonens design (kap. 11), implementasjon (kap. 12) og testing og installasjon (kap. 13). KjApp skal i hovedsak muliggjøre eksperimentering med distribuert KJ og er ikke tenkt som en ferdig applikasjon til benyttelse i næringslivet. Mangler i prototypen og forslag til forbedringer presenteres i kapittel 15, *Videre arbeid*.

## 9.1 Utviklingsmetode

Vi har valgt metoder, verktøy og teknologier vi kjenner fra før for å kunne utvikle en fungerende prototyp raskt, og dermed få bedre tid til å gjennomføre eksperimenter med distribuert arbeid. Utviklingsmetode som benyttes er inkrementell og baseres på Rational Unified Process (RUP) som vi kjenner fra prosjektarbeid ved Hærens forsyningskommando [?]. Bruken av RUP medfører at vi starter med å utvikle en kjørende versjon av programmet. Deretter utvides programmets funksjonalitet ved å implementere og integrere nye use case. Vi har kun benyttet oss av de elementer i rammeverket vi mener passer for et lite utviklingsprosjekt. Utviklingsarbeidet starter med kravspesifikasjonen. Under dette arbeidet er use case diagrammer et viktig hjelpemiddel for å beskrive de funksjonelle kravene til KjApp. Prototypen har en Model View Controller (MVC) arkitektur. Klassediagrammer, viktige algoritmer og sekvensdiagrammer for grunnleggende funksjonalitet ble laget i Rational Rose. Fra Rational Rose genererte vi *skjelettet* til klassene. Det ble et godt utgangspunkt for implementasjonsfasen. Implementasjonen startet med utvikling av brukergrensesnitt og menyer. Deretter ble hver enkelt use case implementert etter tur. Integrasjonstesting ble foretatt etterhvert som ny funksjonalitet ble lagt til.

## 9.2 Utviklingsmiljø

Avsnittet beskriver plattformen KjApp er utviklet for og verktøy som er benyttet i design- og implementasjonsfasen. Videre beskrives støttelitteratur som har vært et viktig hjelpemiddel underveis.

### 9.2.1 Java

KjApp er utviklet med Java 2 Standard Edition versjon 1.4.0 (J2SE). Versjonen er den første som tilbyr XML parser som en del av standardbiblioteket og XML er formatet som benyttes for lagring av tavler. For implementasjon av brukergrensesnittet er stort sett komponenter fra Swing pakken benyttet. Videre informasjon om J2SE er tilgjengelig fra Java Suns hjemmesider [?]. For å kjøre applikasjonen må Java 2 Runtime Environment versjon 1.4.0 (J2RE) installeres. Også J2RE er tilgjengelig fra Java Suns hjemmesider og støtter pakkene som benyttes i prototypen.

### 9.2.2 Støtteverktøy

Rational Rose [?] er benyttet for å lage modeller. Use case diagram er laget i forbindelse med arbeidet med kravspesifikasjonen og klasse- og sekvensdiagram til designspesifikasjonen. Verktøyet støtter alle diagrammer i Unified Modeling Language (UML) [?]. På bakgrunn av klassediagrammer benyttes verktøyet for å generere skjelettet til kildekoden.

JBuilder6 [?] er benyttet som verktøy i arbeidet med kildekoden til prototypen. Verktøyet støtter generering av dokumentasjon, bakovergenerering av klassediagrammer, kompilering og debugging av kildekode.

### 9.2.3 Støttelitteratur

Nettsidene Java Sun [?] er benyttet for oppslag i API (Application Programming Interface) og for å finne eksempler på kildekode. Sidene gir også tilgang til en innholdsrik *tutorial*. Videre har vi benyttet bøkene Core Java Fundamentals [?] og Core Java Advanced Features [?] for å finne effektive løsninger for implementasjonen.



# Kapittel 10

## Kravspesifikasjon

### Innhold

---

10.1 Funksjonelle krav . . . . .	61
10.2 Ikke-funksjonelle krav . . . . .	66
10.3 Skjermbilder . . . . .	68

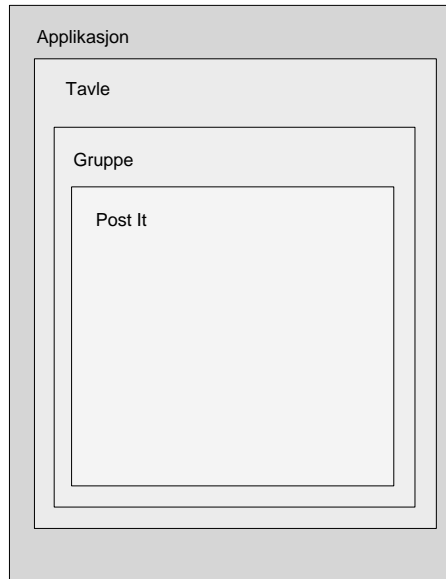
---

Kapittelet beskriver krav til prototypen som skal utvikles. De funksjonelle kravene i avsnitt 10.1 er delt inn i fire nivå. Disse er applikasjonsnivå, tavlenivå, gruppenivå og post-it lapp nivå. Funksjonelle krav beskriver funksjoner som skal være tilgjengelig for brukeren av prototypen. I avsnitt 10.2 beskrives de ikke-funksjonelle kravene til prototypen. Dette er krav som omhandler hvordan prototypen skal samspille med omgivelsene, for eksempel brukervennlighet, ytelse og lagring. Til slutt presenteres konstruerte skjermbilder av prototypen i avsnitt 10.3.

### 10.1 Funksjonelle krav

Avsnittet beskriver de funksjonelle kravene til applikasjonen. Kravene er basert på beskrivelsen av KJ metoden gitt i avsnitt 2.2. Vi har lagt vekt på innspill fra erfarne brukere og egne erfaringer med metoden. Prototypen skal ha tilstrekkelig funksjonalitet til å gjennomføre en KJ sesjon. Vi ønsker likevel minst mulig *ekstra* funksjonalitet som kan forvirre brukerne og dermed forringe kvaliteten på den kreative prosessen. For å strukturere kravspesifikasjonen benyttes fire abstraksjonsnivå. Nivåene beskriver relatert funksjonalitet og et hierarkisk forhold mellom elementene i applikasjonen. Forholdet er vist i figur 10.1.

- Applikasjonsnivå  
Generell funksjonalitet for applikasjonen. Kravene merkes med FA (funksjonelle krav applikasjonsnivå).
- Tavlenivå  
Funksjonalitet som benyttes på tavlen. En tavle må opprettes før funksjonalitet på lavere nivå benyttes (se figur 10.1). Kravene merkes FT (funksjonelle krav tavlenivå).

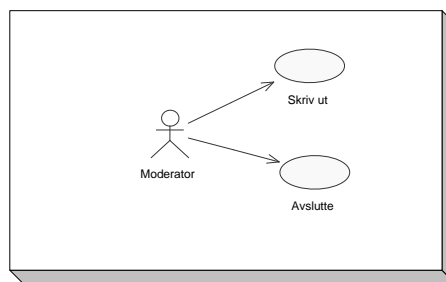


Figur 10.1: Nivåinndeling i applikasjonen

- Gruppenivå  
Funksjonalitet for å håndtere gruppering i KJ metoden beskrives her. Kravene merkes med FG ( funksjonelle krav gruppenivå).
- Post-it nivå  
Funksjonalitet for de enkelte post-it lapper. Kravene merkes med FP (funksjonelle krav post-it nivå).

### 10.1.1 Applikasjonsnivå

Avsnittet beskriver de funksjonelle kravene på applikasjonsnivå. De funksjonelle kravene er vist i use case diagrammet i figur 10.2.



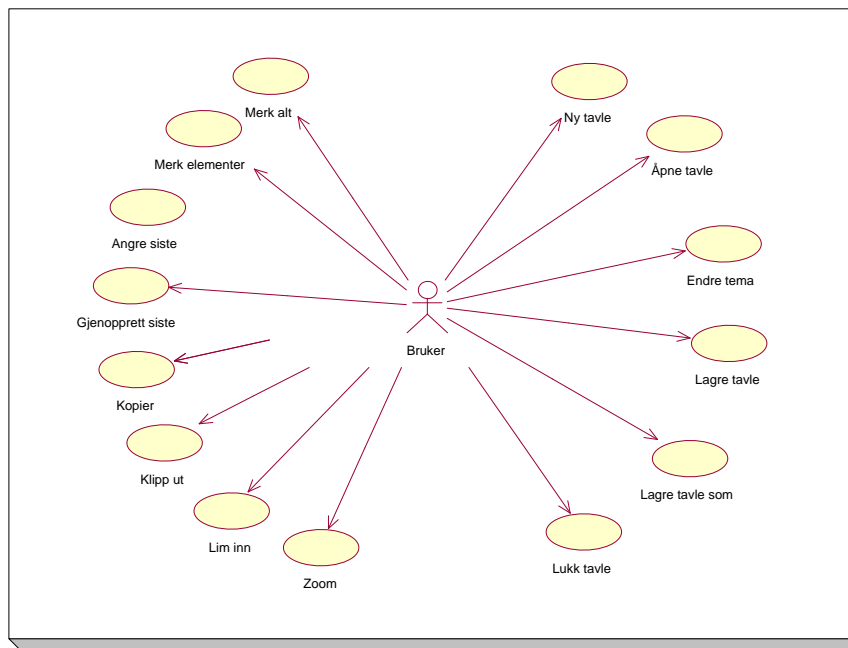
Figur 10.2: Use case diagram av funksjonelle krav på applikasjonsnivå

**FA1-Skriv ut** Skriver ut tavlen på standard skriver.

**FA2-Avslutte** Avslutter KjApp. Dersom tavlen ikke er lagret, skal brukeren spørres om han ønsker å gjøre det.

### 10.1.2 Tavlenivå

Avsnittet beskriver de funksjonelle kravene på tavlenivå. Lesere som er vant med Windows grensesnitt vil gjenkjenne standardfunksjoner fra Windows. De funksjonelle kravene er vist i use case diagrammet i figur 10.3.



Figur 10.3: Use case diagram av funksjonelle krav på tavlenivå

**FT1-Ny tavle** Oppretter en ny, tom tavle. Brukeren gir tavlen et navn eller benytter standardnavn (Min tavle).

**FT2-Åpne tavle** Åpner tidligere lagret tavle.

**FT3-Endre tema** Bruker endrer tematekst på tavlen.

**FT4-Lagre tavle** Lagrer tavle med valgt filnavn og lagringsformat (XML- eller tekstformat).

**FT5-Lagre tavle som** Lagrer tavle med nytt filnavn og/eller lagringsformat (XML- eller tekstformat).

**FT6-Lukk tavle** Lukker tavlen uten å avslutte applikasjonen. Dersom tavlen ikke er lagret, skal brukeren spørres om han ønsker å gjøre det.

**FT7-Merk alt** Merker alle elementer på tavlen.

**FT8-Merk elementer** Merker et eller flere elementer på tavlen.

**FT9-Angre siste** Angrer sist utførte kommando. Tavlen rekonstrueres til slik den var før siste kommando ble kjørt.

**FT10-Gjenopprett siste** Benyttes etter *Angre siste* for å gjenopprette applikasjonen til status før angring.

**FT11-Kopier** Kopierer merkede elementer. Elementene finnes fortsatt i tavlen. Kopien legges på utklippstavlen til KjApp.

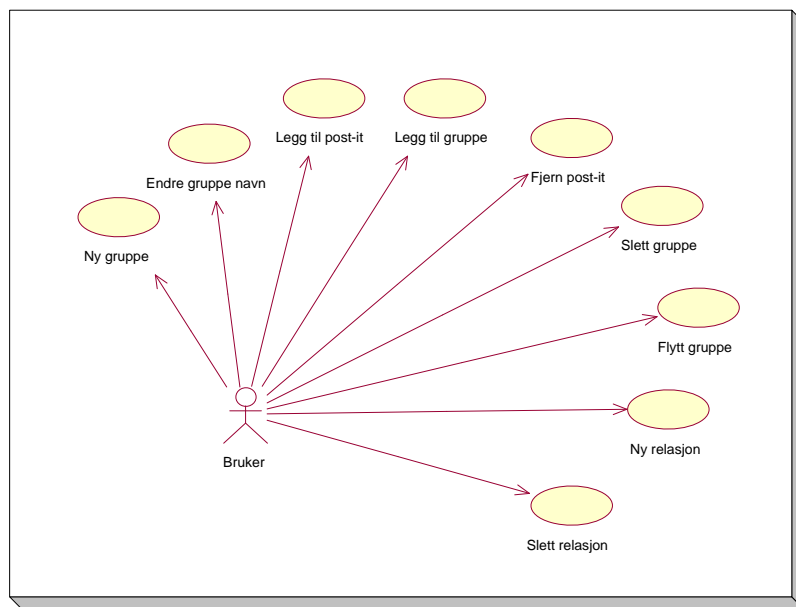
**FT12-Klipp ut** Klipper ut merkede elementer. Elementene forsvinner fra tavlen og legges på utklippstavlen til KjApp.

**FT13-Lim inn** Limer inn elementer fra utklippstavlen.

**FT14-Zoom** Zoomer tavlen inn eller ut (+ og -).

### 10.1.3 Gruppenivå

Avsnittet beskriver funksjonelle krav på gruppenivå. I KJ består en gruppe av en eller flere post-it lapper og eventuelle undergrupper. I KjApp kan tomme grupper opprettes for siden å fylles med post-it lapper. De funksjonelle kravene er vist i use case diagrammet i figur 10.4.



Figur 10.4: Use case diagram av funksjonelle krav på gruppenivå

**FG1-Ny gruppe** Oppretter ny gruppe. Brukeren gir gruppen et navn eller velger å benytte standard navnet *Gruppe [nummer]*.

**FG2-Endre gruppenavn** Endrer navnet på gruppen.

**FG3-Legg til post-it** Legger til en post-it i merket gruppe.

**FG4-Legg til gruppe** Legger til gruppe A i gruppe B. Gruppe B blir *forelder* med A som *barn*, og et hierarki av grupper dannes.

**FG5-Fjern post-it** Fjerner en post-it fra gruppen.

**FG6-Slett gruppe** Sletter gruppen. Innholdet i gruppen legges på den slettede gruppes forelder.

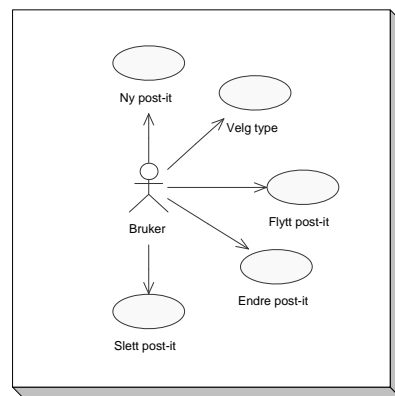
**FG7-Flytt gruppe** Flytter en gruppe med alt innhold til ny posisjon på tavlen.

**FG8-Ny relasjon** Oppretter ny relasjon mellom to grupper.

**FG9-Slett relasjon** Sletter relasjonen mellom to grupper.

#### 10.1.4 Post-it lapp nivå

Avsnittet beskriver funksjonelle krav for post-it lapper. Kravene er vist i use case diagrammet i figur 10.5.



Figur 10.5: Use case diagram av funksjonelle krav på post-it lapp nivå

**FP1-Ny post-it** Ny post-it lapp opprettes. En ide beskrives ved å gi lappen et navn. Standard idetekst er *Post-it [nummer]*.

**FP2-Velg type** Velger om post-it lappen beskriver en positiv, negativ eller nøytral ide. Standard er nøytral.

**FP3-Flytt post-it** Flytter post-it lappen ved å *dra* den til ny posisjon. Post-it lappen blir medlem av gruppe som favner om ny posisjon.

**FP4-Endre post-it** Endre ideteksten og type på post-it lappen.

**FP5-Slett post-it** Sletter merket post-it lapp.

## 10.2 Ikke-funksjonelle krav

Avsnittet beskriver ikke-funksjonelle krav til applikasjonen. Målet er å utvikle en prototyp som muliggjør eksperimentering med distribuert KJ. Brukervennlighet gis størst prioritet. Følgende avsnitt begrunner valgene som er gjort. Kravene merkes med IF (ikke-funksjonelle krav)

### IF1 - Brukervennlighet

Vi vurderer brukervennlighet til å være det viktigste ikke-funksjonelle kravet til prototypen. Ved eksperimentering vil deltakerne få en rask introduksjon til applikasjonen før den skal brukes. Derfor er det viktig at applikasjonen er logisk oppbygd og at deltakerne raskt behersker de mest brukte funksjoner. Brukervennlighet oppnås ved å følge Windows standarden med hensyn på skjermoppsett, hurtigtaster og menyer. I tillegg gjenspeiler applikasjonen lokal gjennomføring av KJ. Vi benytter tavle- og post-it lapp metaforene for å gjøre distribuert og lokal gjennomføring så like som mulig. Personer som kjenner KJ metoden skal enkelt kunne ta i bruk applikasjonen og gjennomføre en KJ sesjon.

### IF2 - Lagring

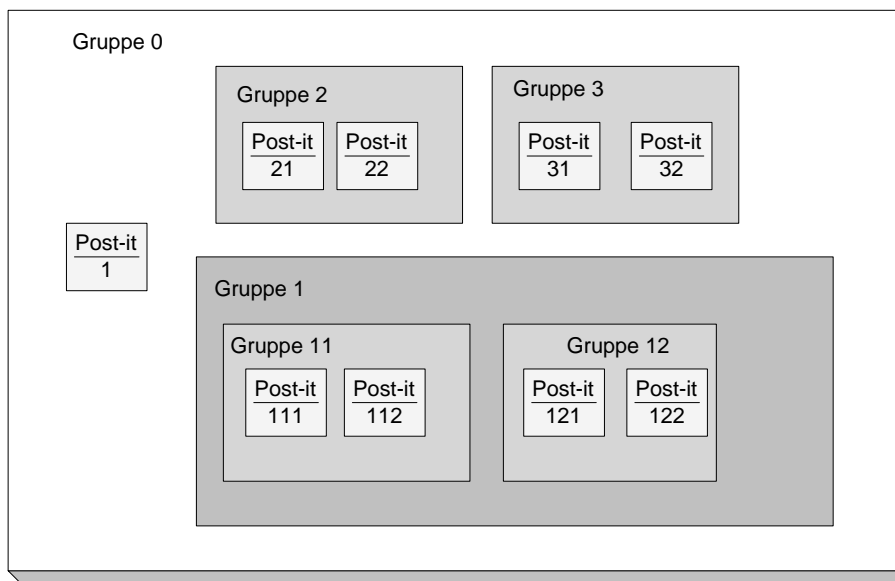
XML (Extended Markup Language) er en teknologi som ofte benyttes i applikasjoner med stor grad av informasjonsoverføring. XML gjør det mulig å kode informasjon på en måte som er lett å forstå, prosessere og generere. Applikasjonen som utvikles i denne oppgaven distribueres ved hjelp av Microsoft NetMeeting. Likevel mener vi det er hensiktsmessig å utforske bruken av XML for å kode applikasjonsinformasjon. Årsaken er at XML kan benyttes som et utvekslingsformat for en applikasjon som håndterer distribusjon uten hjelp av Microsoft Netmeeting. Utviklingen av en *fullstendig distribuert* applikasjon vil være en naturlig forlengelse av arbeidet vårt. Applikasjonen lagrer informasjonen på XML formatet. XML koden som følger beskriver deler av tavlen vist i figur 10.6. Alle elementene på tavlen har en unik id som i XML koden er angitt med *gruppeid* eller *postitid*. I tillegg har elementene navn, kommentar, størrelse og posisjon. For post-it lapper angis også status som kan være nøytral, positiv eller negativ.

```
<?xml version="1.0" encoding="ISO-8859-1"?><tavle>
<tavlenavn>Gruppe 0</tavlenavn>
  <gruppe>
    <gruppeid>105</gruppeid>
    <gruppenavn>Gruppe 2</gruppenavn>
    <gruppeposisjonX>341</gruppeposisjonX>
    <gruppeposisjonY>56</gruppeposisjonY>
    <gruppebredde>165</gruppebredde>
    <gruppehoyde>107</gruppehoyde>
    <postit>
      <postitid>106</postitid>
      <postitnavn>Post-it 21</postitnavn>
      <postitposisjonX>15</postitposisjonX>
      <postitposisjonY>40</postitposisjonY>
      <postitbredde>66</postitbredde>
      <postithoyde>55</postithoyde>
      <postitstatus>2</postitstatus>
    </postit>
  </gruppe>
  <postit>
    <postitid>129</postitid>
    <postitnavn>Post-it 22</postitnavn>
```

```

    <postitposisjonX>89</postitposisjonX>
    <postitposisjonY>42</postitposisjonY>
    <postitbredde>66</postitbredde>
    <postithoyde>55</postithoyde>
    <postitstatus>2</postitstatus>
  </postit>
</gruppe>
.
.
.
.
.
<postit>
  <postitid>25</postitid>
  <postitnavn>Post-it 1</postitnavn>
  <postitposisjonX>256</postitposisjonX>
  <postitposisjonY>104</postitposisjonY>
  <postitbredde>66</postitbredde>
  <postithoyde>55</postithoyde>
  <postitstatus>2</postitstatus>
</postit>
</tavle>

```



Figur 10.6: Layout i tavle for tavleelementer

### IF3 - Ytelse

Det er et krav at applikasjonens ytelse er god nok til at eksperimentet kan gjennomføres på en tilfredsstillende måte. Distribuerings vil føre til økt responstid, men deltakerne skal ikke oppleve dette som plagsomt. Lang responstid er et irritasjonsmoment og vil påvirke resultatene for distribuerte sesjoner i negativ retning. Ytelse vil allikevel ikke prioriteres for prototypen. Årsaken er at det er ressurskrevende å utvikle applikasjoner

med høy ytelse, og dette ville gått på bekostning av brukervennlighet og tid til eksperimentering.

#### IF4 - Distribusjon

Applikasjonen distribueres ved hjelp av Microsoft NetMeeting eller tilsvarende programvare som støtter deling av applikasjoner.

#### IF5 - Plattform

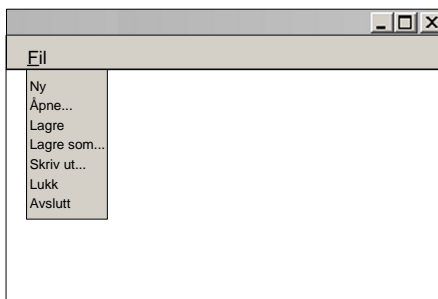
Applikasjonen kjøres på Windows plattform.

### 10.3 Skjermbilder

For å kontrollere at de funksjonelle kravene dekker behovet hos brukere av KJ har vi konstruert skjermbilder av applikasjonen. Skjermbildene forenkler kommunikasjon med brukere av KJ, og vi har benyttet dem for å verifisere kravene overfor erfarne brukere av KJ metoden.

#### Filmeny

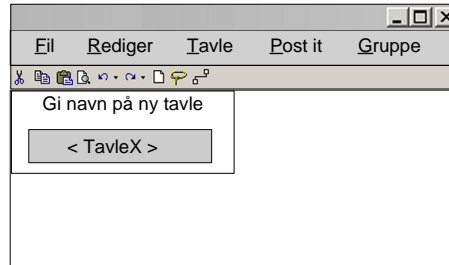
Figur 10.7 viser hvordan applikasjonen ser ut ved start. Brukeren har tilgang til funksjonalitet for å opprette en ny tavle, hente frem en eksisterende tavle eller avslutte programmet.



Figur 10.7: Filmeny

### Opprettelse av ny tavle

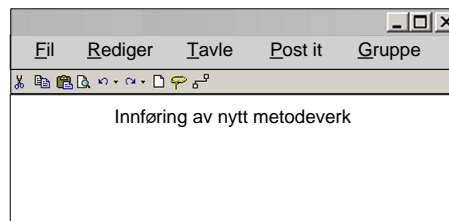
Dersom brukeren velger å opprette en ny tavle vil han få frem en dialogboks som vist i figur 10.8. I dialogboksen skal brukeren angi tema for sesjonen. Temaet vil legges øverst i tavlen som åpnes og være godt synlig for deltakerne i sesjonen (se figur 10.9).



Figur 10.8: Opprettelse av ny tavle

### Resultatet etter at ny tavle er opprettet

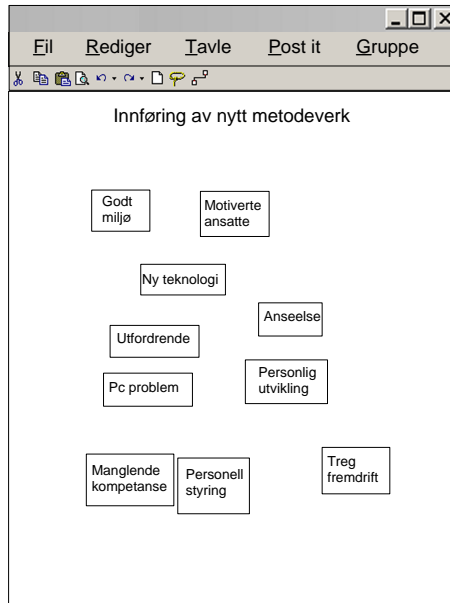
Figur 10.9 viser hvordan applikasjonen ser ut etter at en ny tavle er åpnet. Brukeren får tilgang til en utvidet menylinje med valg for å redigere innholdet i tavlen. Videre får brukeren tilgang til en verktøylinje med knapper for de mest benyttede funksjonene i applikasjonen. For eksempel finnes knapper for klipp ut, lim inn og kopier.



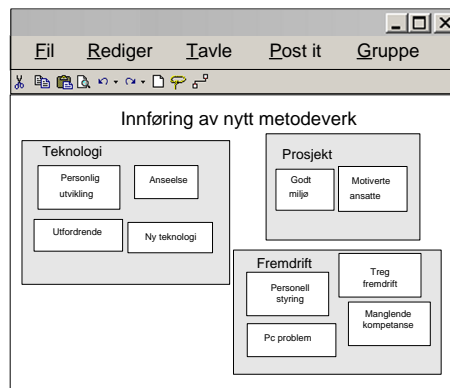
Figur 10.9: Ny tavle

### Post-it lapper fra KJ sesjon

I figur 10.10 vises post-it lapper generert i en KJ sesjon ved Hærens forsyningskommando [?]. På dette tidspunktet er alle ideer generert og presentert, men ideene er ikke gruppert. Det vil være mulig å benytte *drag and drop* på alle nivå under tavle nivå. Det vil si at man kan ta tak i en gruppe eller en post-it lapp og dra denne til et annet sted på tavlen. Man kan også dra en gruppe eller en post-it lapp inn i en eksisterende gruppe. Når alle ideer er presentert på tavlen skal ideene samles i grupper med beskrivende navn. Dette er vist i figur 10.11.



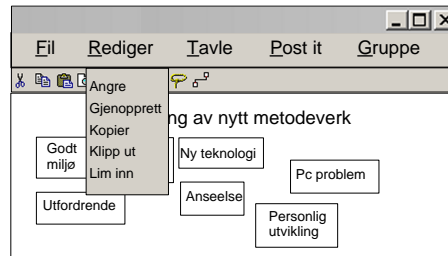
Figur 10.10: Post-it lapper fra Hærens forsyningskommando



Figur 10.11: Gruppering av post-it lapper

### Meny for redigering

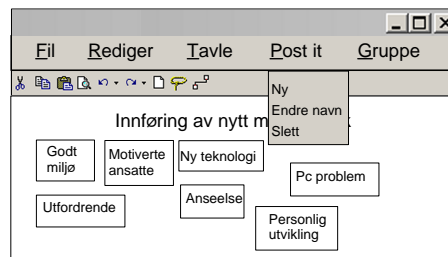
Figur 10.12 viser de valg brukeren har tilgjengelig for redigering på tavlenivå i applikasjonen. Her er vanlige Windows kommandoer tilgjengelig for redigering av både grupper og post-it lapper (angre siste, gjenopprett siste, kopier, klipp ut og lim inn).



Figur 10.12: Meny for redigering

### Meny for post-it

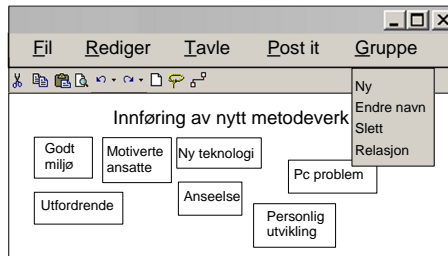
Fra menyen post-it vil brukeren ha tilgang til funksjoner for å lage ny, for å endre navn og type og for å slette post-it lapper (figur 10.13). Funksjonen for å lage ny post-it vil også være tilgjengelig fra verktøylinjen.



Figur 10.13: Meny for post-it

### Meny for gruppe

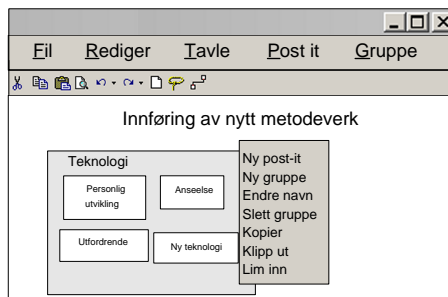
Menyen gir brukeren tilgang til funksjoner for å opprette grupper og endre navn på eksisterende grupper (figur 10.14). Brukeren kan også opprette og slette relasjoner mellom grupper fra denne menyen.



Figur 10.14: Meny for gruppe

### Høyreklikk meny

Brukeren kan benytte høyre musknapp for å åpne menyer i tavlen. Hvilken meny som åpnes er avhengig av hvilket element brukeren har klikket og hva brukeren har gjort tidligere. Figur 10.15 viser høyreklikkmeny der brukeren først har kopiert en post-it lapp og deretter høyreklikket på en gruppe. Høyreklikkmenyen viser tillatte operasjoner.



Figur 10.15: Høyreklikkmeny for gruppe

# Kapittel 11

## Design

### Innhold

---

11.1 Model-View-Controller . . . . .	73
11.2 Composite . . . . .	75
11.3 Klassediagram for KjApp . . . . .	76
11.4 Sekvensdiagrammer . . . . .	81
11.5 Algoritmer . . . . .	86

---

Arkitekturen til KjApp er utarbeidet på bakgrunn av kravspesifikasjonen i kapittel 10. Som en retningslinje for designet har vi valgt å benytte arkitektoniske mønstre og mønstre for design. Arkitektoniske mønstre definerer arkitekturen til applikasjonen på et overordnet nivå og berører hele applikasjonen. Designmønstre benyttes for en mindre del av applikasjonen og beskriver velprøvde løsninger på problemer som går igjen i flere applikasjoner. Ved bruk av mønstre drar vi nytte av eksperters erfaringer på hva som har fungert tidligere.

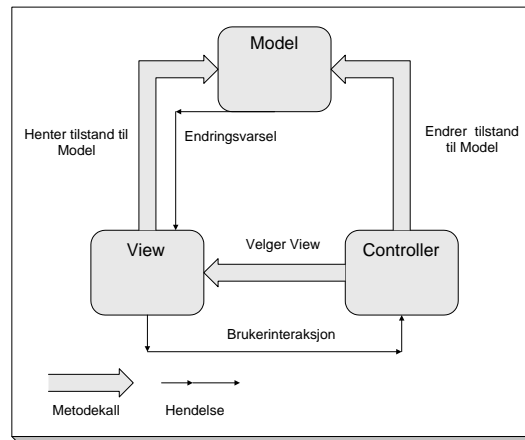
Model-View-Controller (MVC) er valgt som arkitektonisk mønster og presenteres i avsnitt 11.1. Videre presenteres designmønsteret *Composite* som benyttes som en del av modellen i KjApp i avsnitt 11.2. Deretter presenteres klassediagrammer og sekvensdiagrammer som beskriver KjApp i avsnitt 11.3 og 11.4. Kapittelet avsluttes med en beskrivelse av viktige algoritmer i avsnitt 11.5.

### 11.1 Model-View-Controller

MVC er et velprøvd arkitektonisk mønster som blant annet er brukt ved utvikling av Java [?]. MVC benyttes i systemer med høy grad av interaksjon fra brukerne. MVC skiller brukergrensesnittet (*View*) og informasjonen som skal presenteres (*Model*) på en ryddig måte og reduserer ansvaret til delene i systemet. Dermed følger mønsteret et av hovedprinsippene i objektorientert design - *et objekt skal ikke ha ansvaret for for mye*.

MVC er et enkelt mønster som likevel viser seg å være effektivt. MVC tvinger utvikleren til å dele applikasjonen inn i tre moduler - *Model* (Modellen), *View* (Brukergrensesnitt) og *Controller* (Kontroller). Det er flere fordeler med å benytte MVC. MVC passer godt når det er behov for presentere samme informasjon på forskjellige måter, for eksempel tekstlig og grafisk. MVC sørger for klarhet i designet av applikasjonen og gjør implementering og vedlikehold enklere. Ved å beholde grensesnittene mot de andre modulene, kan modulene i MVC byttes ut. Dette gjør MVC applikasjoner modulare. I tillegg tillater dette at MVC applikasjoner vokser. Et eksempel kan være at modellen utvides for å ta inn nye aspekter. Gamle versjoner av brukergrensesnitt og kontroller kan fortsatt benyttes mot en utvidet modell. For å kunne nyttiggjøres seg nye deler av modellen, må kontrolleren og brukergrensesnittet utvides og ta i bruk nye grensesnitt mot modellen.

En applikasjon basert på MVC kan i tillegg distribueres ved bruk av et par *proxies*. *Proxy* er et designmønster beskrevet i [?]. Figur 11.1 viser modulene i MVC og interaksjonen mellom disse. *Model* utgjør kjernen i applikasjonen. *Model* tar vare på



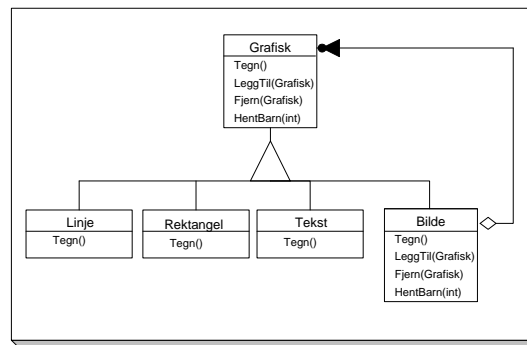
Figur 11.1: Konseptuelt diagram av MVC

tilstand og data til elementene i applikasjonen. Når modellen oppdateres, skal dette reflekteres i *View*. Dette skjer ved at *Model* varsler *View* om at endring har funnet sted. *View* benyttes for å presentere *Model* for brukerne av applikasjonen. Til dette henter *View* nødvendig informasjon fra *Model*. *Controller* manipulerer *Model* og sørger for at den oppdateres på bakgrunn av hendelser fra brukerne av applikasjonen.

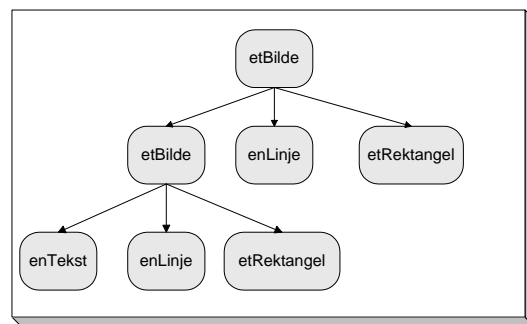
Vi har valgt å benytte MVC fordi applikasjonen som skal utvikles har høy grad av brukerinteraksjon. Videre ser vi det som en fordel å skille ulike deler av applikasjonen på en ryddig måte og dermed begrense ansvaret til komponentene i applikasjonen. Det er også en fordel at en MVC applikasjon kan distribueres ved hjelp av et par *proxies*. Utvikling av en distribuert versjon av KjApp er en naturlig forlengelse av arbeidet og i denne forbindelse kan en løsning med bruk av *Proxy* studeres.

## 11.2 Composite

*Composite* (sammensetning) er et strukturerende mønster for objekter. Mønsteret benyttes for å kunne behandle en trestruktur av objekter uniformt. Med mønsteret kan komplekse strukturer bygges opp av enkle primitiver. Figur 11.2 viser et eksempel fra [?] hvor et grafisk objekt bygges opp ved hjelp av primitivene linje, rektangel og tekst. Klassen *Bilde* fungerer som en kontainer for primitivene. For å sikre uniform behandling av primitiver og kontainer defineres klassen *Grafisk*. *Grafisk* er en abstrakt klasse og representerer både primitiver og kontainer. I *Grafisk* deklarerer metoden *Tegn*, samt fellesmetoder for å aksessere og håndtere barnenoder. Metoden *Tegn* må implementeres spesielt for hvert enkelt primitiv. Primitivene har ingen barnenoder og har derfor ikke metoder for å håndtere og aksessere barn. Klassen *Bilde* implementerer *Tegn* ved å kalle tegnetoden til sine barn. Fordi grensesnittet til *Bilde* er i overensstemmelse med grensesnittet til *Grafisk*, kan man ved hjelp av rekursjon sette sammen bilder ved hjelp av andre bildeobjekter. På denne måten dannes et hierarki av grafiske objekter. En rekursiv komposisjon er vist i figur 11.3.



Figur 11.2: Designmønsteret *Composite*

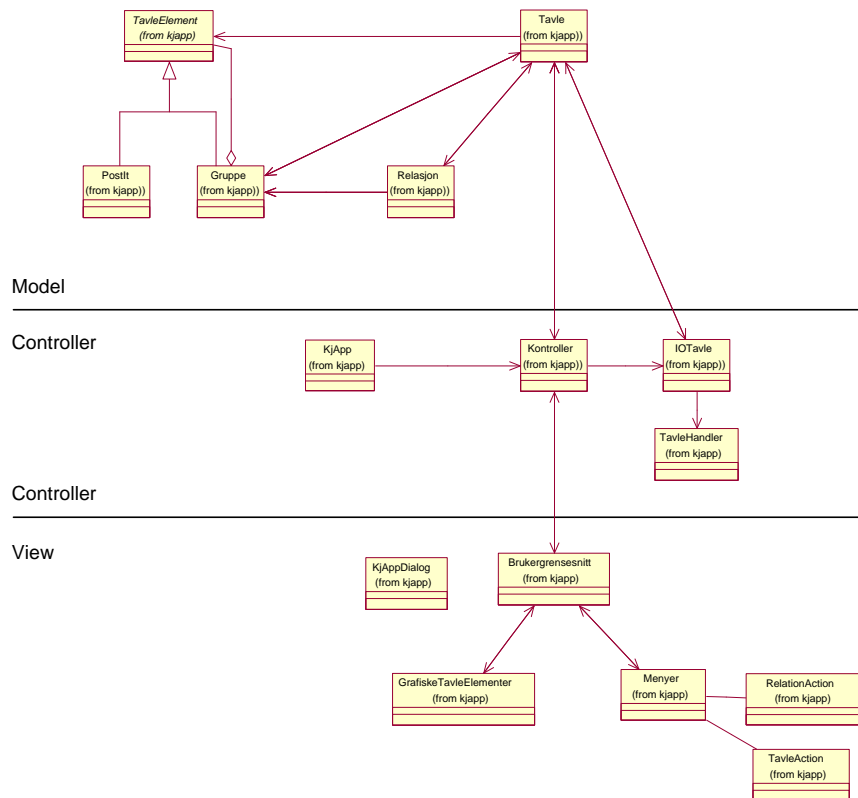


Figur 11.3: Eksempel på en komposisjon av bildeobjekter

I *KjApp* benyttes *Composite* til å representere tavleelementer i en *KJ* sesjon. Klassen *TavleElementer* tilsvarer *Grafisk*, mens klassen *Gruppe* tilsvarer *Bilde*. *PostIt* er en primitiv på lik linje med rektangel, tekst og linje (Se figur 11.4).

## 11.3 Klassediagram for KjApp

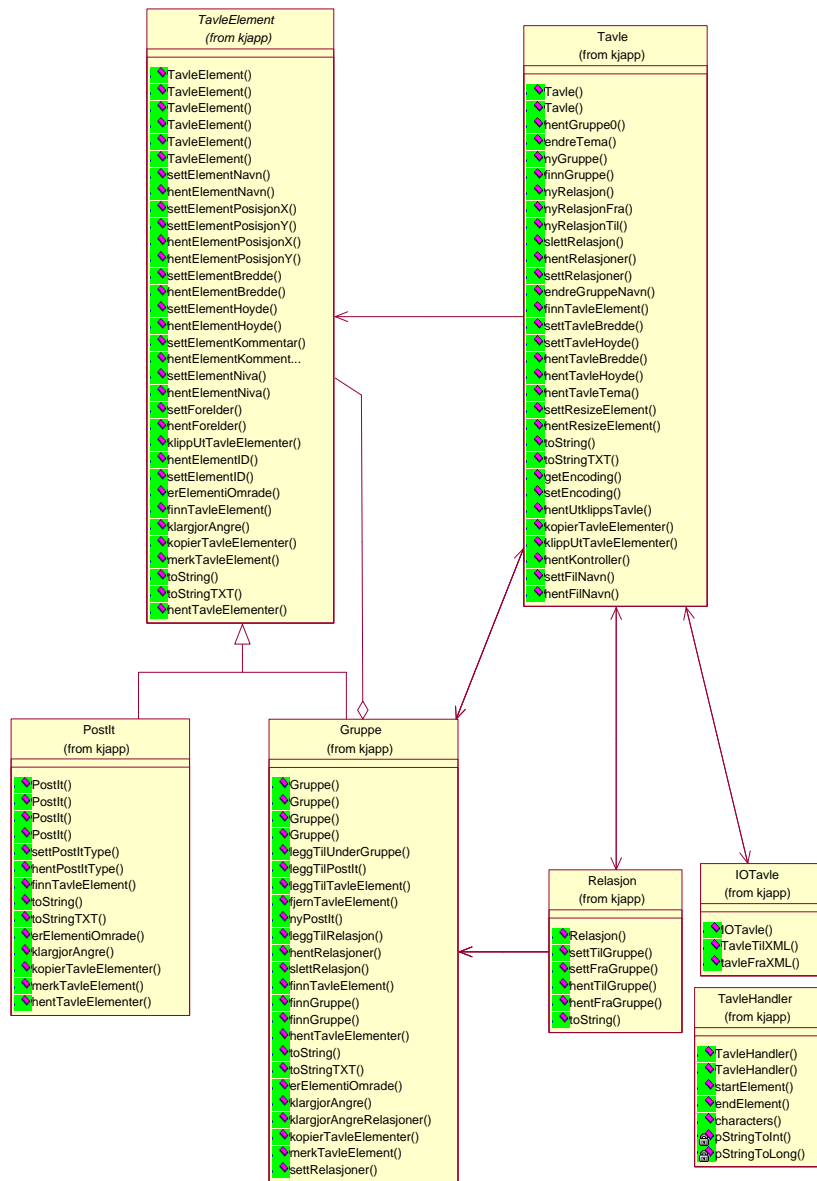
Avsnittet beskriver konseptuelt klassediagram av KjApp vist i figur 11.4. For å skille moduler og klasser fra hverandre benyttes engelske betegnelser når modulene i MVC diskuteres. For eksempel består modulen *View* av klassene *Brukergrensesnitt*, *GrafiskeTavleElementer* og *Menyer*.



Figur 11.4: Overordnet klassediagram for kjApp

### 11.3.1 Model

Klassediagrammet for *Model* i figur 11.5 representerer elementer i en KJ sesjon og det logiske forholdet mellom disse. Designmønsteret *Composite* er brukt mellom *TavleElement*, *PostIt* og *Gruppe*. Med *Composite* dannes en trestruktur med grupper og post-it lapper. Diagrammet viser i tillegg klassen *Relasjon* og *Tavle*. *Relasjon* benyttes for å knytte to grupper sammen, mens *Tavle* representerer tavlen i en KJ sesjon. *Tavle* skal ta vare på alle grupper og post-it lapper som genereres. Vi har valgt å definere en *nullgruppe* hvor alle andre tavleelementer inngår. *Nullgruppen* ligger i *Tavle* og sørger for ensartet prosessering i forbindelse med søk etter tavleelementer. Dette forholdet er ikke vist i klassediagrammet.



Figur 11.5: Klassediagram, Model

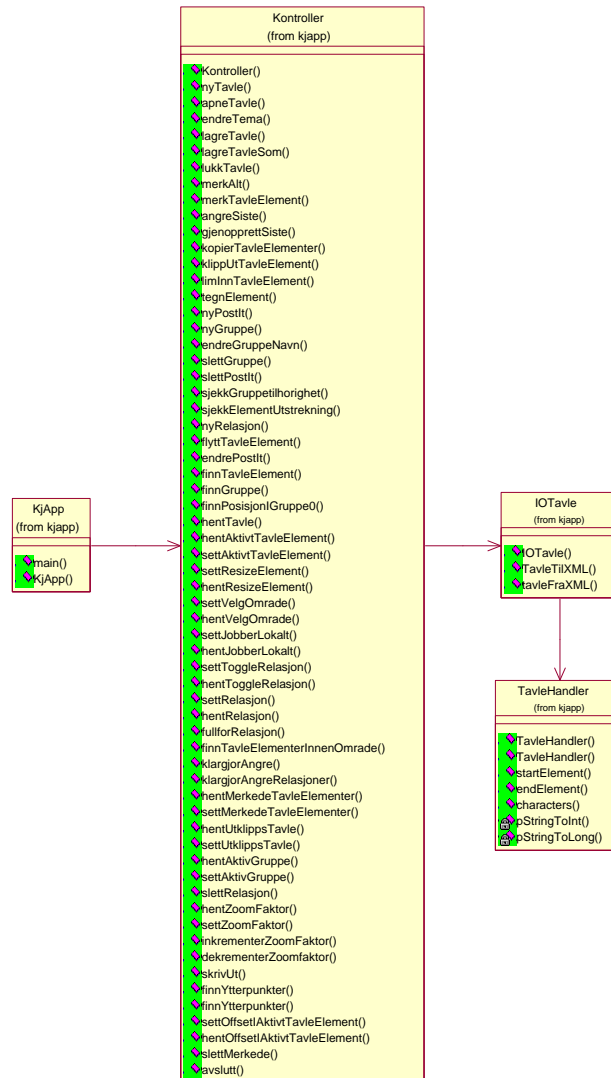
### 11.3.2 View

*View* består av grafiske tavle elementer og menyer. *GrafiskeTavleElementer* visualiserer *Model* ved å tegne post-it lapper, grupper og relasjoner på tavlen. I tillegg implementerer *GrafiskeTavleElementer* tastatur- og muslytter som sørger for at kontrolltaster og musehendelser initierer riktige operasjoner. Når *View* registrerer hendelser som krever oppdatering av *Model* (for eksempel at en post-it lapp klikkes), kontakter *View Controller*. *Controller* kontakter videre *Model* for oppdatering og når dette er gjort, ber *Controller* om at *View* oppdateres (her: post-it lapp merkes). Klassen *Menyer* gir brukeren tilgang til funksjonalitet via meny- og verktøylinje og høyre-klikk menyer. Klassediagram for *View* er vist i figur 11.6.

### 11.3.3 Controller

*Controller* sørger for at *View* til en hver tid reflekterer *Model*. *Controller* tar i mot henvendelser fra *View* og på bakgrunn av disse initieres oppdateringer i *Model*. Når *Model* er endret, skal *Controller* initiere oppdateringer i *View*. På denne måten binder *Controller* *Model* og *View* sammen. Klassediagram for *Controller* er vist i figur 11.7.





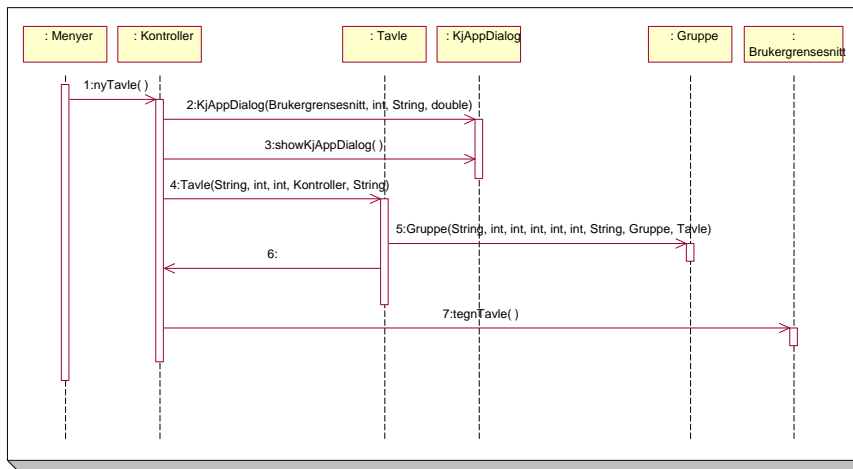
Figur 11.7: Klassediagram, Controller

## 11.4 Sekvensdiagrammer

Avsnittet beskriver et utvalg sekvenser som er viktige for forståelsen av applikasjonen som er implementert. Omtalte sekvenser skal dekke de vanligste forløp som forekommer under kjøring. Sekvensene er beskrevet i henhold til nummerering i hver enkelt figur.

### 11.4.1 Ny tavle

Figur 11.8 beskriver metodekall når en ny tavle opprettes. Sekvensen initieres ved at en bruker velger *Ny tavle* fra filmenyen.



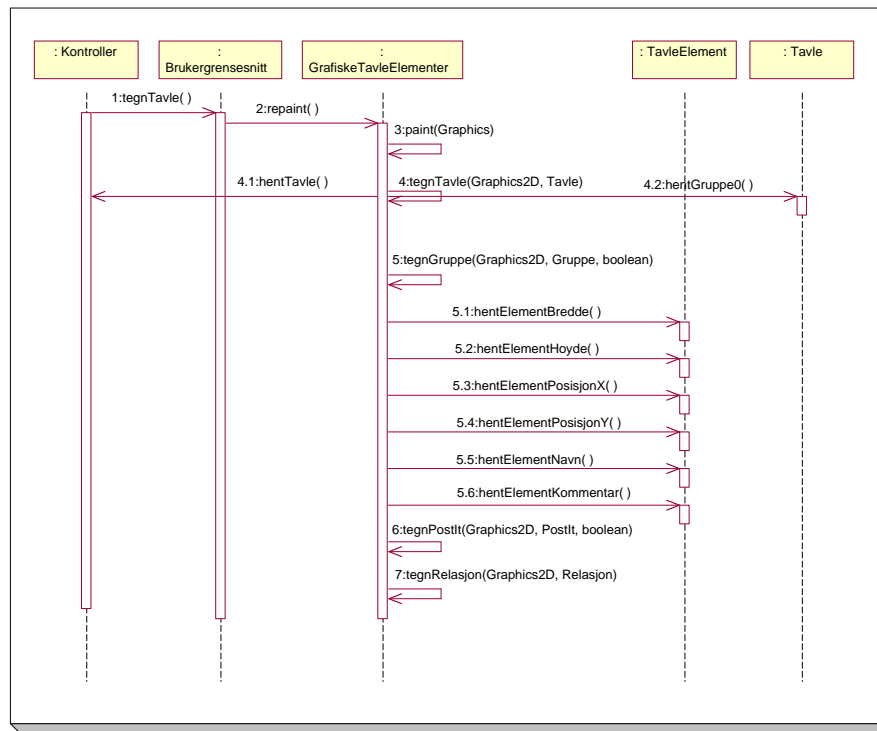
Figur 11.8: Sekvensdiagram for å lage en ny tavle

1. `nyTavle()`  
Kalles i `Kontroller` når lytter i klassen `Menyer` registrerer inngangsverdien *Ny tavle*. Metoden starter en dialog som gir brukeren mulighet til å angi et tema for sesjonen.
2. `KjAppDialog()`  
Oppretter instans av klassen `KjAppDialog` av typen *Egenskaper for Tavle*.
3. `showKjAppDialog()`  
Viser `KjAppDialog` og tar inn tema som benyttes som tema for tavlen
4. `Tavle()`  
Metoden `nyTavle` oppretter en ny tavle ved hjelp av konstruktøren i `Tavle` med angitt tema fra (3) som argument.
5. `Gruppe()`  
Konstruktøren i `Gruppe` kjøres for å opprette nullgruppen. Tema for nullgruppen er lik tema for tavlen.
6. Retur fra `4:Tavle()`  
Funksjonen returnerer nyopprettet tavle.

7. tegnTavle()  
Brukergrensesnittet tegner opp tavlen på nytt.

## 11.4.2 Tegn Tavle

Figur 11.9 viser hvordan en Tavle blir tegnet opp fra Kontroller.



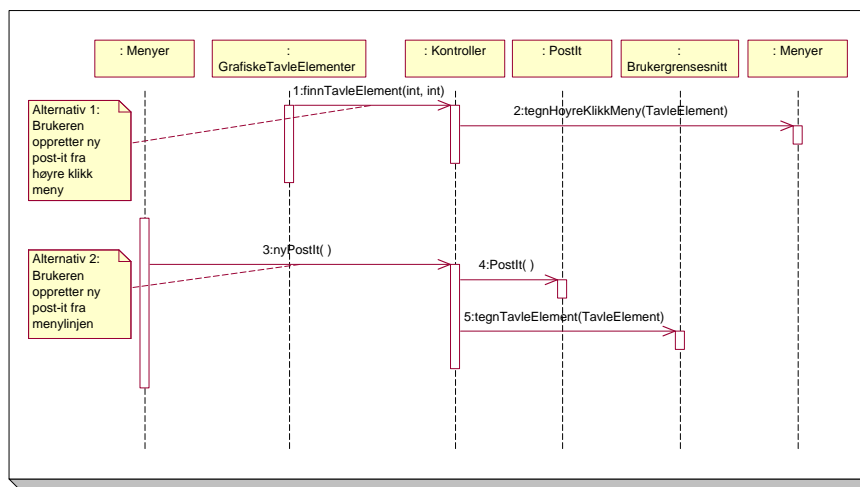
Figur 11.9: Sekvensdiagram for å tegne en tavle

1. tegnTavle()  
Metoden tegnTavle() kalles for eksempel når et tavleelement flyttes i tavlen.
2. repaint()  
repaint() kalles for å oppdatere JPanel (forelderklasse til GrafiskeTavleElementer).
3. paint()  
paint() er overlagret i klassen GrafiskeTavleElementer. Kaller paint() i JPanel før den kaller metoden tegnTavle().
4. tegnTavle()  
Henter tavlen som skal tegnes opp (4.1). Nullgruppen inneholder alle elementer som skal tegnes opp. Nullgruppen hentes derfor fra tavlen (4.2). Metoden tegner opp alle elementer som ligger på laveste nivå i tavlen ved hjelp av metodene tegnGruppe og tegnPostIt.

5. tegnGruppe  
tegnGruppe tar et objekt av typen Gruppe som argument. Informasjon om gruppens bredde (5.1), høyde (5.2), posisjon (5.3 og 5.4), tema (5.5) og kommentar (5.6) hentes fra argumentet før gruppen tegnes på grafikkobjektet til GrafiskeTavleElementer. Dersom gruppen inneholder andre tavleElementer (gruppe eller post-it) tegnes disse opp med metodene tegnGruppe() eller tegnPostIt(). Metoden er dermed rekursiv. Eventuelle underelementer tegnes opp på samme måte som denne gruppen ved at informasjon om posisjon, utsteking og utseende hentes fra argumentet.
6. tegnPostIt()  
tegnPostIt() tar et objekt av typen PostIt som argument. Objektet tegnes opp på samme måte som en gruppe ved at informasjon om elementets posisjon, utsteking og utseende hentes fra argumentet. En PostIt kan imidlertid ikke inneholde andre tavleElementer og metoden er derfor ikke rekursiv.
7. tegnRelasjon  
Når alle tavleElementer er tegnet opp, tegnes relasjoner i tavlen.

### 11.4.3 Ny post-it

En ny post-it lages enten ved at brukeren høyreklikker ett sted i tavlen for så å velge *Ny post-it* fra menyen som kommer opp eller ved at brukeren velger *Ny post-it* fra menylinjen. Begge situasjonene er beskrevet i figur 11.10. Dersom brukeren høyreklikker for å opprette ny post-it gjelder punkt (1) til og med (5) i figur 11.10 mens punkt (3) til (5) gjelder for tilfellet hvor brukeren velger å opprette en ny post-it fra menylinjen.



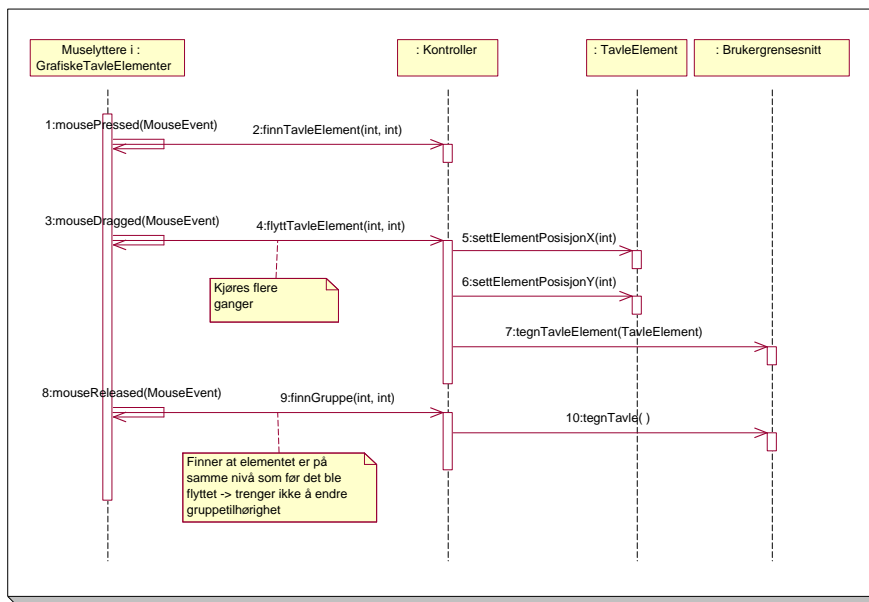
Figur 11.10: Sekvensdiagram for å lage en ny post-it

1. finnTavleElement()  
Muselytteren i GrafiskeTavleElementer registrerer høyreklikk og kjører metoden finnTavleElement i Kontroller for å finne ut hvilket element som er valgt. Metoden finnTavleElement er nærmere beskrevet i avsnitt 11.5.1

2. tegnHoyreKlikkMeny()  
Kontroller benytter Menyer til å tegne opp riktig meny for valgt element. I dette tilfellet tegnes høyreklikkmeny for gruppe.
3. nyPostIt()  
Brukeren velger *Ny post-it* fra menylinje eller høyreklikkmeny og lytteren i klassen Menyer kaller metoden nyPostIt i Kontroller.
4. PostIt()  
Kontroller instansierer et objekt av typen PostIt. PostIt legges inn i gruppen som høyreklikkmenyen ble åpnet i eller i nullgruppen.
5. tegnPostIt()  
Kontroller kjører metoden tegnTavleElement() i Brukergrensesnitt med det nye PostIt objektet som argument.

#### 11.4.4 Flytt tavleelement

Figur 11.11 viser hvordan et tavleelement flyttes. Metoden benytter seg av en sekvens med muselyttere. Først registreres en mousepressed hendelse som benyttes for å finne elementet som skal flyttes. Deretter registreres et antall mousedragged hendelser som fører til animert bevegelse av tavleelementet. Til slutt registreres en moureleased hendelse som *rydder opp* etter bevegelsen. Det innebærer at man kontrollerer hvilken gruppe elementet har havnet i og oppdaterer modellen.



Figur 11.11: Sekvensdiagram for å flytte et TavleElement

1. mousePressed()  
Muselytter registrerer at brukeren trykker ned knappen og holder den inne

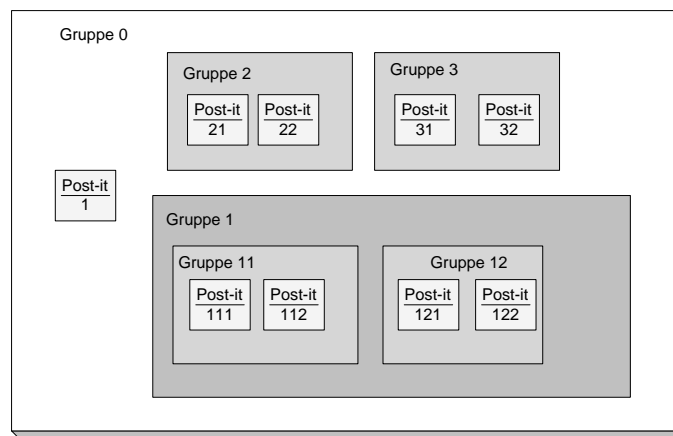
2. `finnTavleElement(int,int)`  
Finner ut hvilket element brukeren har trykket på. I dette tilfellet har brukeren trykket på en gruppe. Metoden `finnTavleElement` er nærmere beskrevet i avsnitt 11.5.1
3. `mouseDragged()`  
Muselyttere registrerer at musen dras med tavleelementet funnet i punkt 2. Denne hendelsen blir kalt helt til brukeren slutter å dra musen
4. `flyttTavleElement()`  
Metoden `flyttTavleElement` i kontroller kjøres for å oppdatere modellen ved hjelp av 5,6 og 7. Metoden blir kalt hver gang `mouseDragged` hendelsen blir kalt
5. `settElementPosisjonX()`  
Oppdaterer gruppens X posisjon.
6. `settElementPosisjonY()`  
Oppdaterer gruppens Y posisjon.
7. `tegnTavleElement()`  
Tegner tavleelementet på nytt for hver oppdatering av posisjon
8. `mouseReleased()`  
`mouseReleased` hendelse detekteres av muselyttere. Dette indikerer at brukeren ikke lenger flytter gruppen
9. `finnGruppe(int,int)`  
Finner gruppen som flyttet tavleelement skal legges inn i. Denne metoden vil returnere nullgruppen dersom tavle elementet ikke legges inn i en av gruppene på tavlen.
10. `tegnTavle()`  
Tegner hele tavlen på nytt etter at bevegelsen er fullført. Alle posisjoner i modellen er nå oppdatert.

## 11.5 Algoritmer

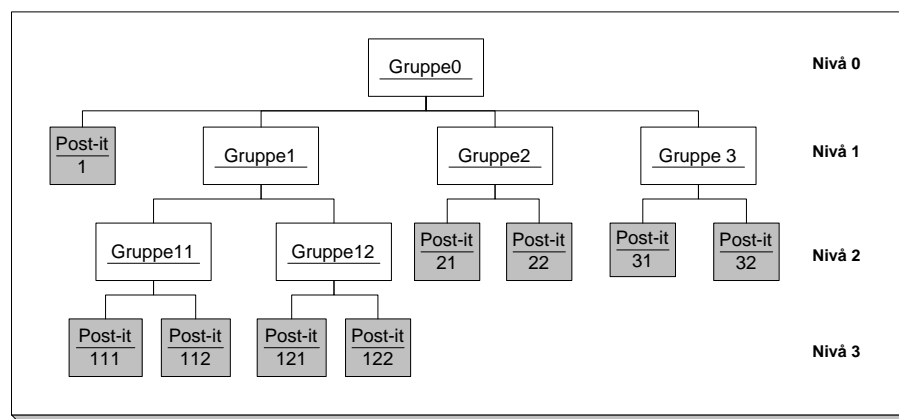
Avsnittet beskriver viktige algoritmer i KjApp. Først beskrives algoritmen for å finne frem til et element på tavla på bakgrunn av brukerens museklikk. Deretter beskrives zooming av tavle og algoritmen for å tegne tavlen.

### 11.5.1 Algoritme for å finne tavle elementer

I KjApp er både post-it lapper og grupper tavleelementer. Grupper kan inneholde post-it lapper og andre grupper, mens post-it lapper ikke kan ha noen barn. Når brukeren trykker på et tavleelement, finnes elementet ved hjelp av museklikk-posisjonen og informasjon i modellen om tavleelementenes posisjon og utstrekning. Tavleelementene vises på tavla som vist i figur 11.12 og danner samtidig trestrukturen som figur 11.13 viser.



Figur 11.12: Layout i tavle for tavleelementer



Figur 11.13: Trestruktur for tavleelementer

Pseudokoden som følger beskriver hvordan et tavleelement finnes. Pseudokoden i *Tavle* kalles en gang, mens pseudokoden i *Gruppe* kalles rekursivt. På tavlen ligger *nullgruppen* som inneholder alle grupper og post-it lapper på nivå en (se figur 11.13). Algoritmen starter ved at *Tavle* setter kandidat til å være nullgruppen. I *Gruppe* leter algoritmen etter andre kandidater. Dette innebærer at algoritmen forsøker å finne andre elementer i nullgruppen som inneholder posisjonen som er trykket. Dersom algoritmen finner en ny gruppe, kjøres algoritmen rekursivt på denne. Algoritmen avsluttes når en post-it lapp er funnet eller når alle elementer i en gruppe er besøkt og ingen ny kandidat er funnet. Gruppen eller post-it lappen som er funnet returneres til *Tavle*.

#### finnTavleElement i Tavle

```
finnTavleElement(posisjon){
  kandidat := nullgruppen.finnTavleElement()
}
```

#### finnTavleElement i Gruppe

Metoden finner ut om en posisjon ligger innefor et element ved å se på elementenes posisjon, høyde og bredde. For å gjøre algoritmen mer lesbar, vises ikke hvordan algoritmen finner ut om posisjonen ligger innenfor eller utenfor elementet som behandles.

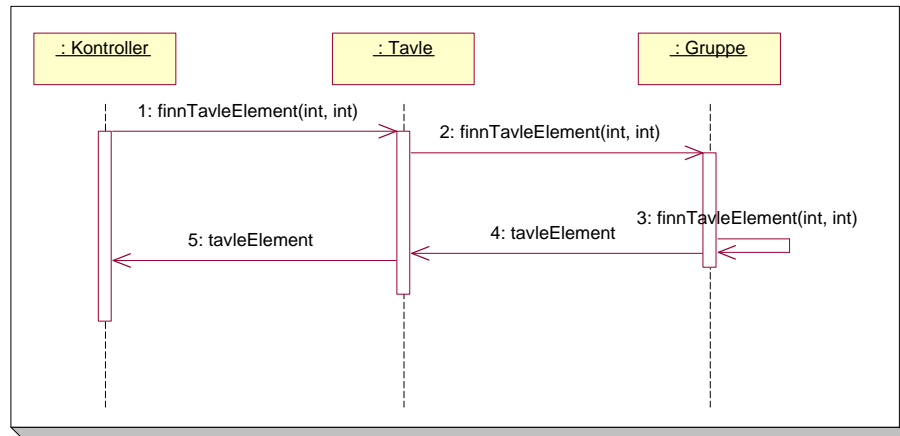
```
finnTavleElement(posisjon){
  kandidat := <denne gruppen>
  for <alle elementer i denne gruppen> do:
    if(posisjon innenfor elementet)
      then kandidat := element;
    end if
  end for

  if kandidat == <denne gruppen> eller kandidat.type() == postit then
    return kandidat
  else
    kandidat.finnTavleElement(posisjon);
  end if
}
```

#### Sekvensdiagram for å finne tavle elementer

Figur 11.14 viser sekvensdiagrammet for metoden `finnTavleElement`. Metoden tar inn en posisjon som parameterverdi som angir hvor på skjermen musen er klikket.

1. `finnTavleElement(int,int)`  
Kontrollerer kjører `finnTavleElement` i aktiv tavle
2. `finnTavleElement(int,int)`  
Tavle kjører `finnTavleElement` på nullgruppen
3. `finnTavleElement(int,int)`  
Gruppen kjører `finnTavleElement` på alle aktuelle undergrupper
4. Retur fra `finnTavleElement` i Gruppe  
`finnTavleElement` i Gruppe returnerer enten en post-it eller en gruppe

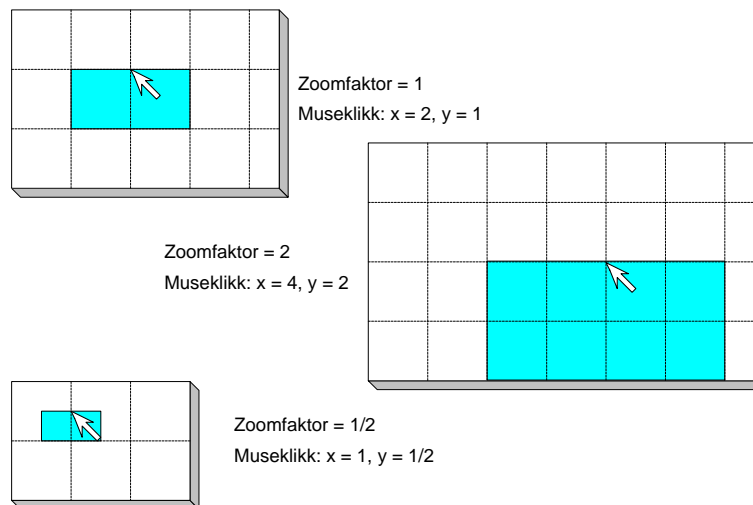


Figur 11.14: Sekvensdiagram for å finne et tavle element

5. Retur fra finnTavleElement i Tavle  
 finnTavleElement i Tavle returnerer enten en post-it eller en gruppe

### 11.5.2 Zooming av tavle

Zooming av innholdet i tavlen foregår ved at en zoomfaktor settes i kontrolleren. Zoomfaktoren benyttes for å representere objekter grafisk og for å håndtere interaksjon fra brukeren. Posisjoner og størrelser som tegnes opp i tavlen multipliseres med zoomfaktoren. Posisjoner angitt ved museklikk fra brukeren divideres med zoomfaktoren. Zoom-



Figur 11.15: Zoomfaktor

faktoren ligger i området fra 1/2 til 2. Dette medfører at den grafiske representasjonen av tavlen kan variere fra halvparten til det dobbelte i størrelse i forhold til modellen.

Standardverdi for zoomfaktoren er 1. Algoritmene i neste avsnitt viser at alle størrelser og posisjoner multipliseres med zoomfaktor før de visualiseres på tavlen. Konseptet er vist i figur 11.15 hvor et rektangel først vises med zoomfaktor = 1, deretter zoomfaktor = 2, og tilslutt zoomfaktor = 1/2. I figuren trykkes musen på det samme stedet i rektangelet for de forskjellige zoomfaktorene, og museklikkets absolutte posisjon er oppgitt. Omregnet posisjon for museklikket vil i de tre tilfellene være den samme:  $x = 2$ ,  $y = 1$ .

### 11.5.3 Algoritme for å tegne tavle

Modellen representeres grafisk ved hjelp av View. Mer spesifikt er det klassen GrafiskeTavleElementer i View som har ansvaret for å tegne opp tavlen. Dette gjøres ved hjelp av metodene tegnTavle, tegnGruppe og tegnPostIt. Metoden tegnGruppe gjør rekursive kall til seg selv for å tegne eventuelle undergrupper med innhold. Algoritmene for metodene presenteres i avsnittene som følger.

#### tegnTavle

Algoritmen for tegnTavle setter tavlens tema til å være navnet til nullgruppen. Videre hentes alle elementer i nullgruppen (det vil si nivå 1 i figur 11.13) og tegnes opp med tegnGruppe eller tegnPostIt. Til slutt tegnes relasjoner i tavlen.

```
tegnTavle(graphics, tavle) {
    sett tavletema := nullgruppens navn

    for (alle tavleElementer i nullgruppen) {
        if (tavleElement er av typen Gruppe) {
            tegnGruppe(graphics, gruppe)
        } else {
            tegnPostIt(graphics, postIt)
        }
    }

    for (alle relasjoner i tavla) {
        tegnRelasjon(graphics, relasjon)
    }
}
```

#### tegnGruppe

Algoritmen for tegnGruppe henter informasjon om gruppen som skal tegnes fra modellen. Posisjoner og størrelser multipliseres med zoomfaktor før gruppen tegnes på tavlen (som beskrevet i avsnitt 11.5.2). Videre hentes tavleelementene i gruppen og tegnes opp med tegnGruppe eller tegnPostIt. Til slutt merkes gruppen dersom den er aktivt tavleelement.

```
tegnGruppe(graphics, gruppe) {
    navn = hent gruppe navn fra modellen
    bredde = hent gruppebredde fra modellen * zoomfaktor
    hoyde = hent gruppehøyde fra modellen * zoomfaktor
    posisjon = hent posisjon fra modellen * zoomfaktor
}
```

```
tegn opp gruppen på graphics objektet
tegn gruppenavn

for (alle tavleElementer i gruppen){
  if (tavleElement er av typen Gruppe){
    tegnGruppe(graphics,gruppe)
  }else{
    tegnPostIt(graphics,postit)
  }
}
merk gruppen dersom den er aktivtTavleElement i kontrolleren
}
```

### **tegnPostIt**

Algoritmen for tegnPostIt henter informasjon om post-it som skal tegnes fra modellen. Posisjoner og størrelser multipliseres med zoomfaktor før post-it tegnes på tavlen. Til slutt merkes post-it dersom den er aktivt tavle element.

```
tegnPostIt(graphics,postit){
  navn = hent postitnavn fra modellen
  bredde = hent postitbredde fra modellen * zoomfaktor
  hoyde = hent postithøyde fra modellen * zoomfaktor
  posisjon = hent posisjon fra modellen * zoomfaktor

  postit farge settes avhengig av postittype
  tegn postit
  tegn postitnavn

  merk postit dersom den er aktivtTavleElement i kontrolleren
}
```

# Kapittel 12

## Implementering

### Innhold

---

12.1 Model . . . . .	91
12.2 View . . . . .	92
12.3 Controller . . . . .	98

---

Implementeringen beskrives med utgangspunkt i det arkitektoniske mønsteret MVC beskrevet i kapittel 11. Først beskrives *Model*, deretter *View* og tilslutt *Controller*.

### 12.1 Model

*Model* representerer elementene i en KJ sesjon. *TavleElement*, *Gruppe* og *PostIt* knyttes sammen med designmønsteret *Composite* som er beskrevet i avsnitt 11.2. *Tavle*

Tabell 12.1: Klasser i Model

---

Klasse	Beskrivelse
Tavle	<i>Tavle</i> representerer tavlen i et møterom. Tavlen inneholder metoder for å manipulere tavleelementer og relasjoner i tavlen. Nullgruppen i tavlen gir tilgang til alle tavleelementer.
TavleElement	<i>TavleElement</i> er en abstrakt klasse som beskriver felles egenskaper for klassene <i>Gruppe</i> og <i>PostIt</i> . Klassen inneholder abstrakte metoder som må implementeres av <i>Gruppe</i> og <i>PostIt</i> og metoder som arves direkte.
Gruppe	Klassen inneholder spesielle egenskaper for grupper. <i>Gruppe</i> har blant annet en vektor med relasjoner som gruppen inngår i, og inneholder den rekursive metoden <i>finnTavleElement</i>
PostIt	Klassen inneholder spesielle egenskaper for <i>PostIt</i> . <i>PostIt</i> har en statusvariabel som kan være negativ, nøytral eller positiv
Relasjon	En relasjon opprettes mellom to grupper. Relasjonen har en referanse til sin fra- og tilgruppe.

---

tar vare på objekter ved hjelp av nullgruppen. Nullgruppen håndteres spesielt i KjApp. For eksempel har brukeren ikke muligheten til å flytte eller endre størrelsen på denne gruppen. Nullgruppen kan heller ikke inngå i andre grupper. *Relasjon* benyttes for å opprette relasjoner mellom grupper. Relasjonsobjekter har referanser til gruppene de knytter sammen. I tillegg har gruppene en referanse til sine relasjoner. Dette er gjort for å forenkle sletting av relasjoner tilhørende en gruppe som slettes. Klassene som utgjør modellen er beskrevet i tabell 12.1.

## 12.2 View

View består av seks klasser. De mest sentrale klassene er *Brukergrensesnitt* og *GrafiskeTavleElementer*. *Brukergrensesnitt* arver fra klassen *JFrame*. Klassen inneholder menyer som er tilgjengelig for brukeren. Menylinje og verktøylinje opprettes ved hjelp av metoder i *JFrame*. Menyer bygges opp og tilpasses med klassen *Menyer*. Knapper i menyene opprettes ved hjelp av Actions definert i klassene *TavleAction* og *RelationAction*. Dette innebærer at samme lytterprosess kan benyttes for flere knapper. Menyene er implementert slik for å sikre at like hendelser håndteres likt, uavhengig av om brukeren trykker på knapp i verktøy-, høyreklikk- eller vanlig meny. Klassene for *View* beskrives i tabell 12.2.

Tabell 12.2: Klasser i View

Klasse	Beskrivelse
Brukergrensesnitt	Arver fra klassen <i>JFrame</i> . En tavle legges til <i>Brukergrensesnitt</i> som en instans av klassen <i>GrafiskeTavleElementer</i> . Funksjonalitet fra <i>JFrame</i> benyttes for å legge til menylinje og verktøylinje
GrafiskeTavleElementer	Arver fra klassen <i>JPanel</i> . Klassen tegner opp elementene i <i>Model</i> . <i>Repaint</i> metoden kaller metoden <i>tegnTavle</i> som ved hjelp av rekursjon tegner elementene i tavlen. Videre implementerer klassen grensesnittene <i>MouseListener</i> , <i>MouseMotionListener</i> , <i>KeyListener</i> og <i>Printable</i>
Menyer	Klassen setter sammen menyer. Metodene i klassen returnerer et menyobjekt som legges til <i>Brukergrensesnitt</i> . Menyobjekter opprettes ved hjelp av Actions som beskrives nedenfor. To knapper kan dermed ha samme lytterprosess
TavleAction	<i>TavleAction</i> arver fra <i>AbstractAction</i> . Klassen benyttes for å håndtere hendelser for alle knapper unntatt relasjonsknappen. Klassen gjør det mulig å tilegne hver Action et navn, et ikon og et tall som benyttes for hendelseshåndtering
RelationAction	Klassen forenkler sletting av relasjoner fra gruppens høyreklikkmeny
KjAppDialog	Klassen benyttes for å åpne dialogbokser. Et tall angir hvilken type dialog som skal opprettes. Eksempler på dialoger er endring av tavletema og egenskaper

### 12.2.1 Komponenter benyttet i View

For å bygge opp brukergrensesnittet i KjApp benyttes komponenter fra Swing pakken som følger med versjon 1.4 av Java2SE. Swing tilbyr komponenter for å bygge opp brukergrensesnittet til en applikasjon. Vi har benyttet JFrame som kontainer med støtte for å legge til meny- og verktøylinje. Når en tavle åpnes opprettes et JPanel som er en annen type kontainer. Grafikkobjektet for dette panelet benyttes til å representere modellen grafisk.

### 12.2.2 Muselyttere i GrafiskeTavleElementer

Avsnittet beskriver flyten i forbindelse med musehendelser fra brukeren. Det er opprettet flytdiagrammer for musehendelser i KjApp. Flytdiagram for mousePressed, mouseDragged og mouseReleased presenteres i avsnittene som følger. Det er viktig å merke seg at metodene kjøres som kombinasjoner av hverandre. En kombinasjon av flytdiagrammene må derfor studeres for å få et bilde av hva som skjer i applikasjonen. Alternativene er:

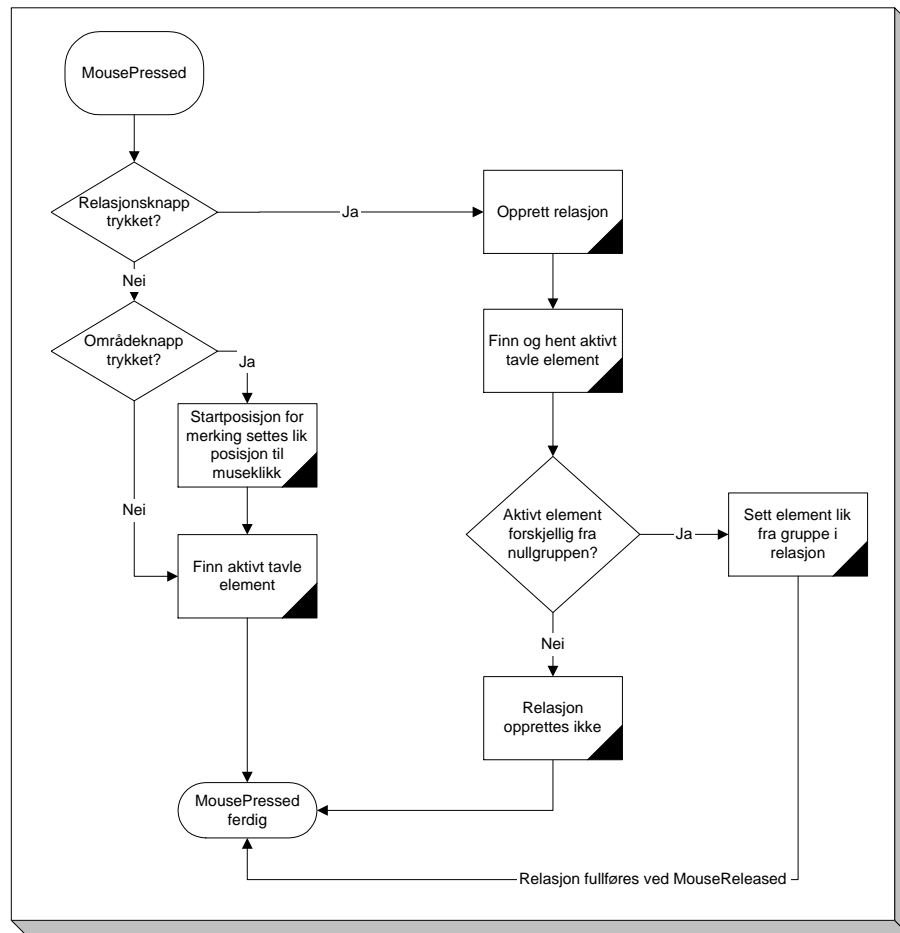
1. mousePressed - mouseDragged  $\times$  n - mouseReleased
2. mousePressed - mouseReleased

#### Flyt for mousePressed

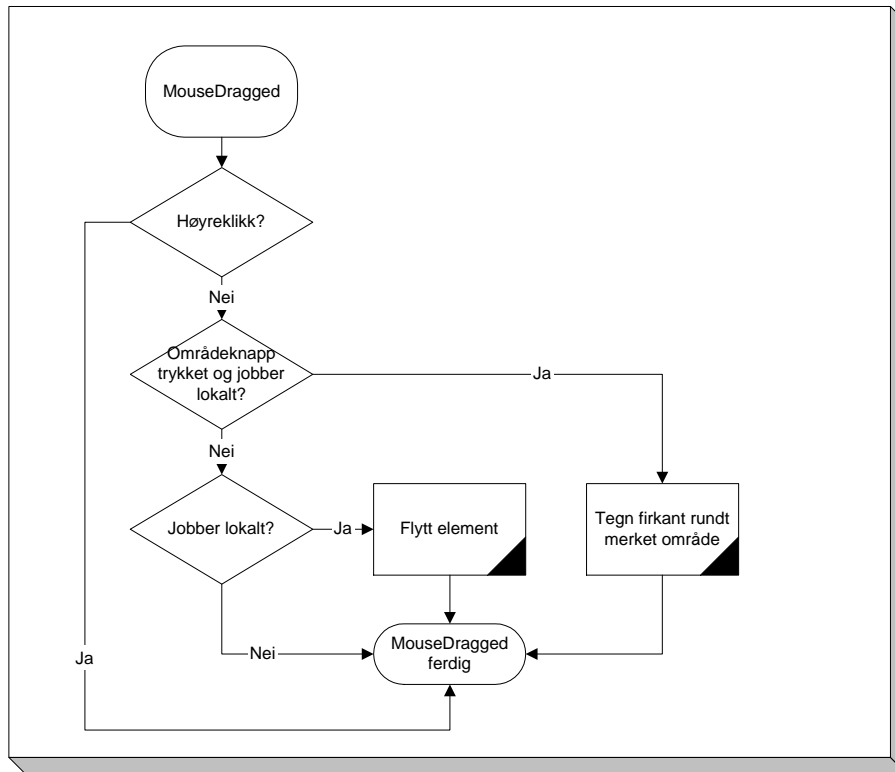
Figur 12.1 viser et flytdiagram for hendelsen mousePressed. Når musen trykkes kontrolleres først status for relasjonsknappen i verktøylinja. Dersom den er trykket, opprettes en relasjon uten fra- og tilgruppe. Deretter kjøres metoden for å finne tavleelementet ved trykket posisjon. Dersom elementet som finnes er forskjellig fra nullgruppen settes elementet som fragruppe i relasjonen som ble opprettet. Hvis ikke droppes relasjonen. Tilgruppen i relasjonenn settes først når museknappen slippes (se figur 12.3). Hvis relasjonsknappen ikke er trykket kontrolleres status for områdeknappen i verktøylinja. Ved aktiv områdeknapp settes startposisjon for merking lik trykket posisjon. Deretter finnes aktivt tavleelement uavhengig av om områdeknappen var trykket eller ikke og mousePressed metoden avsluttes. MousePressed metoden etterfølges alltid av enten mouseDragged og eller mouseReleased metoden.

#### Flyt for mouseDragged

Figur 12.2 viser flytdiagrammet for mouseDragged. Metoden initieres et antall ganger etter mousePressed, og etterfølges av mouseReleased. Dersom klikket ikke er høyreklikk kontrolleres det om områdeknappen er trykket og om man jobber lokalt. I så fall tegnes en firkant som angir det merkede området. Hvis områdeknappen ikke er trykket men man jobber lokalt kjøres en metode for å flytte et element. Dette fører til at bevegelsen på tavlen blir animert ved at elementet som er i bevegelse oppdateres hver gang mouseDragged kalles. Hvis man ikke jobber lokalt avsluttes mouseDragged metoden.



Figur 12.1: Flytdiagram for mousePressed



Figur 12.2: Flytdiagram for mouseDragged

### Flyt for mouseReleased

Figur 12.3 viser flytdiagram for mouseReleased. Metoden initieres etter mousePressed eller mouseDragged. Dersom klikket er høyreklikk settes aktiv posisjon lik trykket posisjon og en høyreklikkmeny tegnes før mouseReleased avsluttes. Hvis klikket ikke er høyreklikk, kontrolleres status for relasjonsknappen. Dersom den er aktiv fullføres relasjonen som ble påbegynnt i mousePressed ved å angi tilgruppen. Ellers kontrolleres status for områdeknappen. Dersom denne er trykket settes knappens status til inaktiv og sluttposisjon for merket område settes lik trykket posisjon. Dersom det finnes elementer i det merkede området, tegnes merking på elementene før metoden avsluttes. Hvis ikke avsluttes metoden uten å gjøre noe mer. Hvis områdeknapp ikke er trykket settes ny aktiv gruppe til gruppen musen ble sluppet i. Deretter flyttes det aktive elementet som ble satt i mousePressed metoden til posisjonen hvor musen ble sluppet. Dersom variabelen resizeElement i kontrolleren ikke er sann kontrolleres gruppetilhørigheten til elementet som ble flyttet. Dersom dette tidligere ikke tilhørte gruppen som ble funnet som aktiv gruppe, settes aktiv gruppe som ny forelder for elementet. Tilslutt sjekkes elementets utstrekning dersom det aktive elementet er forskjellig fra nullgruppen og variabelen resizeElement i kontrolleren settes til ikke sann før metoden avsluttes.

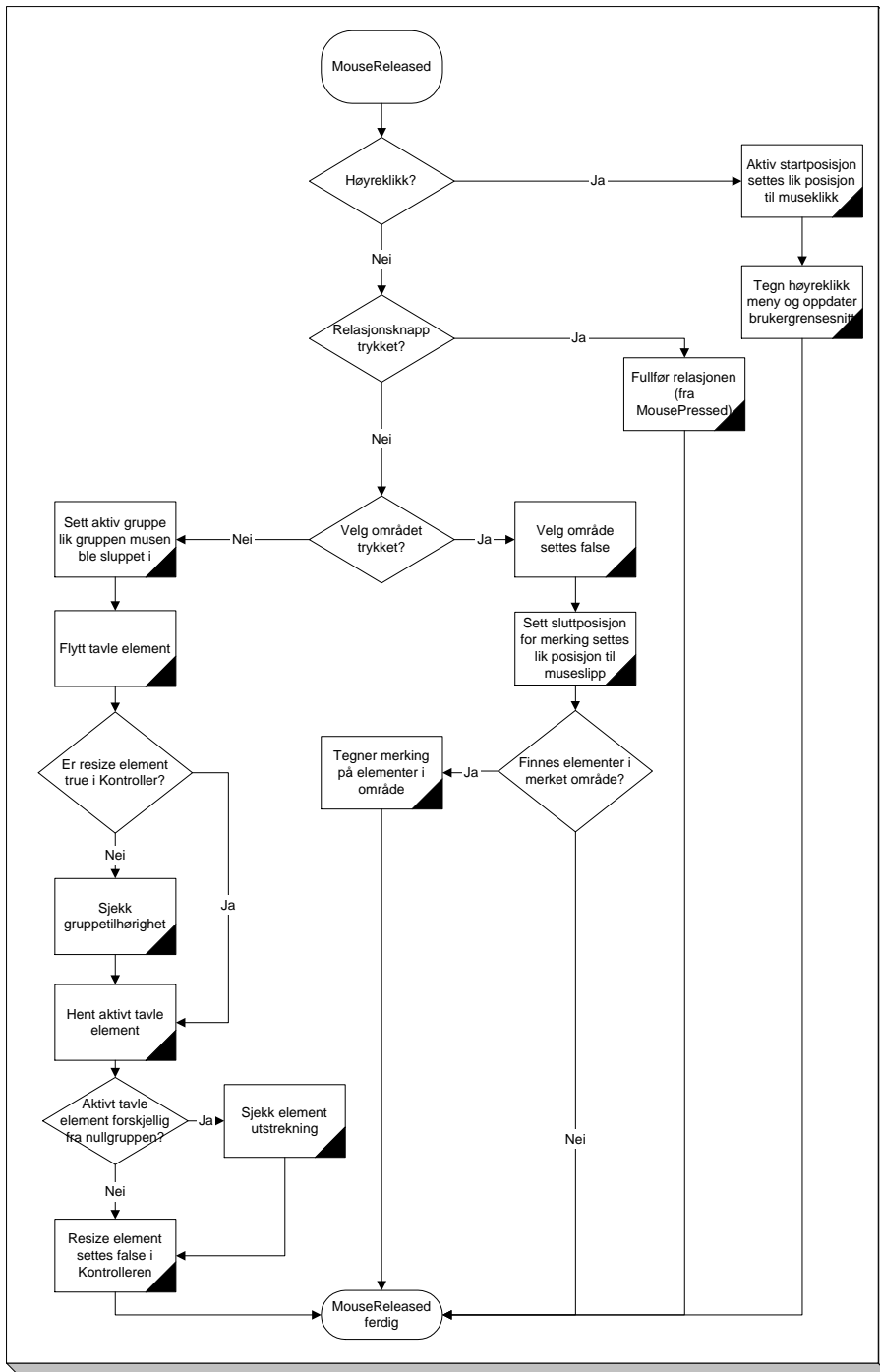
### 12.2.3 Knappelyttere i GrafiskeTavleElementer

For å forenkle bruken av KjApp har vi implementert standard hurtigknapper for applikasjonen. Lytteren for disse hendelsene er lagt til panelet som inneholder den grafiske representasjonen av tavlen. Dette medfører at panelet må være aktivt for at lytteren skal fungere. Lytteren starter metoder i kontrolleren på lik linje med lyttere på menyobjekter. Dette medfører at bruk av hurtigknapper stort sett gjenspeiler bruken av menyer i meny- eller verktøylinja. Hurtigknapper for følgende operasjoner er implementert:

```

ctrl + A          = merk alt
ctrl + C          = kopier
ctrl + X          = klipp ut
ctrl + V          = lim inn
ctrl + S          = lagre
ctrl + O          = åpne tavle
ctrl + Z          = angre
ctrl + shift + Z = gjenopprett
del               = slett merkede elementer / aktivt element

```



Figur 12.3: Flytdiagram for mouseReleased

## 12.3 Controller

*Controller* består av fire klasser. Det meste av funksjonaliteten er plassert i klassen *Kontroller* som håndterer kommunikasjonen mellom *View* og *Model*. *Kontroller* har til enhver tid oversikt over aktivt tavleelement og tilstanden til applikasjonen. Når hendelser fanges opp i *View* vil passende metoder initieres i *Controller*. For eksempel vil metoden *nyPostIt* kalles i *Kontroller* dersom *ny post-it* trykkes i en av menyene. Metoden oppdaterer *Model* ved å opprette en ny *post-it* med posisjonen lik *aktiv posisjon* fra klassen *Bruker grensesnitt*. Dersom *View* registrerer en *mouseDragged* hendelse, oppdaterer *Controller* posisjonen til elementet som dras og sørger for at tavlen tegnes på nytt for å gjenspeile endringene. Videre tar klassen *Kontroller* vare på informasjon for angring og gjenoppretting av de siste operasjoner på tavlen. Klassene som utgjør kontrolleren er beskrevet i tabell 12.3.

Tabell 12.3: Klasser i Controller

Klasse	Beskrivelse
KjApp	I klassen ligger KjApps main metode som instansierer et objekt av typen <i>Kontroller</i> . Videre settes <i>Windows look and feel</i>
Kontroller	Klassen har ansvaret for kommunikasjon mellom <i>View</i> og <i>Model</i> . Den håndterer alle forespørsler fra <i>View</i> om oppdateringer på <i>Model</i> , og ber <i>View</i> oppdatere seg etter at endringene er gjennomført i <i>Model</i> . Har oversikt over tilstanden til applikasjonen. Dette gjelder for eksempel status for tilstandsvariable, aktivt tavleelement (gruppe eller postit), angre/gjenopprett informasjon og zoomfaktor for tavlen
IOTavle	Hjelpeklasse for lesing fra og skriving til fil. Klassen håndterer foreløpig lesing fra XML filer og skriving til filtypene XML og txt.
TavleHandler	Hjelpeklasse for lesing fra XML fil

# Kapittel 13

## Test

### Innhold

---

<b>13.1 Systemtest</b> . . . . .	<b>99</b>
<b>13.2 Brukbarhetstest</b> . . . . .	<b>100</b>
<b>13.3 Distribuert test</b> . . . . .	<b>101</b>

---

Under utvikling av KjApp har vi foretatt metode- og modultesting. Implementasjonen av KjApp har vært use case basert, og funksjonell testing av alle use case er foretatt etterhvert som disse er implementert. Etter at funksjoner for å manipulere elementer i tavlen var implementert gjennomførte vi en test i distribuerte omgivelser for å studere applikasjonens oppførsel og ytelse ved bruk av applikasjonsdeling i NetMeeting. Etter at applikasjonen var ferdig utførte vi systemtest for å verifisere de funksjonelle kravene. I tillegg gjennomførte vi en brukbarhetstest for å finne ut om vi måtte gjøre endringer for å bedre brukbarheten før vi startet eksperimentet. Avsnittene i dette kapitlet beskriver systemtesten, brukbarhetstesten og distribuert test for KjApp.

### 13.1 Systemtest

Systemtesten verifiserer at de funksjonelle kravene til applikasjonen oppfylles av den implementerte prototypen. For hvert krav i avsnitt 10.1 beskrives en test. Testen i sin helhet er lagt ved som tillegg E.1. Gjennomføringen av testen viser at prototypen fungerer best dersom brukere benytter høyreklikkmenyer der disse er tilgjengelig. Dette kommer av at posisjoner og elementer en hendelse skal manipulere ikke alltid er riktig dersom menylinje og verktøylinje benyttes. Vi har allikevel valgt å si at disse tilfellene er godkjent i testen. Dette kommer av at det ble presisert for deltakerne i eksperimentet at de skulle benytte høyreklikkmenyer der disse var tilgjengelige. Videre mener vi at applikasjonen innfrir de krav som stilles når det er mulig å gjennomføre en operasjon på en enkel måte.

#### Applikasjonsnivå

Testen ble ikke bestått for FA1-Skriv ut. Skriv ut funksjonen har fungert i miljø med lokal skriver, men fungerer ikke i nettverksmiljø. Vi har imidlertid ikke prioritert å ferdigstille denne funksjonaliteten, da den ikke er relevant for eksperimentet som skal

gjennomføres. FA2-Avslutt ble bestått, men brukeren blir spurt om lagring selv om tavlen er lagret.

### Tavlenivå

Kravene FT4-Lagre tavle og FT5-Lagre tavle som ble bestått med kommentarer. Spesialtegn som <, > og / må omskrives når XML filen lagres for at det skal være mulig å åpne filen. Imidlertid forsvinner ikke noe informasjon under lagring. Filen kan derfor rettes opp manuelt dersom feilen oppstår. FT10-Gjennopprett siste er godkjent, men angre knappen må trykkes to ganger før gjenopprettknappen gjøres tilgjengelig for brukeren. Feilen kommer av at statusinformasjon for knapper ikke oppdateres på riktig måte når angreknappen trykkes. Funksjonen for å gjenopprette fungerer som den skal. For kravet FT13-Lim inn oppstår det problemer når posisjonsinformasjon ikke er oppdatert. Problemet oppstår ikke dersom lim inn knappen i høyreklikkmenyen benyttes.

### Gruppe- og Post-it lapp nivå

Kravene for gruppe og post-it lapp nivå er godkjent uten kommentarer. Vår erfaring er imidlertid også her at KjApp fungerer best dersom høyreklikkmenyer benyttes der det er mulig. Dette kommer av at aktiv posisjon settes hver gang en høyreklikkmeny åpnes. Det er derfor aldri noen tvil om hvor et element skal opprettes eller hvilket element som skal oppdateres når høyreklikkmenyer benyttes.

## 13.2 Brukbarhetstest

Brukbarhetstesten har som mål å avdekke eventuelle problemområder for brukere og verifisere at applikasjonen er logisk oppbygd. Brukbarhetstesten skal teste de områder av applikasjonen som er relevant i forhold til eksperimentet. Testen som ble benyttet er lagt ved som tillegg E.2. Problemer som ble avdekket er presentert i de følgende avsnitt. Brukbarhetstesten ble gjennomført med fire forsøkspersoner som ikke har vært involvert i utviklingen av KjApp, men som er vant med å bruke Windows programvare.

### Hvor skal ideen skrives

For å gi brukeren mulighet til å utdype ideene sine opprettet vi et kommentarfelt i dialogboksene som angir egenskaper for gruppe og post-it. Dette førte imidlertid til at brukeren ble usikker på hvor ideen skulle skrives for at den skulle bli synlig på skjermen. Årsaken til dette var at feltene opprinnelig het *navn* og *kommentar*. Feilen er nå rettet opp ved å endre ledetekst fra *navn* til *idetekst*. I og med at dette er et *idegenererings program* mener vi at dette nå bør være selvforklarende.

### Knappelyttere på dialogbokser

En av deltakerne i brukbarhetstesten mente at default idetekst burde være merket når dialogboks for egenskaper åpnes. Videre påpekte han at det var tungvint å benytte musen for å klikke OK knappen etter at ideteksten er skrevet inn. Vi implementerte derfor merking av idetekst og knappelyttere for feltet ideteksten skrives inn i. Dette medfører at dersom brukeren trykker linjeskift mens tekstmarkøren står i feltet for ideteksten er dette det samme som å trykke OK knappen med musen. Videre implementerte vi en lytter for tabulatortasten i kommentarfeltet for å forenkle navigasjon i dialogboksen.

### Merking av tavle elementer og oppretting av gruppe

Grupper kan opprettes på flere forskjellige måter i KjApp. Brukeren har tilgang til tre forskjellige måter å opprette en tom gruppe. Grupper kan opprettes fra meny- og verktøylinje eller fra høyreklikkmeny. Dersom brukeren markerer et område før en gruppe opprettes vil alle elementer som inngår i området inkluderes i gruppen som opprettes. Brukeren hadde i dette tilfellet to problemer. Da testen ble foretatt fungerte merking av område bare fra *oppe til venstre og ned mot høyre*. Brukbarhetstesten avdekket imidlertid at merkingen må fungere i alle retninger for ikke å skape problemer for brukeren. Videre fantes ingen indikasjon på at applikasjonen var i *merkemodus*. Etter brukbarhetstesten ble begge disse forholdene rettet opp. Det er nå mulig å markere områder i alle retninger og musepekeren endres når applikasjonen er i merkemodus.

### Endre størrelse på tavle elementer

Størrelsen på tavleelementer kan endres ved å trykke på et skravert felt nederst i høyre hjørne. Brukeren må trykke i det skraverte feltet (ikke på det svarte rektangelet som angir at elementet er merket). Mange windowsbaserte programmer tillater endring av størrelse i alle hjørner av et element. Dette førte til at mange forsøkte denne fremgangsmåten også i KjApp. Vi har ikke gjort noe for å rette opp dette problemet.

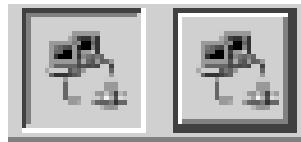
### Relasjoner

Testen avdekket at brukerne hadde problemer med å opprette og slette relasjoner mellom grupper. En relasjon opprettes ved først å klikke på relasjonsknappen på verktøylinjen. Deretter klikkes en gruppe, musknappen holdes nede og dras til relasjonsgruppen. Dersom brukeren utfører andre operasjoner etter at relasjonsknappen er klikket, vil applikasjonen gå ut av *relasjonsmodus*. Dermed må brukeren klikke relasjonsknappen på nytt om en relasjon fortsatt skal opprettes. Sletting av relasjon foregår ved at brukeren høyreklikker i en gruppe med relasjon. Deretter velges slett relasjon fra høyreklikkmenyen og brukeren velger den relasjonen han ønsker å slette. Brukerne i testen forsøkte å slette relasjonen ved å klikke på selve relasjonen på tavlen. Dette fungerer ikke og ga heller ikke brukeren noen tilbakemelding. Problemet er ikke rettet opp, men deltakerne i eksperimentet ble gjort oppmerksom på hvordan funksjonen skal benyttes.

## 13.3 Distribuert test

For at KjApp skal fungere som verktøystøtte for distribuert KJ må applikasjonsdelingsfunksjonen i NetMeeting benyttes. Et møte i NetMeeting settes opp ved at alle som ønsker å være med i møtet starter sin NetMeeting. Deretter *ringer* en av deltakerne de andre som skal være med i sesjonen ved hjelp av deres IP adresser. Når alle som skal være med på møtet har svart på anropet starter en av deltakerne programmet KjApp. KjApp deles deretter i møtet ved hjelp av funksjonen *application sharing* i NetMeeting. Deltakerne kan velge hvem som skal ha kontrollen over den delte applikasjonen ved hjelp av funksjonalitet i NetMeeting. For å teste at KjApp fungerte tilfredsstillende i distribuerte omgivelser satte vi opp et NetMeeting møte med to deltakere og den ene deltakeren startet og delte ut KjApp. Deretter gjennomførte den andre deltakeren brukbarhetstesten (se tillegg E.2) i distribuerte omgivelser. Testen avdekket at animasjonen som benyttes når et tavleelement flyttes på tavlen, og når et område skal markeres

førte til store forsinkelser i dataoverføringen. Årsaken til dette ligger sannsynligvis i at NetMeeting oppdaterer hele skjermbildet hver gang en liten endring forekommer i den delte applikasjonen. Ved animering blir dermed kapasiteten på dataoverføringen for liten, siden skjermbildet oppdateres hver gang mouseDragged hendelsen initieres. Problemet ble løst ved å implementere en ny knapp i verktøylinjen som gir brukeren mulighet til å slå av og på animasjon. Knappen er vist i figur 13.1 både for distribuert (til venstre), og lokal modus. I koden løses dette ved at *View* ikke tegner opp *Model* på nytt før hele bevegelsen er fullført.



Figur 13.1: Knapp for å velge distribuert eller lokal modus for KjApp

# Kapittel 14

## Installasjon og brukerveiledning

### Innhold

---

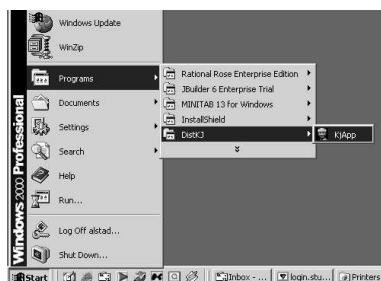
14.1	Installasjon av KjApp	103
14.2	Brukerveiledning	104

---

Kapittelet beskriver først hvordan KjApp installeres i avsnitt 14.1. Deretter følger en brukerveiledning til KjApp i avsnitt 14.2. For en introduksjon til hvordan man kan sette opp et møte ved hjelp av NetMeeting, se avsnitt 3.7.3 og *Help* i NetMeeting.

### 14.1 Installasjon av KjApp

I tillegg G finnes en CD med programvare som må installeres for å kjøre KjApp. Dersom CDen ikke starter automatisk åpnes filen index.html som ligger på roten på CDen. Siden som kommer opp beskriver stegene som er nødvendig for å installere og kjøre KjApp. Installasjonen oppretter programgruppe DistKJ under programmer i startmenyen. I programgruppen og på skrivebordet finnes en snarvei for å starte KjApp (figur 14.1). Applikasjonen kan avinstalleres ved å gå til *Add/Remove programs* i Windows kontrollpanel. Dersom KjApp skal benyttes i distribuerte omgivelser må også



Figur 14.1: Snarvei til KjApp

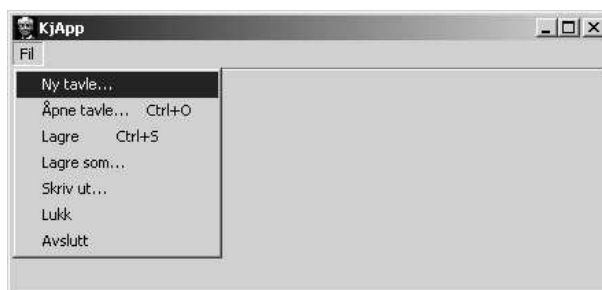
Microsoft NetMeeting være installert på alle deltakeres PC'er. Microsoft NetMeeting er standard programvare i Windows 2000, og finnes under Start -> Programs -> Accessories -> Communications.

## 14.2 Brukerveiledning

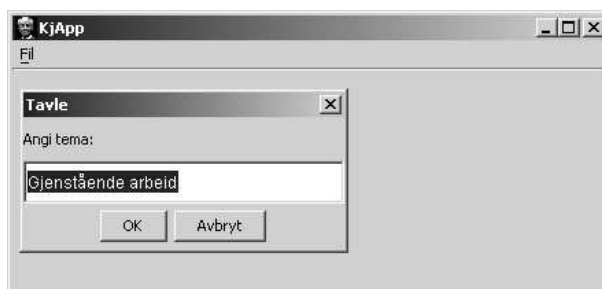
Brukerveiledningen beskriver de mest vanlige funksjonene i KjApp. Hvert avsnitt tar for seg en funksjon som illustreres med en figur. De fleste funksjoner er tilgjengelig fra forskjellige menyer, men brukerveiledningen beskriver den mest effektive fremgangsmåten.

### Ny tavle

Ny tavle opprettes ved å velge *Ny tavle* fra filmenyen som vist i figur 14.2. Dersom brukeren ønsker å åpne en eksisterende tavle velger han *Åpne tavle*. Når ny tavle trykkes får brukeren opp en dialogboks hvor tema for tavlen angis. Tema plasseres senterert øverst i tavlen. Dialogboksen for å angi tema vises i figur 14.3. Dialogboksen er også tilgjengelig ved å høyreklikke i tavlen og velge *Egenskaper for tavle*.



Figur 14.2: Opprette ny tavle



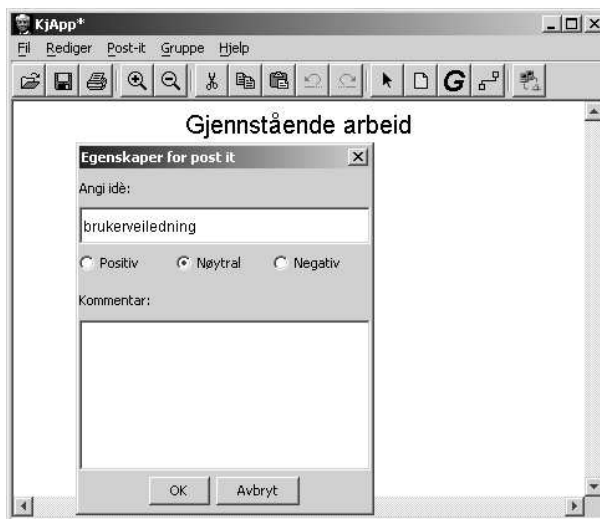
Figur 14.3: Angi tema for ny tavle

### Opprett ny post-it

En ny post-it opprettes ved å velge *Ny post-it* fra høyreklikkemenyen som vist i figur 14.4. Ny post-it plasseres der brukeren har høyreklikket. Dersom høyreklikket er i en gruppe, opprettes post-it i denne gruppen. Når brukeren trykker *Ny post-it* får han opp en dialogboks for å angi egenskaper (se figur 14.5). Brukeren skriver ideen i feltet merket *Angi idé*. Brukeren angir også hvilken type post-it som skal opprettes. Dersom ideteksten beskriver en positiv ide, velges positiv. Post-it får da grønn farge. Dersom brukeren velger nøytral eller negativ blir post-it henholdsvis gul eller rød. Nøytral er standard. Nederst i dialogboksen kan brukeren skrive utfyllende kommentarer. Kommentarer er ikke synlig på tavlen og er mest aktuell for en moderator som ønsker å utdype ideer som fremkommer i en KJ sesjon.



Figur 14.4: Opprett ny post-it



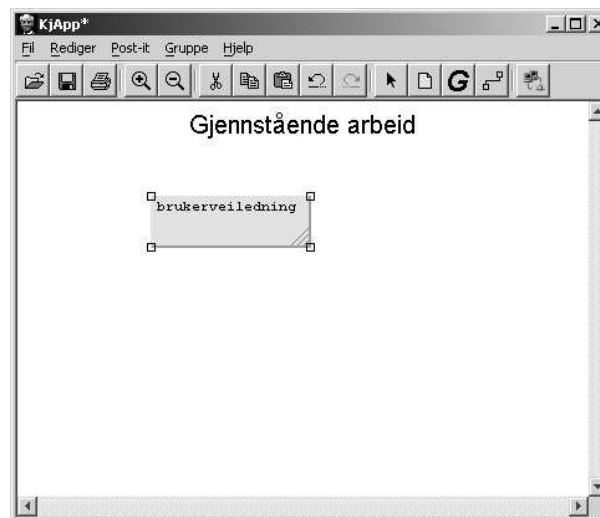
Figur 14.5: Angi egenskaper for ny post-it

### Post-it er opprettet

Nøytral post-it med idetekst *brukerveiledning* tegnes på tavlen som vist i figur 14.6. Legg merke til at teksten brytes der den overstiger bredde på post-it. Størrelsen på post-it endres slik at ideteksten blir mer leselig ved å trykke i det skraverte feltet nederst til høyre på post-it og dra til ønsket størrelse. Se figur 14.7.



Figur 14.6: Post it slik den blir opprettet på tavlen



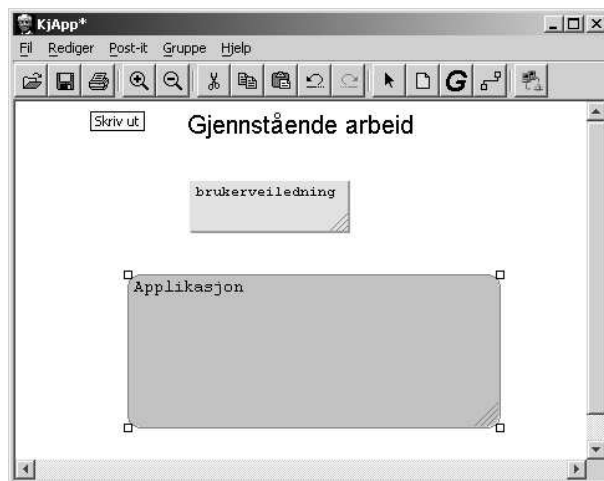
Figur 14.7: Post it etter størrelsen er tilpasset

### Ny gruppe

En tom gruppe opprettes ved å høyreklikke i tavlen og velge *Ny gruppe* vist i figur 14.8. Brukeren får opp en dialogboks for å angi egenskaper for gruppen. Egenskaper er beskrivende navn og kommentarer. Gruppen plasseres der musen er klikket. Gruppen *Gjennstående arbeid* opprettes på tavlen som vist i figur 14.9. Grupper kan inneholde post-it lapper og andre grupper. Størrelsen på en gruppe endres på samme måte som for post-it.



Figur 14.8: Opprett ny gruppe

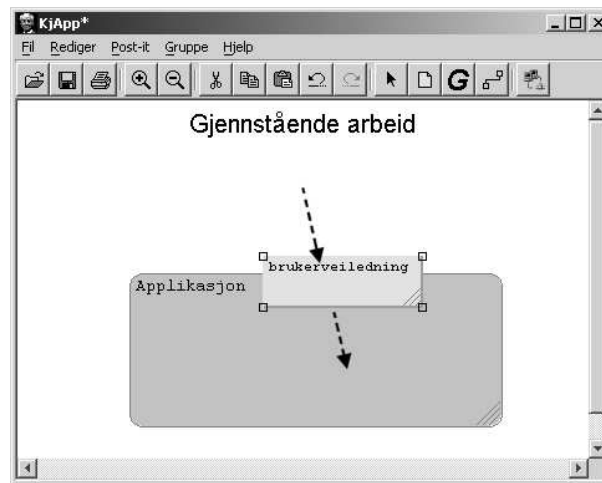


Figur 14.9: Ny gruppe er opprettet på tavlen *Gjennstående arbeid*

### Flytting av post-it

En post-it eller gruppe flyttes ved å klikke på elementet og dra det til ønsket posisjon. Dersom applikasjonen er i lokal modus animeres bevegelsen. I distribuert modus tegnes

elementet på nytt først etter at museknappen er sluppet. Flytting av post-it er illustrert i figur 14.10.



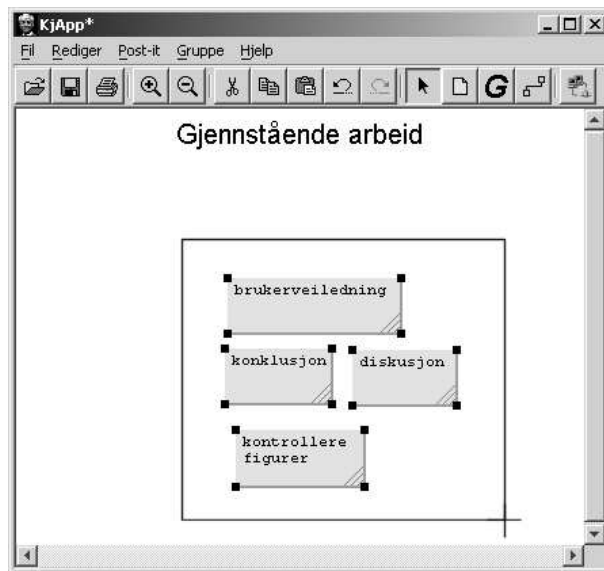
Figur 14.10: Flytting av post-it

### Merking av elementer

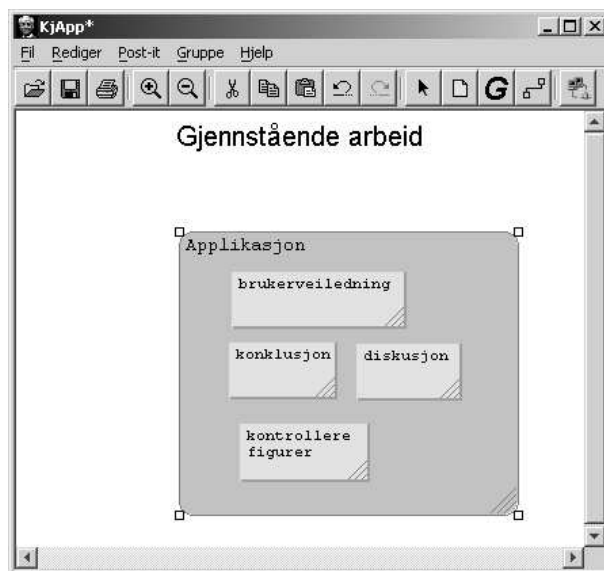
Et område som inneholder flere elementer kan merkes ved å trykke på pilen i verktøylinjen. Når musen så trykkes og dras i tavlen vil et område markeres og elementer innenfor området merkes når musen slippes. Applikasjonen går ut av merkemodus. Merkede elementer kan slettes, kopieres, klippes ut eller inkluderes i en ny gruppe. Merket område er vist i figur 14.11.

### Opprett ny gruppe som inkluderer elementer i merket område

I en lokal KJ sesjon grupperes ideer ved at deltakeren flytter post-it lapper med relaterte ideer til klynger på tavlen. Når deltakerne er enige om plassering av lappene markeres grupper ved at en strek tegnes rundt hver klynge. I KJApp kan brukeren markere en klynge med post-it lapper som beskrevet ovenfor. Deretter opprettes en gruppe som inkluderer alle merkede lapper ved å trykke på knappen **G** i verktøylinjen. Gruppens utsrekning er lik markert område. Gruppen i figur 14.12 inkluderer alle lapper som er markert i figur 14.11.



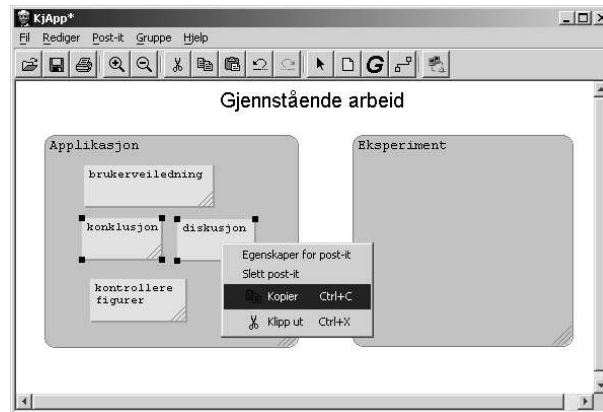
Figur 14.11: Merking av et område



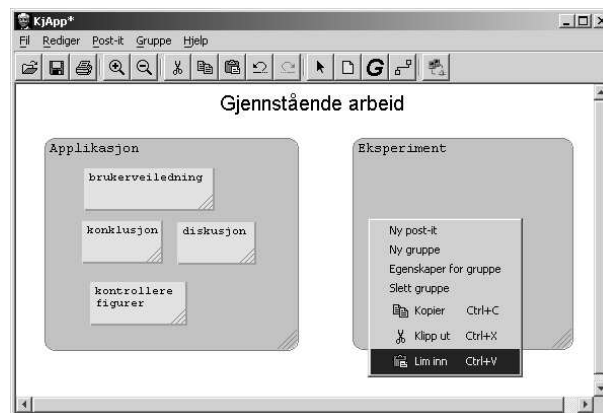
Figur 14.12: Opprett gruppe etter område er valgt

### Kopier merkede elementer

Ett eller flere elementer kan kopieres ved å merke elementene som beskrevet. Deretter velges kopier fra høyreklikkmeny som vist i figur 14.13. Elementene på utklippstavlen limes inn ved å høyreklikke der elementene skal plasseres og velge *Lim inn*. Dersom elementene limes inn i en gruppe inkluderes de i gruppen som vist i figur 14.14.



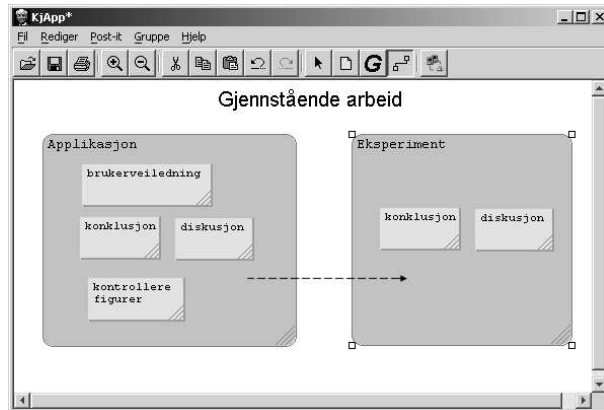
Figur 14.13: Kopier merkede elementer



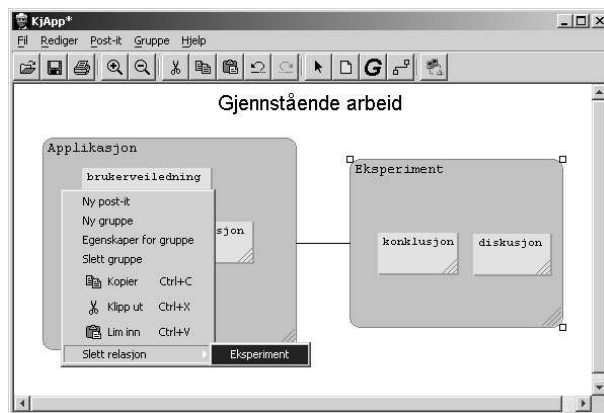
Figur 14.14: Lim inn merkede elementer

### Ny relasjon

KjApp går over i relasjonsmodus når relasjonsknappen trykkes. En ny relasjon opprettes ved å trykke i en gruppe og dra musen over til en annen gruppe som vist med pil i figur 14.15. Når musen slippes går KjApp ut av relasjonsmodus. Dersom brukeren ønsker å opprette en relasjon til, må relasjonsknappen trykkes på nytt. En relasjon slettes ved å høyreklikke i en av gruppene relasjonen knytter sammen. Nederst i høyreklikkmenyen vises valget *Slett relasjon*. Under dette kommer en liste med navn på alle grupper som gruppen har relasjoner til. I figur 14.16 har gruppen kun en relasjon som kan slettes.



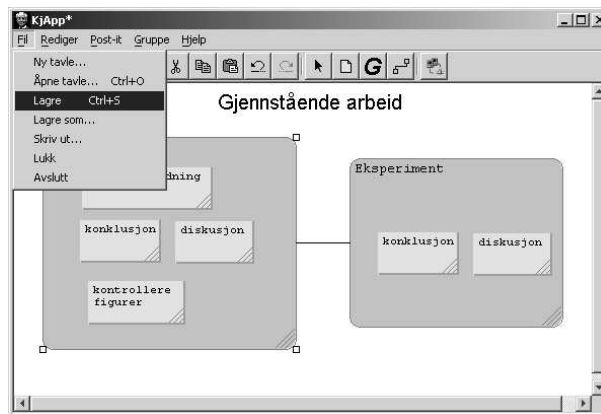
Figur 14.15: Opprett ny relasjon



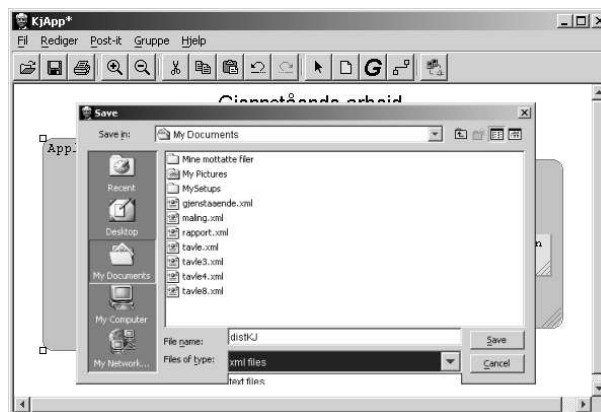
Figur 14.16: Sletting av relasjon

## Lagre tavle

Ved å velge *Lagre tavle* fra filmenyen lagres tavlen. Tavlen kan lagres på XML eller tekstformat, men kun XML filer kan leses av KjApp. Dersom brukeren ikke tidligere har angitt et filnavn, angis dette i dialogboksen som kommer opp (se figur 14.18). Filnavnet vises over menylinjen. Dersom filen oppdateres etter siste lagring, angis dette med en \* etter filnavnet. En tavle kan lagres på tekstformat dersom brukeren ønsker en lesbar eksport av ideer og grupper. Merk imidlertid at KjApp ikke kan åpne tavler som er lagret på tekstformat.



Figur 14.17: Lagring av tavle



Figur 14.18: Angi navn og lagringsformat for tavle

# Kapittel 15

## Videre arbeid

### Innhold

---

15.1 Forbedringer av KjApp . . . . .	113
15.2 Utvidelser av KjApp . . . . .	114

---

Under gjennomføring av brukbarhetstester og eksperimentet (se del I) kom det frem gode forslag til ønsket funksjonalitet. Arbeidsomfanget på forslagene varierte, og vi har valgt å presentere forslagene i to kategorier. Den første kategorien er *Forbedringer av KjApp* og omhandler arbeid av lite til moderat omfang. Dette vil si funksjonalitet som kan implementeres i dagens versjon uten behov for reorganisering av overordnet arkitektur. Den andre kategorien er *Utvidelser av KjApp* og omhandler arbeid med stort omfang. Implementering av funksjonalitene krever mye arbeid og reorganisering er nødvendig.

### 15.1 Forbedringer av KjApp

Enkelte deltakere klaget på tungvint bruk på noen områder. Blant annet bør KjApp ha hurtigtaster for å opprette ny post-it og gruppe. Prototypen støtter hurtigtaster for standard redigeringsfunksjoner som klipp ut (ctrl+x), lim inn (ctrl+v) og kopier (ctrl+c). Deltakerne foreslo at hurtigtaster for ny post-it (ctrl+p) og ny gruppe(ctrl+g) opprettes. Hurtigtaster kan implementeres for all funksjonalitet i KjApp på samme måte som for redigeringsfunksjonene, men ved å kalle andre metoder. I tillegg mangler prototypen støtte for dobbeltklikking på post-it lapper og grupper. Ved dobbeltklikking er det naturlig å få opp egenskaper for valgt element. Dette kan implementeres ved å kalle metoden for egenskaper til valgt element når dobbelklikkehendelse detekteres.

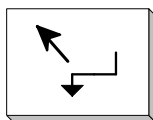
Enkelte deltakere var usikre på hvor ideteksten skulle skrives. Noen skrev først i kommentarfeltet, men skjønnte raskt at dette ikke vises på tavlen. Vi mener derfor at kommentarfeltet bør skjules, og bringes frem med en *Vis/Skjul* knapp.

Relasjoner i KjApp skapte problemer for flere deltakere. Relasjon opprettes ved å klikke på relasjonsknappen på verktøymenyen eller ved å velge relasjon fra gruppemenyen. Deretter klikkes startgruppen for relasjonen, museknappen holdes nede og dras til sluttgruppen for relasjonen. Et problem med funksjonaliteten er at brukeren ikke gis annen

indikasjon på at relasjonsknappen er trykket enn at knappen er inntrykket. Dersom brukeren forsøker å opprette en relasjon mellom to grupper, men slipper museklikket før han kommer på sluttgruppen, vil ingen relasjon opprettes. Brukeren må da trykke på relasjonsknappen på nytt for å opprette relasjonen. Vårt forslag er å endre markør i forbindelse med opprettelse av relasjon. Ved å endre markør får brukeren beskjed om at han jobber i *relasjonsmodus*. Endring av markør kan gjøres på samme måte som for merking av tavle elementer. Ved merking endres musepeker fra hvit pil (se figur 15.1) til et kryss. For relasjon kan markør endres fra hvit pil til svart pil med en relasjonsmarkør (se figur 15.2).



Figur 15.1: Vanlig markør



Figur 15.2: Markør for relasjon

Ved oppstart av KjApp presenteres et tomt grensesnitt til brukeren. Vi mener at KjApp bør startes med en tom tavle slik det gjøres i for eksempel MicroSoft Word med et tomt dokument. Dermed er det bare å begynne å jobbe direkte etter oppstart.

## 15.2 Utvidelser av KjApp

KjApp ble utviklet som en prototyp for å brukes ved eksperimentering med distribuert KJ. Prototypen inneholder mangler som ikke er akseptable for en applikasjon som benyttes aktivt i næringslivet. I tillegg er KjApp utviklet som en frittstående applikasjon og benyttes sammen med NetMeeting for distribuert bruk. En distribuert versjon av KjApp er ønskelig for å oppnå bedre ytelse og større grad av samtidig arbeid. Flere av deltakerne nevnte samtidig arbeid som en viktig egenskap ved en fremtidig versjon av KjApp. Prototypen er heller ikke utviklet med tanke på pålitelighet og sikkerhet, egenskaper som er vesentlige ved en fulldistribuert versjon. I de neste avsnitt vil forslag til utvidelser av KjApp presenteres. Forslagene er ideer på hva som kan gjøres og hvordan.

### 15.2.1 Distribuert versjon

En naturlig forlengelse av arbeidet er å utvikle en distribuert versjon av KjApp. En distribuert versjon kan utvikles med en klient-tjener arkitektur. Klienten finnes hos alle deltakerne i sesjonen og må i tillegg til funksjonaliteten i dagens versjon, håndtere funksjonalitet for pålogging til sentral tjener, oppsett av distribuerte sesjonen og automatisk oppdatering av tavler. Tjeneren håndterer distribuert tavleinformasjon, og sørger

for at deltakere i en sesjon alltid *ser* den samme tavlen. Tjeneren sikrer at data ikke går tapt i sesjonen og at *distribuerte konflikter* løses riktig. En distribuert konflikt kan være at to brukere forsøker å aksessere samme tavleelement samtidig. Kommunikasjonen mellom klient og tjener kan foregå ved hjelp av meldinger. Oppdatering av tavler skjer ved at endret tavleinformasjon på XML format overføres til sentral tjener. Tavleinformatjonen lagres i tjeneren og distribueres videre til de andre deltakerne i sesjonen.

### **Klient**

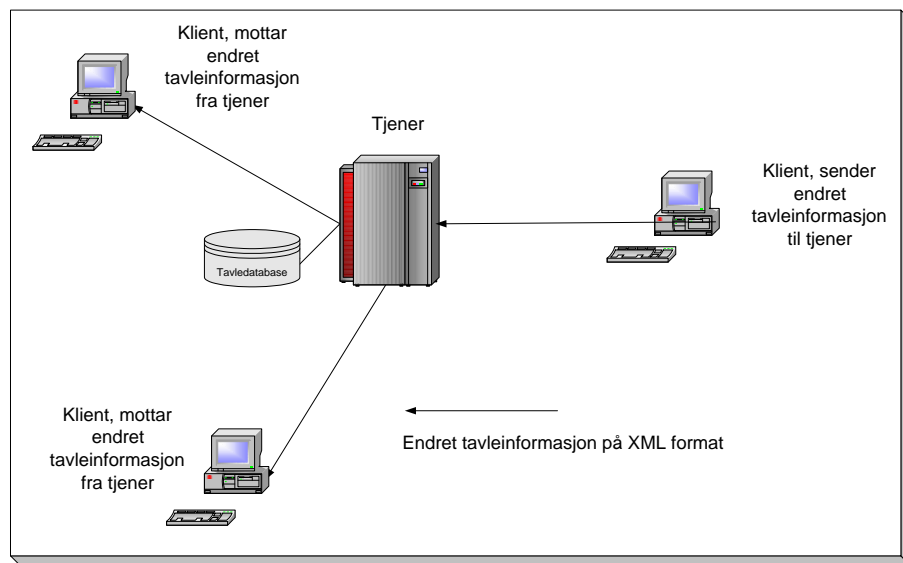
Som nevnt, må klienten tilby funksjonaliteten som dagens versjon av KjApp har. Det må være mulig å opprette nye tavler, generere ideer i form av post-it lapper og gruppere disse. I tillegg må klienten kunne opprette distribuerte sesjoner. Brukeren må gis tilgang til å definere sesjoner, herunder angi start- og sluttidspunkt, invitere deltakere og definere tema for sesjonen. For en distribuert versjon er det viktig å kontrollere at deltakerne i sesjonen virkelig er de de utgir seg for å være. Klienten må derfor sammen med tjeneren tilby sikker pålogging og autentisering av brukerne. Når en deltaker har opprettet en sesjon og invitert deltakere, må klienten sørge for oppkobling mot sentral tjener. Sesjonen starter med at deltakerne forbereder egne ideer i lokalt datalager hos klienten. Lokalt datalager mangler i dagens versjon av KjApp, noe som var et savn hos flere deltakere. Etter at deltakerne er ferdige med å forberede egne ideer presenteres de for de andre i sesjonen. Dette gjøres ved at ideen dras fra lokalt datalager og over i *felles datalager*. Innholdet i felles datalager distribueres til deltakerne via sentral tjener.

### **Tjener**

Tjeneren må samarbeide med klienten for å håndtere pålogging og oppkobling av sesjoner. Pålogging håndteres ved at deltakerne gis unike brukernavn som autentiseres med passord. Etter oppkobling av en sesjon skal tjeneren sørge for at deltakerne til enhver tid *ser* den samme tavlen. Figur 15.3 viser hvordan en deltaker sender endret tavleinformasjon til tjeneren, som videresender dette til de andre deltakerne. Alle elementer på tavlen skal ha en unik identitet. Dagens prototyp genererer en unik lokal identitet til elementene ved opprettelse. En distribuert versjon kan implementere unik identitet ved sammenkobling av lokal identitet og deltakeridentitet. Deltakeridentitet kan for eksempel være et unikt brukernavn.

### **Meldinger mellom klient og tjener**

For å håndtere kommunikasjonen mellom klient og tjener, foreslår vi at det defineres meldinger for lovlige operasjoner i applikasjonen. Meldingene sendes ved hjelp av en protokoll som sikrer pålitelig overføring, for eksempel Transport Control Protocol og Internet Protocol (TCP/IP). Meldinger defineres for pålogging og oppkobling av sesjoner. Kravspesifikasjonen til prototypen er et godt hjelpemiddel for å finne ut hvilke meldinger som må defineres. Eksempler på meldinger er opprettelse og sletting av post-it lapper og grupper. Innholdet i meldingene er avhengig av typen melding og er på XML format. For eksempel krever opprettelse av ny post-it en fullstending definisjon av post-it med idetekst, størrelse, posisjon og status. Sletting av post-it gjøres med melding som kun inneholder identiteten til lappen.



Figur 15.3: Distribuert versjon av KjApp

## Kapittel 16

# Diskusjon KjApp

KjApp er utviklet som en frittstående applikasjon til bruk under eksperimentet. Målsetningen var å utvikle en prototyp med tilstrekkelig funksjonalitet til å støtte en KJ sesjon under distribuerte omgivelser. Deltakerne i eksperimentet har gitt gode tilbakemeldinger på prototypen og den har ikke vært et hinder under idegenereringen.

KjApp har et rent og logisk oppbygd brukergrensesnitt og tilbyr enkel funksjonalitet. Deltakerne i eksperimentet ble raskt i stand til å benytte funksjonalitet for å opprette post-it lapper og grupper, samt å flytte disse. Hurtigtaster for å kopiere og lime inn tavleelementer ble også benyttet av enkelte deltakere. Dermed var deltakerne i stand til å gjennomføre en KJ sesjon under distribuerte omgivelser ved hjelp av KjApp og Microsoft NetMeeting.

Deltakerne i eksperimentet er vant med å benytte PC og Windows basert programvare. Dette har både fordeler og ulemper i forhold til om KjApp har innfridd målsetningen. Erfarne deltakerne finner seg raskt til rette med KjApp og kjenner igjen funksjonalitet fra andre Windows programmer. De behersker bruk av mus og mulighetene dette gir. *Ulempen* er at deltakerne har store forventninger med tanke på tilbudt funksjonalitet. Dette merket vi med enkelte kommentarene vi fikk fra deltakerne. Spesielt gjaldt dette hurtigtaster for å effektivisere bruken av KjApp. Flere deltakere savnet hurtigtaster for å opprette post-it lapper og grupper. Mindre erfarne brukere vil ikke ha samme forventninger til funksjonalitet, men vil også ha større problemer med å benytte Windowsbasert programvare.

Selv om prototypen har nådd målsetningen i forhold til eksperimentet, har vi erfart at den ikke er en fullgod løsning. En distribuert versjon øker muligheten for samtidig arbeid og varigheten på distribuerte sesjoner reduseres. En distribuert løsning kan i tillegg ha et lokalt datalager hvor ideer kan forberedes før presentasjon. Mangelen på et slikt datalager og bruken av NetMeeting for distribusjon er de største ankepunkt med vår løsning. Det er tungvint å overføre kontroll over KjApp mellom deltakerne ved bruk av NetMeeting. Dette problemet oppstår ikke ved en distribuert versjon.

Vi mener det er flere faktorer som gjør det hensiktsmessig å benytte verktøystøtte for KJ metoden. Dette er uavhengig av om sesjonene gjennomføres distribuert eller lokalt. KJ metoden benyttes når et problemområde skal utforskes. En moderator styrer sesjonen og skal i tillegg ta vare på resultatene og produsere en rapport. For en moderator

er det gunstig å kunne lagre tavlen elektronisk fremfor å måtte ta vare på post-it lapper eller ta et bilde av tavlen som er produsert. Moderatoren vil kunne ta tak i resultatene direkte etter gjennomføringen av en sesjon og legge inn kommentarer (i KjApp) på post-it lapper og grupper. Dermed er det lettere å ta vare på inntrykkene fra en KJ sesjon. Et verktøy gjør det i tillegg lettere å strukturere innholdet og vil være en hjelp ved arbeidet med rapporten fra KJ sesjonen.

## Kapittel 17

# Konklusjon KjApp

Utviklingen av prototypen KjApp har etter vår mening innfridd målsetningen. Målet var å utvikle en prototype som med tilstrekkelig funksjonalitet støtter KJ metoden. KjApp ble utviklet som en frittstående applikasjon for å benyttes distribuert ved hjelp av Microsoft NetMeeting. Målet var **ikke** å utvikle en fullstendig applikasjon for bruk i industrien, men en prototype for bruk i eksperimentet og dermed utforske konseptet *Distribuert idegenerering*. Tilbakemeldinger og kommentarer fra deltakerne i eksperimentet viser at KjApp har fungert godt og prototypen har ikke vært et hinder for idegenereringen. Flere deltakere har rost KjApp for å ha et innbydende brukergrensesnitt og for å være et *morsomt* program for å strukturere ideer. Dermed mener vi at KjApp har innfridd kravet om å være brukervennlig og mest mulig lik en lokal gjennomføring av KJ. Dette er momenter ved KjApp som vi er svært fornøyd med.

Likevel er det mangler ved KjApp som gjør at distribuerte KJ sesjoner ikke fungerer optimalt. Fordi KjApp er utviklet som en frittstående applikasjon er det problematisk å oppnå stor grad av samtidighet ved idegenerering. Kun en deltaker kan jobbe med applikasjonen ad gangen og dette begrenser effektiviteten i en distribuert sesjon. I tillegg må ideer skrives inn *på direkten* og dette har skapt en del irritasjon hos deltakerne på grunn av lang vente tid. Vi mener at en distribuert løsning hvor deltakerne kan forberede egne ideer i et lokalt datalager er optimalt. Ved presentasjon kan forberedte ideer dras ut i fellesområdet og presenteres muntlig ved hjelp av telefonkonferanse. Utviklingen av en distribuert løsning er et omfattende arbeid, men vi mener at vårt arbeid i rapportens del II vil være til god hjelp. Vi har i tillegg skissert en løsning for hvordan en distribuert versjon kan utvikles i kapittel 15.



# Tillegg A

## Spørreskjema

### A.1 Spørreskjema til eksperimentet med KJ metoden

30.05.2002

#### 1. Sesjonsopplysninger

1.1 Deltaker identitet «ID»

1.2 Kjønn

Mann

Kvinne

1.3 Alder

1.4 Status

Student

Diplomstudent

Doktorgrad

Yrkesaktiv

1.5 Hvor fortrolig er du med de andre deltakerne i sesjonen?

Ikke fortrolig

Under middels

Middels

Over middels

Veldig fortrolig

1.6 Andre kommentarer

## 2. Distribuert sesjon

Deltaker «id»

2.1 Hvordan opplevde du sesjonen (sett ett kryss for hvert spørsmål)?

2.1.1 Gjennomføring

Nr. 1

Nr. 2

2.1.2 Var det morsomt?

Ikke morsomt

Litt morsomt

Morsomt

Veldig morsomt

Hysterisk morsomt

2.1.3 Følte du at du bidro?

Ikke i det hele tatt

Litt

Bra

Mye

Veldig mye

2.1.4 Hvordan fungerte gruppen?

Ikke i det hele tatt

Under middels

Middels

Over middels

Veldig bra

2.1.5 Følte du at sesjonen hadde trøkk?

(Hadde sesjonen god flyt, kom idéene på løpende bånd eller satt idéene langt inne)

Ingen trøkk

Under middels

Middels

Over middels

Veldig bra trøkk

2.1.6 Setter du pris på å være anonym under sesjonen?

Ikke i det hele tatt

Under middels

Middels

Over middels

Veldig

2.1.7 Hvordan fungerte verktøyene KjApp og NetMeeting?

Ikke i det hele tatt

Under middels

Middels

Over middels

Veldig bra

Kommentér:

2.2 *Hvordan var kvaliteten på resultatet fra sesjonen (ett kryss for hvert spørsmål)?*

2.2.1 I hvor stor grad tror du at du kan benytte resultatet?

Kan ikke benytte resultatet

Kan benytte noen få deler

Kan benytte mange deler

Kan benytte det meste

Kan benytte hele resultatet

2.2.2 Kom det frem nye momenter/idéer?

Ingen nye momenter/idéer

Noen få nye momenter/idéer

En del momenter/idéer

Mange nye momenter/idéer

Veldig mange nye momenter/idéer

2.2.3 Kunne du kommet frem til de samme resultatene på egenhånd?

Nei

Neppe

Kanskje

Trolig

Helt sikkert

2.2.4 Hvor mange enkeltstående ideer mener du ble generert under sesjonen?

### 3. Lokal sesjon

Deltaker «id»

3.1 Hvordan opplevde du sesjonen (sett ett kryss for hvert spørsmål)?

3.1.1 Gjennomføring

Nr. 1

Nr. 2

3.1.2 Var det morsomt?

Ikke morsomt

Litt morsomt

Morsomt

Veldig morsomt

Hysterisk morsomt

3.1.3 Følte du at du bidro?

Ikke i det hele tatt

Litt

Bra

Mye

Veldig mye

3.1.4 Hvordan fungerte gruppen?

Ikke i det hele tatt

Under middels

Middels

Over middels

Veldig bra

3.1.5 Følte du at sesjonen hadde trøkk?

(Hadde sesjonen god flyt, kom idéene på løpende bånd eller satt idéene langt inne)

Ikke noe trøkk

Under middels

Middels

Over middels

- Veldig bra trøkk

3.2 *Hvordan var kvaliteten på resultatet fra sesjonen (ett kryss for hvert spørsmål)?*

3.2.1 I hvor stor grad tror du at du kan benytte resultatet?

- Kan ikke benytte resultatet
- Kan benytte noen få deler
- Kan benytte mange deler
- Kan benytte det meste
- Kan benytte hele resultatet

3.2.2 Kom det frem nye momenter/idéer?

- Ingen nye momenter/idéer
- Noen få nye momenter/idéer
- En del momenter/idéer
- Mange nye momenter/idéer
- Veldig mange nye momenter/idéer

3.2.3 Kunne du kommet frem til de samme resultatene på egenhånd?

- Nei
- Neppe
- Kanskje
- Trolig
- Helt sikkert

## A.2 Oppfølgingsspørsmål

Etter å ha kjørt dette eksperimentet med flere grupper har det dukket opp nye spørsmål vi gjerne vil ha svar på. Håper du tar deg tid til å sette ett kryss for hvert spørsmål og sende inn svarene ved hjelp av *send inn* knappen nederst på siden. På forhånd takk!! Dersom du ønsker å se tavlene som ble generert i din sesjon på nytt er det bare å trykke på følgende linker: *lokal sesjon* og *distribuert sesjon*. Dersom tavlene blir utydelige kan du forsøke å maksimere vinduet du viser de i.

### 1. Ide generering

*1.1 Vurderte du ideene dine på nytt når du skrev de inn i KjApp under distribuert sesjon?*

- vet ikke / husker ikke
- vurderte ingen ideer på nytt
- vurderte noen få av ideene
- vurderte flere av ideene
- vurderte alle ideene

*1.2 Hendte det at du under distribuert sesjon forkastet ideer som var forberedt når ideen skulle presenteres?*

- vet ikke / husker ikke
- nei, forkastet ingen
- ja, forkastet noen få (1-3)
- ja, forkastet mange (4 eller flere)

*1.3 Var du mer opptatt av at det du skrev på lappene under distribuert sesjon skulle være selvforklarende enn under lokal sesjon?*

- vet ikke / husker ikke
- nei, ingen forskjell
- ja, men forskjellen var liten
- ja, forskjellen var stor

*1.4 Tror du at du benyttet flest forkortelser under distribuert eller lokal sesjon?*

- vet ikke / husker ikke
- distribuert hadde flest
- like mange for distribuert og lokal
- lokal hadde flest

*1.5 Synes du den distribuerte eller den lokale sesjon totalt sett genererte de beste ideene?*

- vet ikke / husker ikke
- distribuert var best
- de var like gode
- lokal var best

*1.6 Synes du distribuert eller lokal sesjon ga deg best oversikt over genererte ideer?*

- helt klart distribuert
- distribuert ga litt bedre oversikt enn lokal
- like god oversikt
- lokal ga litt bedre oversikt enn distribuert
- helt klart lokal

*1.7 Distribuert sesjon tok betydelig lengre tid enn lokal sesjon for alle grupper. Hvilken innvirkning tror du dette har på brainstormingen?*

- utelukkende negativt
- mest negativt
- like mye positivt som negativt
- mest positivt
- utelukkende positivt

*1.8 Distribuert KJ sesjon var en ny erfaring for alle deltakerne. Hvordan tror du dette påvirket resultatet?*

- utelukkende negativt
- mest negativt
- like mye positivt som negativt
- mest positivt
- utelukkende positivt

*1.9 Hvordan tror du resultatet ville blitt dersom du hadde mer erfaring med distribuert KJ og bruk av KjApp?*

- vesentlig dårligere
- noe dårligere
- like godt
- noe bedre
- vesentlig bedre

## 2. Flyt og trøkk

2.1 Mener du at telefonkonferansen var et hinder for brainstormingen?

- vet ikke / husker ikke
- nei, ikke i det hele tatt
- ja, til en viss grad
- ja, i stor grad

2.2 Hvor viktig mener du stemmekontakt er i en slik sesjon?

- uvesentlig
- litt viktig
- middels viktig
- svært viktig
- helt nødvendig

2.3 Mener du at KjApp var et hinder for brainstormingen?

- vet ikke / husker ikke
- nei, ikke i det hele tatt
- ja, til en viss grad
- ja, i stor grad

2.4 Mener du at NetMeeting var et hinder for brainstormingen? (overføring av kontroll, oppdatering av skjermbilde)

- vet ikke / husker ikke
- nei, ikke i det hele tatt
- ja, til en viss grad
- ja, i stor grad

2.5 Under eksperimentet ble deltakerne spurt om hvor fortrolige de var med hverandre. Hvor viktig mener du fortrolighet er for at sesjonen skal gi et godt resultat?

- uvesentlig
- litt viktig
- middels viktig
- svært viktig
- helt nødvendig

2.6 I forbindelse med distribuert sesjon ble det kommentert at det kom flere sleivkommentarer enn under lokal sesjon. Mener du at du er tøffere i kjeften på telefon enn ansikt til ansikt med andre deltakere?

- nei, ikke i det hele tatt
- litt tøffere ansikt til ansikt
- like tøff
- litt tøffere på telefon
- ja, helt klart tøffere i telefon

### 3. Fremtidig funksjonalitet og bruk

3.1 I forbindelse med distribuert sesjon ble det ytret et ønske om at KjApp i større grad burde styre tiden i de forskjellige fasene av brainstormingen. F.eks kunne applikasjonen være 5 min i modus for ideforberedelse, deretter 15min i modus for idepresentasjon og tilslutt 10min i modus for gruppering av ideer. På denne måten sikrer man at sesjonen gjennomføres innenfor gitt tidsramme. Hvordan tror du dette vil påvirke resultatet?

- utelukkende negativt
- mest negativt
- like mye positivt som negativt
- mest positivt
- utelukkende positivt

3.2 Under lokal sesjon kunne deltakerne se hvem som har satt opp forskjellige lapper på bakgrunn av deltakernes håndskrift eller ved at deltakerne benyttet lapper med forskjellige farger. Denne muligheten finnes ikke i den distribuerte sesjonen. Hvor viktig mener du det er å vite hvem som har satt opp de forskjellige lappene?

- uvesentlig
- litt viktig
- middels viktig
- svært viktig
- helt nødvendig

3.3 Et ønske fra deltakerne i eksperimentet var et lokalt datalager i KjApp hvor egne ideer legges inn før de skal presenteres. Innholdet i datalagert skal kun være synlig for eieren av ideene. Ved presentasjon drar eieren en ide ut i fellesområdet og presenterer denne. Hvor viktig mener du et lokalt datalager er?

- uvesentlig
- litt viktig
- middels viktig

- svært viktig
- helt nødvendig

*3.4 Du arbeider i et firma i Trondheim og har kunder i London og Paris. Dere har et teknisk problem som skal diskuteres. Kunne du tenke deg å gjøre dette distribuert vha telefonkonferanse, NetMeeting og KjApp?*

- nei, jeg ville reist til london/paris uansett
- ja, dersom jeg kjente kunden godt, ville jeg vurdert dette
- ja, jeg ville uansett forsøkt dette av effektivitetshensyn og for å redusere reise-kostnader

## Tillegg B

# Svar på spørsmål - rådata

Tillegget inneholder data som er samlet inn i løpet av eksperimentet. Dataene danner grunnlaget for presentasjonen i tillegg C og for analysen i tillegg D.1 og tillegg D.2. Først presenteres svarene på spørreskjema som ble benyttet under eksperimentet. Deretter presenteres svarene på fra oppfølgingsspørsmålene. Svarene er indelt på samme måte som spørreskjemaene i tillegg A.

### B.1 Svar på spørreundersøkelse

#### B.1.1 Sesjonsopplysninger

ID	1.2	1.3	1.4	1.5	1.6
1	1	25	1	5	nei
2	1	26	1	5	Ja
3	1	26	1	5	nei
4	1	29	3	5	ja
5	1	32	3	4	nei
6	1	28	3	4	ja
7	0	20	0	3	nei
8	0	19	0	4	ja
9	1	21	0	3	nei
10	1	34	1	5	nei
11	1	25	1	4	nei
12	1	27	1	4	nei
13	0	22	0	3	ja
14	1	22	0	3	ja
15	1	24	0	3	ja

**B.1.2 Distribuert sesjon**

ID	2.1.Opplevelse								2.2 Kvalitet			
	1	2	3	4	5	6	7	8	1	2	3	4
1	2	3	5	5	4	2	4	Ja	4	4	3	19
2	2	4	4	5	4	3	4	Ja	5	4	2	17
3	2	4	3	3	3	1	5	Ja	4	3	3	16
4	1	3	5	5	4	1	5	ja	4	3	3	21
5	1	3	4	5	4	1	3	ja	3	3	2	6
6	1	3	3	4	1	2	4	ja	4	3	3	20
7	2	3	3	3	3	2	4	ja	4	3	4	30
8	2	1	3	2	2	1	3	ja	2	2	4	31
9	2	2	3	3	3	3	3	ja	4	4	2	29
10	1	3	3	4	3	3	3	ja	2	1	3	4
11	1	2	2	4	4	3	3	ja	4	3	2	3
12	1	3	3	3	4	2	4	ja	2	2	2	18
13	2	3	3	4	3	3	4	ja	5	4	1	35
14	2	3	3	4	4	2	5	ja	2	2	4	40
15	2	3	3	3	4	2	3	ja	4	3	4	31

**B.1.3 Lokal sesjon**

ID	3.1.Opplevelse					3.2 Kvalitet			
	1	2	3	4	5	1	2	3	4
1	1	2	4	4	5	4	4	3	23
2	1	3	4	5	5	5	4	2	20
3	1	3	4	5	4	2	3	4	18
4	2	3	5	5	3	4	3	2	23
5	2	2	4	4	3	4	3	2	8
6	2	4	4	4	4	3	3	3	23
7	1	3	3	3	5	2	2	4	56
8	1	4	4	3	5	3	3	4	56
9	1	2	2	3	3	3	4	2	53
10	2	3	4	4	4	2	3	3	4
11	2	2	2	4	4	3	3	2	2
12	2	3	4	4	4	2	3	1	5
13	1	3	3	3	3	4	5	3	25
14	1	3	3	4	4	2	2	5	30
15	1	3	3	3	4	4	2	3	35

## B.2 Svar på oppfølgingsspørsmål

### B.2.1 Idegenerering

ID	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1	2	0	3	0	2	2	3	3	3
2	0	1	2	0	3	3	3	4	4
3	0	2	3	1	2	4	3	4	4
4	1	1	1	3	2	3	2	4	4
5	2	2	2	3	2	4	3	3	3
6	1	1	1	2	3	3	2	3	3
7	3	2	2	2	1	2	4	2	4
8	4	2	1	2	3	3	1	2	4
9	4	3	1	0	1	2	4	3	4
10	3	2	1	3	2	3	3	3	3
11	3	1	2	2	2	2	2	2	4
12	2	2	2	2	3	4	2	2	5
13	1	1	1	2	1	1	4	2	4
14	2	2	1	2	1	1	3	3	4
15	3	2	1	2	2	2	3	3	3

**B.2.2 Flyt og trøkk**

ID	2.1	2.2	2.3	2.4	2.5	2.6
1	2	4	1	0	4	1
2	1	5	1	2	5	5
3	2	5	1	2	4	3
4	1	5	1	2	4	1
5	1	5	2	2	4	4
6	2	4	1	1	4	3
7	2	4	1	2	4	4
8	3	5	2	3	3	1
9	1	4	1	2	2	4
10	1	5	1	2	3	3
11	2	4	1	3	4	3
12	1	4	1	2	2	3
13	1	4	1	2	4	1
14	1	3	1	2	4	2
15	1	4	1	2	3	3

**B.2.3 Fremtidig bruk og funksjonalitet**

<b>ID</b>	<b>3.1</b>	<b>3.2</b>	<b>3.3</b>	<b>3.4</b>
1	2	1	3	2
2	5	5	5	3
3	2	3	4	2
4	4	1	3	2
5	1	3	4	2
6	5	1	4	2
7	3	1	5	2
8	4	1	5	2
9	3	1	4	2
10	2	1	3	3
11	4	2	3	2
12	2	1	4	3
13	4	1	3	2
14	2	2	4	2
15	3	2	4	2



# Tillegg C

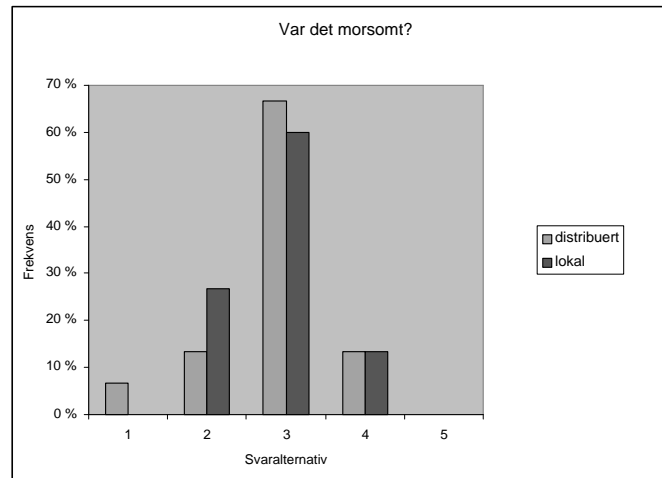
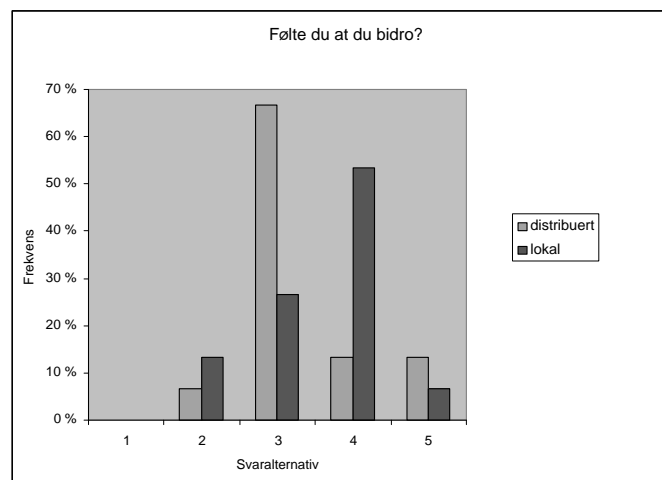
## Grafisk fremstilling av resultater

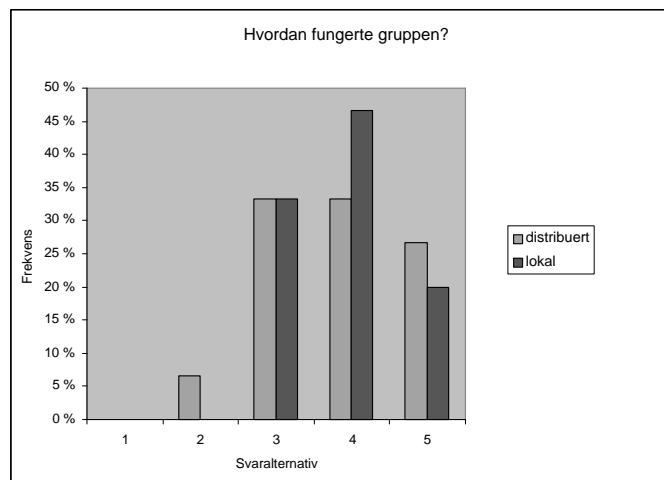
### C.1 Spørreskjema fra eksperimentet

Tillegget viser grafisk fordeling av svarene på spørsmål stilt under eksperimentet. Forklaring til figurene finnes rapportens avsnitt 4.3.2.

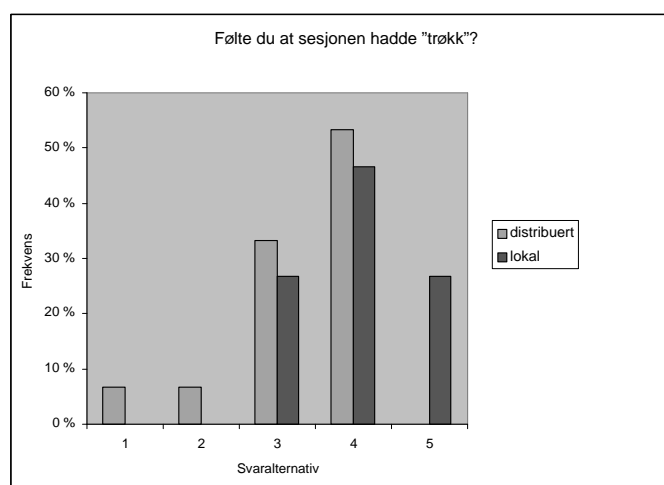


Figur C.1: Fortrolighet med deltakerne i gruppen

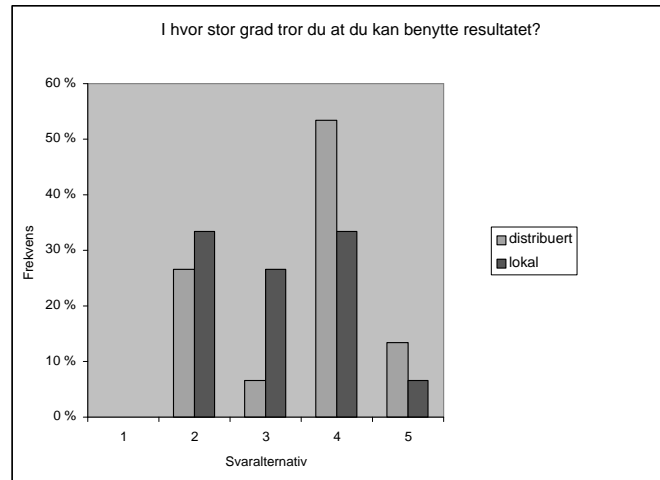
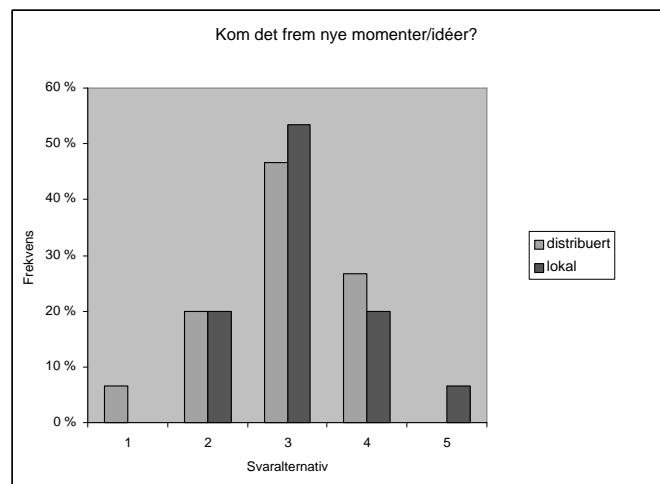
Figur C.2: Fordeling av svar på spørsmålet *Var det morsomt*Figur C.3: Fordeling av svar på spørsmålet *Følte du at du bidro*

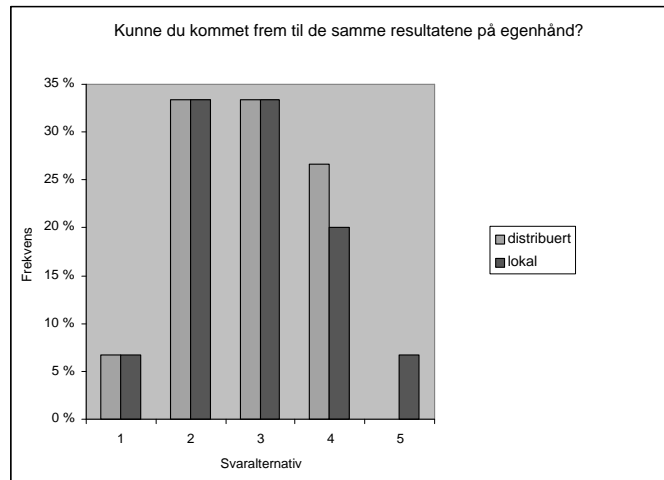


Figur C.4: Fordeling av svar på spørsmålet *Hvordan fungerte gruppen*



Figur C.5: Fordeling av svar på spørsmålet *Hadde sesjonen trøkk*

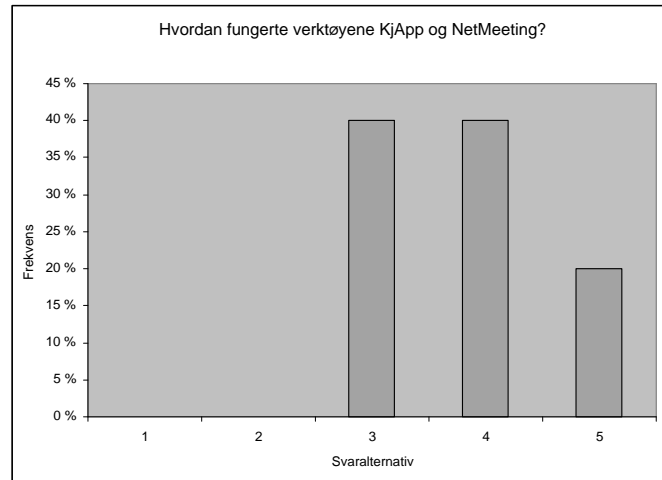
Figur C.6: Fordeling av svar på spørsmålet *Kan du benytte resultatet*Figur C.7: Fordeling av svar på spørsmålet *Kom det frem nye momenter*



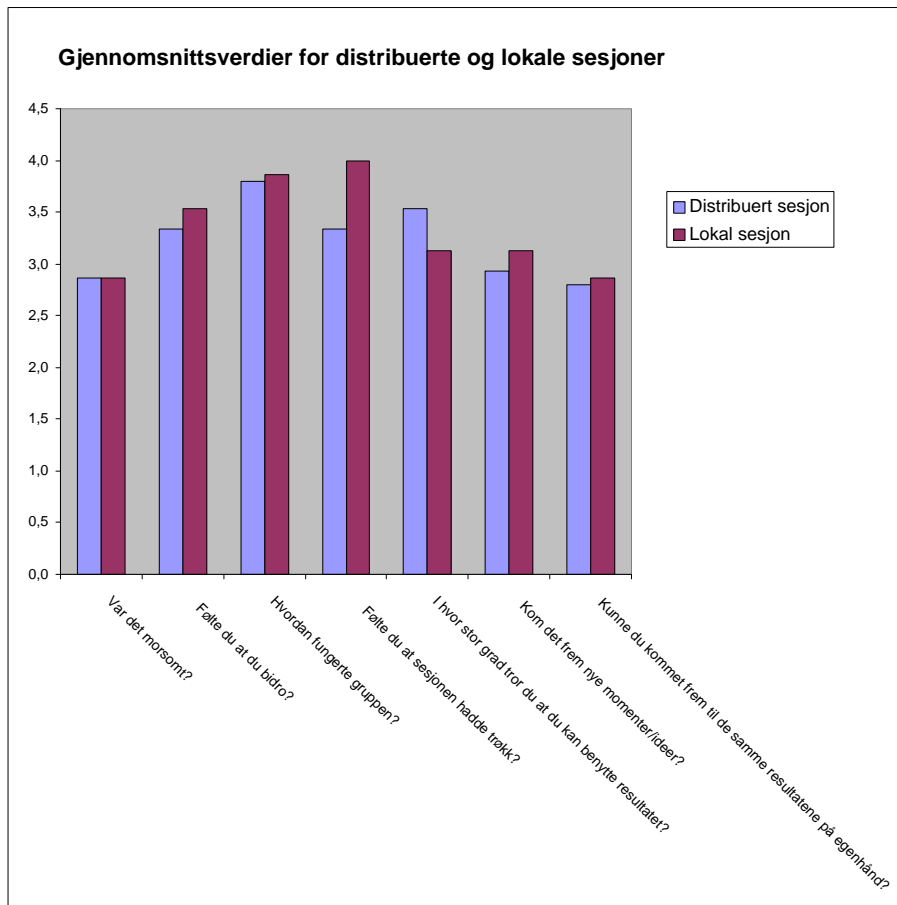
Figur C.8: Fordeling av svar på spørsmålet *Kunne du kommet frem til resultatet på egenhånd*



Figur C.9: Fordeling av svar på spørsmålet *Foretrekker du å være anonym*



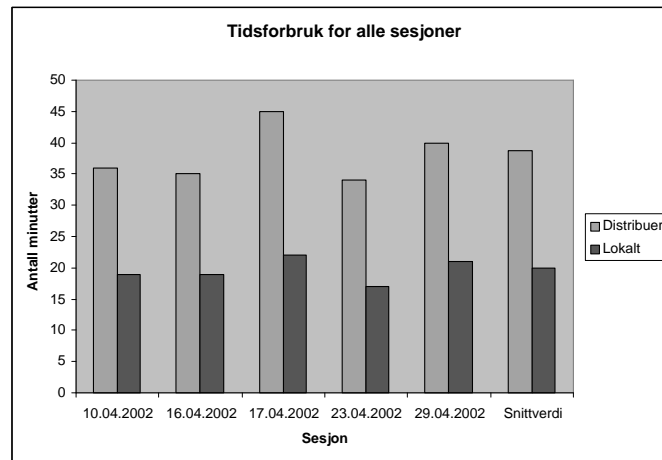
Figur C.10: Fordeling av svar på spørsmålet *Hvordan fungerte KjApp og NetMeeting*



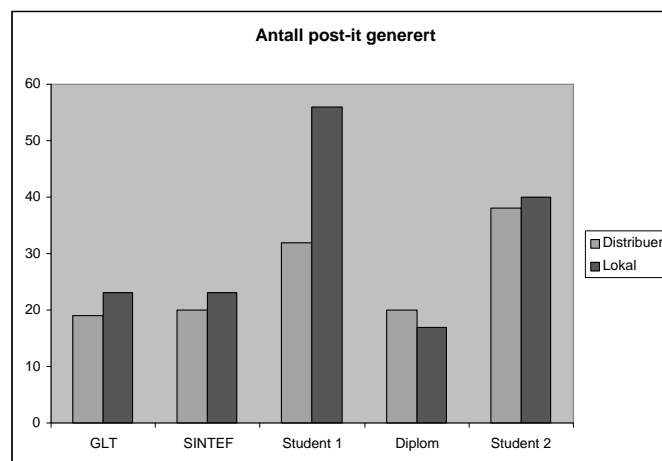
Figur C.11: Gjennomsnittsverdier for spørsmålene fra distribuerte og lokale sesjoner

## C.2 Tidsbruk, antall post-it og grupper generert

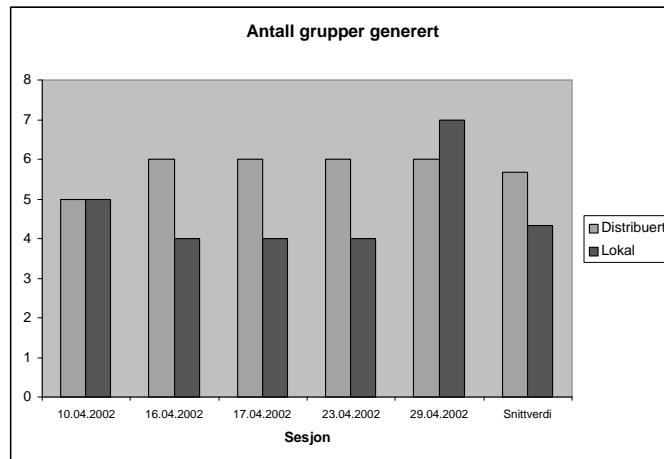
Tillegget viser grafisk fremstilling av tidsforbruket for alle sesjonene. Videre vises antall post-it og grupper generert. Forklaring til figurene er gitt rapportens avsnitt 4.3.3.



Figur C.12: Tidsbruk for alle sesjoner



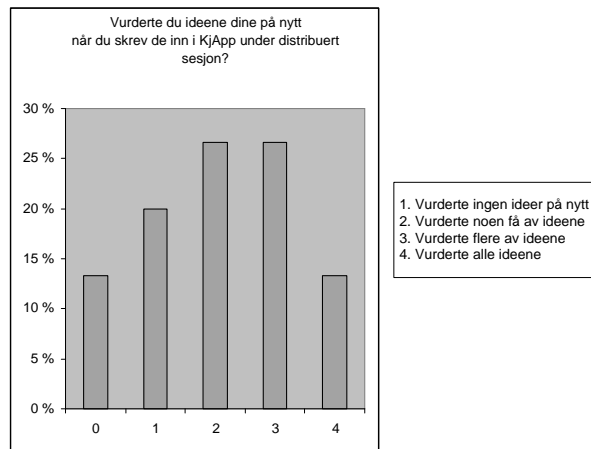
Figur C.13: Post-it lapper generert for alle sesjoner



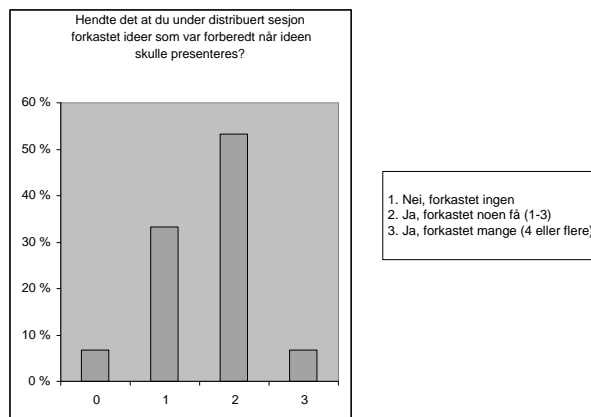
Figur C.14: Antall grupper generert for alle sesjoner

### C.3 Fordeling av svar på oppfølgingsspørsmål

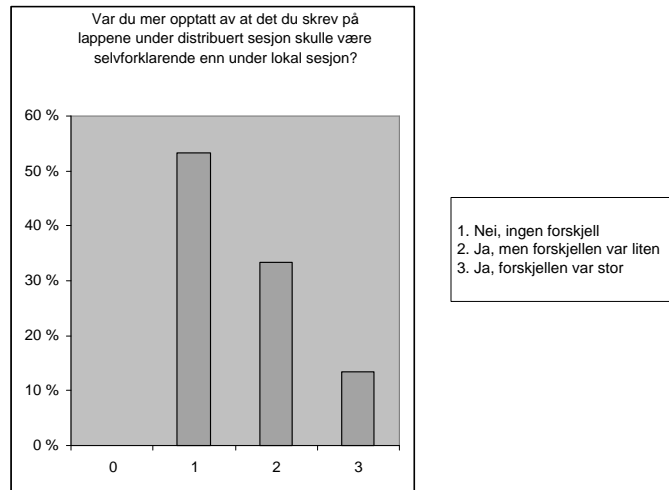
Tillegget viser grafisk fremstilling av svarene på oppfølgingsspørsmålene. Svaralternativene på spørsmålene varierer. Skalaene blir derfor presentert sammen med hvert spørsmål. Svaralternativ null indikerer *vet ikke/husker ikke*. Forklaring til figurene er gitt i rapportens avsnitt 4.3.4.



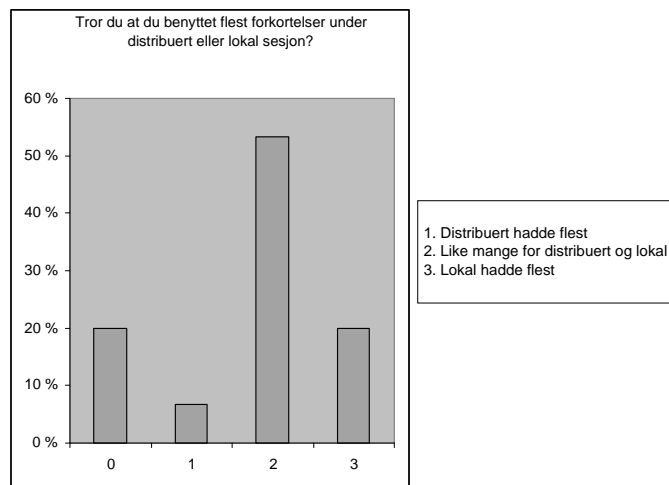
Figur C.15: Fordeling av svar på spørsmålet *Vurderte du ideene på nytt under distribuert sesjon*



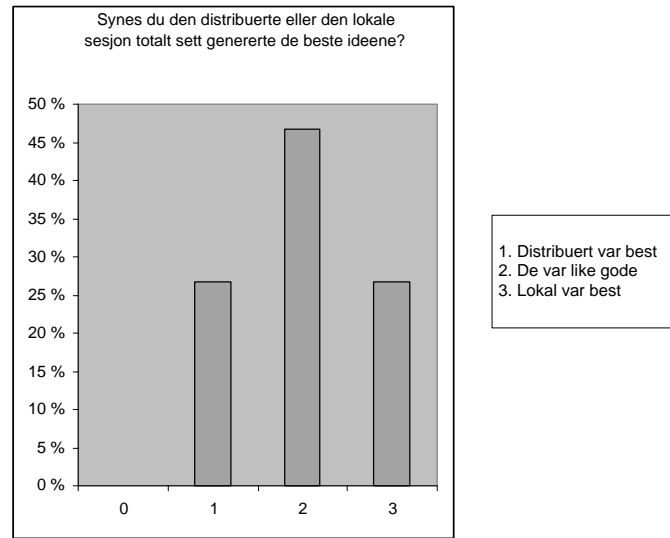
Figur C.16: Fordeling av svar på spørsmålet *Forkastet du noen ideer under distribuert sesjon*



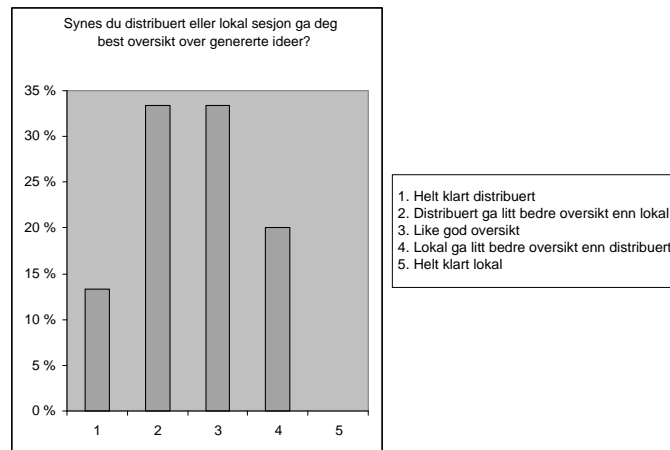
Figur C.17: Fordeling av svar på spørsmålet *Var du mer opptatt av at lappene under under distribuert sesjon skulle være selvforklarende*



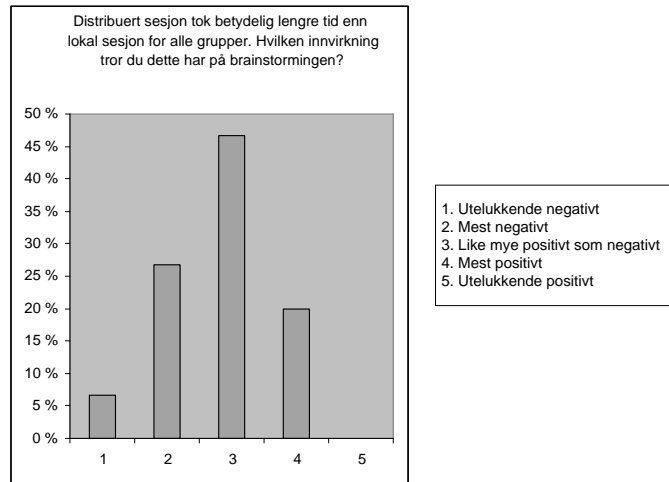
Figur C.18: Fordeling av svar på spørsmålet *I hvilken sesjon benyttet du flest forkortelser*



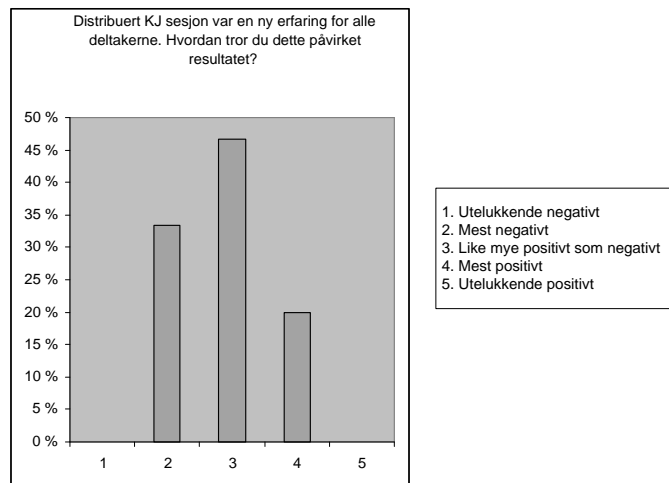
Figur C.19: Fordeling av svar på spørsmålet *Hvilken sesjon genererte de beste ideer*



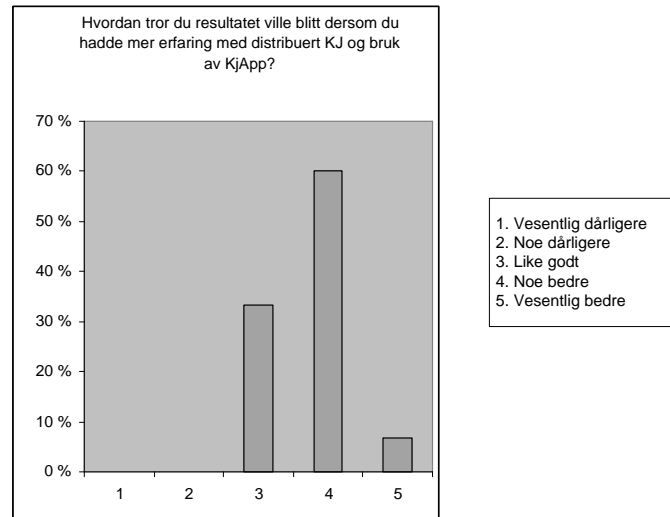
Figur C.20: Fordeling av svar på spørsmålet *Hvilken sesjon ga best oversikt*



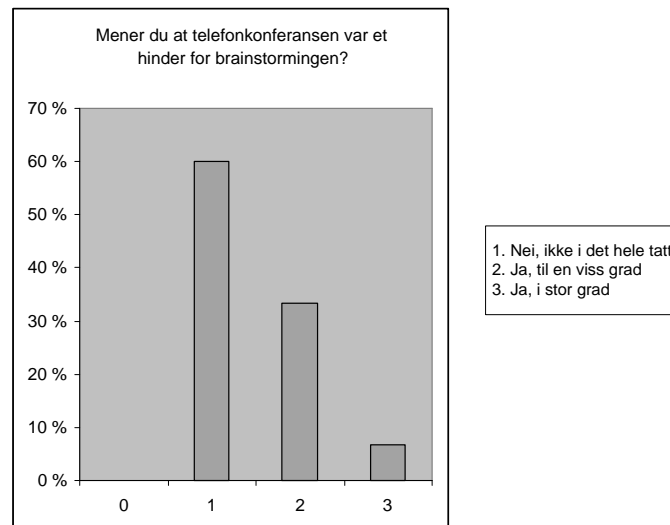
Figur C.21: Fordeling av svar på spørsmålet *Hvordan ble resultatet påvirket av varigheten på distribuert sesjon*



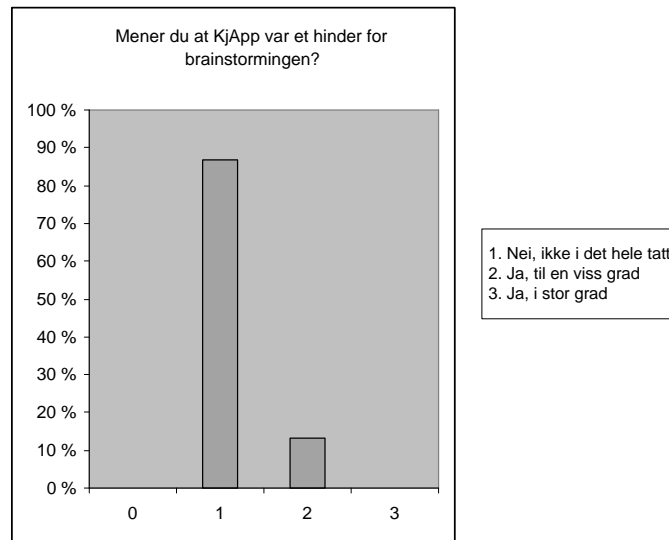
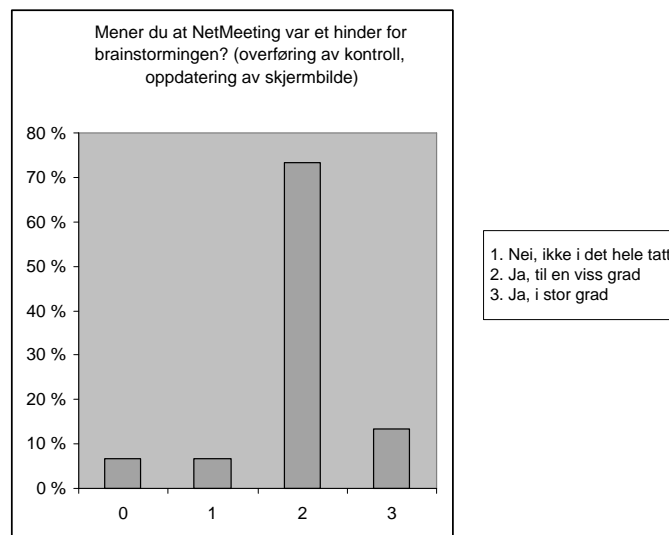
Figur C.22: Fordeling av svar på spørsmålet *Hvordan ble resultatet påvirket av at distribuert KJ var en ny erfaring for alle*

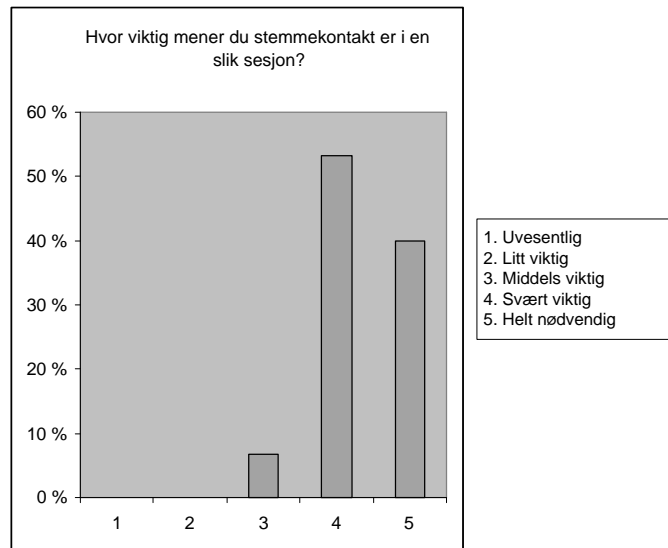


Figur C.23: Fordeling av svar på spørsmålet om *Mer erfaring med distribuert KJ og KjApp*

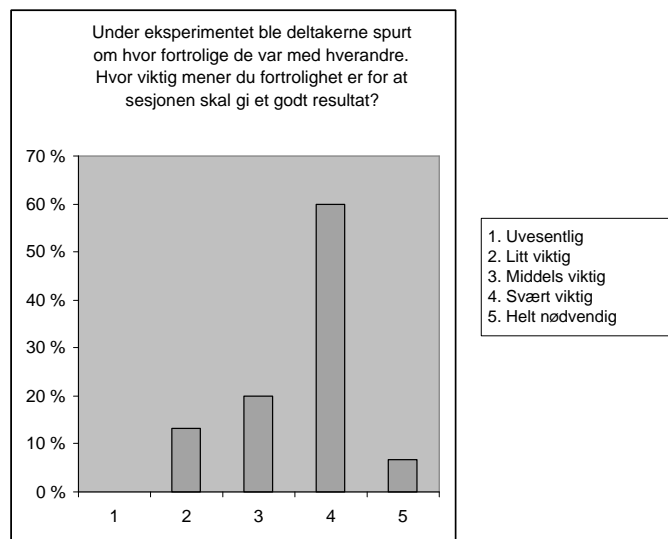


Figur C.24: Fordeling av svar på spørsmålet *Var telefonkonferansen et hinder*

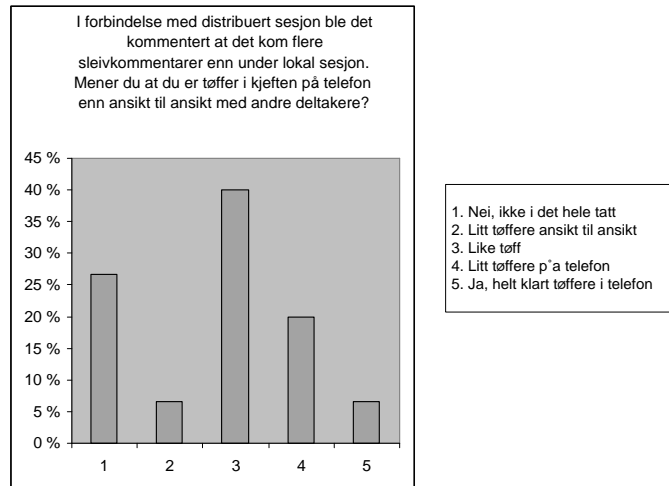
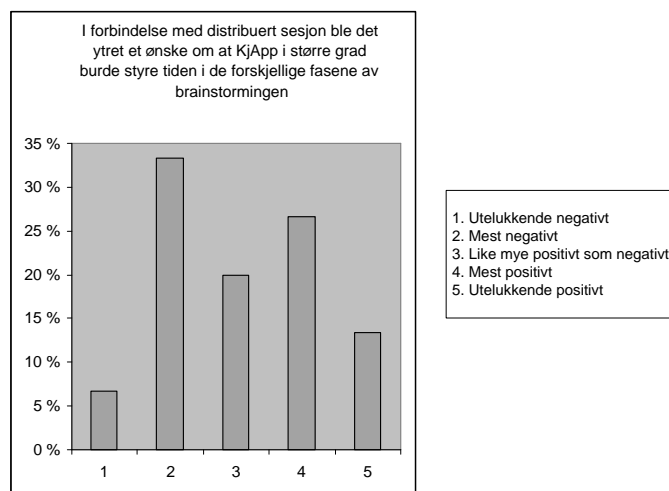
Figur C.25: Fordeling av svar på spørsmålet *Var KjApp et hinder*Figur C.26: Fordeling av svar på spørsmålet *Var NetMeeting et hinder*

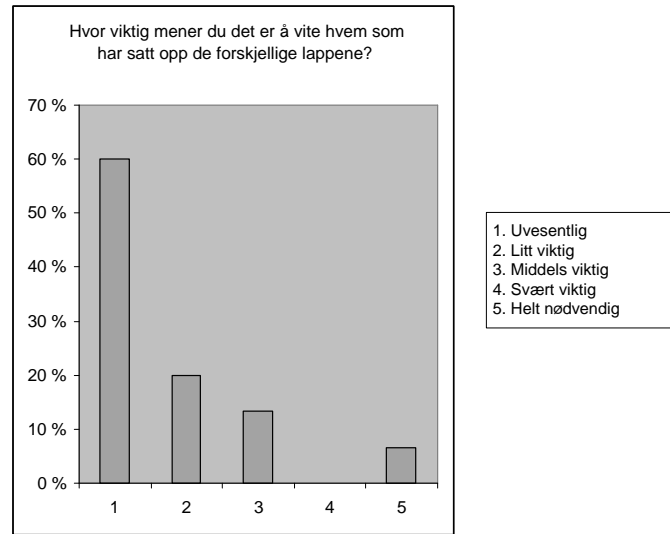


Figur C.27: Fordeling av svar på spørsmålet *Hvor viktig mener du at stemmekontakt er i en slik sesjon*

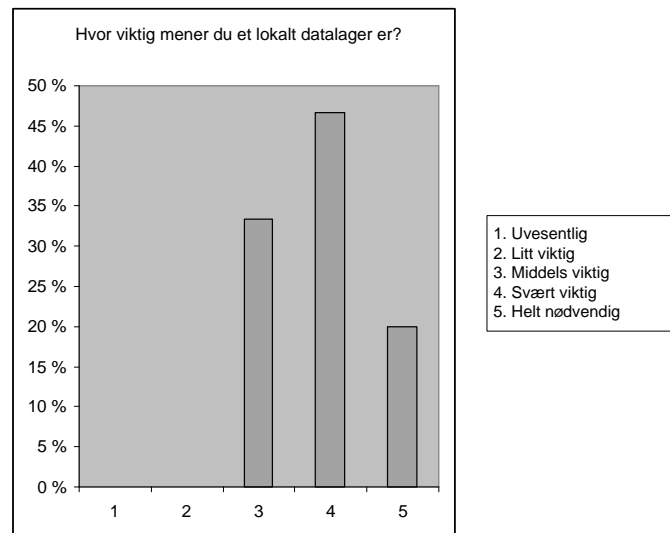


Figur C.28: Fordeling av svar på spørsmålet om *Fortrolighet*

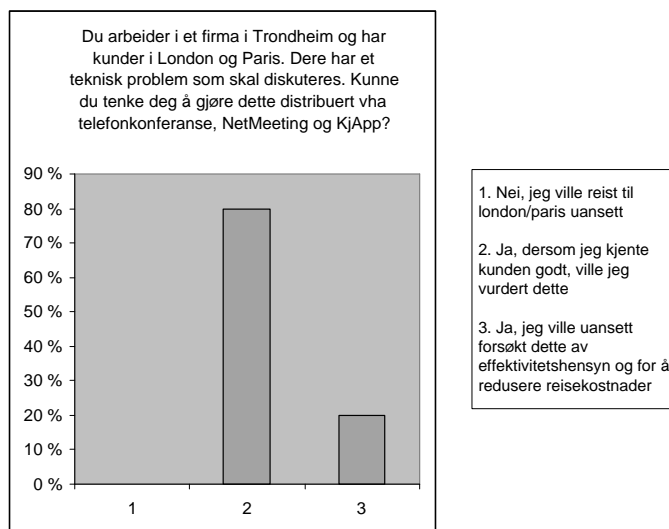
Figur C.29: Fordeling av svar på spørsmålet om *Sleivkommentarer*Figur C.30: Fordeling av svar på spørsmålet om *Tidsstyrt sesjon*



Figur C.31: Fordeling av svar på spørsmålet *Vite hvem som har satt opp lapper*



Figur C.32: Fordeling av svar på spørsmålet om *Lokalt datalager i KjApp*



Figur C.33: Fordeling av svar på spørsmålet om *Teknisk diskusjon Trondheim-Paris-London*



# Tillegg D

## Analyse av resultater

### D.1 ANOVA

Avsnittet presenterer resultater fra statistisk analyse med ANOVA. Vi har analysert spørsmålene som belyser *faktorer ved hypotesetesting* presentert i avsnitt 3.2. Metoden anvendes for to tilfeller. Først sammenlignes data fra lokal sesjon med data fra distribuert sesjon og alle deltakere behandles under ett. Deretter gjennomføres en analyse hvor vi splitter på deltakernes profesjon. Det vil si at vi lar SINTEF og Giant Leap være i en gruppe og de andre i en studentgruppe. I dette tilfellet behandles lokale og distribuerte data under ett på bakgrunn av at resultatene fra første analyse viser at det ikke er noen signifikant forskjell mellom dem. Analysene er gjennomført ved hjelp av MINITAB versjon 13.20.

#### D.1.1 One-way ANOVA: Distribuert mot lokal sesjon, alle data

Sammenligner distribuert og lokal gjennomføring. Alle studiesubjekter behandles under ett. Resultatene viser hvordan spredningen på svarene var fordelt. Distribuert sesjon representeres med 1 og lokal sesjon med 2.

##### Var det morsomt

Analysis of Variance for Morsomt

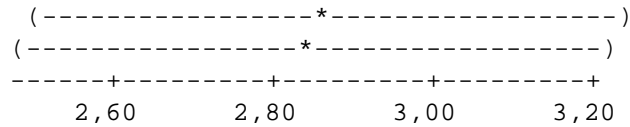
Source	DF	SS	MS	F	P
Distribu	1	0,000	0,000	0,00	1,000
Error	28	13,467	0,481		
Total	29	13,467			

Level	N	Mean	StDev
1	15	2,8667	0,7432
2	15	2,8667	0,6399

Pooled StDev = 0,6935

Individual 95% CIs For Mean  
Based on Pooled StDev

-----+-----+-----+-----+

**Følte du at du bidro**

Analysis of Variance for Bidro du

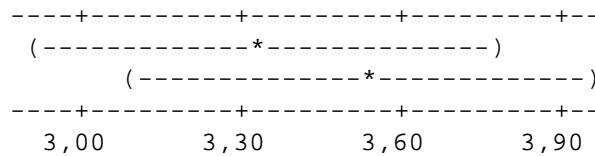
Source	DF	SS	MS	F	P
Distribu	1	0,300	0,300	0,44	0,512
Error	28	19,067	0,681		
Total	29	19,367			

Level	N	Mean	StDev
1	15	3,3333	0,8165
2	15	3,5333	0,8338

Pooled StDev = 0,8252

Individual 95% CIs For Mean

Based on Pooled StDev

**Hvordan fungerte gruppen**

Analysis of Variance for Fungerte gruppen

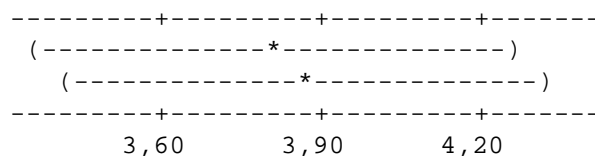
Source	DF	SS	MS	F	P
Distribu	1	0,033	0,033	0,05	0,831
Error	28	20,133	0,719		
Total	29	20,167			

Level	N	Mean	StDev
1	15	3,8000	0,9411
2	15	3,8667	0,7432

Pooled StDev = 0,8480

Individual 95% CIs For Mean

Based on Pooled StDev



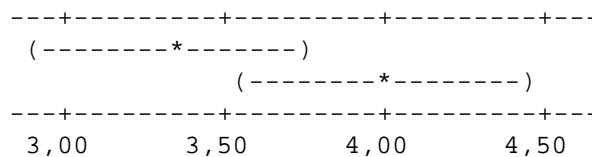
**Følte du at sesjonen hadde trøkk**

Analysis of Variance for Trøkk?

Source	DF	SS	MS	F	P
Distribu	1	3,333	3,333	4,83	0,036
Error	28	19,333	0,690		
Total	29	22,667			

Level	N	Mean	StDev
1	15	3,3333	0,8997
2	15	4,0000	0,7559

Pooled StDev = 0,8309

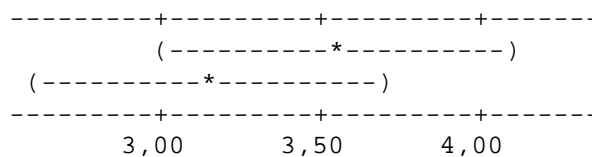
Individual 95% CIs For Mean  
Based on Pooled StDev**I hvor stor grad tror du at du kan benytte resultatet**

Analysis of Variance for Benytte resultatet

Source	DF	SS	MS	F	P
Distribu	1	1,20	1,20	1,14	0,295
Error	28	29,47	1,05		
Total	29	30,67			

Level	N	Mean	StDev
1	15	3,533	1,060
2	15	3,133	0,990

Pooled StDev = 1,026

Individual 95% CIs For Mean  
Based on Pooled StDev**Kom det frem nye momenter/ideer**

Analysis of Variance for Nye momenter

Source	DF	SS	MS	F	P
Distribu	1	0,300	0,300	0,41	0,529
Error	28	20,667	0,738		

Total 29 20,967

Level	N	Mean	StDev
1	15	2,9333	0,8837
2	15	3,1333	0,8338

Pooled StDev = 0,8591

Individual 95% CIs For Mean  
Based on Pooled StDev

```

-----+-----+-----+-----+
(-----*-----)
      (-----*-----)
-----+-----+-----+-----+
          2,70      3,00      3,30

```

### Kunne du kommet frem til de samme resultatene på egenhånd

Analysis of Variance for På egenhånd

Source	DF	SS	MS	F	P
Distribu	1	0,03	0,03	0,03	0,857
Error	28	28,13	1,00		
Total	29	28,17			

Level	N	Mean	StDev
1	15	2,800	0,941
2	15	2,867	1,060

Pooled StDev = 1,002

Individual 95% CIs For Mean  
Based on Pooled StDev

```

-----+-----+-----+-----+
(-----*-----)
      (-----*-----)
-----+-----+-----+-----+
          2,45      2,80      3,15      3,50

```

### D.1.2 One-way ANOVA: GLT og SINTEF mot de andre gruppene

I denne analysen slår vi sammen data for distribuert og lokal sesjon med bakgrunn i resultatene fra forige avsnitt. Vi splitter på deltakernes profesjon. SINTEF og GLT representeres med 1 og studentgruppene med 2. For spørsmålet om *trøkk* kan ikke distribuerte og lokale data behandles sammen, da den første analysen viser at det er en signifikant forskjell mellom disse. For dette spørsmålet splitter vi dataene både på profesjon (SINTEF/GLT mot studentgruppene) og distribuert/lokalt.

#### Var det morsomt

Analysis of Variance for Morsomt

Source	DF	SS	MS	F	P
GLT og S	1	0,939	0,939	2,10	0,159
Error	28	12,528	0,447		
Total	29	13,467			

Level	N	Mean	StDev
1	12	3,0833	0,6686
2	18	2,7222	0,6691

Pooled StDev = 0,6689

Individual 95% CIs For Mean  
Based on Pooled StDev

```

-----+-----+-----+-----
              (-----*-----)
    (-----*-----)
-----+-----+-----+-----
2,40      2,70      3,00      3,30

```

### Følte du at du bidro

Analysis of Variance for Bidro du

Source	DF	SS	MS	F	P
GLT og S	1	8,450	8,450	21,67	0,000
Error	28	10,917	0,390		
Total	29	19,367			

Level	N	Mean	StDev
1	12	4,0833	0,6686
2	18	3,0000	0,5941

Pooled StDev = 0,6244

Individual 95% CIs For Mean  
Based on Pooled StDev

```

-----+-----+-----+-----
              (-----*-----)
    (-----*-----)
-----+-----+-----+-----
      3,00      3,50      4,00

```

### Hvordan fungerte gruppen

Analysis of Variance for Fungerte gruppen

Source	DF	SS	MS	F	P
GLT og S	1	8,889	8,889	22,07	0,000
Error	28	11,278	0,403		
Total	29	20,167			

Level	N	Mean	StDev
-------	---	------	-------

1	12	4,5000	0,6742
2	18	3,3889	0,6077

Pooled StDev = 0,6346

Individual 95% CIs For Mean

Based on Pooled StDev

```

-----+-----+-----+-----
              (-----*-----)
(-----*-----)
-----+-----+-----+-----
              3,50      4,00      4,50

```

### I hvor stor grad tror du at du kan benytte resultatet

Analysis of Variance for Benytte resultatet

Source	DF	SS	MS	F	P
GLT og S	1	5,000	5,000	5,45	0,027
Error	28	25,667	0,917		
Total	29	30,667			

Level	N	Mean	StDev
1	12	3,8333	0,8348
2	18	3,0000	1,0290

Pooled StDev = 0,9574

Individual 95% CIs For Mean

Based on Pooled StDev

```

-----+-----+-----+-----
              (-----*-----)
(-----*-----)
-----+-----+-----+-----
              3,00      3,60      4,20

```

### Kom det frem nye momenter/ideer

Analysis of Variance for Nye momenter

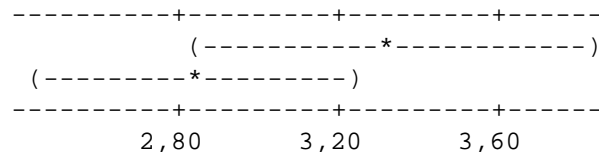
Source	DF	SS	MS	F	P
GLT og S	1	1,800	1,800	2,63	0,116
Error	28	19,167	0,685		
Total	29	20,967			

Level	N	Mean	StDev
1	12	3,3333	0,4924
2	18	2,8333	0,9852

Pooled StDev = 0,8274

Individual 95% CIs For Mean

Based on Pooled StDev



### Kunne du kommet frem til de samme resultatene på egenhånd

Analysis of Variance for På egenhånd

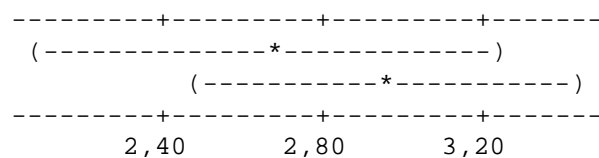
Source	DF	SS	MS	F	P
GLT og S	1	0,556	0,556	0,56	0,459
Error	28	27,611	0,986		
Total	29	28,167			

Level	N	Mean	StDev
1	12	2,6667	0,6513
2	18	2,9444	1,1618

Pooled StDev = 0,9930

Individual 95% CIs For Mean

Based on Pooled StDev



### Følte du at sesjonen hadde trøkk

Det var en signifikant forskjell på resultatene for distribuert og lokale sesjoner for dette spørsmålet. Datamaterialet splittes derfor både på profesjon og på distribuert/lokal. GLT og SINTEF representeres med 1 og 2 for henholdsvis distribuert og lokal sesjon, mens de andre gruppene representeres med 3 og 4.

Analysis of Variance for Trøkk?

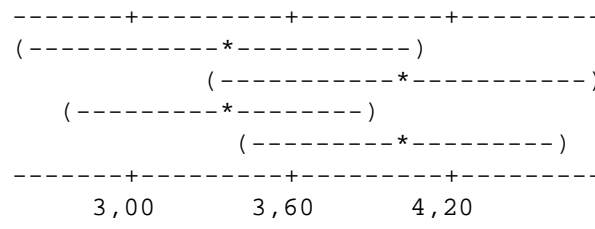
Source	DF	SS	MS	F	P
GLT(D=1/	3	3,333	1,111	1,49	0,239
Error	26	19,333	0,744		
Total	29	22,667			

Level	N	Mean	StDev
1	6	3,3333	1,2111
2	6	4,0000	0,8944
3	9	3,3333	0,7071
4	9	4,0000	0,7071

Pooled StDev = 0,8623

Individual 95% CIs For Mean

Based on Pooled StDev



## D.2 Kruskal-Wallis

Avsnittet beskriver resultater fra statistisk analyse med Kruskal-Wallis test. Kruskal-Wallis er en ikke-parametrisk test, og resultatene vil derfor være med å belyse vårt valg av analyse metode beskrevet i avsnitt 3.5 dersom de sees i sammenheng med resultatene presentert i tillegg D.1.1. Vi har her sammenlignet data for alle studiesubjekter for spørsmål som belyser *faktorer for hypotesetesting* beskrevet i avsnitt 3.2. P-verdien i testen viser hvor sannsynlig det er at medianene i to populasjoner er like. Distribuert sesjon representeres ved 1 og lokal sesjon representeres ved 2.

### Var det morsomt

Distribu	N	Median	Ave Rank	Z
1	15	3,000	15,8	0,19
2	15	3,000	15,2	-0,19
Overall	30		15,5	

H = 0,03 DF = 1 P = 0,852  
 H = 0,05 DF = 1 P = 0,828 (adjusted for ties)

### Følte du at du bidro

Distribu	N	Median	Ave Rank	Z
1	15	3,000	14,0	-0,93
2	15	4,000	17,0	0,93
Overall	30		15,5	

H = 0,87 DF = 1 P = 0,351  
 H = 1,01 DF = 1 P = 0,314 (adjusted for ties)

### Hvordan fungerte gruppen

Distribu	N	Median	Ave Rank	Z
1	15	4,000	15,3	-0,15
2	15	4,000	15,7	0,15
Overall	30		15,5	

H = 0,02 DF = 1 P = 0,885  
 H = 0,02 DF = 1 P = 0,878 (adjusted for ties)

### Følte du at sesjonen hadde trøkk

Distribu	N	Median	Ave Rank	Z
1	15	4,000	12,7	-1,76
2	15	4,000	18,3	1,76
Overall	30		15,5	

H = 3,11 DF = 1 P = 0,078  
 H = 3,67 DF = 1 P = 0,055 (adjusted for ties)

**I hvor stor grad tror du at du kan benytte resultatet**

Distribu	N	Median	Ave Rank	Z
1	15	4,000	17,2	1,06
2	15	3,000	13,8	-1,06
Overall	30		15,5	

H = 1,12 DF = 1 P = 0,290

H = 1,26 DF = 1 P = 0,261 (adjusted for ties)

**Kom det frem nye momenter**

Distribu	N	Median	Ave Rank	Z
1	15	3,000	14,9	-0,37
2	15	3,000	16,1	0,37
Overall	30		15,5	

H = 0,14 DF = 1 P = 0,709

H = 0,16 DF = 1 P = 0,686 (adjusted for ties)

**Kunne du kommet frem til de samme resultatene på egenhånd**

Distribu	N	Median	Ave Rank	Z
1	15	3,000	15,4	-0,08
2	15	3,000	15,6	0,08
Overall	30		15,5	

H = 0,01 DF = 1 P = 0,934

H = 0,01 DF = 1 P = 0,931 (adjusted for ties)

# Tillegg E

## Test

### E.1 Systemtest

Testen består av en test for hvert krav beskrevet i avsnitt 10.1. For hvert krav angir vi om det er innfridd eller ikke. Dersom det er kommentarer til gjennomføringen av testen kommer disse til slutt for hvert krav. I tabellene som oppsummerer hvert nivå angir X at kravet er innfridd, X\* at kravet er innfridd, men at det er en kommentar til resultatet, og - angir at kravet ikke er innfridd.

#### Applikasjonsnivå

Resultatet av testen for applikasjonsnivå er presentert i tabell E.1.

Tabell E.1: Oversikt over resultater fra systemtest

Krav	Godkjent
FA1	-
FA2	X

**FA1-Skriv ut** Kontrollerer at skriv ut knappen i meny- og verktøylinjen gir tilgang på standard skriver, og skriver ut gjeldende tavle på denne

**Godkjent:** Nei

**Kommentarer:** Fungerer ikke i nettverksmiljø. Funksjonen er ikke prioritert som viktig og er derfor gjort utilgjengelig i koden.

**FA2-Avslutte** Kontrollerer at brukeren får spørsmål om tavlen skal lagres dersom den ikke er lagret, og at KjApp avsluttes.

**Godkjent:** Ja

**Kommentarer:** Får spørsmål om lagring selv om tavlen er lagret

### Tavlenivå

Resultatet av testen for tavlenivå er presentert i tabell E.2. Enkelte tester ble godkjent med kommentarer. Disse er merket med X\*.

Tabell E.2: Oversikt over resultater fra systemtest

Krav	Godkjent
FT1	X
FT2	X
FT3	X
FT4	X*
FT5	X*
FT6	X
FT7	X
FT8	X
FT9	X
FT10	X*
FT11	X
FT12	X
FT13	X*
FT14	X*

**FT1-Ny tavle** Kontrollerer at brukeren får opp en dialogboks med spørsmål om tema for tavlen som skal opprettes, og at tavlen opprettes med angitt navn

**Godkjent:** Ja

**FT2-Åpne tavle** Kontrollerer at det går an å åpne en tidligere lagret tavle, og at denne er identisk med tavlen som ble lagret

**Godkjent:** Ja

**FT3-Endre tema** Kontrollerer at det går an å endre tema for tavlen

**Godkjent:** Ja

**FT4-Lagre tavle** Kontrollerer at det er mulig å lagre tavlen både på XML og tekstformat dersom tavlen ikke har vært lagret tidligere. Dersom tavlen er lagret fra før, skal tavlen lagres uten at brukeren angir format for lagringen. Kontrollerer videre at de lagrede filene ser riktige ut, ved visuell inspeksjon for tekstfiler, og ved å åpne filen i KjApp og se at ingen elementer er forsvunnet for XML format

**Godkjent:** Ja

**Kommentarer:** Tekstfiler skifter ikke til ny linje dersom teksten er lang. Kan ikke lagre spesialtegn som XML (<,>)

**FT5-Lagre tavle som** Kontrollerer at det er mulig å lagre tavlen både på XML og tekstformat. Kontrollerer videre at de lagrede filene ser riktige ut, ved visuell inspeksjon for tekstfiler, og ved å åpne filen i KjApp og se at ingen elementer er forsvunnet for XML format

**Godkjent:** Ja

**Kommentarer:** Samme som for Lagre tavle

**FT6-Lukk tavle** Kontrollerer at tavlen lukkes og at menyer som ikke lenger skal være tilgjengelige fjernes. Dersom tavlen ikke er lagret skal brukeren spørres om tavlen skal lagres

**Godkjent:** Ja

**FT7-Merk alt** Kontrollerer at alle tavleelementer i tavlen merkes når kanppen merk alt eller hurtigknappen ctrl + A trykkes

**Godkjent:** Ja

**FT8-Merk elementer** Kontrollerer at det er mulig å merke et utvalg av elementer i tavlen

**Godkjent:** Ja

**Kommentarer:** Alle elementer som er helt innenfor det merkede området markeres

**FT9-Angre siste** Kontrollerer at det er mulig å angre siste gjennomførte interaksjon i tavlen

**Godkjent:** Ja

**Kommentarer:** Angre funksjonen fungerer ikke for flytting av elementer i tavle

**FT10-Gjenoppsett siste** Kontrollerer at det er mulig å gjenopprette tavlen som var før brukeren angret siste kommando

**Godkjent:** Ja

**Kommentarer:** Angre må trykkes to ganger før gjenoppsett gjøres tilgjengelig for brukeren

**FT11-Kopier** Kontrollerer at det er mulig å kopiere tavleelementer. Det skal være mulig å kopiere et tavle element, et utvalg tavle elementer og alle tavleelementer. Etter at tavleelementene er kopiert skal de kopierte elementene fremdeles ligge i tavlen, mens kopiene er lagret i KjApps utklippstavle. Lim inn knappen skal gjøres tilgjengelig etter at et eller flere tavleelementer er kopiert.

**Godkjent:** Ja

**FT12-Klipp ut** Kontrollerer at det er mulig å klippe ut tavleelementer. Det skal være mulig å klippe ut et tavle element, et utvalg tavle elementer og alle tavleelementer. Etter at tavleelementene er klippet ut skal de fjernes fra tavlen, og en kopi lagres i KjApps utklippstavle. Lim inn knappen skal gjøres tilgjengelig etter at et eller flere tavleelementer er klippet ut.

**Godkjent:** Ja

**FT13-Lim inn** Kontrollerer at elementene som er klippet ut eller kopiert blir limt inn i tavlen. Dersom lim inn benyttes fra høyreklikkmeny skal elementene limes inn der brukeren har trykket. Hvis lim inn fra meny- eller verktøylinje benyttes skal elementene limes inn øverst til venstre i tavlen.

**Godkjent:** Ja

**Kommentarer:** Fungerer fint dersom lim inn i høyreklikkmeny benyttes. Posisjonen blir ikke riktig dersom lim inn i verktøymeny eller rediger- og lim inn benyttes

**FT14-Zoom** Kontrollerer at tavlen kan zoomes inn og ut og at scrollbar gjør elementer som kommer utenfor frame tilgjengelig.

**Godkjent:** Ja

**Kommentarer:** Scrollbar blir ikke oppdatert før størelsen på vinduet oppdateres

### Gruppenivå

Resultatet av testen for gruppenivå er presentert i tabell E.3

Tabell E.3: Oversikt over resultater fra systemtest på gruppenivå

Krav	Godkjent
FG1	X
FG2	X
FG3	X
FG4	X
FG5	X
FG6	X
FG7	X
FG8	X
FG9	X

**FG1-Ny gruppe** Kontrollerer at brukeren får opp en dialogboks som gir mulighet for å angi tema for gruppen som skal opprettes og at gruppen opprettes med angitt navn

**Godkjent:** Ja

**FG2-Endre gruppenavn** Kontrollerer at det er mulig å endre navn på merket gruppe fra menylinje eller ved å høyreklikke på en gruppe

**Godkjent:** Ja

**FG3-Legg til post-it** Kontrollerer at det er mulig å legge en post-it inn i en eksisterende gruppe

**Godkjent:** Ja

**Kommentarer:** Ved å dra post-it lapp inn i gruppen den skal inkluderes i. Gruppen vil utvides dersom det ikke er plass til post-it.

**FG4-Legg til gruppe** Kontrollerer at det er mulig å legge en gruppe inn i en gruppe

**Godkjent:** Ja

**Kommentarer:** Ved å dra gruppen inn i gruppen den skal inkluderes i. Gruppen utvides dersom det ikke er plass til gruppen

**FG5-Fjern post-it** Kontrollerer at det er mulig å fjerne en post-it fra en gruppe

**Godkjent:** Ja

**Kommentarer:** Post-it kan fjernes fra en gruppe, enten ved at den dras ut av gruppen, eller ved at man velger slett post-it fra høyreklikkmenyen til post-it

**FG6-Slett gruppe** Kontrollerer at det går an å slette en gruppe, og at innholdet i den slettede gruppen legges inn i forelder til gruppen som ble slettet

**Godkjent:** Ja

**Kommentarer:** Gruppe slettes ved å velge slett gruppe fra gruppens høyreklikkmeny. Innholdet vil da bli lagt inn i foreldregruppen som slettes. Dersom man benytter *Delete* knappen på tastaturet vil gruppen med hele innholdet bli fjernet fra tavlen.

**FG7-Flytt gruppe** Kontrollerer at det er mulig å flytte en gruppe og at alt innhold i gruppen som flyttes følger gruppen

**Godkjent:** Ja

**FG8-Ny relasjon** Kontrollerer at det er mulig å opprette en relasjon mellom to grupper

**Godkjent:** Ja

**Kommentarer:** Relasjon opprettes ved å velge relasjonsknappen (R) i verktøylinja, for så å klikke og holde i en gruppe og dra musen til gruppen relasjonen skal gå til før museknappen slippes. Dersom brukeren slipper knappen i nullgruppen vil ingen relasjon opprettes

**FG9-Slett relasjon** Kontrollerer at det er mulig å slette en relasjon mellom to grupper

**Godkjent:** Ja

**Kommentarer:** En relasjon slettes ved å trykke slett relasjon <navn> i høyreklikkmenyen til en av gruppene relasjonen binder sammen

**Post-it lapp nivå**

Tabell E.4: Oversikt over resultater fra systemtest

Krav	Godkjent
FP1	X
FP2	X
FP3	X
FP4	X
FP5	X

**FP1-Ny post-it** Kontrollerer at brukeren får opp en dialogboks som gir mulighet for å angi idetekst for post-it som skal opprettes og at post-it opprettes med angitt navn

**Godkjent:** Ja

**FP2-Velg type** Kontrollerer at det er mulig å angi om en post-it skal være positiv, nøytral eller negativ og at nøytral er satt som standard.

**Godkjent:** Ja

**FP3-Flytt post-it** Kontrollerer at det er mulig å dra en post-it fra ett sted til et annet ved hjelp av musen, og at post-it innlemmes i en gruppe dersom post-it slippes i en gruppe

**Godkjent:** Ja

**FP4-Endre post-it** Kontrollerer at det er mulig å endre idetekst for en post-it

**Godkjent:** Ja

**Kommentarer:** Idetekst endres ved å velge egenskaper for post-it fra post-its høyreklikkmeny

**FP5-Slett post-it** Kontrollerer at det er mulig å slette en post-it

**Godkjent:** Ja

## E.2 Brukbarhetstest

I de følgende underkapitler beskrives testtilfellene og eventuelle problemer som ble avdekket av brukeren.

### Opprett ny tavle med navnet *Malingsarbeider*

Brukeren blir presentert for applikasjonen rett etter oppstart.

### Opprett åtte nye post-it lapper

Brukeren skal opprette åtte nye post-it lapper med følgende navn:

- Verandamøbler
- Kontormøbler
- Seng
- Stue
- Gang
- Farger
- Trapp
- Verandagjerde

### Endre post-it lapper

Brukeren skal endre følgende post-it lapper:

- Verandamøbler skal ha status positiv
- Kontormøbler skal ha kommentaren *Må vente med dette til Espen er ferdig med studiene*
- Seng skal ha status positiv
- Trapp skal ha status negativ og kommentaren *Må reise bort etter at trappen er malt*
- Stue skal ha status negativ

### Flytting av post-it lapper

Brukeren skal flytte lappene på følgende måte:

- Verandamøbler og Verandagjerde skal plasseres sammen opp og til høyre på tavlen
- Kontormøbler og Seng plasseres nederst til høyre på tavlen
- Gang og trapp plasseres sammen øverst til venstre på tavlen
- Stue plasseres alene nede til høyre på tavlen
- Farger plasseres i midten

**Gruppering av post-it lapper**

Brukeren skal opprette følgende grupper:

- Verandamøbler og Verandagjerde skal med i gruppen *Veranda*
- Kontormøbler og Seng skal med i gruppen *Interiør*
- Gang og trapp skal med i gruppen *Større arbeider*
- Stue skal være eneste medlem av gruppen *Fremtidig malingsprosjekter*
- Farger skal være eneste medlem av gruppen *Maling*

**Relasjoner mellom grupper**

Brukeren skal opprette en relasjon fra gruppen *Maling* til alle de andre gruppene.

**Redigering**

Brukeren skal forsøke følgende redigering:

- Slett gruppen *Veranda*
- Angre operasjonen som slettet gruppen *Veranda*
- Kopier post-it lappen *Farger*
- Lim inn kopien av *Farger* i gruppen *Maling*
- Endre kopien av *Farger* til *Type*
- Slett relasjonen mellom *Maling* og *Fremtidige malingsprosjekter*
- Zoom inn tavlen
- Zoom ut tavlen
- Lagre tavlen med navnet *maling.xml*

# Tillegg F

## Program listing

### F.1 Package

#### F.1.1 Brukergrensesnitt.java

Listing: Brukergrensesnitt.java

```
1 package kjapp;
2
3 import javax.swing.*;
4 import javax.swing.JFrame;
5 import java.awt.event.*;
6 import java.awt.event.MouseListener;
7 import java.awt.event.MouseMotionListener;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.MouseEvent;
10
11
12 import javax.swing.JMenu;
13 import javax.swing.JMenuBar;
14 import javax.swing.JMenuItem;
15 import javax.swing.JScrollBar;
16 import java.awt.*;
17
18 //Klasser for printing
19 import javax.print.*;
20 import javax.print.attribute.*;
21 import java.awt.print.*;
22
23 import java.awt.*;
24 import java.awt.image.*;
25
26 /**
27  * @author aHvard Alstad
28  * @created 13/2 - 2002
29  *
30  * Klasse som skal a'ndre alle gui elementer i kjApp
31  */
32 public class Brukergrensesnitt extends JFrame // implements MouseListener, MouseMotionListener
33 {
34     private GrafiskeTavleElementer theGrafiskeTavleElementer ;
35     private Menyer theMenyer;
36     private boolean komplettMenyLinje = false ;
37     private Kontroller theKontroller ;
38     private JMenuBar menuBar;
39     private JToolBar toolBar;
40     private int aktivXPos;
41     private int aktivYPos;
42     private Posisjon AktivStartPosisjon ;
43     private Posisjon AktivSluttPosisjon ;
44     private Posisjon merkStartPosisjon ;
45     private Posisjon merkSluttPosisjon ;
46     private boolean ctrlErNede = false ;
47     private PrintRequestAttributeSet attributes ;
48     private JScrollPane scrollPane ;
49
50     public Brukergrensesnitt ( Kontroller k)
51     {
52         // instansierer variable som klassen am kjenne til
53         theKontroller = k;
54         theMenyer = new Menyer(theKontroller);
55         addWindowListener(new WindowAdapter(){
56             public void windowClosing(WindowEvent e){
57                 theKontroller . avslutt () ;
58             }
59         }
60     }
61 }
```

```

59     });
60     tegnHovedMeny();
61 }
62
63
64 /**lager menylinje 'ap toppen
65 * komplettMenyLinje = false -> kun fi 1 drop down meny tegnes.
66 * komplettMenyLinje = true -> all drop down menyer tegnes.
67 */
68 public void tegnHovedMeny()
69 {
70     if (menuBar!=null){
71         remove(menuBar);
72     }
73     menuBar = theMenyer.tegnHovedMeny(komplettMenyLinje);
74     setJMenuBar(menuBar);
75     validate ();
76 }
77
78 public void tegnHoyreKlikkMeny(int posX,int posY)
79 {
80     double z = theKontroller .hentZoomFaktor();
81     theKontroller .fi nnTavleElement (posX,posY);
82     TavleElement t = theKontroller .hentAktivTavleElement ();
83     if (t!=null){
84         JPopupMenu p = theMenyer.tegnHoyreKlikkMeny(t);
85         p.show(theGrafiskeTavleElementer , GrafiskeTavleElementer .doubleToInt(posXz),GrafiskeTavleElementer .doubleToInt(posYz));
86     }
87 }
88
89 /**
90 * legger til en overkrymning til framen
91 * Skal kun tegnes opp dersom komplettMenyLinje = true
92 */
93 public void tegnVerktoymeny()
94 {
95     if (komplettMenyLinje){
96         toolBar = theMenyer.tegnVerktoyMeny();
97         Container contentPane = getContentPane();
98         JPanel contentPane2 = new JPanel();
99         contentPane2.setLayout(new BorderLayout());
100        contentPane2.add(toolBar, BorderLayout.NORTH);
101        contentPane2.add(scrollPane = new JScrollPane(contentPane ,JScrollPane .VERTICAL_SCROLLBAR_ALWAYS,JScrollPane .HORIZONTAL_SCROLLBAR_ALWAYS),BorderLayout.CENTER);
102        setContentPane (contentPane2);
103    }
104 }
105
106 /**
107 * Inparameter inneholder 'aogs gruppe 0 som skal
108 * tegnes opp 'anr tavlen instansieres
109 * Gruppe 0 inneholder Tavle tema
110 */
111 public void nyTavle(Tavle t)
112 {
113     theGrafiskeTavleElementer = new GrafiskeTavleElementer (this , theKontroller );
114     setContentPane ( theGrafiskeTavleElementer );
115     //For printing
116     attributes = new HashPrintRequestAttributeSet ();
117 }
118
119 public void tegnTavle () {
120     // theGrafiskeTavleElementer .setTegnbartOmrade(new Posisjon (0,0) ,new Posisjon(10000,10000));
121     theGrafiskeTavleElementer .repaint ();
122     this .setSize (this .getSize ());
123     this .validate ();
124 }
125
126 public void tegnTavleElement(TavleElement t){
127     // int bredde = t.hentElementBredde();
128     // int hoyde = t.hentElementHoyde();
129     int ekstra = 20;
130     // theGrafiskeTavleElementer .setTegnbartOmrade( new Posisjon(t.hentElementPosisjonX() - ekstra,t.hentElementPosisjonY() - ekstra),
131     new Posisjon(t.hentElementBredde()+bredde+ekstra,t.hentElementHoyde() + hoyde + ekstra));
132
133     // theGrafiskeTavleElementer .repaint (10,t.hentElementPosisjonX() - ekstra,t.hentElementPosisjonY() - ekstra,
134     t.hentElementBredde()+bredde+ekstra,t.hentElementHoyde() + hoyde + ekstra);
135
136     theGrafiskeTavleElementer .repaint ();
137     this .setSize (this .getSize ());
138     this .validate ();
139 }
140
141 public void settKomplettMenyLinje(boolean k)
142 {
143     komplettMenyLinje = k;
144 }
145
146 public void settAktivStartPosisjon (Posisjon p)
147 {
148     AktivStartPosisjon =p;
149 }
150
151 public Posisjon hentAktivStartPosisjon ()
152 {
153     if ( AktivStartPosisjon ==null) AktivStartPosisjon = new Posisjon(0,0);
154     return AktivStartPosisjon ;
155 }
156

```

```

157     public void settAktivSluttPosisjon ( Posisjon pos)
158     {
159         AktivSluttPosisjon = pos;
160     }
161
162     public Posisjon hentAktivSluttPosisjon ()
163     {
164         if ( AktivSluttPosisjon ==null) AktivSluttPosisjon = new Posisjon (0,0);
165         return AktivSluttPosisjon ;
166     }
167
168     /**
169     * Metoder som benyttes for å ta vare på start og sluttpunkter for merk 'aomrde ...
170     * @param pos
171     */
172     public void settMerkSluttPosisjon ( Posisjon pos)
173     {
174         merkSluttPosisjon = pos;
175     }
176
177     public Posisjon hentMerkSluttPosisjon ()
178     {
179         return merkSluttPosisjon ;
180     }
181
182     public void settMerkStartPosisjon ( Posisjon pos)
183     {
184         merkStartPosisjon = pos;
185     }
186
187     public Posisjon hentMerkStartPosisjon ()
188     {
189         if ( merkStartPosisjon ==null) merkStartPosisjon = new Posisjon (0,0);
190         return merkStartPosisjon ;
191     }
192
193     public void setSize (int bredde, int hoyde){
194         super. setSize (bredde, hoyde);
195     }
196
197     public GrafiskeTavleElementer hentGrafiskeTavleElement ()
198     {
199         return theGrafiskeTavleElementer ;
200     }
201
202     public void settPilButton (boolean b)
203     {
204         theMenyer. settPilButton (b);
205     }
206
207     public void settRelasjonButton (boolean b)
208     {
209         theMenyer. settRelasjonButton (b);
210     }
211
212     public void settAngre (boolean b)
213     {
214         theMenyer. settAngre (b);
215     }
216
217     public void settGjenopprett (boolean b)
218     {
219         theMenyer. settGjenopprett (b);
220     }
221
222     public boolean hentGjenopprett ()
223     {
224         return theMenyer. hentGjenopprett ();
225     }
226
227     public void skrivUt ( Posisjon ytterPunkt ) throws PrinterException
228     {
229         PrinterJob job = PrinterJob . getPrinterJob ();
230         job. setPrintable ( theGrafiskeTavleElementer );
231         double wMax = ytterPunkt. xPos;
232         double hMax = ytterPunkt. yPos;
233         double zW = (450/wMax);
234         double zH = (450/hMax);
235         double z = zW;
236         double oldZ = theKontroller .hentZoomFaktor();
237         if ( zH < zW) z = zH;
238         theKontroller .setZoomFaktor(z);
239
240         if ( job. printDialog ( attributes )){
241             job. print ( attributes );
242         }
243         theKontroller .setZoomFaktor(oldZ);
244     }
245
246     public void oppdaterMenyForPostIt()
247     {
248         theMenyer. oppdaterMenyForPostIt();
249     }
250
251     public void oppdaterMenyForGruppe()
252     {
253         theMenyer. oppdaterMenyForGruppe();
254     }
255

```

```

256 public void oppdaterMenyForTavle()
257 {
258     theMenyer.oppdaterMenyForTavle();
259 }
260
261 public void oppdaterMenyForDistribuert(boolean b)
262 {
263     theMenyer.oppdaterMenyForDistribuert(b);
264 }
265
266 public void oppdaterScrollPane ()
267 {
268     scrollPane.validate ();
269 }
270
271 }

```

## F.1.2 GrafiskeTavleElementer.java

### Listing: GrafiskeTavleElementer.java

```

1 package kjapp;
2
3 import javax.swing.*;
4 import javax.swing.JPanel.*;
5 import javax.swing.JScrollPane;
6 import java.awt.event.*;
7 import java.awt.*;
8 import java.awt.print.*;
9 import java.util.Vector;
10 import java.lang.String;
11
12 // muselyttene trenger følgende pakker:
13 import java.awt.event.MouseListener;
14 import java.awt.event.MouseMotionListener;
15 import java.awt.event.MouseEvent;
16
17 // pakker for printing
18 import javax.swing.print.*;
19 import javax.swing.print.AttributeSet;
20
21 /**
22  *
23  * <p>Title: GrafiskeTavleElementer </p>
24  * <p>Description: Klasse som benyttes for å tegne opp elementer i tavla. Denne benyttes til å tegne opp modellen som er
25  * lagret i Tavle, TavleElement, Gruppe og PostIT </p>
26  * @author 'aHvard Alstad
27  * @version 1.0
28  */
29 public class GrafiskeTavleElementer extends JPanel implements MouseListener, MouseMotionListener, Printable, KeyListener
30 {
31     private JPanel pane;
32     private Brukergrensesnitt theBrukergrensesnitt;
33     private Kontroller k;
34     private JLabel gTema;
35     private boolean mouseIsPressed = false;
36     private static long paintTeller = 0;
37     private static long paintComponentTeller = 0;
38     private String skriftType = "Courier";
39     private int MaxRight = 0;
40     private int MaxBottom = 0;
41     private Posisjon tegnMin;
42     private Posisjon tegnMax;
43
44     public GrafiskeTavleElementer ( Brukergrensesnitt b, Kontroller k)
45     {
46         gTema = new JLabel("", SwingConstants.CENTER);
47         Font f = new Font("SansSerif", Font.PLAIN, 20);
48         gTema.setFont(f);
49         setPreferredSize (new Dimension(600,800));
50         theBrukergrensesnitt = b;
51         theKontroller = k;
52         this.add(gTema);
53         setBackground(Color.white);
54         this.addMouseListener(this);
55         this.addMouseMotionListener(this);
56         this.addKeyListener(this);
57     }
58
59     public boolean isFocusTraversable ()
60     {
61         return true;
62     }
63
64     public void paint (Graphics g)
65     {
66         super.paint (g);
67     }
68
69     public void paintComponent (Graphics g)
70     {
71         super.paintComponent(g);
72         Tavle t = theKontroller.hentTavle ();
73         t.setTavleBredde (this.getWidth());
74         t.setTavleHoyde (this.getHeight());

```

```

75     Graphics2D g2d = (Graphics2D)g;
76     tegnTavle(g2d,t);
77 }
78
79 public void settTegnbarOmrade(Posisjon min, Posisjon max)
80 {
81     tegnMin=min;
82     tegnMax=max;
83 }
84
85 public boolean skalTegnes(TavleElement t)
86 {
87     boolean returverdi = false;
88     Posisjon pos = theKontroller.fi nnPosisjonGruppe0(t);
89     Posisjon posMax = new Posisjon(pos.xPos + t.hentElementBredde(),pos.yPos+t.hentElementHoyde());
90     Posisjon posMin = new Posisjon(pos.xPos, pos.yPos);
91     if (sjekkPunkt(posMin.xPos,posMin.yPos) returverdi = true;
92     if (sjekkPunkt(posMax.xPos,posMax.yPos) returverdi = true;
93     if (sjekkPunkt(posMax.xPos,posMin.yPos) returverdi = true;
94     if (sjekkPunkt(posMin.xPos,posMax.yPos) returverdi = true;
95     if (sjekkOmradeInnenfor(posMin,posMax)) returverdi = true;
96     return returverdi;
97 }
98
99 public boolean sjekkPunkt(int x, int y)
100 {
101     if (tegnMin.xPos < x && x < tegnMax.xPos && tegnMin.yPos < y && y < tegnMax.yPos ){
102         return true;
103     }else{
104         return false;
105     }
106 }
107
108 public boolean sjekkOmradeInnenfor(Posisjon posMin,Posisjon posMax)
109 {
110     if ((posMin.xPos <= tegnMin.xPos && tegnMin.xPos <= posMax.xPos && posMin.yPos <= tegnMin.yPos && tegnMin.yPos <= posMax.yPos) &&
111         (posMin.xPos <= tegnMin.xPos && tegnMin.xPos <= posMax.xPos && posMin.yPos <= tegnMax.yPos && tegnMax.yPos <= posMax.yPos) &&
112         (posMin.xPos <= tegnMax.xPos && tegnMax.xPos <= posMax.xPos && posMin.yPos <= tegnMin.yPos && tegnMin.yPos <= posMax.yPos) &&
113         (posMin.xPos <= tegnMax.xPos && tegnMax.xPos <= posMax.xPos && posMin.yPos <= tegnMax.yPos && tegnMax.yPos <= posMax.yPos)){
114         return true;
115     }else{
116         return false;
117     }
118 }
119
120 /**
121  * Metode som tegner opp alle elementer i tavla rekursivt . Tavlen inneholder gruppe0 som igjen inneholder alle
122  * grupper og PostIts i tavla.
123  * @param g - Graphics objektet elementene skal tegnes 'ap
124  * @param t - tavle som skal tegnes opp
125  */
126 public void tegnTavle(Graphics2D g,Tavle t)
127 {
128     //Tegner alle tavleElementer i Tavle
129     Gruppe gruppe0 = t.hentGruppe0();
130     String tema = gruppe0.hentElementNavn();
131     int posX = gruppe0.hentElementPosisjonX();
132     int posY = gruppe0.hentElementPosisjonY();
133     g.Tema.setText(tema); //Lagt 'ap pane i ø konstruktren , men med en tom streng
134     Vector tavleElementer = gruppe0.hentTavleElementer();
135     TavleElement tavleElement;
136     MaxRight = 0;
137     MaxBottom = 0;
138
139     if (tavleElementer != null){
140         for (int i = 0; i < tavleElementer.size(); i++){
141             tavleElement = (TavleElement)tavleElementer.elementAt(i);
142             if (tavleElement instanceof Gruppe){
143                 //if ( skalTegnes (tavleElement) ) tegnGruppe(g,(Gruppe)tavleElement, false );
144                 tegnGruppe(g,(Gruppe)tavleElement, false );
145             }else{
146                 //if ( skalTegnes (tavleElement) ) tegnPostIt (g,( PostIt )tavleElement, false );
147                 tegnPostIt (g,( PostIt )tavleElement, false );
148             }
149         }
150         //Tegner alle relasjoner i tavle
151         Vector relasjoner = t.hentRelasjoner();
152         if ( relasjoner != null){
153             for (int i2 = 0; i2 < relasjoner.size(); i2++){
154                 tegnRelasjon (g,( Relasjon ) relasjoner.elementAt(i2));
155             }
156         }
157
158         t.setTavleBredde (MaxRight + 50);
159         t.setTavleHoyde(MaxBottom + 50);
160         this.setPreferredSize (new Dimension(t.hentTavleBredde(), t.hentTavleHoyde()));
161     }
162 }
163
164 public void tegnGruppe(Graphics2D g,Gruppe gruppe,boolean tegnBakgrunn)
165 {
166     double z = theKontroller.hentZoomFaktor();
167     String navn = gruppe.hentElementNavn();
168     int bredde = doubleToInt(gruppe.hentElementBredde()*z);
169     int hoyde = doubleToInt(gruppe.hentElementHoyde()*z);
170     Posisjon pos = theKontroller.fi nnPosisjonGruppe0(gruppe);
171     int x = doubleToInt(pos.xPos*z);
172     int y = doubleToInt(pos.yPos*z);
173     g.setColor (Color.orange);

```

```

174
175 //Lagrer ø strste x og y verdi som forekommer i tavla som tegnes opp ....
176 if ( x + bredde > MaxRight) MaxRight = x + bredde;
177 if ( y + hoyde > MaxBottom) MaxBottom = y + hoyde;
178
179 g. fi llRoundRect(x,y,bredde,hoyde,20,20);
180 g.setColor(Color.gray);
181 g.drawRoundRect(x,y,bredde,hoyde,20,20);
182 g.drawLine(x+bredde-20,y+hoyde-2,x+bredde-2,y+hoyde-20);
183 g.drawLine(x+bredde-15,y+hoyde-2,x+bredde-2,y+hoyde-15);
184 g.drawLine(x+bredde-10,y+hoyde-2,x+bredde-2,y+hoyde-10);
185 g.setColor(Color.black);
186 tegnElementNavn(navn,x,y,bredde,hoyde,g,true);
187
188 //Sjekker om gruppen inneholder flere elementer som skal tegnes opp
189 Vector tavleElementer = gruppe.hentTavleElementer();
190 if ( tavleElementer != null) {
191     int storrels = tavleElementer.size();
192     TavleElement t;
193
194     for (int i = 0; i < storrels; i++) {
195         t = (TavleElement)tavleElementer.elementAt(i);
196         if ( t instanceof Gruppe) {
197             // if ( skalTegnes(t) ) tegnGruppe(g,(Gruppe)t,false); //NB! Rekursjon ...
198             tegnGruppe(g,(Gruppe)t,false); //NB! Rekursjon ...
199         } else {
200             // if ( skalTegnes(t) ) tegnPostIt(g,( PostIt)t,false);
201             tegnPostIt(g,( PostIt)t,false);
202         }
203     }
204 }
205
206 if (gruppe == theKontroller.hentAktivTavleElement())
207 {
208     if (! theKontroller.hentVelgOmrade()) {
209         this.merkTavleElement(gruppe.g);
210     }
211 }
212
213 }
214
215 /**
216  * Metode som tegner opp en post - it lapp
217  * Lappen blir tegnet opp relativt til gruppen den ø tilhør. Ø
218  * vre venstre ø hjrne for gruppen er x0,y0
219  */
220 public void tegnPostIt (Graphics2D g, PostIt p, boolean tegnBakgrunn)
221 {
222     double z = theKontroller.hentZoomFaktor();
223     PostIt postIt = p;
224     String navn = p.hentElementNavn();
225     int bredde = (int)(p.hentElementBredde()*z);
226     int hoyde = (int)(p.hentElementHoyde()*z);
227     Posisjon pos = theKontroller.fi nnPosisjonGruppe0(p);
228
229     int x = (int)(pos.xPos*z);
230     int y = (int)(pos.yPos*z);
231
232     //Lagrer ø strste x og y verdi som forekommer i tavla som tegnes opp ....
233     if ( x + bredde > MaxRight) MaxRight = x + bredde;
234     if ( y + hoyde > MaxBottom) MaxBottom = y + hoyde;
235
236     switch ( p.hentPostItType() ) {
237     case PostIt.STATUS_NOYTRAL:
238         g.setColor( PostIt.COLOR_NOYTRAL);
239         break;
240     case PostIt.STATUS_POSITIV:
241         g.setColor( PostIt.COLOR_POSITIV);
242         break;
243     case PostIt.STATUS_NEGATIV:
244         g.setColor( PostIt.COLOR_NEGATIV);
245         break;
246     }
247     g.draw3DRect(x,y,bredde,hoyde,true);
248     g.fi ll3DRect(x,y,bredde,hoyde,true);
249     g.setColor(Color.gray);
250     g.drawLine(x+bredde-15,y+hoyde-2,x+bredde-2,y+hoyde-15);
251     g.drawLine(x+bredde-10,y+hoyde-2,x+bredde-2,y+hoyde-10);
252     g.drawLine(x+bredde-5,y+hoyde-2,x+bredde-2,y+hoyde-5);
253     g.setColor(Color.black);
254     String s = navn;
255     tegnElementNavn(s,x,y,bredde,hoyde,g,false);
256
257     if ( postIt == theKontroller.hentAktivTavleElement() ) {
258         // Dersom postIt er aktivTavleElement skal denne tegnes opp med merking
259         this.merkTavleElement(postIt.g);
260     }
261 }
262
263 /**
264  * Metode som bryter opp en tekststreng i passe store linjer og tegner denne ^ap
265  * et graphics objekt .
266  * @param s - strengen som skal tegnes opp
267  * @param x - x posisjon for ø vre venstre ø hjrne av tavleelementet teksten skal tegnes i
268  * @param y - y posisjon for ø vre venstre ø hjrne av tavleelementet teksten skal tegnes i
269  * @param g - Graphics - objektet strengen skal tegnes ^ap.
270  */
271 private void tegnElementNavn(String s,int x,int y,int bredde, int hoyde, Graphics2D g,boolean isGruppe)
272 {

```

```

273 double z = theKontroller.hentZoomFaktor();
274 int xoffset ;
275 int yoffset ;
276 int elementBredde;
277 int endCharacter;
278 int tekstHoyde = 0;
279 Font f;
280 int fontstorrelse =14;
281
282 if (isGruppe){
283     xoffset = 5;
284     yoffset = doubleToInt(14);
285     elementBredde = doubleToInt(bredde/8);
286     if (elementBredde < 3) elementBredde = 3;
287     endCharacter = -1;
288     f = new Font(skriftType , Font.PLAIN, fontstorrelse );
289 } else {
290     xoffset = 5;
291     yoffset = doubleToInt(12);
292     elementBredde = doubleToInt(bredde/7.4);
293     if (elementBredde < 3) elementBredde = 3;
294     endCharacter = -1;
295     f = new Font(skriftType , Font.PLAIN, 11);
296 }
297
298 g.setFont(f);
299 tekstHoyde = 2 * yoffset ;
300 if(0.5 < z || isGruppe){
301     while (s.length() > elementBredde){
302         int i1 = s.indexOf(" ");
303         int i2 = i1;
304         if ((i1 < elementBredde) && (i1 != endCharacter)) { //Det finnes mellomrom øfr teksten 'am brytes
305             while (i1 < elementBredde && i1 != endCharacter) { //ØPrver'å fi nne det mellomrommet som er nærmest character 12
306                 i1 = s.indexOf(" ", i1 + 1);
307                 if (i1 < elementBredde && i1 != endCharacter) i2 = i1;
308             }
309             g.drawString(s.substring(0,i2),x + xoffset ,y + yoffset);
310             s = s.substring(i2,s.length());
311             s = s.trim();
312             y = y + yoffset;
313             tekstHoyde = tekstHoyde + yoffset;
314             if (tekstHoyde > hoyde){
315                 s = "  ";
316                 break;
317             }
318         } else { //Det finnes ingen mellomrom øfr teksten 'am brytes
319             i2 = elementBredde - 1;
320             g.drawString(s.substring(0,i2) + " ",x + xoffset ,y + yoffset);
321             s = s.substring(i2,s.length());
322             s = s.trim();
323             y = y + yoffset;
324             tekstHoyde = tekstHoyde + yoffset;
325             if (tekstHoyde > hoyde){
326                 s = "  ";
327                 break;
328             }
329         }
330     }
331     g.setColor(Color.black);
332     g.drawString(s,x+xoffset ,y+yoffset);
333 } else {
334     g.drawString(" ...",x+xoffset ,y+yoffset);
335 }
336 }
337
338
339 /**
340  * Relasjoner tegnes fra en gruppe til en annen gruppe.
341  * - Hvor 'ap gruppen streken skal tegnes øavgjres av hvilke to punkter som er nærmest hverandre.
342  * - Hver gruppe har 8 steder som betegnes som connectorer. Disse er plassert :
343  *   - i alle øhjørnene 'ap gruppen
344  *   - midt 'ap alle sider av gruppen
345  * - Finner øfrist alle mulige punkter.
346  * - Finner øretter det paret som øutgjir den korteste relasjonen
347  */
348 public void tegnRelasjon(Graphics2D g,Relasjon r)
349 {
350     Gruppe g_fra = r.hentFraGruppe();
351     Gruppe g_til = r.hentTilGruppe();
352     if (g_fra != null && g_til != null) {
353         Vector v_fra = new Vector();
354         Vector v_til = new Vector();
355         Posisjon p_fra = theKontroller.fi nnPosisjonGruppe0(g_fra);
356         Posisjon p_til = theKontroller.fi nnPosisjonGruppe0(g_til);
357         int b_fra = g_fra.hentElementBredde();
358         int b_til = g_til.hentElementBredde();
359         int h_fra = g_fra.hentElementHoyde();
360         int h_til = g_til.hentElementHoyde();
361
362         v_fra.add(p_fra);
363         v_fra.add(new Posisjon(p_fra.xPos + b_fra/2, p_fra.yPos));
364         v_fra.add(new Posisjon(p_fra.xPos + b_fra, p_fra.yPos));
365         v_fra.add(new Posisjon(p_fra.xPos + b_fra, p_fra.yPos + h_fra/2));
366         v_fra.add(new Posisjon(p_fra.xPos + b_fra, p_fra.yPos + h_fra));
367         v_fra.add(new Posisjon(p_fra.xPos + b_fra/2, p_fra.yPos + h_fra));
368         v_fra.add(new Posisjon(p_fra.xPos, p_fra.yPos + h_fra));
369         v_fra.add(new Posisjon(p_fra.xPos, p_fra.yPos + h_fra/2));
370
371         v_til.add(p_til);

```

```

372     v.ail.add(new Posisjon( p.ail .xPos + h.ail /2, p.ail .yPos));
373     v.ail.add(new Posisjon( p.ail .xPos + h.ail , p.ail .yPos));
374     v.ail.add(new Posisjon( p.ail .xPos + h.ail , p.ail .yPos + h.ail /2));
375     v.ail.add(new Posisjon( p.ail .xPos + h.ail , p.ail .yPos + h.ail ));
376     v.ail.add(new Posisjon( p.ail .xPos + h.ail /2, p.ail .yPos + h.ail ));
377     v.ail.add(new Posisjon( p.ail .xPos, p.ail .yPos + h.ail ));
378     v.ail.add(new Posisjon( p.ail .xPos, p.ail .yPos + h.ail /2));
379
380     double avstand;
381     double kortesteavstand=1000000;
382     Posisjon kortesteFra = null;
383     Posisjon kortesteTil = null;
384     Posisjon tempFra;
385     Posisjon tempTil;
386
387     for(int i1 = 0; i1 < v.fra.size(); i1++){
388         tempFra = ( Posisjon) v.fra.elementAt(i1);
389         for(int i2 = 0; i2 < v.ail.size(); i2++){
390             tempTil = ( Posisjon) v.ail.elementAt(i2);
391             avstand = Math.sqrt(Math.pow((double)tempFra.xPos - (double)tempTil.xPos,2)+Math.pow((double)tempFra.yPos - (double)tempTil.yPos,2));
392             if ( avstand < kortesteavstand){
393                 kortesteavstand = avstand;
394                 kortesteFra = tempFra;
395                 kortesteTil = tempTil;
396             }
397         }
398     }
399
400     double z = theKontroller.hentZoomFaktor();
401     g.drawLine( doubleToInt( kortesteFra.xPos*z),
402                doubleToInt( kortesteFra.yPos*z),
403                doubleToInt( kortesteTil.xPos*z),
404                doubleToInt( kortesteTil.yPos*z));
405     }
406 }
407
408 public GrafiskeTavleElementer hentGrafiskeTavleElementer ()
409 {
410     return this;
411 }
412
413 public void mouseClicked(MouseEvent arg0)
414 {
415 }
416
417 public void mouseEntered(MouseEvent arg0)
418 {
419 }
420
421 public void mouseExited(MouseEvent arg0)
422 {
423 }
424
425 public void mousePressed(MouseEvent arg0)
426 {
427     double z = theKontroller.hentZoomFaktor();
428     this.requestFocus();
429     int x = doubleToInt(arg0.getX()/z);
430     int y = doubleToInt(arg0.getY()/z);
431
432     if (theKontroller.hentToggleRelasjon()){
433         Relasjon r = theKontroller.hentRelasjon();
434         theKontroller.fi nnTavleElement(x,y);
435         TavleElement t = theKontroller.hentAktivTavleElement();
436         Tavle tavle = theKontroller.hentTavle();
437         if (t != tavle.hentGruppe0()){
438             if ( t instanceof Gruppe){
439                 r.setFraGruppe((Gruppe)t);
440             }else{
441                 r.setFraGruppe(t.hentForelder());
442             }
443         }
444     }else if ( theKontroller.hentVelgOmrade() && !arg0.isPopupTrigger()){
445         theBrukergrensenitt.setMerkStartPosisjon(new Posisjon(x,y));
446         theKontroller.fi nnTavleElement(x,y);
447     }else{
448         theKontroller.fi nnTavleElement(x,y);
449     }
450 }
451
452 public void mouseReleased(MouseEvent arg0)
453 {
454     double z = theKontroller.hentZoomFaktor();
455     int x = doubleToInt(arg0.getX()/z);
456     int y = doubleToInt(arg0.getY()/z);
457
458     //Disse to burde erstattes med en metode i kontrolleren ..... HA
459     Tavle tavle = theKontroller.hentTavle();
460     Gruppe g0 = tavle.hentGruppe0();
461     mouseIsPressed = false;
462     if ( arg0.isPopupTrigger()){
463         theBrukergrensenitt.setAktivStartPosisjon(new Posisjon(x,y));
464
465         theBrukergrensenitt.tegnHoyreklikkMeny(x,y);
466         theBrukergrensenitt.validate();
467     }else if ( theKontroller.hentToggleRelasjon()){
468         theKontroller.fullforRelasjon(x,y);
469     }else if ( theKontroller.hentVelgOmrade()){
470         theKontroller.setVelgOmrade(false);

```

```

471
472     Posisjon posStart = theBrukerGrensesnitt . hentMerkStartPosisjon ();
473     int startX = posStart . xPos;
474     int startY = posStart . yPos;
475     if ( startX > x){
476         int tmpX = startX;
477         startX = x;
478         x = tmpX;
479     }
480     if ( startY > y){
481         int tmpY = startY;
482         startY = y;
483         y = tmpY;
484     }
485
486     theBrukerGrensesnitt . settMerkStartPosisjon (new Posisjon( startX , startY ));
487     theBrukerGrensesnitt . settMerkSluttPosisjon (new Posisjon(x,y));
488
489     Vector v = theKontroller . finnTavleElementerInnenOmrade( startX , startY , x , y);
490     theKontroller . finnGruppe(x,y);
491     if (v!=null){
492         for( int i=0; i<v.size(); i++){
493             TavleElement t = (TavleElement)v.elementAt(i);
494             Graphics g = getGraphics();
495             t.merkTavleElement(g, theKontroller);
496         }
497     }
498 } else {
499     // Setter aktiv gruppe i kontroller — Gruppen museknappen ble sluppet i ...
500     theKontroller . finnGruppe(x,y);
501     // Sjekker om TavleElementet som ble flyttet trenger å f oppdatert sin ø gruppertilhørighet .
502     theKontroller . flyttTavleElement (x,y);
503
504     // Skal kontrollere ø gruppertilhørighet bare dersom man ikke har resizet et element ...
505     if (! theKontroller . hentResizeElement()){
506         theKontroller . sjekkGruppertilhørighet (); //HA
507     }
508
509     TavleElement tavleE = theKontroller . hentAktivTavleElement();
510     if ( tavleE!=g0){
511         theKontroller . sjekkElementUtstrekning( tavleE , tavleE . hentForelder ()); //HA
512     }
513     theKontroller . settResizeElement ( false ); // Denne settes i finnTavleElement dersom brukeren har trykket nederst i øhyre ø hjørne av elementet
514 }
515 }
516
517 public void mouseDragged(MouseEvent arg0)
518 {
519     double z = theKontroller . hentZoomFaktor();
520     int x = doubleToInt(arg0 . getX()/z);
521     int y = doubleToInt(arg0 . getY()/z);
522
523     if (!arg0 . isPopupTrigger()){
524         if ( theKontroller . hentVelgOmrade()){
525             /* Dersom brukeren ønsker å markere et område (med feks en eller flere postit lapper)
526              * settes ø first den boolske variabelen velgOmrade i Kontroller lik true .
527              * å Nr velgOmrade = true, skal det tegnes opp en firkant ved
528              * en mouseDragged hendelse. Elementene som ligger innenfor firkanten
529              * vil bli merket. Dette ø gires med metoden merkTavleElement
530              */
531             Posisjon posStart = theBrukerGrensesnitt . hentMerkStartPosisjon ();
532             int startX = doubleToInt( posStart . hentXPos());
533             int startY = doubleToInt( posStart . hentYPos());
534             int endx = x;
535             int endy = y;
536             Graphics2D g = (Graphics2D) getGraphics();
537
538             if ( startX > endx){
539                 int tmpX = startX;
540                 startX = endx;
541                 endx = tmpX;
542             }
543             if ( startY > endy){
544                 int tmpY = startY;
545                 startY = endy;
546                 endy = tmpY;
547             }
548
549             if ( theKontroller . hentJobberLokalt ()){
550                 this . tegnOmradeFirkant(doubleToInt( startX *z) , doubleToInt( startY *z) , doubleToInt( endx*z) , doubleToInt( endy*z) , g);
551             }
552             else {
553                 if ( theKontroller . hentJobberLokalt ()){
554                     theKontroller . flyttTavleElement (x,y);
555                 }
556             }
557         }
558     }
559 }
560
561 public void mouseMoved(MouseEvent arg0)
562 {
563 }
564
565 public int print(Graphics g , PageFormat pf , int page) throws PrinterException
566 {
567     if ( page >=1) return Printable . NO_SUCH_PAGE;
568     Graphics2D g2 = (Graphics2D)g;
569     g2 . translate (pf . getImageableX(), pf . getImageableY());

```

```

570     Tavle t = theKontroller . hentTavle () ;
571     tegnTavle (g2,t);
572     return Printable .PAGE_EXISTS;
573 }
574
575 public void merkTavleElement(TavleElement t, Graphics2D g)
576 {
577     double z = theKontroller .hentZoomFaktor();
578     Posisjon pos = theKontroller .finnPosisjonGruppe0(t);
579     int x = doubleToInt(pos.hentXPos()*z);
580     int y = doubleToInt(pos.hentYPos()*z);
581     int hoyde = doubleToInt(t.hentElementHoyde()*z);
582     int bredde = doubleToInt(t.hentElementBredde()*z);
583     int hjorneMerking = doubleToInt(3*z); // Gir ø strelsen ̂ap merking av post –it
584     if (hjorneMerking < 2) hjorneMerking=2;
585     g.setColor(Color.black);
586     g.draw3DRect(x-hjorneMerking,y-hjorneMerking,2*hjorneMerking,2*hjorneMerking,true);
587     g.draw3DRect(x+bredde-hjorneMerking,y-hjorneMerking,2*hjorneMerking,2*hjorneMerking,true);
588     g.draw3DRect(x-hjorneMerking,y+hoyde-hjorneMerking,2*hjorneMerking,2*hjorneMerking,true);
589     g.draw3DRect(x+bredde-hjorneMerking,y+hoyde-hjorneMerking,2*hjorneMerking,2*hjorneMerking,true);
590 }
591
592 public void tegnOmradeFirkant(int startx, int starty, int endx, int endy, Graphics2D g)
593 {
594     if (startx < endx && starty < endy) {
595         g.drawRect(startx, starty, endx-startx, endy-starty);
596         repaint ();
597     }
598 }
599
600 public void keyPressed(KeyEvent e)
601 {
602 }
603
604 public void keyTyped(KeyEvent e)
605 {
606 }
607
608 public void keyReleased(KeyEvent e)
609 {
610     int knappKode = e.getKeyCode();
611     if (knappKode == KeyEvent.VK_A && e.isControlDown()){
612         theKontroller .merkAlt();
613         //System.out.println("Ctrl + A er nede ... merker alt");
614     }
615     if (knappKode == KeyEvent.VK_C && e.isControlDown()){
616         theKontroller .kopierTavleElementer ();
617         //System.out.println("Ctrl + C er nede ... Kopierer");
618     }
619     if (knappKode == KeyEvent.VK_X && e.isControlDown()){
620         theKontroller .klippUtTavleElement ();
621         //System.out.println("Ctrl + X er nede ... Klipper ut");
622     }
623     if (knappKode == KeyEvent.VK_V && e.isControlDown()){
624         theKontroller .limInnTavleElement ();
625         //System.out.println("Ctrl + V er nede ... Limer inn");
626     }
627     if (knappKode == KeyEvent.VK_DELETE) {
628         theKontroller .slettMerkeide ();
629     }
630
631     if (knappKode == KeyEvent.VK_S && e.isControlDown()){
632         theKontroller .lagreTavle ();
633     }
634
635     if (knappKode == KeyEvent.VK_O && e.isControlDown()){
636         theKontroller .apneTavle ();
637     }
638
639     if (knappKode == KeyEvent.VK_Z && e.isControlDown()){
640         theKontroller .angreSiste ();
641     }
642
643     if (knappKode == KeyEvent.VK_Z && e.isControlDown() && e.isShiftDown()){
644         theKontroller .gjenoppsettSiste ();
645     }
646 }
647
648 public static int doubleToInt(double d)
649 {
650     Double d2 = new Double(d);
651     int i = d2.intValue ();
652     return i;
653 }
654 }

```

### F.1.3 Gruppe.java

#### Listing: Gruppe.java

```

1 package kjapp;
2
3 import java.util.Vector;
4 import java.io.*;
5 import java.awt.*;

```

```

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104

```

```

/**
 * <p>Title: Gruppe</p>
 * <p>Description: Benyttes for å representere Grupper i modellen. Grupper har de samme egenskaper som TavleElementer, men de har i tillegg en vektor med tavleElementer i gruppe og Relasjoner i gruppe. Dvs at grupper kan inneholde både andre grupper og Post-its ..</p>
 * Post-its ..</p>
 * @author Årvid Alstad
 * @version 1.0
 */
public class Gruppe extends TavleElement
{
    private Vector tavleElementerGruppe;
    private Vector relasjonerGruppe;
    public TavleElement theTavleElement;
    public Tavle theTavle;

    /**
     * Tom økonstruktør for gruppe. Benyttes ifm parsing av xml fil.
     */
    public Gruppe(Tavle t)
    {
        super();
        tavleElementerGruppe = new Vector();
        theTavle = t;
        relasjonerGruppe = new Vector();
    }

    /**
     * ø Konstruktør for å sette navn, posisjon, bredde, øhyde, å niv og kommentar for gruppe
     */
    public Gruppe(String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar, Gruppe g, Tavle t)
    {
        super(navn, x, y, bredde, hoyde, niva, kommentar, g);
        tavleElementerGruppe = new Vector();
        System.out.println("Ny gruppe er opprettet");
        theTavle = t;
        relasjonerGruppe = new Vector();
    }

    /**
     * ø Konstruktør for å sette navn for gruppe
     * Ikke i bruk ..
     */
    public Gruppe(String navn, Tavle t)
    {
        super(navn);
        theTavle = t;
        relasjonerGruppe = new Vector();
    }

    public Gruppe(String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar, Gruppe forelder)
    {
        super(navn, x, y, bredde, hoyde, niva, kommentar, forelder);
        tavleElementerGruppe = new Vector();
        relasjonerGruppe = new Vector();
    }

    public void leggTilUnderGruppe(Gruppe g)
    {
        tavleElementerGruppe.addElement(g);
    }

    public void leggTilPostIt ( PostIt p)
    {
        tavleElementerGruppe.addElement(p);
    }

    /**
     * Legger til et TavleElement i vektoren tavleElementerGruppe
     * @param t - tavleElementet som skal leges til
     */
    public void leggTilTavleElement (TavleElement t)
    {
        tavleElementerGruppe.addElement(t);
    }

    /**
     * Metode som fjerner gitte tavleElement fra vektoren tavleElementerGruppe
     * @param t - elementet som skal fjernes
     */
    public void fjernTavleElement (TavleElement t)
    {
        if (tavleElementerGruppe != null){
            for (int i = 0; i < tavleElementerGruppe.size(); i++){
                if (tavleElementerGruppe.elementAt(i) == t) tavleElementerGruppe.removeElementAt(i);
            }
        }
    }

    public PostIt nyPostIt ()
    {
        PostIt p = new PostIt ();
        return p;
    }

    public void leggTilRelasjon (Relasjon r)

```

```

105 {
106     relasjonerGruppe .addElement(r);
107 }
108
109 public Vector hentRelasjoner ()
110 {
111     return relasjonerGruppe ;
112 }
113
114 public void slettRelasjon (Relasjon r)
115 {
116     if ( relasjonerGruppe !=null) {
117         for (int i = 0; i < relasjonerGruppe.size (); i++){
118             if ( relasjonerGruppe.elementAt(i)==r){
119                 relasjonerGruppe .removeElementAt(i);
120             }
121         }
122     }
123 }
124
125 /**
126  * fi nnTavleElement (int xPos, int yPos) fi mner tavleElementet
127  * som brukeren har trykket 'ap. Dette kan enten være en gruppe eller
128  * en post-it.
129  */
130 public TavleElement fi nnTavleElement (int xPos, int yPos, Kontroller k)
131 {
132     int elementXPos;
133     int elementYPos;
134     int elementBredde;
135     int elementHoyde;
136
137     TavleElement kandidat = this;
138     if ( tavleElementeriGruppe!=null) {
139         if (tavleElementeriGruppe.isEmpty())
140             return kandidat;
141         else
142         {
143             for (int i = 0; i < tavleElementeriGruppe.size (); i++) {
144                 TavleElement element = (TavleElement)tavleElementeriGruppe.elementAt(i);
145                 Posisjon pos = k.fi nnPosisjonGruppe0(element);
146
147                 elementXPos = pos.hentXPos();
148                 elementYPos = pos.hentYPos();
149                 elementBredde = element.hentElementBredde();
150                 elementHoyde = element.hentElementHoyde();
151                 Tavle tavle = k.hentTavle ();
152                 if ((( elementXPos < xPos) && (xPos < elementXPos + elementBredde)) &&
153                     ((elementYPos < yPos) && (yPos < elementYPos + elementHoyde))){
154                     kandidat = element;
155                     //Finn ut om brukeren har trykket i resize ø hjrnet ...
156                     if (((elementXPos + elementBredde - 20 < xPos) && (xPos < elementXPos + elementBredde)) &&
157                         ((elementYPos + elementHoyde - 20 < yPos) && (yPos < elementYPos + elementHoyde))){
158                         //Brukeren har trykket i ø resizehjernet og skal ikke dra elementer, men resize det ...
159                         //Å M sette fbgg i Tavlen — Dette 'am resettes i det brukeren slipper musknappen
160                         tavle . settResizeElement (true);
161                         System.out. println ("resizeelement =true");
162                     }else{
163                         tavle . settResizeElement ( false);
164                         System.out. println ("resizeelement =false");
165                     }
166                 }
167             }
168             if (( kandidat == this) || ( kandidat instanceof PostIt ))
169                 {
170                     return kandidat;
171                 }
172             else
173                 {
174                     Gruppe g = (Gruppe)kandidat; //Ver 'an at dette 'am være en Gruppe
175                     return g.fi nnTavleElement (xPos, yPos,k);
176                 }
177         }
178     }
179     return null; //HA09 ifm sjekk 'ap nullpointer for vektor
180 }
181
182 /**
183  * Samme som fi nnTavleElement, men denne returnerer alltid en Gruppe
184  * Jobber alltid i forhold til absolutt posisjon. Dvs: I forhold til gruppe0
185  * @param xPos — int x Posisjonen for museevent
186  * @param yPos — int y Posisjonen for museevent
187  * @param k — Kontrolleren — gir tilgang 'ap metoden fi nnPosisjonGruppe0
188  * @param aktivTavleElement — TavleElement — sikrer at aktiv gruppe og aktivTavleElement ikke refererer samme tavleElement
189  * @return — Gruppe
190  */
191 public Gruppe fi nnGruppe(int xPos, int yPos, Kontroller k, TavleElement aktivTavleElement )
192 {
193     int elementXPos;
194     int elementYPos;
195     int elementBredde;
196     int elementHoyde;
197
198     Gruppe kandidat = this;
199     if ( tavleElementeriGruppe!=null) {
200         if (tavleElementeriGruppe.isEmpty())
201             return kandidat;
202         else
203         {

```

```

204     for (int i = 0; i < tavleElementerGruppe.size(); i++) {
205         TavleElement element = (TavleElement)tavleElementerGruppe.elementAt(i);
206         if ((element instanceof Gruppe) && (element != aktivTavleElement)) { //Aktiv gruppe skal aldri være det samme elementet som aktivGruppe.
207             Posisjon pos = k.fi nnPosisjonGruppe0(element);
208             elementXPos = pos.hentXPos();
209             elementYPos = pos.hentYPos();
210             elementBredde = element.hentElementBredde();
211             elementHoyde = element.hentElementHoyde();
212             if (((elementXPos < xPos) && (xPos < elementXPos + elementBredde)) &&
213                 ((elementYPos < yPos) && (yPos < elementYPos + elementHoyde)))
214                 kandidat = (Gruppe)element;
215         }
216     }
217
218     if (kandidat == this)
219     {
220         return kandidat;
221     }
222     else
223     {
224         //return (Gruppe)kandidat.fi nnTavleElement(xPos, yPos);
225         Gruppe g = kandidat; //Vei 'an at dette 'an være en Gruppe
226         return g.fi nnGruppe(xPos, yPos, k, aktivTavleElement);
227     }
228 }
229 }
230 return null; //HA09 ifm sjekk 'ap nullpointer for vektor
231 }
232
233 public Gruppe fi nnGruppe(Tavle t, long id) {
234     // Metoden returnerer en gruppe med gitt navn
235     Vector v = this.hentTavleElementer();
236     Gruppe kandidat = null;
237     Gruppe funnetKandidat = null;
238     if (v != null) {
239         for (int i = 0; i < v.size(); i++) {
240             if (v.elementAt(i) instanceof Gruppe) {
241                 Gruppe g = (Gruppe)v.elementAt(i);
242                 if (g.hentElementID() == id) {
243                     funnetKandidat = g;
244                 }
245                 else {
246                     kandidat = g.fi nnGruppe(t, id);
247                     if (kandidat != null) {
248                         funnetKandidat = kandidat;
249                     }
250                 }
251             }
252         }
253     }
254     return funnetKandidat;
255 }
256
257 public Vector hentTavleElementer()
258 {
259     return tavleElementerGruppe;
260 }
261
262 public void toString(PrintWriter fi1, String i)
263 //public void toString(String fi1, String i)
264 {
265     Vector elementer = hentTavleElementer();
266
267     String indenter = i + "\t";
268     fi1.println(i + "<gruppe>");
269     fi1.println(indenter + "<gruppeid>" + hentElementID() + "</gruppeid>");
270     fi1.println(indenter + "<gruppenavn>" + hentElementNavn() + "</gruppenavn>");
271     if (!hentElementKommentar().equalsIgnoreCase("")) fi1.println(indenter + "<gruppekommentar>" + hentElementKommentar() + "</gruppekommentar>");
272     fi1.println(indenter + "<gruppeposisjonX>" + hentElementPosisjonX() + "</gruppeposisjonX>");
273     fi1.println(indenter + "<gruppeposisjonY>" + hentElementPosisjonY() + "</gruppeposisjonY>");
274     fi1.println(indenter + "<gruppebredde>" + hentElementBredde() + "</gruppebredde>");
275     fi1.println(indenter + "<gruppehoyde>" + hentElementHoyde() + "</gruppehoyde>");
276
277     if (relasjonerGruppe != null) {
278         for (int i1 = 0; i1 < relasjonerGruppe.size(); i1++) {
279             ((Relasjon) relasjonerGruppe.elementAt(i1)).toString(fi1, indenter, this);
280         }
281     }
282
283     for (int i2 = 0; i2 < elementer.size(); i2++) {
284         ((TavleElement) elementer.elementAt(i2)).toString(fi1, indenter);
285     }
286     fi1.println(i + "</gruppe>");
287 }
288
289 public void toStringTXT(PrintWriter fi1, String i)
290 {
291     Vector elementer = hentTavleElementer();
292
293     String indenter = i + "\t";
294     fi1.println(indenter + "Gruppenavn_" + hentElementNavn());
295     fi1.println(indenter + "Kommentar_" + hentElementKommentar() + "\n");
296     fi1.println(indenter + "Gruppen_inneholder_øfgende_elementer:");
297     for (int i2 = 0; i2 < elementer.size(); i2++) {
298         ((TavleElement) elementer.elementAt(i2)).toStringTXT(fi1, indenter);
299     }
300     fi1.println(indenter + "\n");
301 }

```

```

302
303 public Vector erElementiOmrade(int startx , int starty , int endx, int endy, Vector v, Posisjon p, Kontroller k)
304 {
305     int x = p.hentXPos();
306     int y = p.hentYPos();
307     int bredde = this.hentElementBredde();
308     int hoyde = this.hentElementHoyde();
309
310     if (x > startx && (x+bredde) < endx && y > starty && (y+hoyde) < endy) {
311         v.addElement(this);
312     } else if (!this.tavleElementerGruppe.isEmpty()) {
313         for (int i=0; i<this.tavleElementerGruppe.size(); i++) {
314             TavleElement t = (TavleElement)tavleElementerGruppe.elementAt(i);
315             Posisjon pos = k.finnPosisjonGruppe0(t);
316             v = t.erElementiOmrade(startx, starty, endx, endy, v, pos, k);
317         }
318     }
319     if (v==null) {
320         return v=new Vector();
321     } else {
322         return v;
323     }
324 }
325
326 public TavleElement klargjorAngre(Gruppe g, Tavle t) {
327     int x = this.hentElementPosisjonX();
328     int y = this.hentElementPosisjonY();
329     int bredde = this.hentElementBredde();
330     int hoyde = this.hentElementHoyde();
331     String navn = this.hentElementNavn();
332     String kommentar = this.hentElementKommentar();
333     int niva = this.hentElementNiva();
334     Gruppe ny_g = new Gruppe(navn,x,y,bredde,hoyde,niva,kommentar,g.t);
335     ny_g.settElementID(this.hentElementID()); // Benytter den samme id til elementet ifm angring
336     Vector nyeGruppeElementer = ny_g.hentTavleElementer();
337     Vector gamleGruppeElementer = this.hentTavleElementer();
338     for (int i=0; i<gamleGruppeElementer.size();i++){
339         TavleElement g_t = (TavleElement)gamleGruppeElementer.elementAt(i);
340         TavleElement ny_t = g.t.klargjorAngre(ny_g, t);
341         nyeGruppeElementer.addElement(ny_t);
342     }
343     return ny_g;
344 }
345
346 public void klargjorAngreRelasjoner (Gruppe fra, Gruppe til, Relasjon r)
347 {
348     TavleElement t;
349     Vector gruppeElementer = this.hentTavleElementer();
350     if (gruppeElementer!=null) {
351         for (int k=0;k<gruppeElementer.size();k++){
352             t = (TavleElement)gruppeElementer.elementAt(k);
353             if (t instanceof Gruppe){
354                 if (t.hentElementID()==fra.hentElementID() || t.hentElementID()==til.hentElementID()){
355                     ((Gruppe)t).leggTilRelasjon(r);
356                 } else {
357                     ((Gruppe)t).klargjorAngreRelasjoner(fra, til, r);
358                 }
359             }
360         }
361     }
362 }
363
364 public Vector kopierTavleElementer(Vector kopi, Kontroller k, Gruppe forelder){
365     int x = this.hentElementPosisjonX();
366     int y = this.hentElementPosisjonY();
367     int bredde = this.hentElementBredde();
368     int hoyde = this.hentElementHoyde();
369     String navn = this.hentElementNavn();
370     String kommentar = this.hentElementKommentar();
371     int niva = this.hentElementNiva();
372     Gruppe g = new Gruppe(navn,x,y,bredde,hoyde,niva,kommentar,forelder);
373     kopi.addElement(g);
374     Vector gruppeKopiElementer = g.hentTavleElementer();
375     Vector gruppeElementer = this.hentTavleElementer();
376     for (int i=0;i<gruppeElementer.size();i++){
377         TavleElement t = (TavleElement)gruppeElementer.elementAt(i);
378         gruppeKopiElementer = t.kopierTavleElementer(gruppeKopiElementer,k,g);
379     }
380
381     return kopi;
382 }
383
384 public void merkTavleElement(Graphics g, Kontroller k)
385 {
386     double z = k.hentZoomFaktor();
387
388     Posisjon pos = k.finnPosisjonGruppe0(this);
389     int x = Grafi.skeTavleElementer.doubleToInt(pos.hentXPos());
390     int y = Grafi.skeTavleElementer.doubleToInt(pos.hentYPos());
391     int hoyde = Grafi.skeTavleElementer.doubleToInt(this.hentElementHoyde());
392     int bredde = Grafi.skeTavleElementer.doubleToInt(this.hentElementBredde());
393     int hjorneMerking = Grafi.skeTavleElementer.doubleToInt(3*z); // Gir ø strrelsen 5ap merking av post - it
394     if (hjorneMerking<2)hjorneMerking=2;
395     g.setColor(Color.black);
396     g.fillRect(x-hjorneMerking, y-hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
397     g.fillRect(x+bredde-hjorneMerking, y-hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
398     g.fillRect(x-hjorneMerking, y+hoyde-hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
399     g.fillRect(x+bredde-hjorneMerking, y+hoyde-hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
400     for (int i=0; i<tavleElementerGruppe.size();i++){

```

```

401     TavleElement t = (TavleElement)tavleElementeriGruppe.elementAt(i);
402     t.merkTavleElement(g,k);
403 }
404 }
405
406 public void settRelasjoner (Vector v){
407     relasjonerGruppe = v;
408 }
409
410 }

```

## F.1.4 IOTavle.java

Listing: IOTavle.java

```

1  package kjapp;
2
3  import java.io.*;
4  import javax.xml.parsers.*;
5  import org.xml.sax.*;
6  import org.xml.sax.helpers.*;
7
8  /**
9   * Title : IOTavle
10  * Description : Klassen benyttes for å å hndtere I/O. Kan skrive til .xml eller .txt , samt lese fra .xml filer
11  * Copyright : Copyright (c) 2002
12  * Company :
13  * @author
14  * @version 1.0
15  */
16
17  public class IOTavle {
18
19     private Tavle theTavle;
20     private Kontroller theKontroller;
21
22     private String file;
23     public IOTavle()
24     {
25
26     }
27
28     public void TavleTilXML(Tavle t, String filename)
29     {
30         boolean funker = true;
31         file = filename.toLowerCase();
32
33         FileOutputStream f;
34         theTavle = t;
35         try{
36             f = new FileOutputStream(file);
37             OutputStreamWriter osw = new OutputStreamWriter(f);
38             PrintWriter fl = new PrintWriter(osw);
39             if (file.endsWith(".xml")){
40                 theTavle.toString(fl);
41             }else if (file.endsWith(".txt")){
42                 theTavle.toStringTXT(fl);
43             }
44             fl.flush();
45             fl.close();
46         } catch (FileNotFoundException e){
47             System.out.println("Noe skår seg ut. returverdi 2");
48             System.out.println(" Feilbeskrivelse : " + e.getMessage());
49         }
50     }
51
52
53     public Tavle tavleFraXML(String f, Kontroller k) throws Exception
54     {
55         theKontroller = k;
56         file = f;
57         theTavle = new Tavle(theKontroller, f);
58         TavleHandler thandler = new TavleHandler(theTavle);
59         SAXParserFactory factory = SAXParserFactory.newInstance();
60         factory.setNamespaceAware(true);
61         SAXParser saxParser = factory.newSAXParser();
62         FileInputStream in = new FileInputStream(file);
63         InputSource is = new InputSource(in);
64         if (is.getEncoding() == null){
65             /** Dersom kodingen til input østrømmen er null , settes denne til ISO-8859-1
66              * defnert i theTavle. Dersom kodingen er forskjellig fra null , settes
67              * koding i tavlen til øinputstrømmens verdi.
68              */
69             is.setEncoding(theTavle.getEncoding());
70         }else{
71             theTavle.setEncoding(is.getEncoding());
72         }
73         saxParser.parse(is, thandler);
74         return theTavle;
75     }
76 }

```

## F.1.5 KjApp.java

## Listing: KjApp.java

```

1 package kjapp;
2
3 import javax.swing.*;
4
5 /**
6  *
7  * <p>Title: KjApp</p>
8  * <p>Description: Main klassen for KjApp. JFrame opprettes `anr` kontrolleren opprettes ...</p>
9  * @author `aHvard Alstad`
10 * @version 1.0
11 */
12 public class KjApp
13 {
14     public static Kontroller theKontroller ;
15     public static void main(String [] args) throws Exception
16     {
17         try {
18             // UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());//merker ingen forskjell
19             // UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");//merker ingen forskjell
20             // UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");//styg
21             UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
22         } catch (Exception e) { }
23         theKontroller = new Kontroller ();
24     }
25
26     public KjApp()
27     {
28     }
29 }

```

## F.1.6 KjAppDialog.java

## Listing: KjAppDialog.java

```

1 package kjapp;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 /**
8  *
9  * <p>Title: KjAppDialog</p>
10 * <p>Description: `aHndterer` alt av dialogbokser mot brukeren . Oppretter en instans av klassen med en av de mulige typene
11 * (endretavletema , endregruppenavn , endrepostitnavn eller yesno) . Avhengig av angitt type vises riktig dialogboks.
12 * Dialogen vises til brukeren ved`a`ø kje`r metoden showKjappDialog). Data hentes fra dialogen alt med hent ... metodene som
13 * gir retur avhengig av hvilken type dialog som vises </p>
14 * @author `aHvard Alstad`
15 * @version 1.0
16 */
17 public class KjAppDialog extends JDialog implements ActionListener, KeyListener {
18     public final static int ENDRETAVLETEMA = 1;
19     public final static int ENDREGRUPPENAVN = 2;
20     public final static int ENDREPOSTITNAVN = 3;
21     public final static int YESNO = 4;
22     private JButton okEndreTavle;
23     private JButton okEndreGruppeNavn;
24     private JButton okEndrePostitNavn;
25     private JButton cancelEndreTavle;
26     private JButton cancelEndreGruppeNavn;
27     private JButton cancelEndrePostitNavn;
28     private JButton okButton;
29     private JButton cancelButton;
30     private JRadioButton noytral ;
31     private JRadioButton positiv ;
32     private JRadioButton negativ ;
33     private ButtonGroup bg;
34     private int status = Postit .STATUS_NOYTRAL;
35     private static int pTeller = 0;
36     private static int gTeller = 0;
37
38     private int type;
39     private JTextField tema;
40     private JTextArea kommentar;
41     private String txtKommentar = "";
42     private boolean ok;
43
44     public KjAppDialog(BrukerGrensensnitt parent ,int t ,String navn,double zoomfaktor)
45     {
46         super(parent, true); //Lager en modal dialog
47         type = t;
48         this . setResizable ( false );
49         Container contentPane = this . getContentPane();
50         contentPane . setLayout ( new BorderLayout());
51         JPanel p1 = null;
52         JPanel p2 = null;
53         JPanel p3 = null;
54         JPanel p4 = null;
55         JScrollPane sp4 = null;
56         JLabel lblKommentar = null;
57         Panel buttonPanel;
58         String okButtonText = "";
59         String cancelButtonText = "";
60         setLocationRelativeTo ( parent );

```

```

61 Posisjon pos = parent .hentAktivStartPosisjon ();
62 int x = Grafi skcTavleElementer .doubleToInt(pos.xPos * zoomfaktor);
63 int y = Grafi skcTavleElementer .doubleToInt(pos.yPos * zoomfaktor);
64 if (x < 100) x = 100;
65 if (y < 40) y = 40;
66
67 pos .setPosisjon (pos.xPos+20,pos.yPos+20);
68 // setLocation (x - 100,y + 40);
69 setLocation (x - 100 + parent .getX(),y + 40 + parent .getY()); //HA1804 — retter feilen med dialogbokser som kommer frem 'ap galt sted ..
70
71 switch (type)
72 {
73   case 1: //Endre tavletema
74     this .setTitle ("Tavle");
75     if (navn.equalsIgnoreCase("")) navn = "Min_Tavle";
76     p1 = new JPanel(new GridLayout(2,1));
77     p1.add(new JLabel("Angi_tema:"));
78     tema = new JTextField (navn);
79     tema .setSelectionStart (0);
80     tema .setSelectionEnd (navn.length());
81     tema.addKeyListener (this);
82     p1.add(tema);
83     contentPane .add(p1,"Center");
84     okButtonText = "OK";
85     cancelButtonText = "Avbryt";
86     setSize (240,120);
87     break;
88   case 2: //Endre gruppenavn
89     this .setTitle ("Egenskaper_for_gruppe");
90     if (navn.equalsIgnoreCase("")){
91       gTeller = gTeller + 1;
92       navn = "Gruppe_" + gTeller ;
93     }
94     p1 = new JPanel(new GridLayout(0,1));
95
96     //legger til label og tekstboks
97     p2 = new JPanel(new GridLayout(0,1));
98     p2.add(new JLabel("Angi_Gruppe_tema:"));
99     p2.add(tema = new JTextField (navn));
100    tema .setSelectionStart (0);
101    tema .setSelectionEnd (navn.length());
102    tema.addKeyListener (this);
103    p1.add(p2);
104
105    //Legger til label for kommentarfelt
106    lblKommentar = new JLabel("Kommentar:");
107    p2.add(lblKommentar);
108
109    //legger til kommentarfelt
110    kommentar = new JTextArea(txtKommentar,3,30);
111    kommentar.setLineWrap(true);
112    kommentar.addKeyListener(this);
113    sp4 = new JScrollPane (kommentar);
114    p1.add(sp4);
115
116    contentPane .add(p1,"Center");
117    okButtonText = "OK";
118    cancelButtonText = "Avbryt";
119    setSize (240,190);
120    break;
121   case 3: //Endre postIT navn
122     this .setTitle ("Egenskaper_for_post_it");
123     if (navn.equalsIgnoreCase("")){
124       pTeller = pTeller + 1;
125       navn = "Post_it_" + pTeller ;
126     }
127     //lager midtre vindu i dialogboksen
128     p1 = new JPanel(new GridLayout(0,1));
129
130     //legger til label og tekstboks
131     p2 = new JPanel(new GridLayout(0,1));
132     p2.add(new JLabel("Angi_aid:"));
133     p2.add(tema = new JTextField (navn));
134     tema .setSelectionStart (0);
135     tema .setSelectionEnd (navn.length());
136     tema.addKeyListener (this);
137     p1.add(p2);
138
139     //legger til radioknapper
140     p3 = new JPanel(new GridLayout(0,3));
141     bg = new ButtonGroup();
142     JRadioButton b;
143     positiv = new JRadioButton("Positiv");
144     positiv .addActionListener (this);
145     bg.add( positiv );
146     p3.add( positiv );
147     noytral = new JRadioButton("øNytral");
148     noytral .addActionListener (this);
149     bg.add( noytral );
150     p3.add( noytral );
151     negativ = new JRadioButton("Negativ");
152     negativ .addActionListener (this);
153     bg.add( negativ );
154     p3.add( negativ );
155     p2.add(p3);
156
157     //Legger til label for kommentarfelt
158     lblKommentar = new JLabel("Kommentar:");
159     p2.add(lblKommentar);

```

```

160
161 //legger til kommentarfelt
162 kommentar = new JTextArea(txtKommentar,6,30);
163 kommentar.setLineWrap(true);
164 kommentar.addKeyListener(this);
165 sp4 = new JScrollPane(kommentar);
166 p1.add(sp4);
167
168 contentPane.add(p1,"Center");
169 okButtonText = "OK";
170 cancelButtonText = "Avbryt";
171 setSize(240,300);
172 break;
173 case YESNO:
174     this.setTitle("Lagre?");
175     p1 = new JPanel(new GridLayout(0,1));
176     p1.add(new JLabel("Gnsker du a lagre ofr du lukker?"));
177     tema = new JTextField("");
178     contentPane.add(p1,"Center");
179     okButtonText = "Ja";
180     cancelButtonText = "Nei";
181     setSize(240,80);
182 break;
183 default:
184     break;
185 }
186
187 buttonPanel = new JPanel();
188 okButton = addButton(buttonPanel, okButtonText);
189 okButton.addKeyListener(this);
190 cancelButton = addButton(buttonPanel, cancelButtonText);
191 cancelButton.addKeyListener(this);
192 contentPane.add(buttonPanel,"South");
193 }
194
195 private JButton addButton(Container c, String name)
196 {
197     JButton button = new JButton(name);
198     button.addActionListener(this);
199     c.add(button);
200     return button;
201 }
202
203
204 public boolean showKjAppDialog()
205 {
206     if (this.type==KjAppDialog.ENDREPOSTITNAVN){
207         kommentar.setText(txtKommentar);
208         settRadioKnapp();
209     }
210     show();
211     return ok;
212 }
213
214 private void settRadioKnapp()
215 {
216     switch (status){
217     case PostIt.STATUS_NOYTRAL:
218         noytral.setSelected(true);
219         break;
220     case PostIt.STATUS_POSITIV:
221         positiv.setSelected(true);
222         break;
223     case PostIt.STATUS_NEGATIV:
224         negativ.setSelected(true);
225         break;
226     }
227 }
228
229 public void actionPerformed(ActionEvent e)
230 {
231     Object source = e.getSource();
232     if (source == okButton){
233         ok = true;
234         setVisible(false);
235     }else if (source == cancelButton){
236         ok = false;
237         setVisible(false);
238     }else if (source == noytral){
239         status = PostIt.STATUS_NOYTRAL;
240         noytral.setSelected(true);
241     }else if (source == positiv){
242         status = PostIt.STATUS_POSITIV;
243         positiv.setSelected(true);
244     }else if (source == negativ){
245         status = PostIt.STATUS_NEGATIV;
246         negativ.setSelected(true);
247     }
248 }
249
250 public void keyPressed(KeyEvent e)
251 {
252     if (e.getSource()==okButton || e.getSource()==tema){
253         if (e.getKeyCode()==10){
254             ok = true;
255             setVisible(false);
256         }
257     }else if (e.getSource()==cancelButton){
258         if (e.getKeyCode()==10){

```

```

259         ok = false ;
260         setVisible ( false );
261     }
262     } else if (e.getSource() ==kommentar){
263         if ( e.getKeyCode()==9){
264             kommentar.transferFocus () ;
265         }
266     }
267 }
268
269 public void keyReleased(KeyEvent e)
270 {
271 }
272
273 public void keyTyped(KeyEvent e)
274 {
275 }
276
277 public String hentTema()
278 {
279     return tema.getText () ;
280 }
281
282 public int hentStatus ()
283 {
284     return status ;
285 }
286
287 public void settStatus (int s)
288 {
289     status = s ;
290 }
291
292 public void settKommentar(String s)
293 {
294     txtKommentar = s ;
295     kommentar.setText(txtKommentar);
296 }
297
298 public String hentKommentar()
299 {
300     return kommentar.getText();
301 }
302 }

```

## F.1.7 Kontrollerer.java

### Listing: Kontrollerer.java

```

1  package kjapp;
2
3  import java .awt. event. MouseEvent;
4  import java .awt. event. KeyEvent;
5  import javax. swing. JFileChooser;
6  import java . util . Vector;
7  import javax. swing. *;
8  import java .awt.*;
9
10 import java .awt. event .*;
11
12 // printing
13 import java .awt. print .*;
14
15
16
17 /**
18 */
19 public class Kontroller
20 {
21     private int aktivTavle ;
22     private boolean velgOmrade = false ; // Variabel som indikerer om brukeren ønsker å merket et eller flere tavleElementer
23     private boolean jobberLokalt = true ; // Variabel som indikerer om brukeren arbeider lokalt eller distribuert
24     private boolean toggleRelasjon = false ; // Variabel som indikerer om brukeren har trykket inn relasjonsknappen eller ikke
25     private Vector merkedeTavleElementer; // Inneholder alle tavleElementer som er merket
26     private Vector utklippTavle; // Inneholder alle tavleElementer som er kopiert/klippet ut.
27     private Tavle theTavle ;
28     private Tavle tavleForAngring;
29     private Tavle tavleForGjenopprett ;
30     private Brukergrensesnitt theBrukergrensesnitt ;
31     private IOTavle theIOTavle;
32     private boolean debug = true;
33     private TavleElement aktivTavleElement ; //Lagrer det tavle elementet som sist ble funnet av finnTavleElement
34     private Posisjon offsetAktivTavleElement ; //Lagrer hvor i aktivt tavle element brukeren har trykket .
35     private Gruppe aktivGruppe; //Lagrer det tavleElementet som sist ble funnet av finnGruppe
36     private Gruppe gammelForelder; //Lagrer forelder til tavleElement som flyttes
37     private Relasjon theRelasjon ;
38     private double zoomFaktor = 1;
39     private double zoomDelta = 0.1;
40     private final String TITTEL = "KjApp";
41     private String aktuellTittel = ""; //Denne skal inkludere Tittel og fi Innavn
42     private boolean fi lEndret =false;
43
44
45
46     private boolean resizeElement ;

```

```

47
48
49  /**
50  * @roseuid 3C57BD700051
51  */
52  public Kontroller ()
53  {
54      theBrukergrensennitt = new Brukergrensennitt (this);
55      theBrukergrensennitt . setTitle (TITTEL);
56      theBrukergrensennitt . setSize (800,700);
57      theBrukergrensennitt . setVisible (true);
58      theBrukergrensennitt . setIconImage(new ImageIcon("images/kja.gif").getImage());
59      int bredde = theBrukergrensennitt . getWidth();
60      int hoyde = theBrukergrensennitt . getHeight();
61      theIOtavle = new IOtavle();
62      merkedeTavleElementer = new Vector();
63  }
64
65  /**
66  * Denne metoden skal a
67  * — ppe en dialog boks som gir brukeren mulighet til a angi tema for tavle
68  * — opprette ny instans av Tavle
69  * — Tegne opp den nye tavla i Brukergrensennitt
70  * @roseuid 3C5693C80386
71  */
72  public void nyTavle()
73  {
74      /*TavleDialog t = new TavleDialog( theBrukergrensennitt );
75      String tema = t.showTavleDialog();*/
76      if ( theTavle != null ) lukkTavle();
77      KjAppDialog k = new KjAppDialog(theBrukergrensennitt, KjAppDialog.ENDRETAVLETEMA, "" , zoomFaktor);
78      if ( k.showKjAppDialog() ) {
79          String tema = k.hentTema();
80          theTavle = new Tavle(tema, theBrukergrensennitt . getWidth(), theBrukergrensennitt . getHeight(), this, "");
81          aktivGruppe = theTavle . hentGruppe0();
82          //Tegn tavle
83          theBrukergrensennitt . nyTavle(theTavle);
84
85          //Tegn menylinje
86          theBrukergrensennitt . settKomplettMenyLinje(true);
87          theBrukergrensennitt . tegnHovedMeny();
88          theBrukergrensennitt . tegnVerktoymeny();
89          settToggleRelasjon ( false );
90          settVelgOmrade( false );
91          settJobberLokalt ( true );
92          theBrukergrensennitt . validate ();
93      }
94  }
95
96  /**
97  * @roseuid 3C56982000D6
98  */
99  public void apneTavle()
100  {
101      boolean funker = false;
102
103      A //1. ppe dialog for innlesing av riktig fi lnavn
104      //2. Kall metode i IOtavle for a opprette tavle fra valgt fi l . Metoden an returnere en Tavle
105      JFileChooser chooser = new JFileChooser();
106      chooser . setFileFilter ( new XMLFilter());
107      if ( theTavle != null ) lukkTavle();
108      int returnVal = chooser . showOpenDialog(theBrukergrensennitt);
109      if ( returnVal == JFileChooser . APPROVE_OPTION ) {
110          String fi lename = chooser . getSelectedFile () . getPath ();
111          System.out . println ("You chose to open this file: " + chooser . getSelectedFile () . getPath ());
112          try {
113              theTavle = theIOtavle . tavleFraXML(fi lename, this);
114              funker = true;
115              aktuellTittel = TITTEL + " - " + fi lename;
116              theBrukergrensennitt . setTitle ( aktuellTittel );
117          } catch (Exception e) {
118              funker = false;
119              System.out . println ("Can't open " + fi lename);
120          }
121      }
122
123      if ( funker ) {
124          aktivGruppe = theTavle . hentGruppe0();
125          theBrukergrensennitt . nyTavle(theTavle);
126
127          //Tegn menylinje
128          theBrukergrensennitt . settKomplettMenyLinje(true);
129          theBrukergrensennitt . tegnHovedMeny();
130          theBrukergrensennitt . tegnVerktoymeny();
131          settToggleRelasjon ( false );
132          settJobberLokalt ( true );
133          settVelgOmrade( false );
134
135          // Oppdater skjerm — Oppdaterer hele JFrame ...
136          theBrukergrensennitt . validate ();
137      }
138  }
139
140  /**
141  * Tema for tavle hentes fra gruppe0.
142  * — endreTema skal a ppe en dialogboks og endre tema for gruppe0
143  */
144  public void endreTema()
145  {
146      tavleForAngring = this . klargjorAngring (tavleForAngring);

```

```

146     Gruppe g = theTavle .hentGruppe(0);
147     KjAppDialog k = new KjAppDialog(theBrukergrensensnitt, KjAppDialog.ENDRETAVLETEMA.g.hentElementNavn(), zoomFaktor);
148     if (k.showKjAppDialog()) {
149         String nyttNavn = k.hentTema();
150         if (nyttNavn != "") g.setElementNavn(nyttNavn);
151         theBrukergrensensnitt .tegnTavleElement(g);
152     }
153 }
154
155 public boolean lagreTavle ()
156 {
157     boolean ok = false ;
158     String fi INavn = theTavle .hentFilNavn();
159     if ( fi INavn .equalsIgnoreCase("")){
160         ok = this .lagreTavleSom();
161     }else {
162         theOTavle .TavleTiXML(theTavle, fi INavn);
163         ok = true;
164     }
165     theBrukergrensensnitt . setTitle ( aktuellTittel );
166     return ok;
167 }
168
169 /**
170  * Tavlen har ikke noe defnert fi lnavn , og brukeren 'am velge ett ved hjelp av
171  * en fi leChooser
172  */
173 public boolean lagreTavleSom()
174 {
175
176     JFileChooser chooser = new JFileChooser();
177     TXTFilter txtFilter ;
178     XMLFilter xmlFilter ;
179     chooser .addChoosableFileFilter ( txtFilter = new TXTFilter());
180     chooser .addChoosableFileFilter ( xmlFilter = new XMLFilter());
181     chooser .setFileFilter ( xmlFilter);
182     chooser .removeChoosableFileFilter (chooser .getAcceptAllFileFilter ());
183     boolean ok = false ;
184     int returnVal = chooser .showSaveDialog( theBrukergrensensnitt );
185     if (returnVal == JFileChooser.APPROVE_OPTION) {
186         String fi lename = chooser .getSelectedFile ( ) .getPath ();
187         if (chooser .getFileFilter ( ) == xmlFilter){
188             if (! fi lename .endsWith(".xml")) fi lename = fi lename + ".xml";
189             theTavle .setFilNavn ( fi lename);
190             aktuellTittel = TITTEL + "—" + fi lename;
191             theBrukergrensensnitt . setTitle ( aktuellTittel );
192         }else if (chooser .getFileFilter ( ) == txtFilter) {
193             if (! fi lename .endsWith(".txt")) fi lename = fi lename + ".txt";
194             // setter ikke fi lnavn 'ap tavla 'anr denne lagres som .txt , fordi denne kan ikke leses inn igjen ...
195         }
196
197         theOTavle .TavleTiXML(theTavle, fi lename);
198         ok = true;
199     }
200     return ok;
201 }
202
203 /**
204  * Kontroller at tavlen er lagret
205  * Hvis ikke:
206  * o Spr brukeren om han/ nsker'a lagre
207  * Hvis ja:
208  * sjekk om fi lnavn er angitt
209  * hvis ja:
210  * lagre med angitt navn vha lagreTavle ()
211  * hvis nei:
212  * o kjr igang lagreTavleSom()
213  * Hvis brukeren ikke o nsker'a lagre tavle ,
214  * sett theTavle =null
215  * Fjern toolBar
216  * reduser menuBar
217  * @roseuid 3C57C14C0003
218  */
219 public void lukkTavle ()
220 {
221     KjAppDialog k = new KjAppDialog(theBrukergrensensnitt, KjAppDialog.YESNO,"", zoomFaktor);
222
223     if ( theTavle != null) {
224         if (k.showKjAppDialog()) {
225             boolean ok = lagreTavle ();
226
227             if ( ok) {
228                 theBrukergrensensnitt .settKomplettMenyLinje( false );
229                 theBrukergrensensnitt .tegnHovedMeny();
230                 theBrukergrensensnitt .setContentPane (new JPanel ());
231                 theTavle = null ;
232                 aktuellTittel =TITTEL;
233                 theBrukergrensensnitt . setTitle ( aktuellTittel );
234                 theBrukergrensensnitt . validate ();
235             }
236
237         } else {
238             theBrukergrensensnitt .settKomplettMenyLinje( false );
239             theBrukergrensensnitt .tegnHovedMeny();
240             theBrukergrensensnitt .setContentPane (new JPanel ());
241             theTavle = null ;
242             aktuellTittel =TITTEL;
243             theBrukergrensensnitt . setTitle ( aktuellTittel );
244             theBrukergrensensnitt . validate ();

```

```

245     }
246     utklippsTavle = new Vector();
247     merkedeTavleElementer = new Vector();
248   }
249 }
250
251 /**
252  * @roseuid 3C57C79300F4
253  */
254 public void merkAlt()
255 {
256     int minsteVertikal = 10000;
257     int størsteVertikal = -10000;
258     int minsteHorisontal = 10000;
259     int størsteHorisontal = -10000;
260     merkedeTavleElementer = new Vector();
261     //merkedeTavleElementer = fi.mTavleElementerInnenOmrade(0,0, theTavle.hentTavleBredde(), theTavle.hentTavleHoyde());
262     merkedeTavleElementer = fi.mTavleElementerInnenOmrade(-10000,-10000,10000,10000)/Elementer kan ligge i tavla selv om de ikke er synlige
263     innenfor JPanel - 10000 er valgt for å dekke hele skjermflaten og mere til
264     Grafi skTavleElementer grafi sk = theBrukergrensesnitt.hentGrafi skTavleElement();
265     Graphics g = grafi sk.getGraphics();
266     for (int i=0; i<merkedeTavleElementer.size();i++) {
267         TavleElement t = (TavleElement)merkedeTavleElementer.elementAt(i);
268         if (t.hentElementPosisjonX()<minsteHorisontal)minsteHorisontal=t.hentElementPosisjonX();
269         if ((t.hentElementPosisjonX()+t.hentElementBredde())>størsteHorisontal) størsteHorisontal=t.hentElementPosisjonX()+t.hentElementBredde();
270         if (t.hentElementPosisjonY()<minsteVertikal) minsteVertikal=t.hentElementPosisjonY();
271         if ((t.hentElementPosisjonY()+t.hentElementHoyde())>størsteVertikal) størsteVertikal=t.hentElementPosisjonY()+t.hentElementHoyde();
272
273         // grafi sk.merkTavleElement(t,(Graphics2D)g);
274         t.merkTavleElement((Graphics2D)g,this);
275     }
276     theBrukergrensesnitt.setMerkStartPosisjon(new Posisjon(minsteHorisontal - 20,minsteVertikal - 20));
277     theBrukergrensesnitt.setMerkSluttPosisjon(new Posisjon(størsteHorisontal + 20, størsteVertikal + 20));
278     this.setAktivTavleElement(theTavle.hentGruppe0());
279 }
280
281 /**
282  * @roseuid 3C57C7610357
283  */
284 public void merkTavleElement()
285 {
286     // Finn tavleElement
287     * Sett funnet tavleElement som aktivt element
288     * Tegn elementet med merking
289     */
290 }
291
292 public void angreSiste ()
293 {
294     //Metoden skal sette aktiv tavle (theTavle) til forrige aktive tavle
295     * (angreTavle). Forrige aktive tavle settes lik null
296     */
297     theBrukergrensesnitt.setAngre(false);
298     theBrukergrensesnitt.setGjenopprett(true);
299     if (tavleForAngring!=null) {
300         tavleForGjenopprett = new Tavle(this,"");
301         tavleForGjenopprett = this.klargjorAngre(tavleForGjenopprett);
302         theTavle = tavleForAngring;
303         tavleForAngring = null;
304         settAktivTavleElement(theTavle.hentGruppe0()); //HA09
305     }
306     theBrukergrensesnitt.tegnTavle(); //Kjrer repaint `ap theGrafi skTavleElementer
307 }
308
309 public void gjenopprettSiste ()
310 {
311     theBrukergrensesnitt.setAngre(true);
312     theBrukergrensesnitt.setGjenopprett(false);
313
314     if (tavleForGjenopprett !=null) {
315         tavleForAngring = this.klargjorAngre(tavleForAngring);
316         theTavle = tavleForGjenopprett;
317         tavleForGjenopprett = null;
318         settAktivTavleElement(theTavle.hentGruppe0()); //HA09
319     }
320     theBrukergrensesnitt.tegnTavle(); //Kjrer repaint `ap theGrafi skTavleElementer
321 }
322
323 public void kopierTavleElementer ()
324 {
325     utklippsTavle = theTavle.kopierTavleElementer(this.hentMerkedeTavleElementer());
326     this.setAktivGruppe(theTavle.hentGruppe0());
327     theBrukergrensesnitt.setAktivStartPosisjon(new Posisjon(10,10)); //Setter default plassering i tilfelle brukeren ikke angir en posisjon ved`a
328     klikke i tavla
329 }
330
331 public void klippUkTavleElement()
332 {
333     tavleForAngring = this.klargjorAngre(tavleForAngring);
334     /**
335      * Merkede elementer skal legges `ap utklippstavlen og fjernes
336      * fra tavlen. Kopierer ` frste vector merkedeTavleElementer til
337      * vector utklippsTavle ;
338      */
339     utklippsTavle = theTavle.kopierTavleElementer(this.hentMerkedeTavleElementer());
340     theTavle.klippUkTavleElementer(this.hentMerkedeTavleElementer());
341     this.setMerkedeTavleElementer(null);

```

```

342     this .settAktivGruppe (theTavle .hentGruppe0());
343     theBrukergrensesnitt . settAktivStartPosisjon (new Posisjon(10,10)); // Setter default plassering i tilfelle brukeren ikke angir en posisjon ved
344         a klikke i tavla
345     theBrukergrensesnitt . tegnTavle ();
346 }
347
348 public void limInnTavleElement()
349 {
350     /** Limer inn innholdet 'ap utklippstavlen i aktiv gruppe
351     *
352     */
353     tavleForAngring = this .klargjorAngre (tavleForAngring);
354     //Gruppe aktiv = this .hentAktivGruppe(); //HA
355     Gruppe aktiv = null;
356     if (aktivTavleElement instanceof Gruppe){
357         aktiv = (Gruppe)aktivTavleElement;
358     } else {
359         aktiv = aktivTavleElement . hentForelder ();
360     }
361
362     Posisjon posStart = theBrukergrensesnitt . hentAktivStartPosisjon ();
363     int x = posStart .hentXPos();
364     int y = posStart .hentYPos();
365     int forsteX=0;
366     int forsteY=0;
367     Vector elementerAktiv = aktiv .hentTavleElementer ();
368     Vector utklipp = this .hentUtklippstavle ();
369     Posisjon elementPos = null;
370     Posisjon forelderPos = fi nnPosisjonGruppe0 (aktiv);
371     if ( utklipp != null){
372
373         for (int i=0; i < utklipp.size (); i++){
374             // Lim inn elementene fra utklippstavlen
375             TavleElement t = (TavleElement)utklipp .elementAt(i);
376             elementPos = fi nnPosisjonGruppe0 (t);
377             if (i==0){
378                 /** Dersom dette er ø frste tavleElement som skal limes inn
379                 * a s settes elementPosisjonen til aktiv X og Y pos.
380                 * Ellers beregnes innsettingsposisjonen i forhold til
381                 * forrige elements posisjon.
382                 */
383                 forsteX = elementPos .xPos;
384                 forsteY = elementPos .yPos;
385                 t .settElementPosisjonX (x - forelderPos .xPos);
386                 t .settElementPosisjonY (y - forelderPos .yPos);
387                 t .settForelder (aktiv);
388                 sjekkElementUtstrekning (t, aktiv); //HA09
389             } else {
390                 // int diffX = t .hentElementPosisjonX() -forsteX;
391                 // int diffY = t .hentElementPosisjonY() -forsteY;
392                 int diffX = elementPos .xPos - forsteX;
393                 int diffY = elementPos .yPos - forsteY;
394                 t .settElementPosisjonX (x+diffX - forelderPos .xPos);
395                 t .settElementPosisjonY (y+diffY - forelderPos .yPos);
396                 t .settForelder (aktiv);
397                 sjekkElementUtstrekning (t, aktiv); //HA09
398             }
399             elementerAktiv .addElement(t);
400
401             if (t instanceof Gruppe){
402                 Vector gruppeElementer = t .hentTavleElementer ();
403                 for (int n=0; n < gruppeElementer.size(); n++){
404                     TavleElement tElement = (TavleElement)gruppeElementer .elementAt(n);
405                     this .tegnElement(tElement);
406                 }
407             }
408             theBrukergrensesnitt .tegnTavleElement (t);
409         }
410         Vector nyKopi = new Vector ();
411         nyKopi = theTavle .kopierTavleElementer (utklipp);
412         this .settUtklippstavle (nyKopi);
413     }
414 }
415
416
417 public void tegnElement(TavleElement t){
418     theBrukergrensesnitt .tegnTavleElement (t);
419     if (t instanceof Gruppe){
420         Vector gruppeElementer = t .hentTavleElementer ();
421         for (int i=0; i < gruppeElementer.size(); i++){
422             TavleElement element = (TavleElement)gruppeElementer .elementAt(i);
423             this .tegnElement(element);
424         }
425     }
426 }
427
428 public void nyPostIt ()
429 {
430     /**
431     * aNr brukeren skal opprette en ny post - it kan dette skje ved:
432     * 1 - Brukeren ø hyreklikker i en gruppe og velger ny post - it fra popupmeny
433     * 2 - Bruker velger ny post - it fra menylinjen
434     * 3 - Brukeren velger ny post - it fra ø verktøylinjen
435     * pkt 1 ø frer til at post - it skal plasseres 'ap samme sted som popupmeny er tegnet opp
436     * - hentAktivXPos og hentAktivYPos i Brukergrensesnitt
437     * - Disse settes 'anr brukeren ø hyreklikker et sted i tavlen
438     * - Den nye post - it skal plasseres i aktiv gruppe som 'aogs
439     * settes 'anr brukeren 'a pner popup meny (vha fi nnTavleElement i kontroller )

```

```

440     * pkt 2 og 3 ø frer til at post—it skal plasseres 'ap default plassering
441     *
442     * PS: Dersom angiPos=true er aktivTavleElement alltid en Gruppe
443     */
444     tavleForAngring = this.klargjorAngre(tavleForAngring);
445
446     int posX = 0;
447     int posY = 0;
448     Gruppe gForelder = theTavle.hentGruppe0();
449
450     Posisjon posStart = theBrukergrensennitt.hentAktivStartPosisjon();
451     posX = posStart.hentXPos();
452     posY = posStart.hentYPos();
453     theBrukergrensennitt.settAktivStartPosisjon(new Posisjon(posX + 20, posY + 20));
454
455     if (aktivTavleElement != null) {
456         if (aktivTavleElement instanceof PostIt) {
457             gForelder = aktivTavleElement.hentForelder();
458         } else {
459             gForelder = (Gruppe)aktivTavleElement;
460         }
461
462         //a M regne om aktivXPos til relativ posisjon i gruppen
463         Posisjon pos = finnPosisjonIGruppe0(gForelder);
464         int x0 = pos.xPos;
465         int y0 = pos.yPos;
466         posX = posX - x0;
467         posY = posY - y0;
468     }
469
470     KjAppDialog k = new KjAppDialog(theBrukergrensennitt, KjAppDialog.ENDREPOSTITNAVN, "", zoomFaktor);
471     if (k.showKjAppDialog()) {
472         String tema = k.hentTema();
473         int status = k.hentStatus();
474         String kommentar = k.hentKommentar();
475
476         PostIt p = new PostIt(tema, posX, posY, 80, 100, 0, kommentar, gForelder);
477         p.settPostItType(status);
478         gForelder.leggTilPostIt(p);
479
480         //Dersom ny Post—it kommer utenfor gruppen, 'am gruppen utvides slik at hele post—it'en kommer innenfor ...
481         sjekkElementUtstrekning(p, gForelder);
482
483         // Setter den nye post—it som aktivTavleElement
484         settAktivTavleElement(p); //HA09
485     }
486 }
487
488 public void nyGruppe()
489 {
490     /**
491     * 'a Nr brukeren skal opprette en ny gruppe kan dette skje ved:
492     * 1 — Brukeren ø hyreklikker i en gruppe og velger ny gruppe fra popupmeny
493     * 2 — Bruker velger ny gruppe fra menylinjen
494     * 3 — Brukeren velger ny grupe fra ø verktøylinjen
495     * pkt 1 ø frer til at gruppe skal plasseres 'ap samme sted som popupmeny er tegnet opp
496     * — hentAktivXPos og hentAktivYPos i Brukergrensennitt
497     * — Disse settes 'a nr brukeren ø hyreklikker et sted i tavlen
498     * — Den nye gruppen skal plasseres i aktiv gruppe som 'a ogs
499     * settes 'a nr brukeren 'a pner popup meny (vha finnTavleElement i kontrollen)
500     * pkt 2 og 3 ø frer til at gruppen skal plasseres 'ap default plassering
501     *
502     * PS: Dersom angiPos=true er aktivTavleElement alltid en Gruppe
503     */
504     tavleForAngring = this.klargjorAngre(tavleForAngring);
505     Gruppe gForelder = theTavle.hentGruppe0();
506     String kommentar = "";
507     if (merkedeTavleElementer.size() == 0 || theBrukergrensennitt.hentMerkSluttPosisjon() == null) {
508         int posX = 0;
509         int posY = 0;
510
511         Posisjon posStart = theBrukergrensennitt.hentAktivStartPosisjon();
512         posX = posStart.hentXPos();
513         posY = posStart.hentYPos();
514
515         if (aktivTavleElement != null) {
516             if (aktivTavleElement instanceof PostIt) {
517                 gForelder = aktivTavleElement.hentForelder();
518             } else {
519                 gForelder = (Gruppe)aktivTavleElement;
520             }
521         }
522
523         //a M regne om aktivXPos til relativ posisjon i gruppen
524         Posisjon pos = finnPosisjonIGruppe0(gForelder);
525         int x0 = pos.xPos;
526         int y0 = pos.yPos;
527         posX = posX - x0;
528         posY = posY - y0;
529     }
530
531     KjAppDialog k = new KjAppDialog(theBrukergrensennitt, KjAppDialog.ENDREGRUPPENAVN, "", zoomFaktor);
532
533     if (k.showKjAppDialog()) {
534         String tema = k.hentTema();
535         kommentar = k.hentKommentar();
536
537         Gruppe g = new Gruppe(tema, posX, posY, 300, 200, 0, kommentar, gForelder, theTavle);
538         settAktivTavleElement(g); //HA09

```

```

539
540 //Dersom ny Post—ii kommer utenfor gruppen, "am gruppen utvides slik at hele gruppen kommer innenfor ...
541 gForelder.leggTilUnderGruppe(g);
542 sjekkElementUtrekning(g,gForelder);
543 }
544
545 }else{//end if merkedeTavleElementer == 0
546 int posX = 0;
547 int posY = 0;
548 TavleElement t1 = (TavleElement)merkedeTavleElementer.elementAt(0);
549 gForelder=t1.hentForelder();
550
551 Posisjon posStart = theBrukergrensensnitt.hentMerkStartPosisjon();
552 posX = posStart.hentXPos();
553 posY = posStart.hentYPos();
554 Posisjon posSlutt = theBrukergrensensnitt.hentMerkSluttPosisjon();
555 Posisjon posForelder = fi nnPosisjonGruppe0(gForelder);
556
557 KjAppDialog k = new KjAppDialog(theBrukergrensensnitt,KjAppDialog.ENDREGRUPPENAVN,"",zoomFaktor);
558
559 if(k.showKjAppDialog()){
560 String tema = k.hentTema();
561 kommentar = k.hentKommentar();
562
563 Gruppe g = new Gruppe(tema,posX—posForelder.xPos,posY—posForelder.yPos,posSlutt.xPos—posX,posSlutt.yPos—posY,0,kommentar,gForelder,
564 theTavle);
565 settAktivTavleElement(g);//HA09
566 Vector v = g.hentTavleElementer();
567 TavleElement t;
568 Posisjon pos_g = fi nnPosisjonGruppe0(g);
569 for (int i = 0; i < merkedeTavleElementer.size();i++){
570 t = (TavleElement)merkedeTavleElementer.elementAt(i);
571 Posisjon t_pos = fi nnPosisjonGruppe0(t);
572 t.hentForelder().fjernTavleElement(t);
573 t.setForelder(g);
574 //am oppdatere posisjoner i forhold til ny forelder ...
575 t.setElementPosisjonX(t_pos.xPos—pos_g.xPos);
576 t.setElementPosisjonY(t_pos.yPos—pos_g.yPos);
577 v.addElement(t);
578 }
579 sjekkElementUtrekning(g,gForelder);
580 gForelder.leggTilUnderGruppe(g);
581 }
582 }
583
584 public void endreGruppeNavn()
585 {
586 tavleForAngring = this.klargjorAngre(tavleForAngring);
587 KjAppDialog k = new KjAppDialog(theBrukergrensensnitt,KjAppDialog.ENDREGRUPPENAVN,aktivTavleElement.hentElementNavn(),zoomFaktor);
588 k.setKommentar(aktivTavleElement.hentElementKommentar());
589 if(k.showKjAppDialog()){
590 String nyttNavn = k.hentTema();
591 String kommentar = k.hentKommentar();
592 if (nyttNavn!="") aktivTavleElement.setElementNavn(nyttNavn);
593 aktivTavleElement.setElementKommentar(kommentar);
594 theBrukergrensensnitt.tegnTavleElement(aktivTavleElement);
595 }
596 }
597
598 public void slettGruppe()
599 {
600 tavleForAngring = this.klargjorAngre(tavleForAngring);
601 if (aktivTavleElement!=theTavle.hentGruppe0()){
602 //Fjerner alle relasjoner som er tilknyttet denne gruppen
603 Vector vRelasjoner = ((Gruppe)aktivTavleElement).hentRelasjoner();
604 if (vRelasjoner!=null){
605 System.out.println ("Antall relasjoner tilknyttet aktivTavleElement:" + vRelasjoner.size());
606 while(vRelasjoner.size() > 0){
607 int teller = 0; //Skal fjerne alle relasjonene i gruppen, tar alltid den øfrste som ligger i vektoren
608 this.slettRelasjon((Relasjon)vRelasjoner.elementAt(teller));
609 System.out.println ("ØForsker å slette relasjon nummer " + teller);
610 }
611 }
612
613 Gruppe forelder = aktivTavleElement.hentForelder();
614 Vector vForelder = forelder.hentTavleElementer();
615 for (int i = 0; i < vForelder.size(); i++){
616 if (vForelder.elementAt(i)==aktivTavleElement){
617 Gruppe g = (Gruppe)vForelder.elementAt(i);//== aktivTavleElement ...
618 Vector v = g.hentTavleElementer();
619 //Flytter over alle elementer i gruppa som slettes til foreldre gruppa.
620 for (int i2 = 0; i2 < v.size(); i2++){
621 TavleElement e = (TavleElement)v.elementAt(i2);
622 //aM oppdatere elementet anr det flyttes med ny forelder og ny relativ posisjon
623 e.setForelder(forelder);
624 e.setElementPosisjonX(e.hentElementPosisjonX()+g.hentElementPosisjonX()); //Setter riktig pos i forhold til forelder
625 e.setElementPosisjonY(e.hentElementPosisjonY()+g.hentElementPosisjonY());
626 vForelder.addElement(v.elementAt(i2));
627 }
628 vForelder.removeElementAt(i);
629 theBrukergrensensnitt.tegnTavleElement(forelder);
630 break;//Muligens noe stygg denne, men det er ikke behov for a a g lenger ...
631 }
632 }
633 }
634 }
635
636 public void slettPostIt()

```

```

637     {
638         tavleForAngring = this.klargjorAngre(tavleForAngring);
639
640         Gruppe forelder = aktivTavleElement.hentForelder();
641         Vector vForelder = forelder.hentTavleElementer();
642
643         if (aktivTavleElement != theTavle.hentGruppe0()) {
644             for (int i = 0; i < vForelder.size(); i++) {
645                 if (vForelder.elementAt(i) == aktivTavleElement) {
646                     vForelder.removeElementAt(i);
647                     break; // Muligens noe stygg denne, men det er ikke behov for 'a' 'a' g lenger ....
648                 } //if
649             } //for
650             theBrukergrensensnitt.tegnTavleElement(aktivTavleElement);
651         } //if
652     }
653
654
655     /**
656     * Hver gang et TavleElement er flyttet 'am' ø gruppetilhorigheten kontrolleres .
657     * — Dersom deler av elementet fjernes er innenfor forelder, skal forelder utvides.
658     * — Dersom hele elementet er utenfor forelder skal elementet gis ny forelder og ny posisjon relativ til ny forelder
659     */
660     public void sjekkGruppetilhorighet()
661     {
662         Gruppe gruppe0 = theTavle.hentGruppe0();
663         if (aktivTavleElement != gruppe0) {
664             Gruppe forelder = (Gruppe)aktivTavleElement.hentForelder();
665             int xLeft = aktivTavleElement.hentElementPosisjonX();
666             int xRight = aktivTavleElement.hentElementPosisjonX() + aktivTavleElement.hentElementBredde();
667             int yTop = aktivTavleElement.hentElementPosisjonY();
668             int yBottom = aktivTavleElement.hentElementPosisjonY() + aktivTavleElement.hentElementHoyde();
669             // Dersom aktivGruppe (som settes i det musknappen slippes) ikke er forelder til aktivTavleElement fra øfr og
670             // dersom aktivGruppe ikke er aktivTavleElement 'am forelder til aktivTavleElement oppdateres
671             if ((forelder != aktivGruppe) && (aktivTavleElement != aktivGruppe)) {
672                 // Elementet ligger utenfor forelder, og ø gruppetilhorighet 'am endres ...
673                 // — Ny forelder vil da være aktivGruppe, som blir satt i det brukeren slipper musknappen.
674                 // — 'aNr ny forelder settes, 'am koordinatene for tavleelementet omregnes til 'a stemme med ny forelder.
675                 // — Dette øgjes ved ø frst 'a omregne til gruppe0 sine koordinater for 'as 'a regne om til ny forelder sine koordinater.
676                 //
677                 Gruppe tempGruppe = forelder;
678
679                 // Finner ø frst hvor aktivTavleElement ligger i forhold til gruppe0
680                 Posisjon posAktivT = fi.nnPosisjonGruppe0(aktivTavleElement);
681
682                 // Finner deretter hvor aktivGruppe ligger i forhold til gruppe0
683                 Posisjon posAktivG = fi.nnPosisjonGruppe0(aktivGruppe);
684
685                 // Setter ny posisjon for aktivTavleElement
686                 aktivTavleElement.settElementPosisjonX(posAktivT.xPos — posAktivG.xPos);
687                 aktivTavleElement.settElementPosisjonY(posAktivT.yPos — posAktivG.yPos);
688
689                 // Finn og fjern aktivTavleElement fra forelder
690                 forelder.fjernTavleElement(aktivTavleElement);
691
692                 // Sett ny Forelder i aktivTavleElement
693                 aktivTavleElement.settForelder(aktivGruppe);
694
695                 // Legg til aktivTavleElement i aktivGruppe
696                 aktivGruppe.leggTilTavleElement(aktivTavleElement);
697             } // If aktivGruppe != forelder
698
699             // 'aS 'am vi kontrollere at det er plass til det nye elementet i sin nye forelder
700             // Dette 'am øgjes uansett om tavleElementet har 'a fit ny forelder eller ikke ...
701             // Regner med absolutte posisjoner
702
703             // sjekkElementUtstrekning (aktivTavleElement, aktivTavleElement.hentForelder());
704             theBrukergrensensnitt.tegnTavle();
705
706         } // If (aktivTavleElement != gruppe0)
707     } // metoden
708
709
710     /**
711     * Metode som kontrollerer at hele det flyttede elementet fremdeles er innenfor utstrekningen til foreldrelementet
712     * Kalles rekursivt helt til forelder er gruppe0
713     * @param t — TavleElementet som er flyttet
714     * @param forelder — forelder til tavleElementet som er flyttet
715     */
716     public void sjekkElementUtstrekning(TavleElement t, Gruppe forelder) {
717         Gruppe gruppe0 = theTavle.hentGruppe0();
718         if (forelder != gruppe0) {
719             Gruppe forelderTilForelder = forelder.hentForelder();
720             // Henter utstrekning for tavleElementet
721             Posisjon posAT = fi.nnPosisjonGruppe0(t);
722             int breddeAT = t.hentElementBredde();
723             int hoydeAT = t.hentElementHoyde();
724             // Henter utstrekning for Forelder til tavleElementet
725
726             Posisjon posForelder = new Posisjon();
727
728             posForelder = fi.nnPosisjonGruppe0(forelder);
729
730             int breddeForelder = forelder.hentElementBredde();
731             int hoydeForelder = forelder.hentElementHoyde();
732             Vector vForelder = forelder.hentTavleElementer();
733             TavleElement te;
734
735             if (posAT.xPos < posForelder.xPos) {

```

```

736 // AktivTavleElement ligger utenfor forelder til venstre
737 // Flytt forelder mot venstre og k forelder bredde tilsvarende
738 // Oppdater posisjon for samtlige elementer i forelder
739 int dx = posForelder.xPos - posAT.xPos;
740 forelder.setElementPosisjonX( forelder.hentElementPosisjonX() - dx);
741 forelder.setElementBredde( breddeForelder + dx);
742 for ( int i = 0; i < vForelder.size(); i++) {
743     te = (TavleElement)vForelder.elementAt(i);
744     te.setElementPosisjonX( te.hentElementPosisjonX() + dx);
745 }
746 }
747 if ( posAT.xPos + breddeAT > posForelder.xPos + breddeForelder ) {
748     // AktivTavleElement ligger utenfor forelder til høyre
749     // Unvid forelder mot høyre ved å utvide bredden i forelder
750     // Dersom forelder ligger i en annen gruppe! = grupe0 am aogs undergrupper kontrolleres ....
751     forelder.setElementBredde( forelder.hentElementBredde() + ( posAT.xPos + breddeAT ) - ( posForelder.xPos + breddeForelder ) );
752 }
753 if ( posAT.yPos < posForelder.yPos ) {
754     // AktivTavleElement ligger utenfor forelder - ovenfor
755     // Flytt forelder oppover og k ø forelderhøyde tilsvarende
756     // Oppdater posisjon for samtlige elementer i forelder tilsvarende
757     int dY = posForelder.yPos - posAT.yPos;
758     forelder.setElementPosisjonY( forelder.hentElementPosisjonY() - dY);
759     forelder.setElementHoyde( hoydeForelder + dY);
760     for ( int i = 0; i < vForelder.size(); i++) {
761         te = (TavleElement)vForelder.elementAt(i);
762         te.setElementPosisjonY( te.hentElementPosisjonY() + dY);
763     }
764 }
765 if ( posAT.yPos + hoydeAT > posForelder.yPos + hoydeForelder ) {
766     // AktivTavleElement ligger utenfor forelder - under
767     // Unvid forelder nedover ved å ø ke øyden i forelder
768     // Dersom forelder ligger i en annen gruppe! = grupe0 am aogs undergrupper kontrolleres ....
769     forelder.setElementHoyde( forelder.hentElementHoyde() + ( posAT.yPos + hoydeAT ) - ( posForelder.yPos + hoydeForelder ) );
770 }
771 if ( forelderTilForelder != grupe0 ) {
772     sjekkElementUtrekning( forelder , forelderTilForelder );
773 }
774 }
775 theBrukergrensensnitt.tegnTavleElement( forelder );
776 }
777
778 public void nyRelasjon()
779 {
780     tavleForAngring = this.klargjorAngring( tavleForAngring );
781     theRelasjon = new Relasjon();
782 }
783
784 /**
785  * Oppdaterer modellen øfr tavla tegnes ap nytt.
786  * Metoden tar seg av to ting:
787  * Resizing av tavleElementer og Flytting av tavleElementer. TavleElementer kan kan resized
788  * nede i øhyre øhjrne ...
789  * Vet at elementet skal resized dersom resi:Element = true.
790  *
791  * Elementet er i bevegelse, og ø tilhrer dermed grupe0.
792  * - Setter grupe0 = forelder (am smatidig fjerne referansen fra til TavleElementet fra " gammel " forelder
793  * - Benytter absolutt posisjon (... i forhold til grupe0)
794  * - aNr musen slipes ( mouseDropped event ) økjres sjekk ø gruppeTilhrighet for a finne ny foreldregruppe
795  * @param posX - posisjonen hvor elementet skal tegnes opp
796  * @param posY - posisjonen hvor elementet skal tegnes opp
797  */
798 public void flyttTavleElement( int posX, int posY )
799 {
800     this.setMerkedeTavleElementer( null );
801     theBrukergrensensnitt.setTittel( aktueltTittel + "*" );
802     Gruppe g0 = theTavle.hentGruppe0();
803
804     if ( aktivTavleElement != g0 ) { // Det skal ikke øvre mulig a flytte grupe0
805         int bredde = aktivTavleElement.hentElementBredde();
806         int hoyde = aktivTavleElement.hentElementHoyde();
807
808         if ( toggleRelasjon ) {
809
810         } else if ( resizeElement ) {
811             Posisjon pos = finnPosisjonIGruppe0( aktivTavleElement );
812             if ( aktivTavleElement instanceof Gruppe ) {
813                 Posisjon posMin = finnYtterpunkter( ( Gruppe ) aktivTavleElement );
814                 if ( ( posMin.xPos + pos.xPos < posX ) && ( posX - pos.xPos > 30 ) ) aktivTavleElement.setElementBredde( posX - pos.xPos );
815                 if ( ( posMin.yPos + pos.yPos < posY ) && ( posY - pos.yPos > 40 ) ) aktivTavleElement.setElementHoyde( posY - pos.yPos );
816             } else {
817                 if ( posX - pos.xPos > 30 ) aktivTavleElement.setElementBredde( posX - pos.xPos );
818                 if ( posY - pos.yPos > 40 ) aktivTavleElement.setElementHoyde( posY - pos.yPos );
819             }
820         } else {
821             // Endrer forelder slik at jeg kan jobbe med absolutt posisjon
822             Gruppe g = aktivTavleElement.hentForelder();
823             g.fjernTavleElement( aktivTavleElement );
824             aktivTavleElement.setForelder( g );
825
826             if ( aktivTavleElement instanceof Gruppe ) {
827                 g0.leggTilUnderGruppe( ( Gruppe ) aktivTavleElement );
828             } else {
829                 g0.leggTilPostt( ( Postt ) aktivTavleElement );
830             }
831             if ( posX - offsetAktivTavleElement.xPos > 0 ) {
832                 aktivTavleElement.setElementPosisjonX( posX - offsetAktivTavleElement.xPos );
833             } else {
834                 aktivTavleElement.setElementPosisjonX( 0 );

```

```

835     }
836     if ( posY - offsetAktivTavleElement.yPos > 0){
837         aktivTavleElement . setElementPosisjonY(posY - offsetAktivTavleElement.yPos);
838     } else {
839         aktivTavleElement . setElementPosisjonY(0);
840     }
841     }
842     sjekkElementUtstrekning (aktivTavleElement , aktivTavleElement . hentForelder ());
843     theBrukergrensesnitt . tegnTavleElement( aktivTavleElement );
844     }
845     }
846
847     public void endrePostIt ()
848     {
849         tavleForAngring = this . klargjørAngring (tavleForAngring);
850         KjAppDialog k = new KjAppDialog(theBrukergrensesnitt , KjAppDialog.ENDREPOSTITNAVN,aktivTavleElement.hentElementNavn(),zoomFaktor);
851         k. setStatus ((( PostIt )aktivTavleElement ) . hentPostItType ());
852         k. settKommentar(aktivTavleElement . hentElementKommentar());
853         if (k. showKjAppDialog()){
854             String nyttNavn = k.hentTema();
855             int status = k.hentStatus ();
856             String kommentar = k.hentKommentar();
857             if (nyttNavn!="")aktivTavleElement . setElementNavn(nyttNavn);
858             (( PostIt )aktivTavleElement ) . setPostItType (status );
859             aktivTavleElement . settElementKommentar(kommentar);
860             theBrukergrensesnitt . tegnTavleElement ( aktivTavleElement );
861         }
862     }
863
864     public void fi nnTavleElement (int xPos, int yPos)
865     {
866         Gruppe g0 = theTavle . hentGruppe0();
867         TavleElement t = g0.fi nnTavleElement (xPos,yPos, this);
868         settAktivTavleElement (t);
869         Posisjon posG0 = fi nnPosisjonGruppe0(t);
870         setOffsetAktivTavleElement (new Posisjon(xPos - posG0.xPos,yPos - posG0.yPos));
871     }
872
873     /**
874     * Benytter fi nnTavleElement for å finne en Gruppe.
875     * Dersom fi nnTavleElement returnerer en PostIt settes
876     * Forelder til Post-it'en som aktivTavleElement
877     * @roseuid 3C627B9303CD
878     */
879     public void fi nnGruppe (int xPos, int yPos)
880     {
881         Gruppe g0 = theTavle . hentGruppe0();
882         Gruppe g = g0.fi nnGruppe(xPos,yPos, this , aktivTavleElement );
883         aktivGruppe = g;
884     }
885
886     /**
887     *
888     * @param element — TavleElementet man ønsker å finne den absolutte posisjonen til
889     * @return Posisjon — Objekt som inneholder både x og y verdi for posisjonen
890     */
891     public Posisjon fi nnPosisjonGruppe0 (TavleElement element)
892     {
893         Gruppe g0 = theTavle . hentGruppe0();
894         Posisjon pos = new Posisjon ();
895         if (element!=g0){
896             Gruppe g = element . hentForelder ();
897             pos. settPosisjon (element.hentElementPosisjonX(),element.hentElementPosisjonY());
898             while (g!=g0){
899                 pos.xPos = pos.xPos + g.hentElementPosisjonX();
900                 pos.yPos = pos.yPos + g.hentElementPosisjonY();
901                 g = g. hentForelder ();
902             }
903         }
904         return pos;
905     }
906
907     public Tavle hentTavle ()
908     {
909         return theTavle ;
910     }
911
912     public TavleElement hentAktivTavleElement ()
913     {
914         return aktivTavleElement ;
915     }
916
917     public void settAktivTavleElement (TavleElement t)
918     {
919         aktivTavleElement = t;
920         Gruppe g0 = theTavle . hentGruppe0();
921         if (aktivTavleElement instanceof PostIt){
922             theBrukergrensesnitt . oppdaterMenyForPostIt();
923         } else {
924             if (aktivTavleElement == g0){
925                 theBrukergrensesnitt . oppdaterMenyForTavle();
926             } else {
927                 theBrukergrensesnitt . oppdaterMenyForGruppe();
928             }
929         }
930     }
931
932     public void settResizeElement (boolean b)
933     {

```

```

934     resizeElement = b;
935 }
936
937 public boolean hentResizeElement()
938 {
939     return resizeElement;
940 }
941
942 public void settVelgOmrade(boolean b)
943 {
944     velgOmrade = b;
945     theBrukergrensensnitt . settPilButton (b);
946     if (b==true){
947         theBrukergrensensnitt . setCursor(Cursor.CROSSHAIR_CURSOR);
948     }else{
949         theBrukergrensensnitt . setCursor(Cursor.DEFAULT_CURSOR);
950     }
951 }
952
953 public boolean hentVelgOmrade()
954 {
955     return velgOmrade;
956 }
957
958 public void settJobberLokalt (boolean b)
959 {
960     jobberLokalt = b;
961     theBrukergrensensnitt . oppdaterMenyForDistribuert(b);
962 }
963
964 public boolean hentJobberLokalt ()
965 {
966     return jobberLokalt;
967 }
968
969 public void settToggleRelasjon (boolean b)
970 {
971     toggleRelasjon = b;
972     if (b){
973         nyRelasjon();
974     }else{
975         theRelasjon = null;
976     }
977     theBrukergrensensnitt . settRelasjonButton (b);
978 }
979
980 public boolean hentToggleRelasjon ()
981 {
982     return toggleRelasjon;
983 }
984
985 public void settRelasjon (Relasjon r)
986 {
987     theRelasjon = r;
988 }
989
990 public Relasjon hentRelasjon ()
991 {
992     return theRelasjon;
993 }
994
995 public void fullforRelasjon (int x,int y)
996 {
997     finnTavleElement(x,y);
998     Gruppe g0 = theTavle .hentGruppe0();
999     if ( aktivTavleElement!=g0 &&
1000         !(( aktivTavleElement instanceof PostIt) &&
1001             aktivTavleElement . hentForelder ()==g0) &&
1002         aktivTavleElement !=theRelasjon .hentFraGruppe() &&
1003         theRelasjon .hentFraGruppe()!=null){
1004         Gruppe fra;
1005         Gruppe til;
1006         if ( aktivTavleElement instanceof Gruppe){
1007             til = ( Gruppe)aktivTavleElement;
1008         } else {
1009             til = aktivTavleElement . hentForelder ();
1010         }
1011         if ( til !=g0){
1012             theRelasjon . settTilGruppe ( til );
1013             fra = theRelasjon .hentFraGruppe();
1014             fra . leggTilRelasjon ( theRelasjon );
1015             til . leggTilRelasjon ( theRelasjon );
1016             theTavle .nyRelasjon(theRelasjon);
1017         } else {
1018             theRelasjon = null;
1019         }
1020     }else {
1021         theRelasjon = null;
1022     }
1023     settToggleRelasjon ( false );
1024     theBrukergrensensnitt .tegnTavle ();
1025 }
1026
1027 public Vector finnTavleElementerInnenOmrade(int startx , int starty , int endx , int endy)
1028 {
1029     merkedeTavleElementer = new Vector();
1030     Gruppe g0 = theTavle .hentGruppe0();
1031     Vector alleTavleElementer = g0.hentTavleElementer ();
1032     if (! alleTavleElementer .isEmpty()){

```

```

1033     for(int i=0; i<alleTavleElementer.size(); i++){
1034         if(alleTavleElementer.elementAt(i) instanceof Gruppe){
1035             Gruppe g = (Gruppe)alleTavleElementer.elementAt(i);
1036             merkedeTavleElementer = g.erElementOmrade(startx, starty, endx, endy, merkedeTavleElementer, this, fi nnPosisjonGruppe0(g), this);
1037         } else if(alleTavleElementer.elementAt(i) instanceof PostIt){
1038             PostIt p = (PostIt)alleTavleElementer.elementAt(i);
1039             merkedeTavleElementer = p.erElementOmrade(startx, starty, endx, endy, merkedeTavleElementer, this, fi nnPosisjonGruppe0(p), this);
1040         }
1041     }
1042 }
1043 return merkedeTavleElementer;
1044 }
1045 /** Lagrer en kopi av theTavle i angreTavle.
1046  * Behandler øfrst alle grupper og postit i tavlen
1047  */
1048 public Tavle klargjorAngre(Tavle angreTavle) {
1049     theBrukergrensesnitt.setTitel("aktuellTitel"+"*");
1050     int bredde = theTavle.hentTavleBredde();
1051     int hoyde = theTavle.hentTavleHoyde();
1052     String fi lNavn = theTavle.hentFi lNavn();
1053     // Oppretter ny tavle som det skal øvre mulig'a angre til
1054     angreTavle = new Tavle("", bredde, hoyde, this, fi lNavn);
1055     Gruppe g0_angre = angreTavle.hentGruppe0();
1056     Gruppe g0_1javle = theTavle.hentGruppe0();
1057     g0_angre.setElementNavn(g0_1javle.hentElementNavn());
1058     g0_angre.setElementID(g0_1javle.hentElementID());
1059     Vector v_g0_angre = g0_angre.hentTavleElementer();
1060     // Henter frem gruppe0 og Vector tavleElementerGruppe0 fra tavlen
1061
1062     Vector v_g0_1javle = g0_1javle.hentTavleElementer();
1063     for(int i=0; i<v_g0_1javle.size(); i++){
1064         TavleElement t = (TavleElement)v_g0_1javle.elementAt(i);
1065         t = t.klargjorAngre(g0_angre, angreTavle);
1066         v_g0_angre.addElement(t);
1067     }
1068     theBrukergrensesnitt.setAngre(true);
1069     if ( theBrukergrensesnitt.hentGjenopprett() theBrukergrensesnitt.setGjenopprett(false); //HA09
1070
1071     // Behandler alle relasjonene i tavlen.
1072     this.klargjorAngreRelasjoner(angreTavle);
1073     return angreTavle;
1074 }
1075
1076 /**Finner alle relasjoner i tavlen og skal kopiere disse til angreTavle
1077  * øGjres ved'a hente ut alle gruppene i tavlen og 'ag gjennom
1078  * deres relasjoner. Relasjonen som finnes i gruppen
1079  * skal inn i angretavlens tilsvarende grupper.
1080  * Metoden fi nnGruppe(Tavle t, long id) leter gjennom gruppene
1081  * som ligger i angreTavlen til den finner en med riktig id. Deretter
1082  * legges relasjonene som finnes inn i gruppen som er funnet i angreTavle.
1083  * Metoden gir pr 14.03.02 kl 1400 NullPointerException
1084  * Espen øGjs
1085  *
1086  */
1087 public void klargjorAngreRelasjoner(Tavle angreTavle){
1088     Gruppe g0_1javle = theTavle.hentGruppe0();
1089     Gruppe g0_angre = angreTavle.hentGruppe0();
1090     Vector tavleRelasjoner = theTavle.hentRelasjoner();
1091     Vector angreRelasjoner = angreTavle.hentRelasjoner();
1092     Vector angreElementer = g0_angre.hentTavleElementer();
1093     Relasjon tavle_relasjon;
1094     Relasjon angre_relasjon;
1095     TavleElement t;
1096     if (tavleRelasjoner != null){
1097         for(int j=0; tavleRelasjoner.size()>j;j++){
1098             tavle_relasjon = (Relasjon)tavleRelasjoner.elementAt(j);
1099             Gruppe tavle_gfra = tavle_relasjon.hentFraGruppe();
1100             Gruppe tavle_gtil = tavle_relasjon.hentTilGruppe();
1101             angre_relasjon = new Relasjon();
1102             Gruppe angre_gfra = angreTavle.fi nnGruppe(tavle_gfra.hentElementID());
1103             Gruppe angre_gtil = angreTavle.fi nnGruppe(tavle_gtil.hentElementID());
1104
1105             angre_relasjon.settFraGruppe(angre_gfra);
1106             angre_relasjon.settTilGruppe(angre_gtil);
1107
1108             angre_gfra.leggTilRelasjon(angre_relasjon);
1109             angre_gtil.leggTilRelasjon(angre_relasjon);
1110
1111             angreRelasjoner.addElement(angre_relasjon);
1112         }
1113     }
1114 }
1115
1116 public Vector hentMerkedeTavleElementer(){
1117     Gruppe g0 = theTavle.hentGruppe0();
1118     if(merkedeTavleElementer==null){
1119         merkedeTavleElementer = new Vector();
1120         if (aktivTavleElement!=g0)merkedeTavleElementer.addElement(this.hentAktivTavleElement());
1121         return merkedeTavleElementer;
1122     } else{
1123         if (merkedeTavleElementer.size()==0){
1124             if (aktivTavleElement!=g0)merkedeTavleElementer.addElement(this.hentAktivTavleElement());
1125         }
1126         return merkedeTavleElementer;
1127     }
1128 }
1129
1130 public void settMerkedeTavleElementer(Vector v){
1131     if (v==null){

```

```

1132     merkedeTavleElementer = new Vector();
1133     } else {
1134     merkedeTavleElementer = v;
1135     }
1136     }
1137
1138     public Vector hentUtklippsTavle ()
1139     {
1140     return utklippsTavle ;
1141     }
1142
1143     public void settUtklippsTavle (Vector utklipp){
1144     utklippsTavle = utklipp;
1145     }
1146
1147     public Gruppe hentAktivGruppe()
1148     {
1149     return aktivGruppe;
1150     }
1151
1152     public void settAktivGruppe(Gruppe g){
1153     aktivGruppe = g;
1154     }
1155
1156     public void slettRelasjon (Relasjon r)
1157     {
1158     theTavle . slettRelasjon (r);
1159     theBrukergrensesnitt . tegnTavle ();
1160     }
1161
1162     public double hentZoomFaktor()
1163     {
1164     return zoomFaktor;
1165     }
1166
1167     public void settZoomFaktor(double f)
1168     {
1169     zoomFaktor = f;
1170     }
1171
1172     public void inkremitterZoomFaktor()
1173     {
1174
1175     if (zoomFaktor < 2) {
1176     zoomFaktor = zoomFaktor + zoomDelta;
1177     theBrukergrensesnitt . oppdaterScrollPane ();
1178     theBrukergrensesnitt . tegnTavle ();
1179     }
1180     //System.out.println("Zoomfaktor er: " + zoomFaktor);
1181     }
1182
1183     public void dekrementerZoomfaktor()
1184     {
1185     if (zoomFaktor > 0.5){
1186     zoomFaktor = zoomFaktor - zoomDelta;
1187     theBrukergrensesnitt . oppdaterScrollPane ();
1188     theBrukergrensesnitt . tegnTavle ();
1189     }
1190
1191     System.out.println("Zoomfaktor er: " + zoomFaktor);
1192     }
1193
1194     public void skrivUt()
1195     {
1196     Gruppe g0 = theTavle .hentGruppe0();
1197     Posisjon ytterPunkt = fi nnYtterpunkter (g0);
1198     try {
1199     theBrukergrensesnitt . skrivUt (ytterPunkt);
1200     } catch (PrinterException e) {
1201     System.out.println (e.getMessage());
1202     }
1203     }
1204
1205     public Posisjon fi nnYtterpunkter (Gruppe g)
1206     {
1207     //Gruppe g0 = theTavle .hentGruppe0();
1208     Posisjon storsteHitil = new Posisjon (0,0);
1209     Vector v = g .hentTavleElementer ();
1210     TavleElement t;
1211     for (int i=0; i < v.size(); i++){
1212     t = (TavleElement)v .elementAt(i);
1213     if ( ( t .hentElementPosisjonX() + t .hentElementBredde() > storsteHitil .xPos) storsteHitil .xPos = t .hentElementPosisjonX() + t .
1214     hentElementBredde());
1215     if ( ( t .hentElementPosisjonY() + t .hentElementHoyde() > storsteHitil .yPos) storsteHitil .yPos = t .hentElementPosisjonY() + t .
1216     hentElementHoyde());
1217     }
1218     return storsteHitil ;
1219     }
1220
1221     public void fi nnYtterpunkter (Vector v)
1222     {
1223     //Gruppe g0 = theTavle .hentGruppe0();
1224     Posisjon storsteHitil = new Posisjon (0,0);
1225     Posisjon minsteHitil = new Posisjon(10000,10000);
1226     TavleElement t;
1227     for (int i=0; i < v.size(); i++){
1228     t = (TavleElement)v .elementAt(i);

```

```

1228         if ( t.hentElementPosisjonX() + t.hentElementBredde() > størsteHitil .xPos) størsteHitil .xPos = t.hentElementPosisjonX() + t.
1229             hentElementBredde();
1230         if ( t.hentElementPosisjonY() + t.hentElementHoyde() > størsteHitil .yPos) størsteHitil .yPos = t.hentElementPosisjonY() + t.
1231             hentElementHoyde();
1232         if ( t.hentElementPosisjonX() < minsteHitil .xPos) minsteHitil .xPos = t.hentElementPosisjonX();
1233         if ( t.hentElementPosisjonY() < minsteHitil .yPos) minsteHitil .yPos = t.hentElementPosisjonY();
1234     }
1235     theBrukergrensesnitt . settMerkStartPosisjon ( minsteHitil );
1236     theBrukergrensesnitt . settMerkSluttPosisjon ( størsteHitil );
1237 }
1238
1239 public void settOffsetAktivTavleElement ( Posisjon p)
1240 {
1241     offsetAktivTavleElement = p;
1242 }
1243
1244 public Posisjon hentOffsetAktivTavleElement ()
1245 {
1246     return offsetAktivTavleElement ;
1247 }
1248
1249 public void sluttMerkede ()
1250 {
1251     this .klippUtTavleElement(); // Klipper ut merkede tavleElementer
1252     this . settUtklippTavle ( new Vector()); // Øtmer utklippstavlen med ny vektor
1253 }
1254
1255 public void avslutt ()
1256 {
1257     if ( jobberLokalt){
1258         this .lukkTavle();
1259         System .exit (0);
1260     }
1261 }

```

## F.1.8 Menyer.java

### Listing: Menyer.java

```

1 package kjapp;
2
3 import javax .swing .*;
4 import java .awt .*;
5 import java .awt .event .*;
6 import java .util . Vector;
7
8 /**
9  *
10  * <p>Title: Menyer</p>
11  * <p>Description: Klasse som benyttes for å tegne opp alle menyelementer i KjApp. Alle knapper
12  * opprettes ved hjelp av Actions som er definert i ø konstruktoren til denne klassen . Dette omfatter at
13  * f. eks "ny post it" fra menylinja trigger den samme aksjonen som ny post it ø hyreklipp meny ..</p>
14  * @author aHvard Alstad
15  * @version 1.0
16  */
17 public class Menyer implements ActionListener
18 {
19     public Brukergrensesnitt theBrukergrensesnitt ;
20     private Kontroller theKontroller ;
21     JMenu menu;
22     JMenu subMenu;
23     JMenuItem menuItem;
24     JMenuBar menuBar;
25     JToolBar toolBar;
26
27     // variable som benyttes for å identifisere eventer ifm . drop menyer
28     // fi lmeny
29     public static final int E_NY = 1;
30     public static final int E_APNE = 2;
31     public static final int E_LAGRE = 3;
32     public static final int E_LAGRE_SOM = 4;
33     public static final int E_SKRIV_UT = 5;
34     public static final int E_LUKK = 6;
35     public static final int E_AVSLUTT = 7;
36     private Action nyTavle;
37     private Action apneTavle;
38     private Action lagreTavle ;
39     private Action lagreTavleSom;
40     private Action skrivUt;
41     private Action lukk;
42     private Action avslutt ;
43
44     // rediger meny
45     public static final int R_ANGRE = 8;
46     public static final int R_GJENOPPRETT = 9;
47     public static final int R_KOPIER = 10;
48     public static final int R_KLIIPP_UT = 11;
49     public static final int R_LIM_INN = 12;
50     public static final int R_MERK_ALT = 26;
51     public static final int R_SLETTMERKET = 27;
52     private Action angre;
53     private Action gjenopprett ;
54     private Action kopier;
55     private Action klippUt;

```

```

56 private Action limInn;
57 private Action merkAlt;
58 private Action slettMerket ;
59 private JMenuItem angreltem = null;
60 private JButton angreButton = null;
61 private JMenuItem gjenoppsettItem = null;
62 private JButton gjenoppsettButton = null;
63
64 // post – it meny
65 public static final int P_NY = 13;
66 public static final int P_ENDRE = 14;
67 public static final int P_SLETT = 15;
68 private Action nyPostIt ;
69 private Action endrePostIt ;
70 private Action slettPostIt ;
71
72 //Gruppeny
73 public static final int G_NY = 16;
74 public static final int G_ENDRE = 17;
75 public static final int G_SLETT = 18;
76 public static final int G_RELASJON = 19;
77 private Action nyGruppe;
78 private Action endreGruppe;
79 private Action slettGruppe;
80 private Action relasjon ;
81 private JToggleButton relasjonButton ;
82
83 // Generelt
84 public static final int PIL = 20;
85 public static final int CONNECTED = 21;
86 public static final int T_ENDRE = 22;
87 public static final int ZOOMUP = 23;
88 public static final int ZOOMDOWN = 24;
89 public static final int OMKJAPP = 25;
90
91 //26 er tatt ..... ( ligger sammen elementer i rediger menyen)
92 //27 er tatt ..... ( ligger sammen elementer i rediger menyen)
93 private Action pil ;
94 private Action connected;
95 private Action endreTavle;
96 private Action zoomup;
97 private Action zoomdown;
98 private Action omkjapp;
99 private JToggleButton pilButton;
100
101
102 public Menyer(Kontroller k)
103 {
104     theKontroller = k;
105
106     //Genererer menyelementer for fil menyen
107     nyTavle = new TavleAction("Ny_tavle ...", null, new Integer(F_NY), theKontroller);
108     apneTavle = new TavleAction("Apne_tavle ..... Ctrl+O", new ImageIcon("images/open.gif"), new Integer(F_APNE), theKontroller);
109     lagreTavle = new TavleAction("Lagre_tavle ..... Ctrl+S", new ImageIcon("images/save.gif"), new Integer(F_LAGRE), theKontroller);
110     lagreTavleSom = new TavleAction("Lagre_som... ", null, new Integer(F_LAGRE_SOM), theKontroller);
111     skrivUt = new TavleAction("Skriv_ut ...", new ImageIcon("images/print.gif"), new Integer(F_SKRIV_UT), theKontroller);
112     lukk = new TavleAction("Lukk", null, new Integer(F_LUKK), theKontroller);
113     avslutt = new TavleAction("Avslutt", null, new Integer(F_AVSLUTT), theKontroller);
114
115     //Genererer menyelementer for rediger menyen
116     angre = new TavleAction("Angre_tavle ..... Ctrl+Z", new ImageIcon("images/undo.gif"), new Integer(R_ANGRE), theKontroller);
117     gjenoppsett = new TavleAction("Gjenoppsett ..... Ctrl+Shift+Z", new ImageIcon("images/redo.gif"), new Integer(R_GJENOPPRETT), theKontroller);
118     kopier = new TavleAction("Kopier ..... Ctrl+C", new ImageIcon("images/copy.gif"), new Integer(R_KOPIER), theKontroller);
119     klippUt = new TavleAction("Klipp_ut ..... Ctrl+X", new ImageIcon("images/cut.gif"), new Integer(R_KLIPP_UT), theKontroller);
120     limInn = new TavleAction("Lim_inn ..... Ctrl+V", new ImageIcon("images/paste.gif"), new Integer(R_LIM_INN), theKontroller);
121     merkAlt = new TavleAction("Merk_alt ..... Ctrl+A", new ImageIcon("images/paste.gif"), new Integer(R_MERK_ALT), theKontroller);
122     slettMerket = new TavleAction("Slett_merke ..... Del", new ImageIcon("images/paste.gif"), new Integer(R_SLETTMERKET), theKontroller);
123
124     //Genererer menyelementer for post – it menyen
125     nyPostIt = new TavleAction("Ny_post – it", new ImageIcon("images/nypostit.gif"), new Integer(P_NY), theKontroller);
126     endrePostIt = new TavleAction("Egenskaper_for_post – it", null, new Integer(P_ENDRE), theKontroller);
127     slettPostIt = new TavleAction("Slett_post – it", null, new Integer(P_SLETT), theKontroller);
128
129     //Genererer menyelementer for gruppe menyen
130     nyGruppe = new TavleAction("Ny_gruppe", new ImageIcon("images/nygruppe.gif"), new Integer(G_NY), theKontroller);
131     endreGruppe = new TavleAction("Egenskaper_for_gruppe", null, new Integer(G_ENDRE), theKontroller);
132     slettGruppe = new TavleAction("Slett_gruppe", null, new Integer(G_SLETT), theKontroller);
133
134     relasjon = new TavleAction("Relasjon", new ImageIcon("images/relasjon.gif"), new Integer(G_RELASJON), theKontroller);
135
136     //Genererer generelle menyelementer
137     pil = new TavleAction("Merk", new ImageIcon("images/arrow.gif"), new Integer(PIL), theKontroller);
138     connected = new TavleAction("Distribuert /Lokal", new ImageIcon("images/connected.gif"), new Integer(CONNECTED), theKontroller);
139     endreTavle = new TavleAction("Egenskaper_for_tavle", null, new Integer(T_ENDRE), theKontroller);
140     zoomup = new TavleAction("Zoom_opp", new ImageIcon("images/zoomup.gif"), new Integer(ZOOMUP), theKontroller);
141     zoomdown = new TavleAction("Zoom_ned", new ImageIcon("images/zoomdown.gif"), new Integer(ZOOMDOWN), theKontroller);
142     omkjapp = new TavleAction("Om_kjApp", null, new Integer(OMKJAPP), theKontroller);
143
144 }
145
146 public JMenuItem tegnHovedMeny(boolean tegnHele)
147 {
148     menuBar = new JMenuItem();
149     if (tegnHele) {
150         //Filmeny
151         menuBar.add(fiIDropMeny());
152         //Redigermeny
153         menuBar.add(redigerDropMeny());
154         //Post – it meny

```

```

155     menuBar.add(postItDropMeny());
156     // Gruppemeny
157     menuBar.add(gruppeDropMeny());
158     // Hjelpemeny
159     menuBar.add(hjelpDropMeny());
160 } else {
161     menuBar.add(fi IDropMeny());
162 }
163 }
164 }
165 }
166
167 /**
168  Her skal ø fjende elementer inn:
169  - NyÅ
170  - pne
171  - Lagre
172  - Lagre Som
173  - Skriv ut
174  - Lukk
175  - Avslutt
176 */
177 public JMenu fi IDropMeny()
178 {
179     menu = new JMenu("Fi");
180     menu.setMnemonic(KeyEvent.VK_F);
181     JMenuItem m = null;
182
183     m = menu.add(myTavle);
184     m.setIcon (null);
185     m = menu.add(apneTavle);
186     m.setIcon (null);
187     m = menu.add(lagreTavle);
188     m.setIcon (null);
189     m = menu.add(lagreTavleSom);
190     m.setIcon (null);
191     m = menu.add(skrivUt);
192     m.setIcon (null);
193     m = menu.add(lukk);
194     m.setIcon (null);
195     m = menu.add(avslutt);
196     m.setIcon (null);
197     return menu;
198 }
199
200 /**
201  Her skal ø fjende elementer inn:
202  - Angre
203  - Gjenopprett
204  - Kopier
205  - Klipp ut
206  - Lim inn
207 */
208
209 public JMenu redigerDropMeny()
210 {
211     menu = new JMenu("Rediger");
212     menu.setMnemonic(KeyEvent.VK_R);
213     JMenuItem m = null;
214
215     m = menu.add(angre);
216     m.setIcon (null);
217     angreltem = m;
218     m = menu.add(gjenopprett);
219     m.setIcon (null);
220     gjenopprettItem = m;
221     m = menu.add(kopier);
222     m.setIcon (null);
223     m = menu.add(klippUt);
224     m.setIcon (null);
225     m = menu.add(limInn);
226     m.setIcon (null);
227     m = menu.add(merkAlt);
228     m.setIcon (null);
229     m = menu.add(slettMerket);
230     m.setIcon (null);
231     return menu;
232 }
233
234 /**
235  Her skal ø fjende elementer inn:
236  - Ny
237  - Endre
238  - Slett
239 */
240
241 public JMenu postItDropMeny()
242 {
243     menu = new JMenu("Post -it");
244     menu.setMnemonic(KeyEvent.VK_P);
245     JMenuItem m = null;
246
247     m = menu.add(nyPostIt);
248     m.setIcon (null);
249     m = menu.add(endrePostIt);
250     m.setIcon (null);
251     m = menu.add(slettPostIt);
252     m.setIcon (null);
253     return menu;
254 }

```

```

254
255 public JMenu hjelpDropMeny()
256 {
257     menu = new JMenu("Hjelp");
258     menu.setMnemonic(KeyEvent.VK_H);
259     JMenuItem m = null;
260
261     m = menu.add(omkjapp);
262     m.setIcon (null);
263     return menu;
264 }
265
266 /**
267  * Her skal øftgende elementer inn
268  * — Opprett
269  * — Endre navn
270  * — Relasjon
271  */
272 public JMenu gruppeDropMeny()
273 {
274     menu = new JMenu("Gruppe");
275     menu.setMnemonic(KeyEvent.VK_G);
276     JMenuItem m = null;
277
278     m = menu.add(nyGruppe);
279     m.setIcon (null);
280     m = menu.add(endreGruppe);
281     m.setIcon (null);
282     m = menu.add(slettGruppe);
283     m.setIcon (null);
284     m = menu.add(relasjon);
285     m.setIcon (null);
286     return menu;
287 }
288
289 /**
290  * Metode som tegner opp popup meny dersom brukeren trykker øhyre
291  * museknapp i en gruppe
292  */
293 public JPopupMenu gruppePopupMeny()
294 {
295     JPopupMenu pMenu = new JPopupMenu();
296     JMenuItem m = null;
297
298     Tavle t = theKontroller . hentTavle ();
299     Gruppe gruppe = ( Gruppe)theKontroller . hentAktiviTavleElement ();
300     Gruppe g0 = t . hentGruppe ();
301     if ( gruppe!=g0){
302         m = pMenu.add(nyPostf);
303         m.setIcon (null);
304         m = pMenu.add(nyGruppe);
305         m.setIcon (null);
306         m = pMenu.add(endreGruppe);
307         m.setIcon (null);
308         m = pMenu.add(slettGruppe);
309         m.setIcon (null);
310         slettGruppe . setEnabled (true);
311     } else {
312         m = pMenu.add(nyPostf);
313         m.setIcon (null);
314         m = pMenu.add(nyGruppe);
315         m.setIcon (null);
316         m = pMenu.add(endreTavle);
317         m.setIcon (null);
318         m = pMenu.add(slettGruppe);
319         m.setIcon (null);
320         slettGruppe . setEnabled (false);
321     }
322
323     //Dersom elementet ikke er gruppe0 skal det kunne kopieres og klippes ut ved hjelp av øhyreklikk menyen
324     if ( gruppe!=g0){
325         m = pMenu.add(kopier);
326         m = pMenu.add(klippUt);
327     }
328
329     //Dersom det finnes elementer som kan limes inn skal knapp for lim inn øvre med i popup — menyen
330     Vector v_lim_inn = theKontroller . hentUtKlippsTavle ();
331     if ( v_lim_inn !=null) {
332         if ( v_lim_inn . size ()!=0){
333             m = pMenu.add(limInn);
334         }
335     }
336
337     //Dersom det finnes relasjoner tilknyttet gruppa, skal knapp for sletting av disse øvre med i popup — menyen
338     Vector v_relasjoner = gruppe . hentRelasjoner ();
339     if ( v_relasjoner != null) {
340         if ( v_relasjoner . size ()!=0){
341             JMenuItem slettRelasjon = new JMenuItem("Slett_ørelasjon");
342             pMenu.add(slettRelasjon);
343             Gruppe g;
344             Relasjon r;
345             for ( int i = 0; i < v_relasjoner . size (); i++) {
346                 r = ( Relasjon) v_relasjoner . elementAt(i);
347                 g = r . hentTilGruppe ();
348                 if ( g==gruppe) g=r . hentFraGruppe ();
349                 slettRelasjon . add(new JMenuItem(new RelationAction(g . hentElementNavn(),r , theKontroller )));
350             }
351         }
352     }

```

```

353     return pMenu;
354 }
355
356
357 /**
358  * Metode som tegner opp popup meny dersom brukeren trykker øhyre
359  * museknapp i en gruppe
360  */
361 public JPopupMenu postItPopupMenu()
362 {
363     JPopupMenu pMenu = new JPopupMenu();
364     JMenuItem m = null;
365
366     m = pMenu.add(endrePostIt);
367     m.setIcon( null );
368     m = pMenu.add( slettPostIt );
369     m.setIcon( null );
370
371     //legger til kopier og klipp ut ...
372     m = pMenu.add(kopier);
373     m = pMenu.add(klippU);
374
375     return pMenu;
376 }
377
378 public JToolBar tegnVerktoyMeny()
379 {
380     JToolBar = new JToolBar();
381     JButton button = null;
382     JToggleButton tButton = null;
383
384     button = toolBar.add(apneTavle);
385     button.setMargin(new Insets (0,0,0,0) );
386     button.setToolTipText("Apne_tavle");
387
388     button = toolBar.add(lagreTavle);
389     button.setMargin(new Insets (0,0,0,0) );
390     button.setToolTipText("Lagre_tavle");
391
392     button = toolBar.add(skrivU);
393     button.setMargin(new Insets (0,0,0,0) );
394     button.setToolTipText("Skriv_ut");
395
396     toolBar.addSeparator();
397     button = toolBar.add(zoomup);
398     button.setMargin(new Insets (0,0,0,0) );
399     button.setToolTipText("Zoom_+");
400
401     button = toolBar.add(zoomdown);
402     button.setMargin(new Insets (0,0,0,0) );
403     button.setToolTipText("Zoom_-");
404
405     toolBar.addSeparator();
406
407     button = toolBar.add(klippU);
408     button.setMargin(new Insets (0,0,0,0) );
409     button.setToolTipText("Klipp_ut");
410
411     button = toolBar.add(kopier);
412     button.setMargin(new Insets (0,0,0,0) );
413     button.setToolTipText("Kopier");
414
415     button = toolBar.add(limInn);
416     button.setMargin(new Insets (0,0,0,0) );
417     button.setToolTipText("Lim_inn");
418
419     button = toolBar.add(angre);
420     button.setMargin(new Insets (0,0,0,0) );
421     button.setToolTipText("Angre");
422     angreButton = button;
423     angreButton.setEnabled( false );
424
425     button = toolBar.add(gjenopprett);
426     button.setMargin(new Insets (0,0,0,0) );
427     button.setToolTipText("Gjenopprett");
428     gjenopprettButton = button;
429     gjenopprettButton.setEnabled( false );
430
431     toolBar.addSeparator();
432
433     tButton = new JToggleButton(pil);
434     tButton.setText("");
435     tButton.setMargin(new Insets (1,1,1,1) );
436     tButton.setToolTipText("Merk");
437     toolBar.add(tButton);
438     pilButton = tButton;
439
440     button = toolBar.add(nyPostIt);
441     button.setMargin(new Insets (0,0,0,0) );
442     button.setToolTipText("Ny_post-it");
443
444     button = toolBar.add(nyGruppe);
445     button.setMargin(new Insets (0,0,0,0) );
446     button.setToolTipText("Ny_gruppe");
447
448     tButton = new JToggleButton(relasjon);
449     tButton.setText("");
450     tButton.setToolTipText("Relasjon");
451     tButton.setMargin(new Insets (1,1,1,1) );

```

```

452     toolBar.add(tButton);
453     relasjonButton = tButton;
454
455     toolBar.addSeparator();
456
457     tButton = new JToggleButton(connected);
458     tButton.setText("");
459     tButton.setToolTipText(" Distribuert /Lokal");
460     tButton.setMargin(new Insets(1,1,1,1));
461     toolBar.add(tButton);
462
463     //Forhindrer at brukeren kan flytte 'ap overkrymenyen
464     toolBar.setFloatable(false);
465     return toolBar;
466 }
467
468 public JPopupMenu tegnHoyreKlikkMeny(TavleElement t)
469 {
470     JPopupMenu m = null;
471     if (t instanceof Gruppe){
472         m = gruppePopupMenu();
473     }else if (t instanceof PostIt){
474         m = postItPopupMenu();
475     }
476     return m;
477 }
478
479 public void actionPerformed(ActionEvent e)
480 {
481     menuItem = (JMenuItem)e.getSource();
482 }
483
484 public void settPilButton(boolean b)
485 {
486     pilButton.setSelected(b);
487 }
488
489 public void settRelasjonButton(boolean b)
490 {
491     relasjonButton.setSelected(b);
492 }
493
494 public void settAngre(boolean b)
495 {
496     angreButton.setEnabled(b);
497     angreItem.setEnabled(b);
498 }
499
500 public void settGjenopprett(boolean b)
501 {
502     gjenopprettButton.setEnabled(b);
503     gjenopprettItem.setEnabled(b);
504 }
505
506 public boolean hentGjenopprett()
507 {
508     return gjenopprettButton.isEnabled();
509 }
510 public void oppdaterMenyForPostIt()
511 {
512     nyPostIt.setEnabled(true);
513     endrePostIt.setEnabled(true);
514     slettPostIt.setEnabled(true);
515     nyGruppe.setEnabled(true);
516     endreGruppe.setEnabled(false);
517     slettGruppe.setEnabled(false);
518 }
519
520 public void oppdaterMenyForGruppe()
521 {
522     nyGruppe.setEnabled(true);
523     endreGruppe.setEnabled(true);
524     slettGruppe.setEnabled(true);
525     nyPostIt.setEnabled(true);
526     endrePostIt.setEnabled(false);
527     slettPostIt.setEnabled(false);
528 }
529
530 public void oppdaterMenyForTavle()
531 {
532     nyGruppe.setEnabled(true);
533     endreGruppe.setEnabled(false);
534     slettGruppe.setEnabled(false);
535     nyPostIt.setEnabled(true);
536     endrePostIt.setEnabled(false);
537     slettPostIt.setEnabled(false);
538 }
539
540 public void oppdaterMenyForDistribuert(boolean b)
541 {
542     nyTavle.setEnabled(b);
543     apneTavle.setEnabled(b);
544     lagreTavleSom.setEnabled(b);
545     skrivUt.setEnabled(b);
546     lukk.setEnabled(b);
547     avslutt.setEnabled(b);
548 }
549
550 }

```

```

551
552 class TavleAction extends AbstractAction{
553     private Kontroller kontrollert;
554
555     public TavleAction( String name,Icon icon, Integer aksjon, Kontroller k){
556         putValue(Action.NAME,name);
557         putValue(Action.SMALL_ICON,icon);
558         putValue("aksjon",aksjon);
559         kontrollert = k;
560     }
561
562     public void actionPerformed(ActionEvent evt)
563     {
564         int aksjon = (( Integer )getValue("aksjon")).intValue();
565         switch ( aksjon )
566         {
567             case Menyer.F_NY:
568                 kontrollert.nyTavle();
569                 break;
570             case Menyer.F_APNE:
571                 kontrollert.apneTavle();
572                 break;
573             case Menyer.F_LAGRE:
574                 kontrollert.lagreTavle();
575                 break;
576             case Menyer.F_LAGRE_SOM:
577                 kontrollert.lagreTavleSom();
578                 break;
579             case Menyer.F_SKRIV_UT:
580                 // kontrollert.skrivUt();
581                 break;
582             case Menyer.F_LUKK:
583                 kontrollert.lukkTavle();
584                 break;
585             case Menyer.F_AVSLUTT:
586                 kontrollert.avslutt();
587                 break;
588             case Menyer.P_NY:
589                 kontrollert.nyPostIt();
590                 break;
591             case Menyer.P_ENDRE:
592                 kontrollert.endrePostIt();
593                 break;
594             case Menyer.P_SLETT:
595                 kontrollert.slettPostIt();
596                 break;
597             case Menyer.G_NY:
598                 kontrollert.nyGruppe();
599                 break;
600             case Menyer.G_ENDRE:
601                 kontrollert.endreGruppeNavn();
602                 break;
603             case Menyer.G_SLETT:
604                 kontrollert.slettGruppe();
605                 break;
606             case Menyer.G_RELASJON:
607                 kontrollert.setToggleRelasjon(! kontrollert.hentToggleRelasjon());
608                 break;
609             case Menyer.PIL:
610                 // Switch funksjon - velgOmrade i Kontroller settes lik !velgOmrade
611                 kontrollert.setVelgOmrade(! kontrollert.hentVelgOmrade());
612                 break;
613             case Menyer.R_ANGRE:
614                 kontrollert.angreSiste();
615                 break;
616             case Menyer.R_GJENOPPRETT:
617                 kontrollert.gjenopprettSiste();
618                 break;
619             case Menyer.R_KOPIER:
620                 kontrollert.kopierTavleElementer();
621                 break;
622             case Menyer.R_KLIPP_UT:
623                 kontrollert.klippUTavleElement();
624                 break;
625             case Menyer.R_LIM_INN:
626                 kontrollert.limInnTavleElement();
627                 break;
628             case Menyer.R_MERK_ALT:
629                 kontrollert.merkAlt();
630                 break;
631             case Menyer.R_SLETTMERKET:
632                 kontrollert.slettMerke();
633                 break;
634             case Menyer.CONNECTED:
635                 kontrollert.setJobberLokalt(! kontrollert.hentJobberLokalt());
636                 break;
637             case Menyer.T_ENDRE:
638                 kontrollert.endreTema();
639                 break;
640             case Menyer.ZOOMDOWN:
641                 kontrollert.dekrementerZoomfaktor();
642                 break;
643             case Menyer.ZOOMUP:
644                 kontrollert.inkrementerZoomFaktor();
645                 break;
646             case Menyer.OMKJAPP:
647                 break;
648         } //switch
649     } //metoden

```

```

650
651
652 class ActionButton extends JButton{
653
654     public ActionButton(Action a)
655     {
656         setText ((String)a, getValue(Action.NAME));
657         Icon icon = (Icon)a, getValue(Action.SMALL_ICON);
658         if (icon!=null) setIcon (icon);
659         addActionListener (a);
660     }
661 }
662
663 class RelationAction extends AbstractAction{
664     private Kontroller kontroll;
665
666     public RelationAction (String name,Relasjon r, Kontroller k){
667         putValue(Action.NAME,name);
668         putValue(" relasjon ",r);
669         kontroll = k;
670     }
671
672     public void actionPerformed (ActionEvent evt)
673     {
674         Relasjon r = (( Relasjon)getValue(" relasjon "));
675         kontroll . slettRelasjon (r);
676     }
677 }

```

## F.1.9 Posisjon.java

Listing: Posisjon.java

```

1 package kjapp;
2
3 /**
4  * <p>Title: Posisjon</p>
5  * <p>Description: Klasse som benyttes for'a kunne utveksle x og y komponenter for en posisjon.<br>
6  * <p>Muliggjør utveksling av hele posisjonsobjekter istedenfor'a å mitte hente x og y posisjon hver for seg<br>
7  * xPos og yPos er public slik at de skal kunne aksesseres ved hjelp av . operatoren</p>
8  *
9  * @author 'aHvard Alstad
10 * @version 1.0
11 */
12
13 public class Posisjon {
14     public int xPos;
15     public int yPos;
16
17     public Posisjon ()
18     {
19     }
20
21     public Posisjon (int x,int y)
22     {
23         xPos = x;
24         yPos = y;
25     }
26
27     public Posisjon hentPosisjon ()
28     {
29         return this;
30     }
31
32     public int hentXPos()
33     {
34         return xPos;
35     }
36
37     public int hentYPos()
38     {
39         return yPos;
40     }
41
42     public void settXPos(int x)
43     {
44         xPos = x;
45     }
46
47     public void settYPos(int y)
48     {
49         yPos = y;
50     }
51
52     public void settPosisjon (int x,int y)
53     {
54         xPos = x;
55         yPos = y;
56     }
57 }

```

## F.1.10 PostIt.java

## Listing: PostIt.java

```

1  package kjapp;
2
3  import java.io.*;
4  import java.util.*;
5  import java.awt.*;
6  /**
7   */
8  public class PostIt extends TavleElement
9  {
10
11     /**
12      * 0 = Negativ
13      * 1 = Positiv
14      * 2 = 0Nytral ( Default)
15      */
16     public static final int STATUS.NEGATIV = 0;
17     public static final int STATUS.POSITIV = 1;
18     public static final int STATUS.NOYTRAL = 2;
19     public static final Color COLOR.NEGATIV = new Color(255,105,66); //255.67.9
20     public static final Color COLOR.POSITIV = new Color(20,175,120); //20.168.107
21     public static final Color COLOR.NOYTRAL = Color.YELLOW;
22
23     private int postItType;
24     public TavleElement theTavleElement;
25
26     /** 0 Konstrukt for debugging
27      *
28      */
29     public PostIt ()
30     {
31         super();
32     }
33
34     public PostIt (String navn, int x, int y, int niva)
35     {
36         super(navn, x, y, niva);
37     }
38
39     public PostIt (String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar,Gruppe g)
40     {
41         super(navn, x, y, bredde, hoyde, niva, kommentar,g);
42     }
43
44     public PostIt (String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar)
45     {
46         super(navn, x, y, bredde, hoyde, niva, kommentar);
47     }
48
49     public void settPostItType (int type)
50     {
51         postItType = type;
52     }
53
54     public int hentPostItType ()
55     {
56         return postItType;
57     }
58
59     public TavleElement finnTavleElement (int x,int y, Kontroller k)
60     {
61         return this;
62     }
63
64     public void toString (PrintWriter fi1, String i)
65     {
66         String indenter = i + "\t";
67         fi1.println (i + "<postit>");
68         fi1.println (indenter + "<postitid>" + hentElementID() + "</postitid>");
69         fi1.println (indenter + "<postitnavn>" + hentElementNavn() + "</postitnavn>");
70         if (!hentElementKommentar().equalsIgnoreCase("")) fi1.println (indenter + "<postitkommentar>" + hentElementKommentar() + "</
71         postitkommentar>");
72         fi1.println (indenter + "<postitposisjonX>" + hentElementPosisjonX() + "</postitposisjonX>");
73         fi1.println (indenter + "<postitposisjonY>" + hentElementPosisjonY() + "</postitposisjonY>");
74         fi1.println (indenter + "<postitbredde>" + hentElementBredde() + "</postitbredde>");
75         fi1.println (indenter + "<postithoyde>" + hentElementHoyde() + "</postithoyde>");
76         fi1.println (indenter + "<postitstatus>" + hentPostItType() + "</postitstatus>");
77         fi1.println (i + "</postit>");
78     }
79
80     public void toStringTXT(PrintWriter fi1, String i)
81     {
82         String indenter = i + "\t";
83         fi1.println (indenter + "Postitnavn " + hentElementNavn());
84         fi1.println (indenter + "Kommentar " + hentElementKommentar());
85         switch (hentPostItType()){
86         case 0:
87             fi1.println (indenter + "Postit er negativ \n");
88             break;
89         case 1:
90             fi1.println (indenter + "Postit er 0nytral \n");
91             break;
92         case 2:
93             fi1.println (indenter + "Postit er positiv \n");
94             break;
95         }
96     }

```

```

97
98
99 public Vector erElementiOmrade(int startx , int starty , int endx, int endy, Vector v, Posisjon p, Kontroller k)
100 {
101     int x = p.hentXPos();
102     int y = p.hentYPos();
103     int bredde = this.hentElementBredde();
104     int hoyde = this.hentElementHoyde();
105     if (x > startx && (x+bredde) < endx && y > starty && (y+hoyde) < endy) {
106         v.addElement(this);
107     }
108     return v;
109 }
110
111 public TavleElement klagjorAngre(Gruppe g, Tavle t){
112     int x = this.hentElementPosisjonX();
113     int y = this.hentElementPosisjonY();
114     int bredde = this.hentElementBredde();
115     int hoyde = this.hentElementHoyde();
116     String kommentar = this.hentElementKommentar();
117     String navn = this.hentElementNavn();
118     int niva = this.hentElementNiva();
119     int type = this.hentPostlType();
120     Postl p = new Postl (navn,x,y,bredde,hoyde,niva,kommentar,g);
121     p.setElementID(this.hentElementID()); // Benytter samme id 'ap postit ifm angring
122     p.setPostlType (type);
123     return p;
124 }
125
126 public Vector kopierTavleElementer(Vector kopi , Kontroller k, Gruppe forelder){
127     int x = this.hentElementPosisjonX();
128     int y = this.hentElementPosisjonY();
129     int bredde = this.hentElementBredde();
130     int hoyde = this.hentElementHoyde();
131     String kommentar = this.hentElementKommentar();
132     String navn = this.hentElementNavn();
133     int niva = this.hentElementNiva();
134     int type = this.hentPostlType();
135     Postl post = new Postl (navn,x,y,bredde,hoyde,niva,kommentar,forelder);
136     post.setPostlType (type);
137     kopi.addElement(post);
138     return kopi;
139 }
140
141 public void merkTavleElement(Graphics g, Kontroller k){
142     double z = k.hentZoomFaktor();
143
144     Posisjon pos = k.fi nnPosisjonGruppe0( this );
145     int x = Grafi skeTavleElementer .doubleToInt(pos.hentXPos()*z);
146     int y = Grafi skeTavleElementer .doubleToInt(pos.hentYPos()*z);
147     int hoyde = Grafi skeTavleElementer .doubleToInt( this .hentElementHoyde()*z);
148     int bredde = Grafi skeTavleElementer .doubleToInt( this .hentElementBredde()*z);
149     int hjorneMerking = Grafi skeTavleElementer .doubleToInt(3*z); // Gir 9 strelsen 'ap merking av post—it
150     if (hjorneMerking<2)hjorneMerking=2;
151     g.setColor(Color.black);
152     g.fi l3DRect (x—hjorneMerking, y—hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
153     g.fi l3DRect (x+bredde—hjorneMerking, y—hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
154     g.fi l3DRect (x—hjorneMerking, y+hoyde—hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
155     g.fi l3DRect (x+bredde—hjorneMerking, y+hoyde—hjorneMerking, 2*hjorneMerking, 2*hjorneMerking, true);
156 }
157
158 public Vector hentTavleElementer () {
159     return null;
160 }
161 }

```

## F.1.11 Relasjon.java

Listing: Relasjon.java

```

1 package kjapp;
2
3 import java .io .*;
4
5
6 public class Relasjon
7 {
8     private Gruppe fraGruppe;
9     private Gruppe tilGruppe;
10
11     /**
12      * relasjonType gir muligheten til'a angi hvordan relasjonen skal se ut 'ap Tavlen . Dette kan for eksempel være :
13      * 0 = positiv relasjon med pil i begge ender
14      * 1 = positiv relasjon med pil bare 'ap tilGruppe
15      * 2 = positiv relasjon uten pil
16      * 3 = negativ relasjon med pil i begge ender
17      * 4 = negativ relasjon med pil bare 'ap tilGruppe
18      * 5 = negativ relasjon uten pil
19
20      * Hvor positive relasjoner har heltrukken linje og negative relasjoner har stiplede linje .
21      */
22     private int relasjonsType ;
23
24     public Gruppe theGruppe;
25     public Tavle theTavle ;

```

```

26
27     public Relasjon()
28     {
29     }
30
31     public void settTilGruppe(Gruppe g)
32     {
33         tilGruppe = g;
34     }
35
36     public void settFraGruppe(Gruppe g)
37     {
38         fraGruppe = g;
39     }
40
41     public Gruppe hentTilGruppe()
42     {
43         return tilGruppe;
44     }
45
46     public Gruppe hentFraGruppe()
47     {
48         return fraGruppe;
49     }
50
51     public void toString(PrintWriter fi1, String i, Gruppe g)
52     {
53         Gruppe skrivG = this.hentTilGruppe();
54         String indent = i + "\t";
55         if (skrivG==g){//Denne gruppen er tilGruppe i relasjonen — skriver ut navnet 'ap fra gruppen
56             skrivG = this.hentFraGruppe();
57             fi1.println(i + "< relasjontil >" + skrivG.hentElementID() + "</ relasjontil >");
58         }else{//Denne gruppen er fraGruppen i relasjonen — skriver ut navnet 'ap til gruppen
59             fi1.println(i + "< relasjonfra >" + skrivG.hentElementID() + "</relasjonfra >");
60         }
61     }
62 }
63 }

```

## F.1.12 Tavle.java

### Listing: Tavle.java

```

1     package kjapp;
2
3     import java.util.Vector;
4     import java.io.*;
5
6     /**
7     */
8     public class Tavle
9     {
10        private String tavleNavn;
11        private String fiINavn;
12        private String encoding = "ISO-8859-1";
13        private String tavleTema;
14        private Gruppe gruppe0;
15        private Vector relasjoneriTavle;
16        private int tavleHoyde;
17        private int tavleBredde;
18
19        /**
20         * Tar vare 'ap aktivt tavle element. Dette er enten en Gruppe eller en Postlt.
21         */
22        private TavleElement aktivTavleElement;
23        private Vector utklippTavle;
24        private IOTavle theOTavle;
25        private Relasjon theRelasjon;
26        private Kontroller theKontroller;
27        private Gruppe theGruppe;
28
29        //Dersom brukeren har trykket nederst i venstre øhjerne 'ap et tavleElement
30        //Settes et fugg som angir at elementet skal resize og ikke flyttes ...
31        public boolean resizeElement;
32
33        public Tavle(Kontroller k, String fi1)
34        {
35            theKontroller = k;
36            gruppe0 = new Gruppe("Tema", 0, 0, 800, 600, 0, "Gruppe0 opprettet 'ap tavlen", null, this);
37            relasjoneriTavle = new Vector();
38            fiINavn = fi1;
39        }
40
41        /**
42         * 'ø Konstruktør for'a opprette ny tavle med tavletema og gruppe0.
43         * gruppe0 skal inneholde alle andre tavleelementer og har
44         * utstrekning lik tavleBredde og tavleHoyde.
45         * gruppe0 opprettes i posisjon 0,0.
46         */
47        public Tavle(String tema, int bredde, int hoyde, Kontroller k, String f)
48        {
49            tavleBredde = bredde;
50            tavleHoyde = hoyde;
51            tavleTema = tema;
52            theKontroller = k;

```

```

53     System.out.println("Ny tavle er opprettet med navn: " + tavleTema);
54     relasjoneriTavle = new Vector();
55     fiINavn = f;
56
57     gruppe0 = new Gruppe(tavleTema, 0, 0, tavleBredde, tavleHoyde, 0, "Gruppe 0 opprettet på tavlen", null, this);
58 }
59
60 public Gruppe hentGruppe0()
61 {
62     return gruppe0;
63 }
64
65 /**
66  * @roseuid 3C57C70F00FB
67  */
68 public void endreTema(String tema)
69 {
70     tavleTema = tema;
71 }
72
73 /**
74  * @roseuid 3C5696BB0334
75  */
76 public Gruppe nyGruppe(String gruppeNavn)
77 {
78     Gruppe g = new Gruppe(gruppeNavn, this);
79     gruppe0.leggTilUnderGruppe(g);
80     return g;
81 }
82
83 public Gruppe fi nnGruppe(long id){
84     // Finner en gruppe med id
85     Gruppe g = this.hentGruppe0();
86     Vector v = g.hentTavleElementer();
87     Gruppe kandidat = null;
88     for(int i=0; i<v.size();i++){
89         if(v.elementAt(i) instanceof Gruppe){
90             Gruppe g1 = (Gruppe)v.elementAt(i);
91             if(g1.hentElementID()==id){
92                 kandidat = g1;
93             }else{
94                 Gruppe temp_kandidat = g1.fi nnGruppe(this, id);
95                 if(temp_kandidat!=null) kandidat = temp_kandidat;
96             }
97         }
98     }
99     return kandidat;
100 }
101
102 public void nyRelasjon(Relasjon r)
103 {
104     relasjoneriTavle.addElement(r);
105 }
106
107 /**
108  * Metode som benyttes fra TavleHandler.
109  * 1. dette tilfellet er en <relasjonfra> tag lest, og gruppen skal settes som fra element i relasjonen
110  * 1.1. Hvis tilsvarende relasjon er opprettet tidligere, returneres den relasjonen
111  * 2. Hvis ikke opprettes en ny relasjon med gruppe g som fra gruppe og fra_navn som
112  * @param fra_navn
113  * @param g
114  * @return
115  */
116 public Relasjon nyRelasjonFra(Gruppe gFra, long tilID)
117 {
118     Gruppe fra=null;
119     Gruppe til=null;
120     Relasjon r = null;
121     Relasjon returRelasjon=null;
122     if ( relasjoneriTavle !=null){
123         for(int i=0;i<relasjoneriTavle.size();i++){
124             r = (Relasjon) relasjoneriTavle.elementAt(i);
125             fra = r.hentFraGruppe();
126             til = r.hentTilGruppe();
127             if ( til !=null){
128                 if ( til.hentElementID()==tilID){//Det ligger allerede en relasjon i gruppa som har til gruppe med id = tilID
129                     if ( fra==gFra){//Og i den relasjonen er til gruppen lik g — denne relasjonen er lik den jeg skal ha ...
130                         return r;
131                     }else{//Denne relasjonen har en annen eller ingen fra gruppe
132                         if ( fra==null){//Denne relasjonen har ingen til gruppe. Setter "min" gruppe (g) som tilgruppe og returnerer relasjonen
133                             returRelasjon = r;
134                             break;
135                         }
136                     }
137                 }
138             }
139         }
140     }
141     if ( returRelasjon !=null){
142         returRelasjon.settFraGruppe(gFra);
143         gFra.leggTilRelasjon ( returRelasjon );
144     }else{
145         r = new Relasjon();
146         returRelasjon = r;
147         returRelasjon.settFraGruppe(gFra);
148         gFra.leggTilRelasjon ( returRelasjon );
149         this.nyRelasjon ( returRelasjon );
150     }
151     return returRelasjon ;

```

```

152     }
153
154     /**
155     * Metode som benyttes fra TavleHandler.
156     * I dette tilfellet er en <relasjon til> tag lest, og gruppen skal settes som fra element i relasjonen
157     * 1. Sjekker om en tilsvarende relasjon er opprettet tidligere
158     * 1.1. Hvis tilsvarende relasjon er opprettet tidligere, returneres den relasjonen
159     * 2. Hvis ikke opprettes en ny relasjon med gruppe g som fra gruppe og fra_navn som
160     * @param fra_navn
161     * @param g
162     * @return
163     */
164     public Relasjon nyRelasjonTil(long fraID, Gruppe gTil)
165     {
166         Gruppe fra=null;
167         Gruppe til=null;
168         Relasjon r=null;
169         Relasjon returRelasjon=null;
170         if ( relasjoneriTavle !=null){
171             for(int i=0;i<relasjoneriTavle.size();i++){
172                 r=(Relasjon) relasjoneriTavle.elementAt(i);
173                 fra=r.hentFraGruppe();
174                 til=r.hentTilGruppe();
175                 if ( fra !=null){
176                     if ( fra.hentElementID()==fraID){//Det ligger allerede en relasjon i gruppa som har fra gruppe med id = fraID
177                         if ( til ==gTil){//Og i den relasjonen er til gruppen lik g — denne relasjonen er lik den jeg skal ha ...
178                             return r;
179                         }else{//Denne relasjonen har en annen eller ingen til gruppe
180                             if ( til ==null){//Denne relasjonen har ingen til gruppe. Setter "min" gruppe (g) som tilgruppe og returnerer relasjonen
181                                 returRelasjon = r;
182                                 break;
183                             }
184                         }
185                     }
186                 }
187             }
188         }
189         if ( returRelasjon !=null){
190             returRelasjon.setTilGruppe(gTil);
191             gTil.leggTilRelasjon( returRelasjon );
192         }else{
193             r = new Relasjon();
194             returRelasjon = r;
195             returRelasjon.setTilGruppe(gTil);
196             gTil.leggTilRelasjon( returRelasjon );
197             this.nyRelasjon( returRelasjon );
198         }
199         return returRelasjon ;
200     }
201
202     public void slettRelasjon (Relasjon r)
203     {
204         for(int i = 0; i<relasjoneriTavle.size();i++){
205             if ( relasjoneriTavle.elementAt(i)==r){
206                 relasjoneriTavle.removeElementAt(i);
207                 System.out.println(" Sletter _relasjon _fra _tavle" + r);
208             }
209         }
210         Gruppe fra = r.hentFraGruppe();
211         Gruppe til = r.hentTilGruppe();
212         fra.slettRelasjon (r);
213         til.slettRelasjon (r);
214     }
215
216     public Vector hentRelasjoner ()
217     {
218         return relasjoneriTavle ;
219     }
220
221     public void settRelasjoner (Vector v)
222     {
223         relasjoneriTavle = v;
224     }
225
226     public void endreGruppeNavn(Gruppe g, String navn)
227     {
228         g.settElementNavn(navn);
229     }
230
231     public TavleElement fi nnTavleElement(int xPos, int yPos)
232     {
233         return gruppe0.fi nnTavleElement(xPos, yPos, theKontroller );
234     }
235
236     public void settTavleBredde(int b)
237     {
238         tavleBredde = b;
239     }
240
241     public void settTavleHoyde(int h)
242     {
243         tavleHoyde = h;
244     }
245
246     public int hentTavleBredde()
247     {
248         return tavleBredde;
249     }
250

```

```

251 public int hentTavleHoyde()
252 {
253     return tavleHoyde;
254 }
255
256 public String hentTavleTema()
257 {
258     return tavleTema;
259 }
260
261 public void settResizeElement (boolean b){
262     theKontroller . settResizeElement (b);
263 }
264
265 public boolean hentResizeElement () {
266     return theKontroller .hentResizeElement();
267 }
268
269 public void toString (PrintWriter f){
270     Gruppe gruppe0 = hentGruppe0();
271     Vector elementer = gruppe0.hentTavleElementer ();
272     String indenter = "\t";
273     if (getEncoding() != null) {
274         f. print ("<?xml version='1.0'>");
275         f. print (" encoding=\'' + this.getEncoding() + '\'?>");
276     } else {
277         f. println ("<?xml version='1.0'?>");
278     }
279     f. println ("<tavle>");
280     f. println ("<tavlenavn>" + gruppe0.hentElementNavn() + "</tavlenavn>");
281     for (int i = 0; i < elementer.size(); i++) {
282         ((TavleElement)elementer .elementAt(i)) .toString (f, indenter);
283     }
284     f. println ("</tavle>");
285 }
286
287 public void toStringTXT(PrintWriter f){
288     Gruppe gruppe0 = hentGruppe0();
289     Vector elementer = gruppe0.hentTavleElementer ();
290     String indenter = "\t";
291     f. println ("Tavlenavn: " + gruppe0.hentElementNavn());
292     f. println ("");
293     f. println ("Tavlen inneholder følgende elementer:");
294     for (int i = 0; i < elementer.size(); i++) {
295         ((TavleElement)elementer .elementAt(i)) .toStringTXT(f, indenter);
296     }
297 }
298
299 public String getEncoding() {
300     return encoding;
301 }
302
303 public void setEncoding (String e){
304     encoding = e;
305 }
306
307 public Vector hentUtklippTavle () {
308     return utklippTavle ;
309 }
310
311 public Vector kopierTavleElementer (Vector merkede){
312     Vector kopi = new Vector();
313     for (int i = 0; i < merkede.size(); i++) {
314         TavleElement t = (TavleElement)merkede .elementAt(i);
315         Gruppe forelder = this .hentGruppe0();
316         kopi = t .kopierTavleElement (kopi, theKontroller , forelder);
317     }
318     return kopi;
319 }
320
321 public void klippUtTavleElementer (Vector merkede){
322     for (int i = 0; i < merkede.size(); i++) {
323         TavleElement t = (TavleElement)merkede .elementAt(i);
324         if (t instanceof Gruppe) {
325             /*Dersom t er en gruppe, skal relasjonen i gruppen fjernes ved klippUt
326             */
327             Gruppe g = (Gruppe)merkede .elementAt(i);
328             Vector rel = g .hentRelasjoner ();
329             if (rel != null) {
330                 while (rel .size () > 0) {
331                     int n = 0;
332                     // Relasjonene fjernes øfrst i tavlen.
333                     // Deretter fjernes relasjonene i til og fra gruppene
334                     // Dette øgjøres via metode i tavlen
335                     Relasjon r = (Relasjon)rel .elementAt(n);
336                     this .slettRelasjon (r);
337                 } // end while
338             } // end if
339         } // end if
340     } // end for
341     t .klippUtTavleElementer ();
342 } // end metode klippUtTavleElementer
343
344 public Kontroller hentKontroller () {
345     return theKontroller ;
346 }
347
348 public void settFilNavn (String f)

```

```

350     {
351         fi lNavn = f;
352     }
353
354     public String hentFilNavn()
355     {
356         return fi lNavn;
357     }
358 }

```

## F.1.13 TavleElement.java

### Listing: TavleElement.java

```

1  package kjapp;
2
3  import java.io.*;
4  import java.util.*;
5  import java.awt.*;
6
7  public abstract class TavleElement
8  {
9      private String elementNavn = "";
10     private int xPos;
11     private int yPos;
12     private int elementBredde;
13     private int elementHoyde;
14     private int elementNiva;
15     private long elementID;
16     private String elementKommentar = "";
17     private Gruppe forelderGruppe;
18     public PostIt thePostIt;
19     private static long tavleElementID = 0;
20
21
22     public TavleElement(String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar, Gruppe g)
23     {
24         elementNavn = navn;
25         xPos = x;
26         yPos = y;
27         elementBredde = bredde;
28         elementHoyde = hoyde;
29         elementNiva = niva;
30         elementKommentar = kommentar;
31         forelderGruppe = g;
32         tavleElementID=tavleElementID+1;
33         elementID = tavleElementID;
34     }
35
36     public TavleElement(String navn, int x, int y, int niva)
37     {
38         elementNavn = navn;
39         xPos = x;
40         yPos = y;
41         elementNiva = niva;
42         tavleElementID=tavleElementID+1;
43         elementID = tavleElementID;
44     }
45
46     public TavleElement(int x, int y, int niva)
47     {
48         xPos = x;
49         yPos = y;
50         elementNiva = niva;
51         tavleElementID=tavleElementID+1;
52         elementID = tavleElementID;
53     }
54
55     public TavleElement(String n)
56     {
57         elementNavn = n;
58         tavleElementID=tavleElementID+1;
59         elementID = tavleElementID;
60     }
61
62     public TavleElement()
63     {
64         tavleElementID=tavleElementID+1;
65         elementID = tavleElementID;
66     }
67
68     public TavleElement(String navn, int x, int y, int bredde, int hoyde, int niva, String kommentar)
69     {
70         elementNavn = navn;
71         xPos = x;
72         yPos = y;
73         elementBredde = bredde;
74         elementHoyde = hoyde;
75         elementNiva = niva;
76         elementKommentar = kommentar;
77         tavleElementID=tavleElementID+1;
78         elementID = tavleElementID;
79     }
80
81     public void settElementNavn(String navn)

```

```
82     {
83     elementNavn = navn;
84     }
85
86     public String hentElementNavn()
87     {
88     return elementNavn;
89     }
90
91     public void settElementPosisjonX(int x)
92     {
93     xPos = x;
94     }
95
96     public void settElementPosisjonY(int y)
97     {
98     yPos = y;
99     }
100
101     public int hentElementPosisjonX()
102     {
103     return xPos;
104     }
105
106     public int hentElementPosisjonY()
107     {
108     return yPos;
109     }
110
111     public void settElementBredde(int bredde)
112     {
113     elementBredde = bredde;
114     }
115
116     public int hentElementBredde()
117     {
118     return elementBredde;
119     }
120
121     public void settElementHoyde(int hoyde)
122     {
123     elementHoyde = hoyde;
124     }
125
126     public int hentElementHoyde()
127     {
128     return elementHoyde;
129     }
130
131     public void settElementKommentar(String kommentar)
132     {
133     elementKommentar = kommentar;
134     }
135
136     public String hentElementKommentar()
137     {
138     return elementKommentar;
139     }
140
141     public void settElementNiva(int niva)
142     {
143     elementNiva = niva;
144     }
145
146     public int hentElementNiva()
147     {
148     return elementNiva;
149     }
150
151     public void settForelder (Gruppe g)
152     {
153     forelderGruppe = g;
154     }
155
156     public Gruppe hentForelder ()
157     {
158     return forelderGruppe;
159     }
160
161     public void klippUt(TavleElementer () {
162     Gruppe forelder = this.hentForelder ();
163     Vector forelderElementer = forelder.hentTavleElementer ();
164     for (int i=0; i < forelderElementer.size (); i++){
165     TavleElement t = (TavleElement) forelderElementer.elementAt(i);
166     if (t==this) {
167     forelderElementer.removeElementAt(i);
168     }
169     }
170     }
171
172     public long hentElementID()
173     {
174     return elementID;
175     }
176
177     /**
178     * Metoden settElementID benyttes kun ifm innlesing fra xml fil
179     * @param l
180     */
```

```

181 public void settElementID(long l)
182 {
183     elementID=l;
184 }
185
186
187 public abstract Vector erElementiOmrade(int startx, int starty, int endx, int endy, Vector v, Posisjon p, Kontroller k);
188 public abstract TavleElement finnTavleElement(int xPos, int yPos, Kontroller k);
189 public abstract TavleElement klargjorAngre(Gruppe g, Tavle t);
190 public abstract Vector kopierTavleElementer(Vector v, Kontroller k, Gruppe forelder);
191 public abstract void merkTavleElement(Graphics g, Kontroller k);
192 public abstract void toString(PrintWriter fi1, String indenter);
193 public abstract void toStringTXT(PrintWriter fi1, String indenter);
194 public abstract Vector hentTavleElementer();
195 }

```

## F.1.14 TavleHandler.java

### Listing: TavleHandler.java

```

1 package kjapp;
2
3 import javax.xml.parsers.*;
4 import org.xml.sax.*;
5 import org.xml.sax.helpers.*;
6 import java.util.Stack;
7 import java.io.*;
8
9 public class TavleHandler extends DefaultHandler
10 {
11     private int debug = 0;
12     private String tagVerdi;
13     private Tavle theTavle;
14     private String status;
15     private Stack theStack;
16     private Gruppe g;
17     private Gruppe g2;
18     private PostIt p;
19
20     public TavleHandler()
21     {
22         super();
23     }
24
25     public TavleHandler(Tavle t)
26     {
27         super();
28         theTavle = t;
29         theStack = new Stack();
30     }
31
32     public void startElement (String namespaceURI, String lname, String qname, Attributes attrs)
33     {
34         if (qname.equalsIgnoreCase("tavle"))
35         {
36             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
37             g = theTavle.hentGruppe0();
38             theStack.push(g);
39             g = null;
40         }
41
42         else if (qname.equalsIgnoreCase("tavlenavn"))
43         {
44             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
45         }
46         else if (qname.equalsIgnoreCase("gruppe"))
47         {
48             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
49             g = (Gruppe)theStack.pop();
50             g2 = new Gruppe(theTavle);
51             g.leggTilTavleElement(g2);
52             g2.setForelder(g);
53             theStack.push(g);
54             theStack.push(g2);
55             g = null;
56             g2 = null;
57         }
58         else if (qname.equalsIgnoreCase("gruppenavn"))
59         {
60             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
61         }
62         else if (qname.equalsIgnoreCase("gruppekommentar"))
63         {
64             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
65         }
66
67         else if (qname.equalsIgnoreCase("gruppeposisjonX"))
68         {
69             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
70         }
71         else if (qname.equalsIgnoreCase("gruppeposisjonY"))
72         {
73             if (debug > 1) {System.out.println("Start tag_qualified_name" + qname);}
74         }
75         else if (qname.equalsIgnoreCase("gruppehoyde"))

```

```

76     {
77     if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
78     }
79
80     else if ( qname.equalsIgnoreCase("gruppebredde"))
81     {
82     if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
83     }
84     else if ( qname.equalsIgnoreCase("postit"))
85     {
86     if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
87     g = ( Gruppe)theStack.pop();
88     p = new PostIt();
89     g.leggTilTavleElement(p);
90     p.setForelder(g);
91     theStack.push(g);
92     theStack.push(p);
93     }
94     else if ( qname.equalsIgnoreCase("postitnavn"))
95     {
96     if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
97     }
98
99     else if ( qname.equalsIgnoreCase("postitkommentar"))
100    {
101    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
102    }
103
104    else if ( qname.equalsIgnoreCase("postitposX"))
105    {
106    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
107    }
108    else if ( qname.equalsIgnoreCase("postitposY"))
109    {
110    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
111    }
112    else if ( qname.equalsIgnoreCase("postithoyde"))
113    {
114    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
115    }
116    else if ( qname.equalsIgnoreCase("postitbredde"))
117    {
118    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
119    }
120    else if ( qname.equalsIgnoreCase(" relasjonfra "))
121    {
122    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
123    }
124    else if ( qname.equalsIgnoreCase(" relasjontil "))
125    {
126    if ( debug > 1 ) { System.out.println(" Start _tag_qualified_name_ " + qname); }
127    }
128
129    }
130
131    public void endElement(String uri , String lname, String qname)
132    {
133    if ( qname.equalsIgnoreCase("tavle"))
134    {
135    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
136    theStack.pop(); // Dette skal øvre det siste elementet 'ap stacken
137    }
138
139    else if ( qname.equalsIgnoreCase("tavlenavn"))
140    {
141    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
142    // theTavle . endreTema(tagVerdi);
143    g = ( Gruppe)theStack.pop();
144    g.settElementNavn(tagVerdi);
145    theStack.push(g);
146    }
147    else if ( qname.equalsIgnoreCase("gruppe"))
148    {
149    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
150    theStack.pop();
151    }
152    else if ( qname.equalsIgnoreCase("gruppeid"))
153    {
154    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
155    g = ( Gruppe)theStack.pop();
156    g.settElementID(pStringToLong(tagVerdi));
157    theStack.push(g);
158    }
159    else if ( qname.equalsIgnoreCase("gruppenavn"))
160    {
161    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
162    g = ( Gruppe)theStack.pop();
163    g.settElementNavn(tagVerdi);
164    theStack.push(g);
165    }
166    else if ( qname.equalsIgnoreCase("gruppekommentar"))
167    {
168    if ( debug > 1 ) { System.out.println("End _tag_qualified_name_ " + qname); }
169    g = ( Gruppe)theStack.pop();
170    g.settElementKommentar(tagVerdi);
171    theStack.push(g);
172    }
173
174    else if ( qname.equalsIgnoreCase("gruppeposisjonX"))

```

```

175     {
176         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
177         g = (Gruppe)theStack .pop();
178         g .setElementPosisjonX( pStringToInt ( tagVerdi ));
179         theStack .push(g);
180     }
181     else if ( qname.equalsIgnoreCase("gruppeposisjonY"))
182     {
183         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
184         g = (Gruppe)theStack .pop();
185         g .setElementPosisjonY( pStringToInt ( tagVerdi ));
186         theStack .push(g);
187     }
188     else if ( qname.equalsIgnoreCase("gruppehoyde"))
189     {
190         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
191         g = (Gruppe)theStack .pop();
192         g .setElementHoyde( pStringToInt ( tagVerdi ));
193         theStack .push(g);
194     }
195
196     else if ( qname.equalsIgnoreCase("gruppebredde"))
197     {
198         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
199         g = (Gruppe)theStack .pop();
200         g .setElementBredde( pStringToInt ( tagVerdi ));
201         theStack .push(g);
202     }
203     else if ( qname.equalsIgnoreCase(" postit "))
204     {
205         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
206         theStack .pop();
207     }
208     else if ( qname.equalsIgnoreCase(" postitid "))
209     {
210         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
211         p = ( PostIt)theStack .pop();
212         p .setElementID( pStringToLong( tagVerdi ));
213         theStack .push(p);
214     }
215     else if ( qname.equalsIgnoreCase(" postinavn "))
216     {
217         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
218         p = ( PostIt)theStack .pop();
219         p .setElementNavn( tagVerdi );
220         theStack .push(p);
221     }
222
223     else if ( qname.equalsIgnoreCase("postitkommentar"))
224     {
225         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
226         p = ( PostIt)theStack .pop();
227         p .setElementKommentar( tagVerdi );
228         theStack .push(p);
229     }
230
231     else if ( qname.equalsIgnoreCase(" postitposisjonX "))
232     {
233         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
234         p = ( PostIt)theStack .pop();
235         p .setElementPosisjonX( pStringToInt ( tagVerdi ));
236         theStack .push(p);
237     }
238     else if ( qname.equalsIgnoreCase(" postitposisjonY "))
239     {
240         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
241         p = ( PostIt)theStack .pop();
242         p .setElementPosisjonY( pStringToInt ( tagVerdi ));
243         theStack .push(p);
244     }
245     else if ( qname.equalsIgnoreCase(" postithoyde "))
246     {
247         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
248         p = ( PostIt)theStack .pop();
249         p .setElementHoyde( pStringToInt ( tagVerdi ));
250         theStack .push(p);
251     }
252     else if ( qname.equalsIgnoreCase(" postitbredde "))
253     {
254         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
255         p = ( PostIt)theStack .pop();
256         p .setElementBredde( pStringToInt ( tagVerdi ));
257         theStack .push(p);
258     }
259     else if ( qname.equalsIgnoreCase(" postitstatus "))
260     {
261         if ( debug > 1) {System.out.println ("End_tag_qualifi ed_name_ " + qname);}
262         p = ( PostIt)theStack .pop();
263         p .setPostItType ( pStringToInt ( tagVerdi ));
264         theStack .push(p);
265     }
266     else if ( qname.equalsIgnoreCase(" relasjonfra "))
267     {
268         g = (Gruppe)theStack .pop();
269         Relasjon r = theTavle .nyRelasjonFra( g, pStringToLong( tagVerdi ));
270         theStack .push(g);
271     }
272     else if ( qname.equalsIgnoreCase(" relasjontil "))
273     {

```

```

274     g = (Gruppe)theStack.pop();
275     Relasjon r = theTavle.nyRelasjonTil(pStringToLong(tagVerdi),g);
276     theStack.push(g);
277 }
278 }
279
280 public void characters(char[] data, int start, int length)
281 {
282     tagVerdi = new String(data, start, length);
283 }
284
285 private int pStringToInt(String s)
286 {
287     int i = 0;
288     try{
289         i = Integer.parseInt(s);
290     }catch(NumberFormatException e){
291         System.out.println("Kan ikke gjre om " + s + " til en integer");
292     }
293     return i;
294 }
295
296 private long pStringToLong(String s)
297 {
298     long i = 0;
299     try{
300         i = Long.parseLong(s);
301     }catch(NumberFormatException e){
302         System.out.println("Kan ikke gjre om " + s + " til en integer");
303     }
304     return i;
305 }
306 }

```

## F.1.15 XMLFilter.java

Listing: XMLFilter.java

```

1  package kjapp;
2
3
4  import java.io.File;
5  import javax.swing.*;
6  import javax.swing.filechooser.*;
7  /**
8   * Title:
9   * Description:
10 * Copyright: Copyright (c) 2002
11 * Company:
12 * @author
13 * @version 1.0
14 */
15
16 public class XMLFilter extends FileFilter {
17
18     public XMLFilter() {
19     }
20
21     public boolean accept(File f) {
22         if (f.isDirectory()) {
23             return true;
24         }
25
26         String extension = getExtension(f);
27         if (extension != null) {
28             if (extension.equalsIgnoreCase("xml")) {
29                 return true;
30             } else {
31                 return false;
32             }
33         }
34         return false;
35     }
36
37     public String getDescription() {
38         return "xml-filer";
39     }
40
41     private String getExtension(File f) {
42         String ext = null;
43         String s = f.getName();
44         int i = s.lastIndexOf('.');
45
46         if (i > 0 && i < s.length() - 1) {
47             ext = s.substring(i+1).toLowerCase();
48         }
49         return ext;
50     }
51 }
52 }

```

## F.1.16 TXTFilter.java

## Listing: TXTFilter.java

```
1 package kjapp;
2
3
4 import java.io. File;
5 import javax.swing.*;
6 import javax.swing. filechooser *;
7 /**
8  * Title :
9  * Description :
10 * Copyright: Copyright (c) 2002
11 * Company:
12 * @author
13 * @version 1.0
14 */
15
16 public class TXTFilter extends FileFilter {
17
18     public TXTFilter() {
19     }
20
21     public boolean accept(File f) {
22         if (f.isDirectory()){
23             return true;
24         }
25
26         String extension = getExtension(f);
27         if (extension != null){
28             if (extension.equalsIgnoreCase("txt")){
29                 return true;
30             }else{
31                 return false;
32             }
33         }
34         return false;
35     }
36
37     public String getDescription() {
38         /**@todo: implement this javax.swing. filechooser . FileFilter abstract method*/
39         return "text files ";
40     }
41
42     private String getExtension(File f) {
43         String ext = null;
44         String s = f.getName();
45         int i = s.lastIndexOf('.');
46
47         if (i > 0 && i < s.length() - 1) {
48             ext = s.substring(i+1).toLowerCase();
49         }
50         return ext;
51     }
52
53 }
```

## **Tillegg G**

# **Installsjons CD**

Tillegget inneholder en CD med programvare som er nødvendig for å installere og kjøre KjApp. Videre inneholder cd'en brukerveiledning, javadoc for kildekode og rapporten i elektronisk format. Dersom cd'en ikke starter automatisk får du tilgang til innholdet ved å åpne filen index.html som ligger på roten på cd'en.



# Bibliografi

- [1] Jbuilder6, enterprise 6.0.438.0, 1997-2001.
- [2] Rational rose enterprise edition release version 2001.03.00, 2001. <http://www.rational.com>.
- [3] Asbjørn Aune. Kvalitetstyrte bedrifter, 1996. Kapittel 5.2.2 Kreative (ideskapende) teknikker, s.224-229.
- [4] J. Baroudi and W. Orlikowski. The problem of statistical power in mis research, 1989. Vol 13, No. 1, side 87 - 106.
- [5] Jacobsen Booch, Rumbaugh. The unified modeling language user guide, 1999.
- [6] G.L. Rein C.A Ellis. Groupware:some issues and experiences, 1992.
- [7] Horstmann Cornell. Core java 1.2 - fundamentals, 1999. Volume I.
- [8] Horstmann Cornell. Core java 2 - advanced features, 2002. Volume II.
- [9] T. Dybå. Enabling software process improvement, 2001. An Investigation of the Importance of Organizational Issues, side 150 - 155.
- [10] Johnson Vlissides Gamma, Helm. Design patterns. Elements of Reusable Object-Oriented Software.
- [11] E. Gjøs H. Alstad. Innføring av rational unified process ved hærens programvaresenter, 2001.
- [12] Massachusetts institute of technology, 1996. TQM Tool: Affinity Diagram (KJ).
- [13] Java, 2002. <http://www.java.sun.com/>.
- [14] Liam Bannon Kjeld Schmidt. Taking cscw seriously: Supporting articulation work, 1992.
- [15] R.C. Serlin L.A Marascuilo. Statistical methods for the social behavioral sciences, 1988.
- [16] C. Robson. Real world reasearch: A resource for social scientists and practitioners-researchers, 1993.
- [17] J. Castellan S. Siegel. Nonparametric statistics for the behavioral sciences, 1988. 2nd edition.

- [18] Berghout Solingen. The goal/question/metric method: A practical guide for quality improvement and software development, 1999.
- [19] David Straker. A toolbox for quality improvement and problem solving, 1995. Chapter 9 Affinity Diagram, s. 89-97.
- [20] H.D. Rombach V.R. Basili, G. Caldiera. Goal question metrics paradigm, encyclopedia of software engineering, 1994. ed. J.J. Marciniak, Vol.I, pp. 528-532.
- [21] Myers Walpole, Myers. Probability and statistics for engineers and scientists. Sixth edition.
- [22] Host Ohlsson Regnell Wesslen Wohlin, Runeson. Experimentation in software engineering - an introduction, 2000.