Prosjektnavn

Kravdokument

Revidert for faget Objektorientert programmering med systemarbeid

For Navn på systemet

Versjon <1.0>

I kravdokumentasjonen skal du beskrive funksjonelle krav til systemet som er laget. Begrunnelsene for valgene som er tatt hører generelt hjemme i Sluttrapporten.

Revisjonshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Dato** | **Versjon** | **Beskrivelse av endring** | **Forfatter** |
| <dd/mmm/yy> | <x.x> | <beskrivelse> | <navn> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Innhold

1. Innledning 4

1.1 Dokumentets hensikt 4

1.2 Avgrensning 4

1.3 Definisjoner og forkortelser 4

1.4 Referanser 4

1.5 Oversikt over innholdet 4

2. Bakgrunn og oversikt 4

2.1 Use Case UML diagram 4

2.2 Forutsetninger og avhengigheter 4

3. Detaljerte krav 5

3.1 Use Case tekstlig beskrivelse 5

3.2 Tilleggsspesifikasjon av krav (Supplementary Specification) 5

4. Domenemodell 5

5. EER-modell 6

6. Relasjonsskjema 7

7. Tilleggsinformasjon 7

8. Oversikt over brukergrensesnitt (skjermbilder) 7

Kravdokument

# Innledning

Innledningen til dokumentet skal fortelle leseren hva som er hensikten med dokumentet og hva det inneholder. Han/hun skal også forstå i hvilken forbindelse dette dokumentet er skrevet. Innledningen har følgende underpunkter:

## Dokumentets hensikt

Hensikten med dette dokumentet er å gi en logisk beskrivelse av det systemet som skal utvikles. Beskrivelsen skal vise hvilke krav brukerne har til systemet.

## Avgrensning

Her finner vi en kortfattet beskrivelse av det aktuelle prosjektet slik at det går klart frem hva dette dokumentet dekker. Hvis dokumentet dekker deler av det fremtidige systemet må det gå klart fram. Eventuelle relasjoner til andre systemer eller delsystemer beskrives.

## Definisjoner og forkortelser

Alle begreper og forkortelser må defineres når dette er nødvendig for å forstå innholdet i dokumentet.   
NB! Husk at dette dokumentet skal leses også av brukerne.

Vurder å veldikeholde en *ordbok* i prosjektet. Dette bør da være et eget dokument som refereres her.

## Referanser

Alle dokumenter som refereres andre steder i dette dokumentet skal listes opp her.

## Oversikt over innholdet

Her forteller du leserne hva dette dokumentet inneholder og hvordan det er organisert (hva han finner hvor)

# Bakgrunn og oversikt

Her skal du beskrive alt som er viktig å vite om bakgrunnen for de kravene som dokumenteres i resten av dokumentet. Du tar med informasjon om brukerne, om rammebetingelser, om forutsetninger og om av­hengigheter. Hensikten er å gjøre det lettere å forstå brukerkravene og sette dem inn i den rette konteksten. Hensikten er også å kunne vurdere brukerkravene på nytt om noe av det som beskrives her endrer seg.

## Use Case UML diagram

Her finner du det komplette Use Case UML diagrammet for den delen av systemet som dokumentet dekker. Det vil si alle aktører og alle Use Case og kommunikasjon mellom aktører og Use Case i diagramform. Det kan være ett eller flere diagrammer, avhengig av hva du finner hensiktsmessig.   
NB! Use Case tekstlig beskrivelse kommer senere i dokumentet. Husk å ta med en innledning som forklarer hva dette diagrammet viser og konsistensen med andre deler i dokumentet. Vurder om aktører og Use Case må defineres eller forklares.

## Forutsetninger og avhengigheter

Her beskriver du så nøyaktig som mulig de forutsetningene som beskrivelsene i dette dokumentet bygger på. Videre beskriver du så nøyaktig som mulig hvilke avhengigheter som eksisterer, for eksempel til andre systemer.

# Detaljerte krav

Her skal du beskrive alle brukerkravene til systemet. Beskrivelsene skal være så detaljerte at du på bak­grunn av dem kan utvikle systemet. Beskrivelsene skal også senere være utgangspunkt for testing, slik at du gjennom testing kan vurdere om systemet oppfyller brukernes krav.   
NB! Dette står ikke i motsetning til at både beskrivelsen av brukerkravene og utvikling av systemet skjer inkrementelt og i iterasjon­er.

Beskrivelsen består av følgende deler:

## Use Case tekstlig beskrivelse

For hver av Use Case-ene i Use Case diagrammet skal det utformes en Use Case tekstlig beskrivelse. De ikke-funksjonelle kravene som naturlig hører til en Use Case, beskrives også her. Husk å ta med en kort innledning som forklarer hva beskrivelsene gjelder og beskriv sammenhengen mellom disse beskrivelsene og andre deler av dokumentet.

## Tilleggsspesifikasjon av krav (Supplementary Specification)

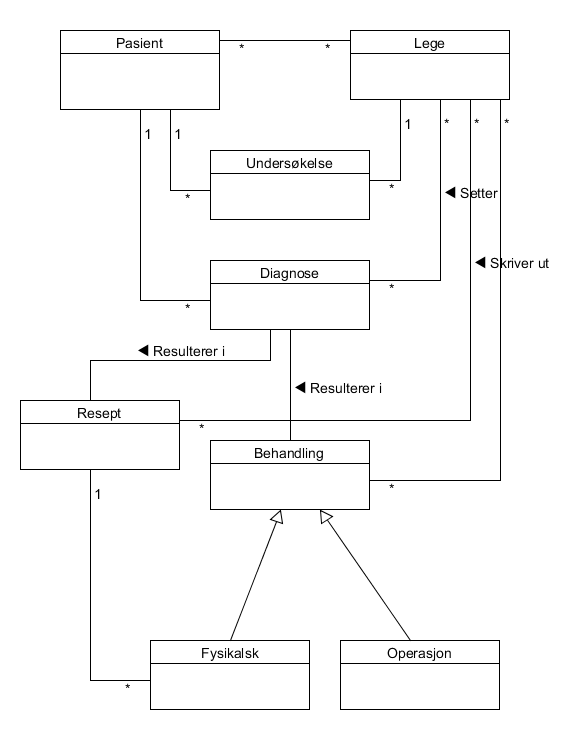
Alle krav som ikke beskrives gjennom en Use Case skal beskrives her. Det vil være alle ikke-funksjonelle krav som gjelder hele systemet. Det kan også være funksjonelle krav som ikke naturlig beskrives som Use Case. Hvis det er mange slike krav kan det være naturlig å ha tilleggsspesifikasjonen som et eget dokument, det refereres da her.

# Domenemodell

Domenemodellen er en viktig del av kravdokumentet i UP. Domenemodellens funksjon er å få utviklere og domeneeksperter til å enes om en felles forståelse av problemdomenet. Utviklerne må forstå domenet de skal jobbe med og alle involverte må snakke om det samme når man bruker begreper fra domenet. Dette er viktig for å unngå misforståelser underveis.

Bruk domenemodell hvis prosjektet opererer i et ikke-trivielt problemdomene og/eller for å oppnå best mulig kommunikasjon med oppdragsgiver.

Under er et eksempel på en domenemodell for domenet Pasient, lege og undersøkelse. En pasient kan behandles av x-antall leger. En doktor kan behandle n-antall pasienter.



# EER-modell

Tegn et EER-diagram (utvidet ER-modell) som viser hvilke data som skal lagres i systemet, dvs. entitets­typer med attributter og sammenhengstyper mellom entitetene som viser intensjonen og ekstensjonen.

Ha med en kort tekstlig beskrivelse av EER-diagrammet inkl. evt. forutsetninger som er gjort.

# Relasjonsskjema

EER-modellen i pkt. 5 skal oversettes til et relasjonsskjema på angitt format og som dokumenterer entitets- og referanseintegriteten.

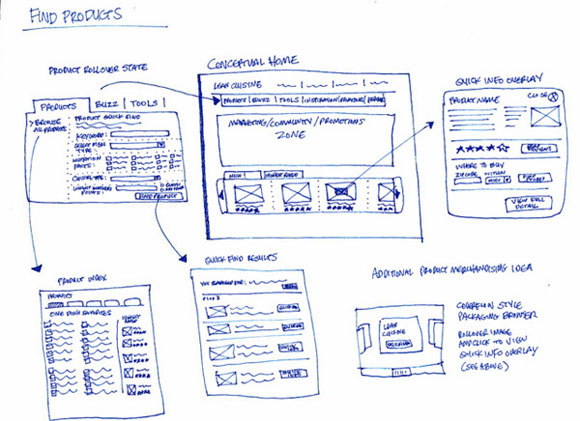
# Tilleggsinformasjon

Evt. annen informasjon som vi ønsker å ta med for å øke forståeligheten og tilgjengeligheten av det som er dokumentert.

# Oversikt over brukergrensesnitt (skjermbilder)

Hvis du utvikler en applikasjon hvor grensesnittet er en viktig del bør du vurdere å bruke en eller annen form for prototyping for å beskrive krav. En mulige måte å gjøre dette på er å bruke Wireframes.

Wireframes er enten frihåndstegninger eller skisser laget av et av mange wireframe-verktøy som finnes på nett. Med nettbaserte wireframe-verktøy kan man lage klikkbare wire-frames som gjør at man også kan navigere mellom skjermbildene. Dette gjør dem godt egnet til å brukes i brukertester.



Hvis du har laget wireframes for hånd kan du lime de inn her. Hvis du har brukt online-verktøy kan du eventuelt vise lenker til dem.

Ha med utskrift (kopi) av alle skjermbildene (men ingen programkode) slik at oppdragsgiver og andre interessenter kan få tilgang til og vurdere brukergrensesnittet.

For hvert skjermbilde skal det være en tekstlig beskrivelse av hvilke funksjonalitet skjermbildet implementerer. Funksjonaliteten bør knyttes direkte opp mot Use Case-ene beskrevet tidligere i dokumentet. Ett skjermbilde kan knyttes til ett eller flere Use Case, og ett Use Case kan knyttes til ett eller flere skjermbilder.