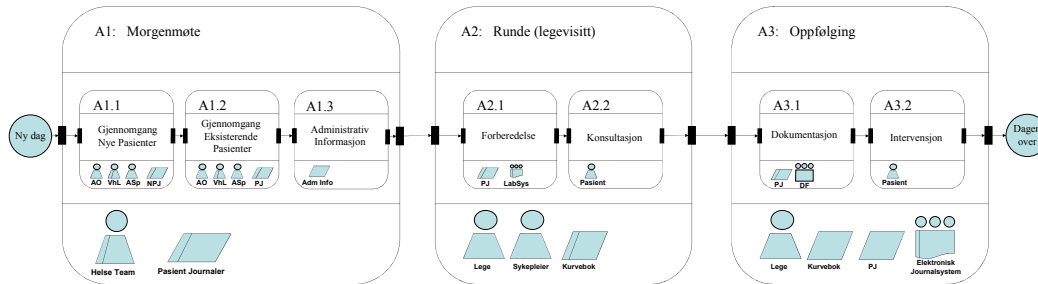


Løsningsforslag, eksamen i SIF8035 Informasjonssystemer, mai 2002

Oppgave 1 – Modellering (30%)



A1: Morgenmøte

En ny dag på avdelingen starter med at morgenmøtet avholdes. De viktigste ressursene er hele avdelingens *helseteam*, og *pasientjournalene* til alle innlagte pasienter. Med helseteam, mener vi alle pleiere og leger som er tilstede på avdelingen denne dagen. Dette er modellert som en *composite* ressurs, dvs. en gruppe-ressurs som kan dekomponeres på underliggende nivåer. Tilsvarende for pasientjournalene. Selve møterommet/kontrollrommet er ikke tatt med som ressurs – dette er kanskje en vurderingssak, men det er ikke i caset beskrevet noe med rommet som gjør dette spesielt viktig – bare at de har et sted og møtes.

A1.1: Gjennomgang av nye pasienter

Nye pasienter behandles spesielt nøye.

Ressurser:

AO: Avdelingens overlege

VhL: Alle vakthavende leger

ASp: Avdelingssykepleier

NPJ: PasientJournalene til alle nye pasienter

A1.2: Gjennomgang av eksisterende pasienter

Eksisterende pasienter kan gjøres unna litt fortere, med mindre det er spesielle hendelser eller notater. Samme aktør-ressurser som A1.1, men her med Eksisterende Pasienters Journaler (EPJ).

A1.3: Administrativ informasjon

Administrativ informasjon tas til slutt. Her har vi utelatt å spesifisere aktør-ressursene, vi nøyer oss med å arve HelseTeam fra den overordnede prosessen. Administrativ informasjon er modellert som informasjonsobjekt-ressurs.

A2: Runden (også kalt legevisitten)

Legene, sammen med sykepleierne forbereder og går på konsultasjonsrunde til alle pasientene. Når de går runden har de med seg *kurveboken* – dette er den eneste informasjonsressursen de har tilgjengelig under hele denne aktiviteten.

A2.1: Forberedelse

Lege og sykepleier går gjennom pasientjournalene for de aktuelle pasientene og forbereder dagens behandlingsplan, som overføres på kurveboken.

Ressurser:

PJ: PasientJournalen – papirutgave

EPJ: PasientJournalsystemet (elektronisk utgave) – litt andre data....

Labsystemer: composit ressurs som indikerer de ulike labinformasjonssystemer legen har tilgang til.

A2.2: Konsultasjon

Lege med hjelp av sykepleier undersøker pasienten. Her er ingen andre informasjonsressurser enn kurveboken (arves fra overordnet nivå). Konsultasjon kan dekomponeres videre ihht. legens arbeidsmåte (observasjon, diagnose, tiltak/intervensjon) – men dette er ikke spesifisert i case't.

A3: Oppfølging

Vel tilbake på kontoret, må legen dokumentere de funn og tiltak som er kommet til gjennom runden. Informasjonsressurser her er først og fremst kurveboken og pasientjournalen (papir), men legen er nå tilbake på kontoret sitt har tilgang til det elektroniske journalsystemet – med en rekke informasjonskilder.

A3.1: Dokumentasjon

En viktig oppgave for legen er å oppdatere informasjonsressursene – dvs. få dokumentert og begrunnet de funn som er gjort og evt. de tiltak som er iverksatt. Med utgangspunkt i kurveboken og den papirbaserte journalen, brukes en diktafon (DF) for å diktere nye tillegg til journalen. Denne kunne vært dekomponert videre for å få med spesialtilfeller slik som skrivning av epikrise (ved utskrivning) eller laging av spesielle notat (ved spesielle funn eller hendelser med pasienten).

A3.2: Intervensjon

Dette betegner den aktiviteten der legen "rekvirerer" tiltak på vegne av pasienten, dvs. skriver ut resepter, bestiller nye prøver, ordinerer medisiner etc. Dette må som oftest dokumenteres/bestilles på egne skjemaer – vi har her antatt at disse skjemaene er tilgjengelig elektronisk gjennom det elektroniske journalsystemet.

Oppgave 2 – Kravspesifikasjon (20%)

Kvaliteten på use case beskrivelsen ihht generelle retningslinjer for use cases

- positivt: oppsettet tilfredsstillende stort sett et template som er anbefalt i pensum, og mange av punktene er uttrykt om trent som man forventer i en use case. Selve handlingsflyten er også rimelig sammenhengende og forståelig (se dog kommentarer nedenfor).
- negativt: diverse feil / brudd på retningslinjer for hvordan use cases bør skrives, både mhp struktur og innhold. Nedenstående liste skulle dekke det aller meste, men utelukker ikke at det kan finnes enda noe å sette fingeren på.
 - *hvis-ledd direkte i basis-sti* (steg 7 og 8) er ikke anbefalt, da det gjør use cases tyngre å lese og forstå. Slikt bør enten flyttes til unntakstier eller komme fram som antagelser / prebetingelser. Hvis-leddet i steg 7 synes dessuten feilplassert, er mer naturlig som et unntak på steg 6 (dvs hvis det er kort tid siden forrige test, varsler systemet om dette før legen bes om å fylle inn bestillingen).
 - *lange og litt tunge beskrivelser* i flere av stegene, bl.a. pga islett av argumentasjon heller enn ren fokus på HVA bruker og system skal gjøre. Også noe uklart hvem som skal gjøre hva, f.eks. hvis-leddet i steg 7: er det systemet som varsler legen om at det er kort tid siden forrige test, eller er dette noe legen selv må vite på en eller annen måte? Eller alternativt steg 8, som blander to ting: utskrift av skjemaet (som er noe systemet kan gjøre) og sending av skjemaet med internpost (som det er mindre sannsynlig at systemet gjør).
 - *brukergrensesnittdesign foregripes* (f.eks. steg 3: ”taste inn pasientens fødselsnummer”, 5: ”taste inn en trebokstavs forkortelse”). Her burde formuleringene vært mer abstrakte, igjen for å få rent fokus på HVA som skal gjøres, heller enn å binde opp brukergrensesnittdesignet i antagelser om inntasting fra tastatur, trebokstavs forkortelser, osv. Til dels er det antatte brukergrensesnittet også temmelig dårlig, bl.a. at legen må fylle inn pasientens navn og adresse i steg 5 selv om dette burde kunne utledes automatisk hvis pasienten allerede er identifisert i steg 3.
 - *internt systemdesign foregripes* (steg 8) – i en use case skal det fokuseres på systemets respons ut mot brukeren, ikke hva systemet internt foretar seg (f.eks. hvis nettet er nede).
 - *manglende presisering av utfallet av unntakstier* (I steg 5, I steg 9: nevner kun unntaket, ikke hva som da skal gjøres)

- *dårlig kohesjon*, enkelte deler av beskrivelsen synes utenfor naturlig skop for ”Bestill lab-test”. Den påfølgende *utførelsen* av testen er ikke en del av bestillingen. Steg 9 (eller unntak steg 9) hører derfor ikke hjemme her, en mer fornuftig slutt ville vært at systemet fortalte brukeren at bestilling var utført. Likeledes kan man spørre om unntak steg 1 er innenfor skopet; hvis legen ikke har greid å autentisere seg, skulle man tro at han overhodet ikke er i stand til å starte opp funksjonen ”Bestill lab-test”. Feil passord (f.eks.) burde heller vært satt som et mulig unntak på en annen use case som har med selve innloggingen å gjøre.
- *feil i oppsettet*: To felter er kalt **Prebetingelser**, den siste av disse skal være **Postbetingelser**. Prebetingelser inneholder et spørsmål ”Hmm... hva er egentlig forskjellen på dette og antagelser?”, ikke en betingelse. Kandidatene behøver ikke besvare det spørsmålet som står her (dette er en del av den teksten som skal kritiseres), hvis i tilfelle noen likevel gjør det (og svarer riktig), teller dette selvsagt ikke negativt: Forskjellen er at prebetingelser er slikt som systemet selv er i stand til å kontrollere, mens antagelser er forhold som systemet ikke har kontroll over, men som likevel må være tilfredsstillt for at use casen skal kunne utføres normalt.
- annen fornuftig kritikk kan også komme som ikke nødvendigvis behøver å passe inn i de ovennevnte kategoriene, f.eks. at sending av bestilling pr epost kan ha dårlig sikkerhet (siden det kan dreie seg om sensitive personopplysninger)

Samsvar med casebeskrivelsen på side 4: use casen ”Bestill lab-test” burde for såvidt passe godt inn i denne. Utifra passasjen ”Når runden er ferdig går legen gjennom pasientinformasjonen på ny og sørger for at tiltakene dokumenteres og iverksettes. Her er det mye informasjon som skal oppdateres. Aktuelle prøver må bestilles og evt. endret medisinerings tilføyes.” ser man at bestilling av tester er en aktivitet som nevnes i caset, og use case beskrivelsen er således simpelthen på et langt mer detaljert nivå som ikke bryter med case beskrivelsen (i allfall hvis man ser bort ifra skop-feilene som diskutert ovenfor).

Samsvar med den tilhørende prosessmodellen kandidaten laget i oppgave 1: Dette kommer alt an på hva vedkommende gjorde i sin prosessmodell, derfor ikke lett å gi noe fasitsvar. Vurderingen må derfor gå på om det hver enkelt svarer virker fornuftig i forhold til det som ble gjort på oppgave 1. Et tilforlatelig svar vil være at use case fra oppgave 2 ikke er i motstrid med case-beskrivelsen fra oppgave 1, men at use casen beskriver i langt større detalj noe som bare var så vidt nevnt som en liten delaktivitet i oppgave 1. Kandidaten bør på dette punktet klare å vise til hvilken delprosess i sin løsning fra oppgave 1 som use casen kunne passe inn under.

Kategorisere i henhold til kvalitetsrammeverket. Selve rammeverket (behøver ikke forklares av studentene, kan antas kjent av den som svarer):

- syntaktisk kvalitet: om spesifikasjonen er i samsvar med det valgte spesifikasjonsspråkets syntaks
- semantisk kvalitet: om spesifikasjonen er i samsvar med det problemdomenet den prøver å forestille (dvs om modellen inneholder de ting som bør sies, og ikke inneholder ting som ikke bør sies)
- pragmatisk kvalitet: om ulike interessenter i prosjektet / lesere av spesifikasjonen er i stand til å forstå den

En klassifisering av feil ihht ovenstående er ikke like lett for use cases (hvor det meste er tillatt) som for mer formelle modelleringsspråk, fordi det kan herske tvil om hva som er ”syntaks” for use cases – er det kun rene språkfeil, eller også feil i oppsettet, hvis-setninger i basis-sti, osv.? Ulike klassifiseringer kan derfor godtas, bare kandidatene viser at de forstår forskjellen mellom kategoriene. Her er ett mulig forslag: To felt kalt ”Prebetingelser” kan

kalles en syntaksfeil, siden den går direkte på oppsettet (trenger ikke sammenholde beskrivelsen med ”verden” for å vurdere dette). Lange, tunge setninger og hvis-ledd i basis-sti kan kalles pragmatiske svakheter, siden dette vil redusere forståeligheten av spesifikasjonen (man kunne også ha kalt hvis-ledd i basis-sti en syntaks-feil hvis det var eksplisitt definert et begrenset antall setningsstrukturer som var lov å bruke i basis-stien, og hvor hvis-setninger ikke var blant disse). Ellers faller de fleste feilene i den semantiske kategorien, enten ved at spesifikasjonen inneholder ting den ikke burde ha inneholdt (design-antagelser, handlinger utenfor skopet til ”Bestill lab-test”), eller at det mangler noe som burde ha vært nevnt (presiseringer av hva som skal gjøres ved unntak – igjen kunne dette alternativt ha vært kalt syntaks-feil hvis det var eksplisitt definert at unntakstier skulle spesifiseres ved en setning om unntakssituasjonen + en setning om unntakshandlingen).

Oppgave 3 – Brukergrensesnitt - MMI (25%)

Denne oppgaven skal besvares i en egen tabell, angitt på egne svarark bakerst i oppgaven (Side 5-7). For å gi et inntrykk av hva slags svar vi ber om, har vi fylt ut første rad.

Merk at vi i denne raden har referert til feil delsystem, altså byttet om 1) og 2) :-)

Forslag:

Poenget er som nevnt å finne vesentlig trekk ved problemstillingen, som er relevant for hvert begrep, ikke gi en uttømmende beskrivelse. Det er forsåvidt fint om de forsøker å definere begrepet (formålet med aktiviteten) før de går inn på ulikheter. Det er viktigere med fornuftige resonnementer som viser at de kjenner pensum, enn god dekning av alle poenger jeg har kommet på.

Det er rimerlig å anta at jobb-brukerne (de som skal bruke 1) kjenner problemstillingen godt, har et ferdig begrepsapparat osv. Det ligger også i kortene at de (tror de) kjenner hjemmebrukerne godt og har en ide om hva som har skapt problemer for dem før.

Det er vanskelig å si noe generelt om hjemmebrukerne, bortsett fra at en kan anta at de identifiserer seg med ”senior”-begrepet og har forventninger om en tilpasset, dersom de velger å bruke den.

Brukskvalitet (eksempel)	
Funksjonalitet	<i>Siden det underliggende domenet er det samme, kan forvente at de underliggende funksjonene vil være de samme. Kanskje vil også funksjonaliteten til 1) kunne formuleres som sammensetninger av 2) sine funksjoner.</i>
Effektivitet	<i>Forskjellen i effektivitet for 1) og 2) vil være knyttet til hyppighet i bruk og antall utførelser av relevante oppgaver. Med sjelden og lite bruk vil effektivitet i 1) i større grad være knyttet til læring og antall feil som gjøres, mens det for 2) i større grad vil være høyest oppnåelig fart for eksperter og utmatting ved kontinuerlig bruk.</i>
Tilfredsstillelse	<i>For begge er tilfredsstillelse knyttet til det å bli tatt hensyn til som senior. Forskjellen ligger i at 1) er for hjemmebruk og 2) for jobben, og dette bør gjenspeiles i funksjonaliteten, effektivitet og verdiene som formidles forøvrig.</i>

Scenariebygging

Prosess	<p>Formålet med denne er å sikre at utviklerne har dannet seg et rikt og "riktig" bilde av brukerne og deres verden.</p> <p>For jobb-brukerne er scenariebyggingen, om ikke underordnet så mer forutsigbart, siden en kan regne med at de selv har bra oversikt over "sin verden". Det er riktignok viktig at utviklerne setter seg inn i denne og får bekreftet sin forståelse, men kanskje ikke så trolig at de vil oppdage ting jobb-brukerne selv ikke er bevisst.</p> <p>Hjemmebrukerne må kartlegges i større bredde og dybde, siden det er en mer ukjent og mindre homogen gruppe. Det er aktuelt med hjemmebesøk, kanskje gjennomgang av statistikk før en prøver å beskrive "typiske" brukere.</p>
Innhold	<p>Hvem er brukeren: aldersfordeling, nedsatte fysiske "evner" f.eks. dårlig syn, hørsel. Hvordan ser en typisk arbeidsplass ut i hjemmet/jobben. Hvordan brukere hjemmebrukeren banktjenester i dag, f.eks. sjekkhefte, bruk av skranke, brevgiro etc. Hva slags økonomi har de (mye/lite penger, lønn/pensjon, mange/få transaksjoner etc.) For jobb-brukeren er dagens system (det som er i banken(e) de sluttet i) og arbeidsplass viktig og evt. ny organisering/nye arbeidsomgivelser.</p>

Oppgaveanalyse	
Prosess	<p>Dette er en mer formalisert beskrivelse fokusert på dagens bruk av banktjenester hjemme og på jobb. For begge grupper er det en fordel å observere brukerne i deres naturlige element. Jobb-brukerne har i tillegg lover og instruksjoner som regulerer hva de gjør.</p>
Representasjon	<p>For jobb-brukeren er det mer trolig at en kan modellere vha. mer formelle metoder, spesielt siden en bør kunne fremskaffe en dyp domenemodell. Hjemmebrukerne har enklere oppgaver og varierende dypde i "domene"-modellen og antageligvis mindre forhold til mer formelle beskrivelser, så tekst og HTA er passe.</p>

Konseptuell design	
Metaforer	<p>Tanken er å basere grensesnittet på kjente begreper. F.eks. har sjekkhefte-metaforen vært vellykket for hjemmeøkonomi-applikasjoner. Giro'er og mappesystem, er kanskje aktuelle for hjemmebrukeren. For jobb-brukeren burde det være greit å legge seg opp til det de er vant med, og evt. utnytte at deres banktjenester er mer fokusert enn de gammel-banken deres kom fra. Metaforer er uansett mindre viktig når en bruker systemer hver dag.</p>
Overordnet struktur	<p>Jobb-brukerne tåler mer opplæring og kompleksitet siden de er profesjonelle og bruker systemet hver dag. Hjemmebrukerne kan en anta trenger trygghet, oversikt og enkelhet, og heller færre og enklere funksjoner enn mange.</p>

Konkret design

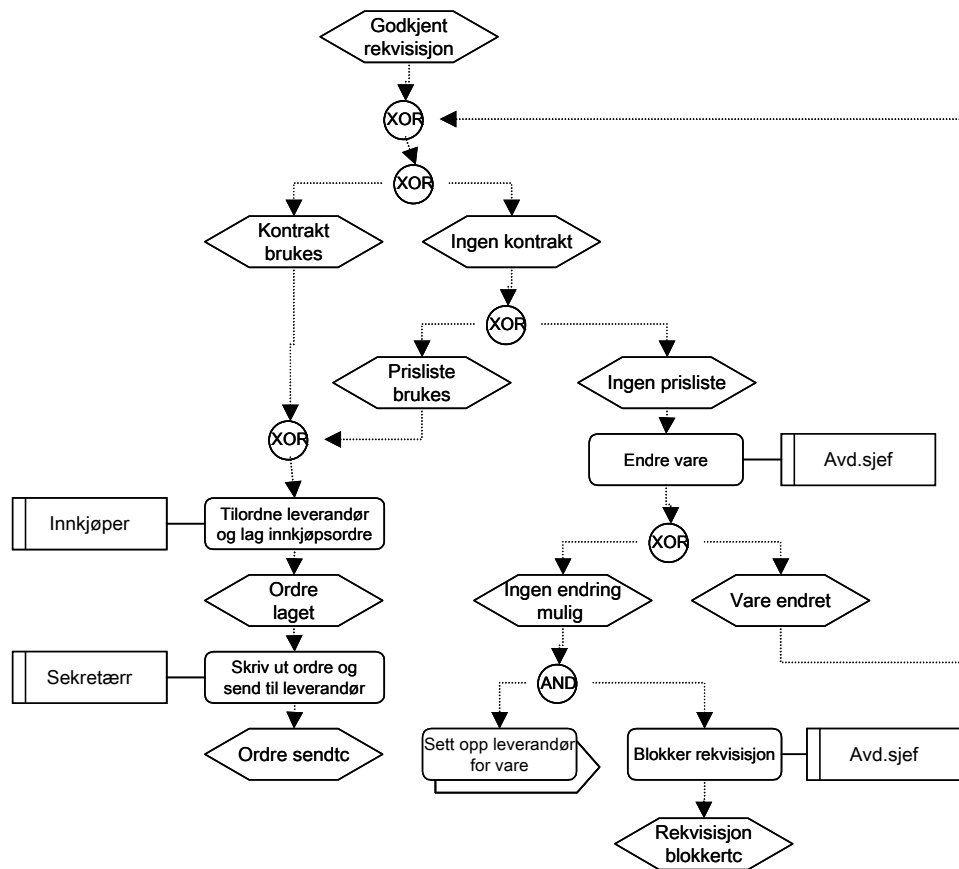
Interaksjonsstil	Her er plattformen sentral: Hva passer til typen applikasjon og plattformen. Windows-grensesnitt og skjemastil hører sammen, men hierarkiske views, lister, innfyllingsskjema etc. Tror ikke direkte manipulasjon og kommandostil passer. For browser-grensesnitt er mulighetene færre, en har kanskje et valg mellom mye bruk av forms, mange lenker, mye tekst i naturlig språk, forklaringer og bekreftelser etc.
Designkunnskap (f.eks. retningslinjer, standarder og regler)	Her kommer en inn på design ift. brukergruppens spesielle trekk, f.eks. skrift og farge, krav om finmotorikk, tydelig ift. spennende grafikk etc. Dette er mer generelt for begge gruppene, selv om hjemmebrukerne kanskje er mer sammensatt som gruppe enn jobb-brukerne. Jobb-brukerne er kanskje vant til bransjestandarder og konvensjoner fra transaksjonsorienterte applikasjoner.

Prototyping og evaluering:	
Type	Utgangspunktet er at det er færre ukjente for jobb-brukerne og at en derfor kan gå mer direkte på prototyper med høyere "fidelitet". Det er antageligvis enklere og viktigere å lage funksjonelle vertikale prototyper siden de er nærmere utviklerne og har mer innsyn i prosessen, og mer fokus er på effektivitet. Hjemmebrukerne vil antageligvis trenge større bredde i typen prototyper, fra papir til mer funksjonelle, lav til høy fidelitet, etc.
Hypighet	For jobb-brukerne kan en kanskje nøye seg med en oppnevnt referansegruppe, som kan møtes ofte, og evaluere i større bredde etterhvert. En bør ihvertfall kunne anta at de er motiverte for dette. For hjemmebrukerne kreves det mere for å samle dem, det betyr færre og mer omfattende brukertester, mer eksperimentaktig. Jobb-brukerne kan brukes til testing av testene, siden de har noen av de samme karakteristika som hjemme-brukerne og kanskje kjenner denne gruppen.

OPPGAVE 4 – ERP (25%)

Svar 4a

Modell av innkjøpsprosessen til DIB:



Svar 4b

Følgende typer tabeller brukes i SAP R/3:

- *Systemadministrasjonstabeller:* Dette er tabeller som brukes til å administrere hele SAP-systemet, inkludert diverse arbeidsbenker og data som deles av alle klienter. Eks. valutatarater og måleenheter, ABAP/4-programmer
- *Kontrolltabeller:* Kontrolltabeller brukes til å konfigurere systemet til brukernes behov. Eks. toleransegrenser for prisavvik mellom innkjøpsordre og faktura
- *Masterdatatabeller:* Dette er data som gjenbrukes i transaksjoner og typisk knyttes til visse organisatoriske enheter. Eks. kundedata og leverandørdato
- *Transaksjonsdatatabeller:* Disse inneholder dokumentene/dataene som produseres når transaksjoner kjøres. Eks. innkjøpsordre.