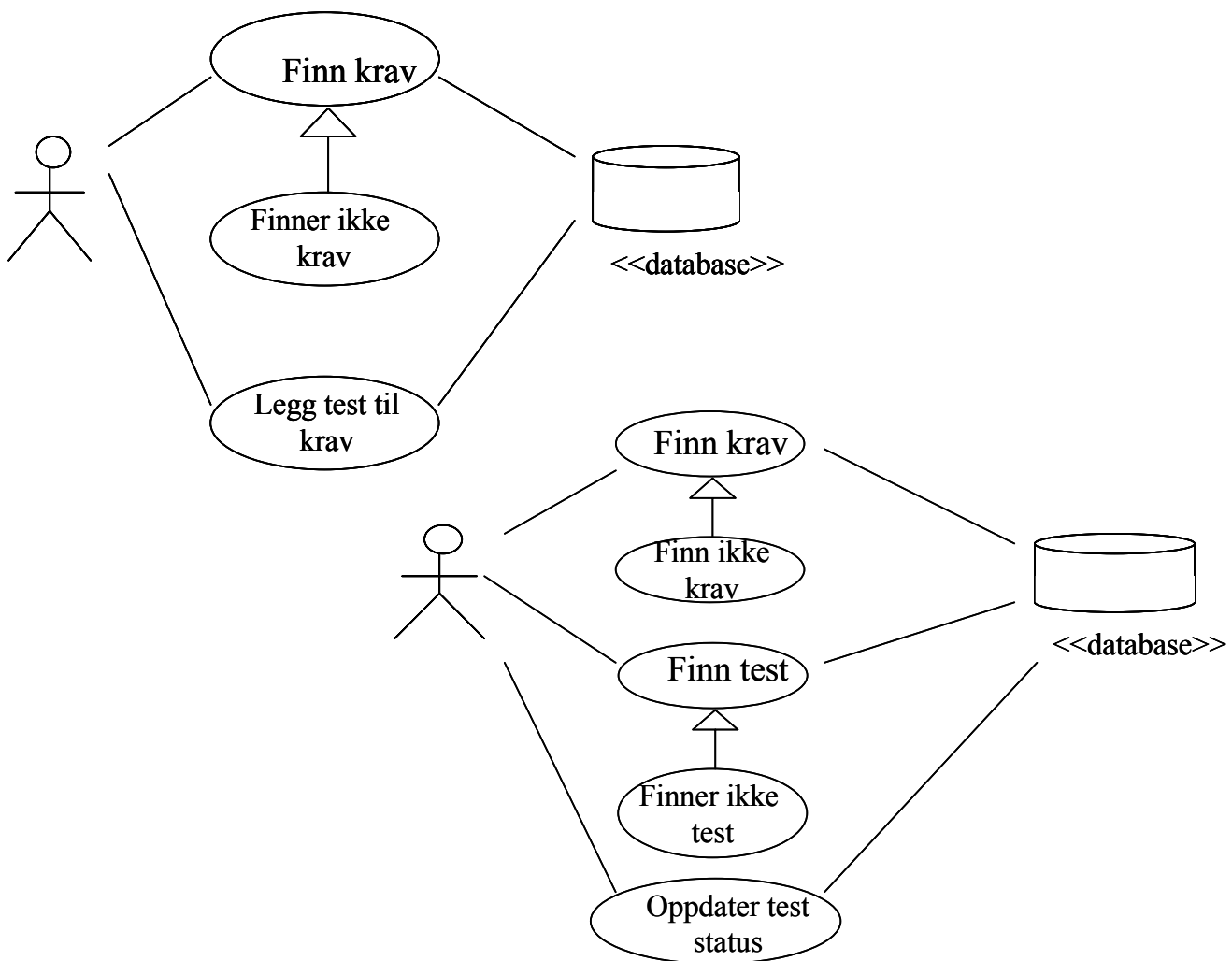


Oppgave 1

1. Hensikten med use case er å oppnå en felles forståelse av krav til systemet mellom brukere / kunder og utviklere. Et use case er et scenario, ikke en komplett, detaljert kravspesifikasjon. Viktige momenter:
 - Kundene må delta
 - Fokus på kundens behov – hva systemet skal gjøre, ikke hvordan
 - Må detaljeres ned til et nivå der kunden og utvikleren har en felles forståelse av hva funksjonen skal gjøre
 - Use case er ikke design – ingen designbeslutninger
2. Use case for "Legg inn test til et kundekrav" og "Oppdater status for en test".



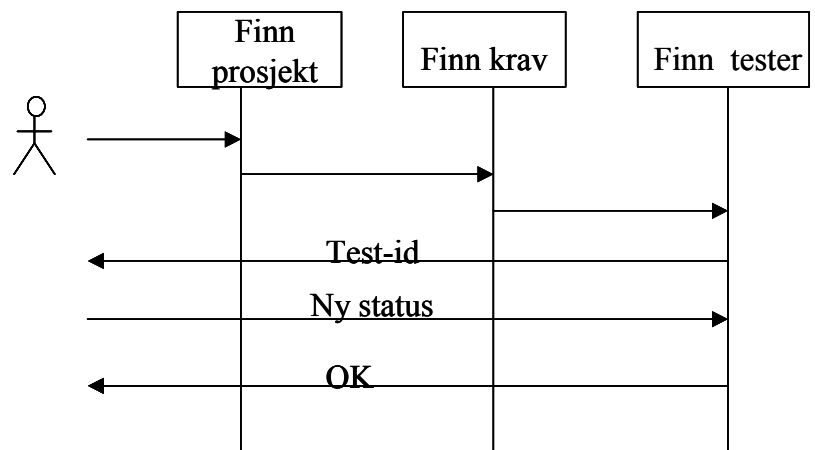
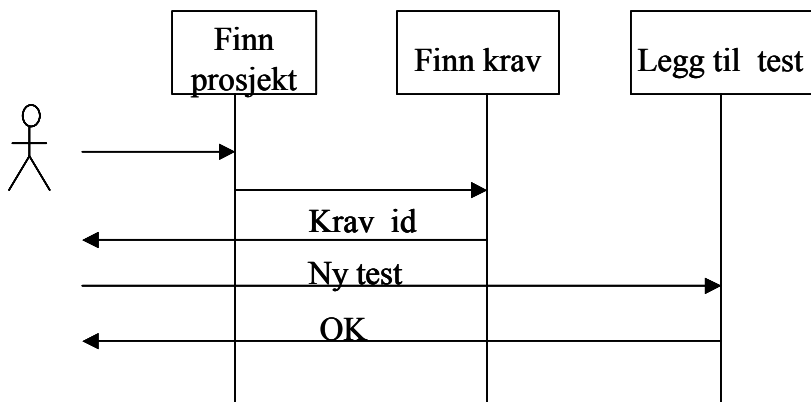
3. Tekstlige use case for ”Legg inn et kundekrav” og ”legg inn forventet resultat for en test”.

Tittel: Legg inn et kundekrav til et prosjekt
Aksjoner: 1. Finn prosjektet 2. Legg inn et kundekrav
Avvik: 1.1 Finner ikke prosjekt 1.2 Gi feilmelding 1.3 Avslutt

Tittel: Legg inn forventet resultat for en test
Aksjoner: 1. Finn prosjekt 2. Finn test 3. Legg til forventet testresultat
Avvik: 1.1 Finner ikke prosjekt 1.2 Gi feilmelding 1.3 Avslutt 2.1 Finner ikke test 2.2 Gi feilmelding 2.3 Avslutt

Det finnes mange måter å lage tekstlige use case på. De skiller seg fra hverandre ved bruk av ulike tabellformater og nøkkelord. Dette er imidlertid ikke viktig for besvarelsen. Det som derimot er viktig er at use caseen skal vise sekvenser – normaltilfellene – og hvordan man behandler unntak fra normaltilfellene.

4. Diagramform – interaksjon med brukerne
Tekstform – overgang til MSC og klassediagrammer. Viser rekkefølgen i interaksjonene.
5. Det er flere måter å lage sekvensdiagrammene på. I det som er vist under er det prosjektet som henter kravene og testene – viss disse er tilgjengelige. Alternativet er å la brukeren hente all denne informasjonen i separate trinn.



6. MSC'er viser hvordan systemdeler – ikke nødvendigvis klasser – kommuniserer med hverandre, utveksler informasjon osv. Dette kommer i tillegg til å vise scenarier – som finnes både i diagrammer og i tekstlige use case – og sekvensering – som finnes i tekstlige use case.

Oppgave 2

1. Vi trenger følgende arbeidspakker i tillegg:
 - a. Prosjektledelse og prosjektplan
 - b. Systemdesign
 - c. Databasedesign
 - d. Subsystemdesign - hovedfunksjoner
 - e. Integrasjon for hver hovedfunksjon
 - f. Systemintegrasjon
 - g. Testing av hovedfunksjoner
 - h. Systemtest, inklusiv brukertest
 - i. Prosjektrisikoplan

Mange hadde problemer med å forstå dette spørsmålet. I ettertankens bleke skjær ser jeg at det ikke er så lett å forstå hva slags svar jeg var på jakt etter. To viktige momenter:

- *Studentene skal ikke trekkes for manglende arbeidspakker både i spørsmål 2.1, 2.2 og 2.3 – de skal ikke straffes flere ganger for manglen.*
 - *Vi må diskutere hva vi skal gjøre med de som har misforstått,, ikke svart eller kommet med "på jorde"-svar til spørsmål 2.1.*
2. Strengt tatt er det bare nødvendig å ha med signifikante avhengigheter i tabellen. Viss f.eks. aktivitet 6 er avhengig av aktivitetene b, d, 1, 2, 3, 4, 5 holder det med å registrere avhengigheten til aktivitet 4 og 5. Disse er allerede avhengig av aktivitetene b, d, 1, 2, 3.

Nr.	Arbeidspakke	Avhengig av
1	Legg inn prosjekt	d
2	Legg inn kundekrav til et prosjekt	1
3	Legg inn en test til et kundekrav	2
4	Legg inn forventet resultat for en test	3
5	Oppdater status for en test	3
6	A	3
7	B	A (6)
8	C	A (6)
9	D	3
10	E	3
11	F	5
12	G	5
a	Prosjektledelse	Ingen
b	Systemdesign	Ingen
c	Databasesdesign	Ingen
d	Subsystemdesign	c
e1 – e4	Integrasjon for hver hovedfunksjon	Alle arbeidspakker som inngår i denne funksjonen
f	Systemintegrasjon	b, d, 1 – 17
g1 – g4	Testing av hver hovedfunksjon	b, d, Alle arbeidspakker som inngår i denne funksjonen
h	Systemtest, inklusive brukertest	b, d, 1 – 17, f
i	Prosjektrisikoenalyse	a, b, c, d

Legg merke til at det er en trykkfeil i oppgaven. Det står arbeidspakkene 1 – 4 i stedet for 1 – 5. Pass på dette når du retter.

3. Gantt-diagram. Pass på at det bare er fem personer tilgjengelig inklusive PM.

Aktivitet	1	2	3	4	5	6	7	8	9	10	11
a	1	1	1	1	1	1	1	1	1	1	1
b	2										
c	2										
d		2									
i			2								
1			2								

Oppgave 3

1. Viser klassene i systemet samt klasseattributter og metoder.
2. Valg av metode for å lage klassediagrammer.
Siden vi har et veldefinert sett av krav bør man velge OMT. Det viktigste for kvaliteten av svaret er imidlertid argumentasjonen for å velge den metoden studenten har valgt.
3. Klasse for testinfo.

Klasse: Testinfo
Viktige attributter: <ul style="list-style-type: none">• Testid• Hvilke(t) krav tester denne testen• Ansvarlig for denne testen• Status på testen – f.eks. definert men ennå ikke kjørt, kjørt med feil, kjørt OK
Viktige metoder <ul style="list-style-type: none">• Kobl test til krav• Oppdater teststatus

Det er mange måter å beskrive en klasse, attributter og metoder på. Det overstående er nbare et eksempel. Represnetasjonen er ikke viktig for krakteren.

4. Tilstandsdiagram
Viktige tilstandsvariable er ”koblet til krav” og ”tilstand for kjøring av test”.
Tilstandsdiagrammet fokusere på disse to.
5. Pakkediagram

Oppgave 4

1. Brukervennlighet: Hvor lett er det å lære å bruke systemet og hvor lett er det å få gjort de oppgaven mna skal bruke systemet til
Robusthet: Systeemt skal tåle feil input, manglenmde input og temporære feil i oprogram,vare og maskinvare uten å miste inforasjon som ellere er lasgt inn. Det skal ikke være mulig for en vanlig bruker å få systemet til å ”krasje”.
2. Test for brukervennlighet. Velg ut et sett av brukere fra kunden. Definer:
 - Hva slags opplæring brukere skal ha
 - Definer et sett av oppgaver som er relvante for brukerne – tar utgangspunkt i deres daglige arbeid
 - Definer akseptanskriterier – f.eks. minst 90% må si de er fornøyde, ingen bruker mer enn 10 minutter på å gjøre jobben eller liknenede.
3. Test for robusthet. Definer et sett med feil / gale data og systemtilstander der systemet ikke vil kunne bruke data som ellers er ”riktige”. Systemet skal kunne få disse dataene, gi feilmelding og deretter kunne akseptere riktige data i riktig tilstand. Det er viktig å på forhånd tenke på hva mna må passe spesilet på – det er ikke alle typer feil det er like viktige å kunne tolerere.

For svar på spørsmålene 4.2 og 4.3 er det viktigste at de har med testbeskrivelse og hva osm må skje når de kjører testen for at de skal godkjenne resultatet.

Vedlegg A

Krav for verktøy til testadministrasjon

- **Definer test**
 1. Legg inn et prosjekt
 2. Legg inn kundekrav til et prosjekt
 3. Legg inn en test til et kundekrav
 4. Legg inn forventet resultat for en test
 5. Oppdater status for en test
- **Utfør risikoanalyse av testplanen**
 6. A
 7. B
 8. C
- **Legg tidsplan for testing**
 9. D
 10. E
- **Vis status for testingen**
 11. F
 12. G

Avhengigheter:

Nr.	Aktivitet	Avhengig av:
1	Legg inn et prosjekt	
2	Legg inn kundekrav til et prosjekt	
3	Legg inn en test til et kundekrav	
4	Legg inn forventet resultat for test	
5	Oppdater status for en test	
6	A	3
7	B	A
8	C	A
9	D	3
10	E	3
11	F	5
12	G	5