

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for fysikk,
informatikk og matematikk

BOKMÅL

Institutt for datateknikk
og informasjonsvitenskap



Sensurfrist: 22. juni

Eksamen i fag
SIF8018 Systemutvikling

Fredag 25. mai 2001
kl 0900 - 1400

Hjelpemidler A1:

Kalkulator ikke tillatt

Ingen trykte eller håndskrevne hjelpemidler tillatt:

Faglig kontakt under eksamen:

Professor Tor Stålhane, tlf. 94484

Prosentsetser viser hvor mye hver oppgave teller innen settet. Innen en oppgave teller de ulike deloppgaver likt, med mindre annet er angitt.

Lykke til!

Innledning

Vi vil i denne oppgaven flere ganger referere til ”systemet”. Dette er et system for å handtere eksamensavviklingen ved NTNU – nærmere bestemt den delen som gjelder registrering og sensurering av eksamensoppgaver. Systemet består av to hoveddeler – en mobil del som finnes i hvert eksamenslokale og en stasjonær del som finnes på NTNU – sentralsystemet. Her ligger også en database med alle nødvendige data.

Den mobile delen benyttes av inspektørene til å registrere oppmøte i eksamenslokalet og innlevering av ferdig oppgave. Den stasjonære delen benyttes av faglærer og sensor for å registrere karakterer, av studieadministrasjonen for å lage eksamensutskrifter og av den enkelte student for å sjekke resultatene av eksamen i hvert enkelt fag.

Den mobile delen av systemet skal være knyttet til den stasjonære delen via det vanlige mobiltelefonnettet.

Generelt for hele eksamen gjelder at viss du har bruk for informasjon du ikke finner i oppgaveteksten så skal du:

- Dokumentere de nødvendige antakelsene.
- Forklare hvorfor du trenger de.

Oppgave 1 – UML (30%)

1. Lag use case diagram for det systemet som er beskrevet i innledningen.
Se figur 1. Legg merke til at figuren også inneholder eksamenskontoret (E), selv om dette ikke er en del av oppgaven.
2. Lag tekstlige use case for følgende situasjoner – pass på å få med eventuelle varianter:
 - Inspektøren registrerer fram møte i eksamenslokalet.
 - Faglærer legger inn resultatene for et fag.
 - En student sjekker alle sine karakterer for et semester.

<i>Trigger: Ny student ankommer eksamenslokalet</i>
<ol style="list-style-type: none">1. <i>Studenten viser fram studentbevis</i>2. <i>Inspektøren sjekker i studentdatabasen for faget at</i><ul style="list-style-type: none">• <i>Studenten finnes</i>• <i>Studenten er oppmeldt til eksamen i faget</i>3. <i>Inspektøren markerer i databasen at studenten har møtt fram til eksamen i dette faget.</i>4. <i>Systemet sender kvittering - OK</i>
<i>Alternativ: Finner ikke studenten i studentdatabasen</i> <i>2a: Studenten undertegner nødvendige dokumenter for å få midlertidig adgang til eksamen.</i>
<i>Alternativ: Studenten er ikke oppmeldt til eksamen i faget</i> <i>2b: Studenten undertegner nødvendige dokumenter for å få midlertidig adgang til eksamen.</i> <i>Forstsett med trinn 3</i>

Alternativ: Systemet sender ikke kvittering – OK
4a: Overgang til manuelt system.
4b: Inspektør sender melding til driftsavdeling

Trigger: Faglærer skal legge inn resultatene for et fag

- 1. Faglærer finner studenten i studentdatabasen ut fra stud. nummer**
- 2. Faglærer sjekker karakter og studentnummer mot sensors lister for å se at de er enige**
- 3. Faglærer legger inn karakteren til studenten i studentdatabasen**
- 4. Systemet sender kvittering - OK**

Alternativ: Finner ikke studenten i studentdatabasen
1a: Studentnummer lagres i reserveområdet for senere behandling.
Forsett med trinn 2

Alternativ: Systemet sender ikke kvittering – OK
4a: Faglærer sender melding til driftsavdelingen
4b: Avslutt prosessen

Trigger: Student vil sjekke karakterene sine for dette semesteret

- 1. Studenten finner sitt område i studentdatabasen ut fra studentnummeret**
- 2. Studenten leser / sjekker karakterene**
- 3. Avslutt prosessen**

Alternativ: Finner ikke sitt område
1a: Studenten sender melding til driftsavdelingen
1b: Avslutt prosessen

Alternativ: Finner ikke ett eller flere fag:
2a: Studenten sender melding til eksamenskontoret
2b: Avslutt prosessen

- 3. Når egner diagramformen for use case seg best og når egner tekstformen av use case seg best?**

Diagram: Egner seg best for å kommunisere og diskutere med brukerne, særlig på et tidlig stadium i utviklingsprosessen. Nyttig for å identifisere hvilke funksjoner systemet skal tilby brukerne

Tekst: Egner seg best når vi utarbeide en komplett beskrivelse av hver enkelt funksjon – ofte sammen med kundens eksperter - og skal bruke use-case til å gå over til ett eller flere sekvensdiagram

- 4. Lag sekvensdiagrammer for de situasjonene som er beskrevet i ovenstående punktliste.**
Se figur 2, 3 og 4.
- 5. Lag konseptuelle klassediagram for systemet, basert på use case og sekvensdiagrammene.**
Se figur 5

Oppgave 2 – Arkitektur (35%)

Systemet skal realiseres som en tre-lags modell i en klient-tjener arkitektur.

1. Forklar hva en klient-tjener arkitektur er

En klient-tjener arkitektur har følgende komponenter:

- En eller flere servere som yter tjenester til andre subsystemer.*
- Et sett av klienter som kan bruke de tjenestene som tjenerne tilbyr*
- Et nettverk som forbinder tjenerne og klientene.*

2. Forklar forskjellen på en tynn og en tykk klient.

Tynn klient: All prosessering i applikasjonen og all dataadministrasjon foregår på tjeneren.

Tykk klient: Tjeneren er bare ansvarlig for dataadministrasjon.

3. Diskuter fordeler og ulemper ved de to løsningene i punkt 2.

Tykk klient:

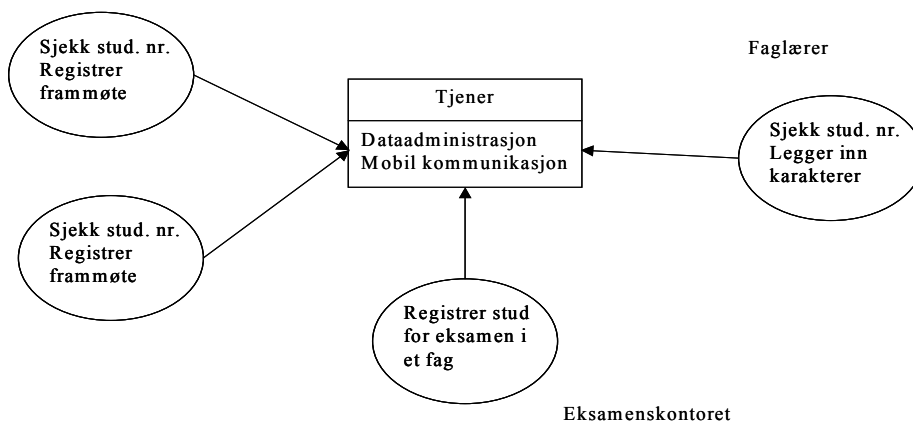
- Fordeler: når vi har en PC el. Som klient vil vi få maksimalt ut av den prosesseringskapasiteten som finnes her. Mindre trafikk på nettet.*
- Ulemper: mer kompleks logikk siden det er flere prosesser som manipulerer felles data, vanskeligere å oppnå høy sikkerhet.*

Tynn klient:

- Fordeler: passer godt når vi skal legge moderne brukergrensesnitt på en legacy applikasjon, kan brukes når vi bare har en terminal og ikke en full PC på klientsida.*
- Ulemper: fører til mye arbeid på tjenersida og mye trafikk på nettet. Dersom klienten er en PC el. Vil vi bare bruke en liten del av de mulighetene den tilbyr.*

4. Lag en klient-tjener arkitektur for systemet og forklar grunnene til de valgene du gjør underveis.

Eksamenslokalene



5. ”Noen i NTNU” har uttrykt engstelse for at trøbbel med mobiltelefonnettet kan skape problemer for eksamensavviklingen. Selv om sannsynligheten er liten, er konsekvensene så store at man velger å ta hensyn til problemet når man lager systemet. Hvordan kan man redusere faren for at trøbbel med mobiltelefonnettet lager problemer for eksamensavviklingen?

Ved å legge den delen av databasen som angår de studentene som skal være der over på

det mobile systemet før eksamen vil man være uavhengig av mobiltelefonnettet. Alternativt kan man sørge for å ha det manuelle opplegget inntakt, men dette er en dårligere løsning

Oppgave 3 – Planlegging (35%)

Ut fra use-case analysen har vi estimert kostnad og varighet for følgende arbeidspakker:

Nr.	Tittel	Kostnad - ukeverk	Varighet - uker	Før-aktiviteter
1	A	12	3	-
2	B	5	3	A
3	C	12	7	A
4	D	7	4	A
5	Test av sentralsystem – E	4	1	B, C
6	F	3	5	C, D
7	Test av delsystem for eksamenslokalet – G	4	8	F, E
8	Integrasjon – H	2	1	E, G
12	Test av hele systemet - I	4	3	H

Utviklingsprosjektet skal starte 1. juni, 2001. For å kunne teste deler av systemet i realistiske omgivelser skal en del av systemet kjøres parallelt med de manuelle rutinene til kontinuasjonseksamen i august samme år - prøveinstallasjon. Hele systemet skal være klart til eksamen i høstsemesteret 2001. Bruk de arbeidspakkene, kostnadene, avhengigheter og gjennomføringstidene som er gitt i tabellen over til å lage:

1. Risikoanalyse.

Det viktigste er å få fram risikotabellen og velge en fornuftig måte å evaluere konsekvens, risiko og sannsynlighet på. Eksempler på mulige måter er Høy, Middels og Lav eller gradering fra 1 til 10. Det er ikke så viktig å ha de "riktige risikofaktorene" med.

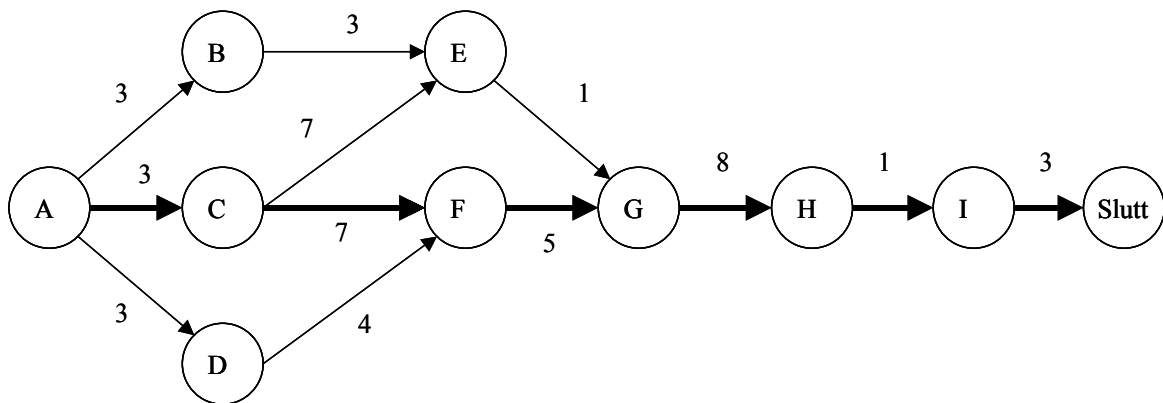
<i>Hendelse</i>	<i>Sannsynlighet</i>	<i>Konsekvens</i>	<i>Risiko</i>	<i>Tiltak</i>	<i>Ansvarlig</i>

2. Gantt-diagram.

Det som er helt svart er aktivitetene. De lysegrå feltene viser slakk på aktivitetene B, D og E. Dette behøver ikke være med, men er et ekstra pluss viss det er tatt med. Legg merke til at med denne planen ligger starten på høsteksamen (kontinuasjoneksamen) i uke 11.

Akt.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
A	■	■	■																									
B				■	■	■	■	■	■	■																		
C				■	■	■	■	■	■	■																		
D				■	■	■	■	■	■	■	■	■																
E											■	■	■	■	■													
F											■	■	■	■	■	■												
G																	■	■	■	■	■	■	■	■	■	■	■	■
H																												
I																												

3. Aktivitetsdiagram – inklusive kritisk vei.



Kritisk vei er markert med tykke piler.

Det er noen som mener at en del av estimatene er vel optimistiske. For å være på den sikre siden skal det derfor lages en prioritetsliste over funksjonene basert på den funksjonelle beskrivelsen i innledningen til denne oppgaven.

Planen slik den er her gjør at det ikke er mulig å komme i mål med verken sentralsystem eller mobilsystem – systemet for eksamenslokalet. Hovedhensikten med besvarelsen må være å definere et sett av funksjonalitet som det er mulig å teste i realistiske omgivelser. Det er relativt mange kombinasjoner som kan settes dammen til testing så det er ikke viktig hvilken av disse studenten velger i sin besvarelse.

4. Lag en prioritetsliste for funksjonaliteten til systemet. Listen skal beskrive hvilke funksjoner som må være til stede i:

- Testversjonen – i forbindelse med kontinuasjoneksamen i august, 2001

Følgende må være til stede:

Studentdatabase med studentnummer og informasjon om de er oppmeldt til eksamen eller ikke for hvert enkelt fag

Nødvendig programvare for kommunikasjon med mobil enhet i eksamenslokalet.

Programvare på mobil enhet som kan sjekke at studenten er oppmeldt og helst også registrere at han har møtt opp.

- Første versjon som er i drift – i forbindelse med høsteksamen 2001.

Følgende må være til stede:

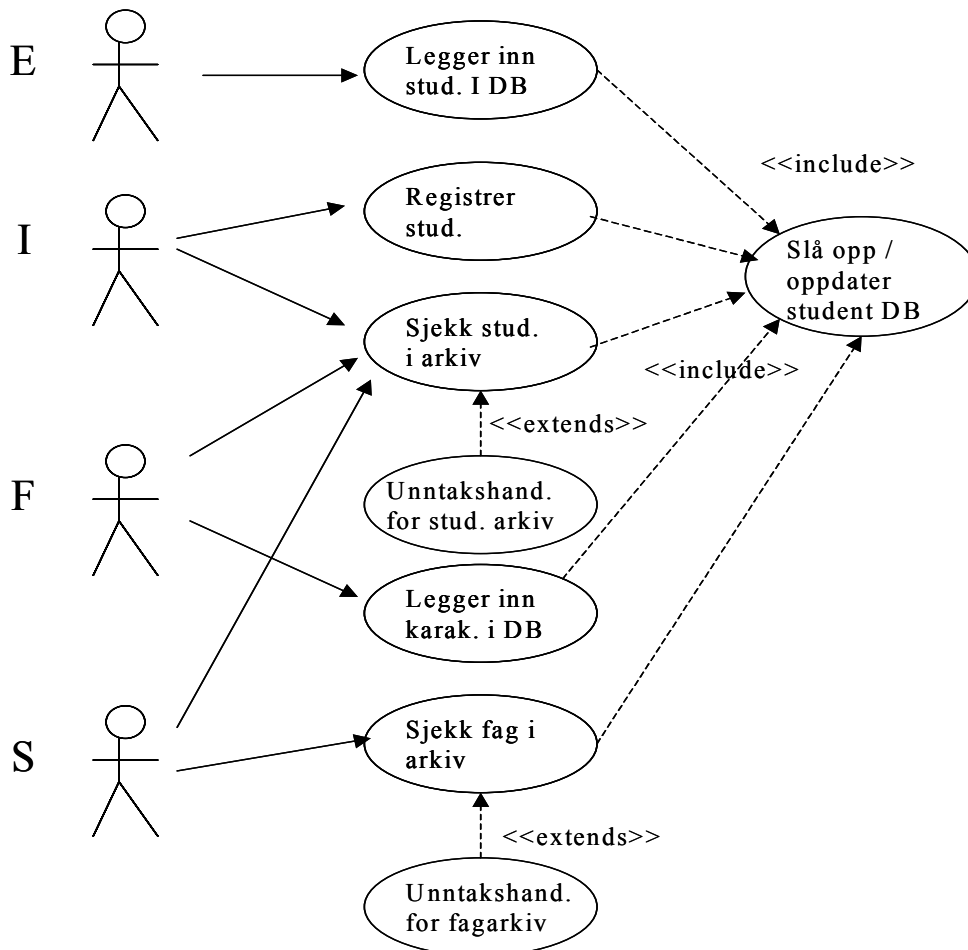
Alt fra testversjonen i punktet over. Dersom ikke selve registreringen var på plass må denne lages nå.

Programvare for at sensor skal kunne finne en student og sette karakter i et gitt fag.

Programvare for at en student skal kunne sjekke sine karakterer.

5. Mange av de planlagte brukerne av dette systemet har liten eller ingen erfaring med bruk av Pc-er. Dette stiller store krav til brukervennlighet og opplæring. Forklar hvordan du vil ta hensyn til disse to forhold i design, testingen og prøveinstallasjon av systemet.

- **Design: lage en prototyp / "mock-up" av brukergrensesnittet for å teste det mot virkelige brukere – tidligere eller nåværende eksamensinspektører.**
- **Testing: teste det ferdige brukergrensesnittet med hensyn til brukervennlighet mot tidligere eller nåværende eksamensinspektører.**
- **Prøveinstallasjon: sørge for at alle brukerne får et kurs i å bruke systemet.**

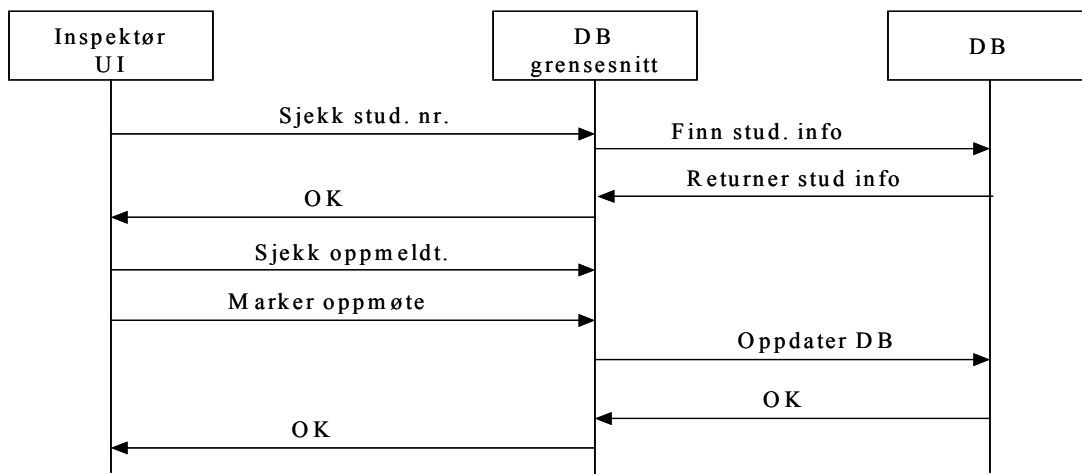


Figur 1: Use case diagram – oppgave 1.1

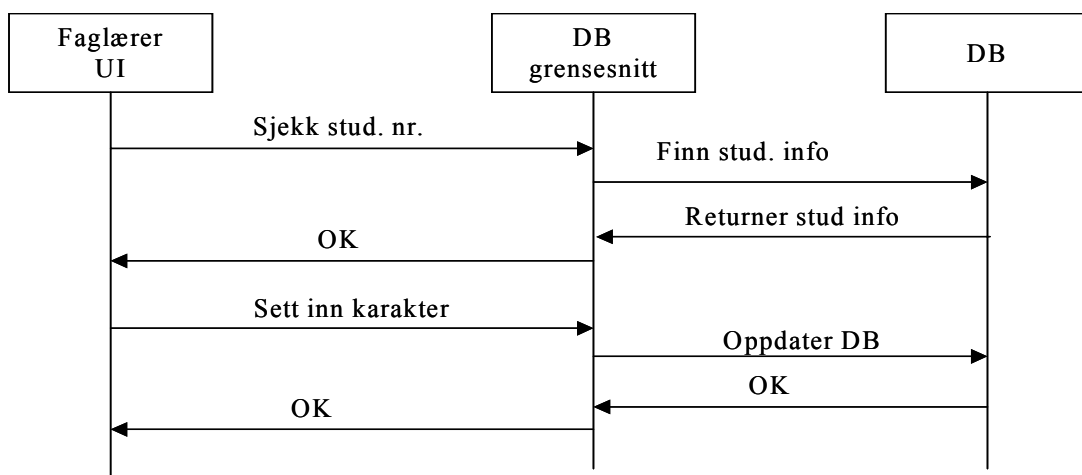
Kommentarer til sekvensdiagrammene:

- **Det som er vist er minimum. En meget god besvarelse bør også ha med spesialtilfellene – use case "Alternativ", helst som egne sekvensdiagrammer**

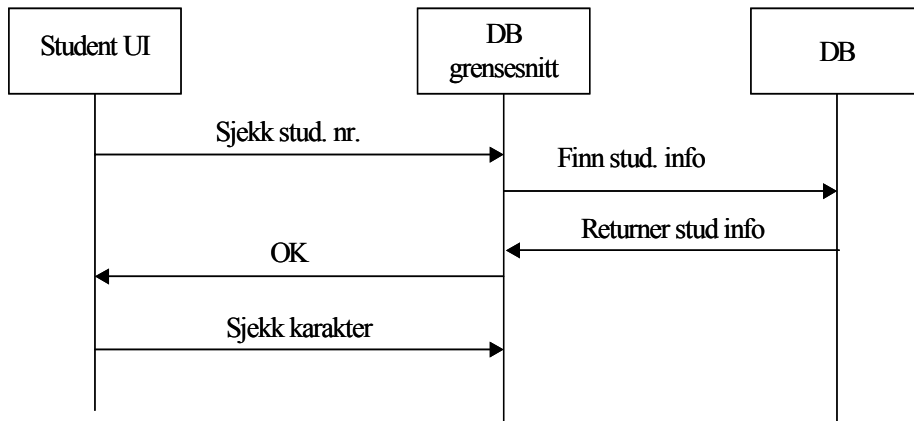
- *Vi opererer med følgende objekter:*
 - *DB – databasen der all informasjon ligger*
 - *DB grensesnitt – objektet som er ansvarlig for alle endringer i databasen. Den er også ansvarlig for å hente all nødvendig informasjon fra DB og lagre all endret informasjon*
 - *Grensesnitt for brukeren – Inspektør UI, Faglærer UI og Student UI*
- *Kvittering – OK – er strengt tatt ikke nødvendige og det skal ikke trekkes viss de ikke er med når diagrammene ellers er bra.*
- *Det er mulig å ta med den delen av systemet som sørger for kommunikasjon mellom sentral del og mobil del. Dette bør være et pluss for bevarelsen.*



Figur 2: Inspektøren registrerer frammøte i eksamenslokalet



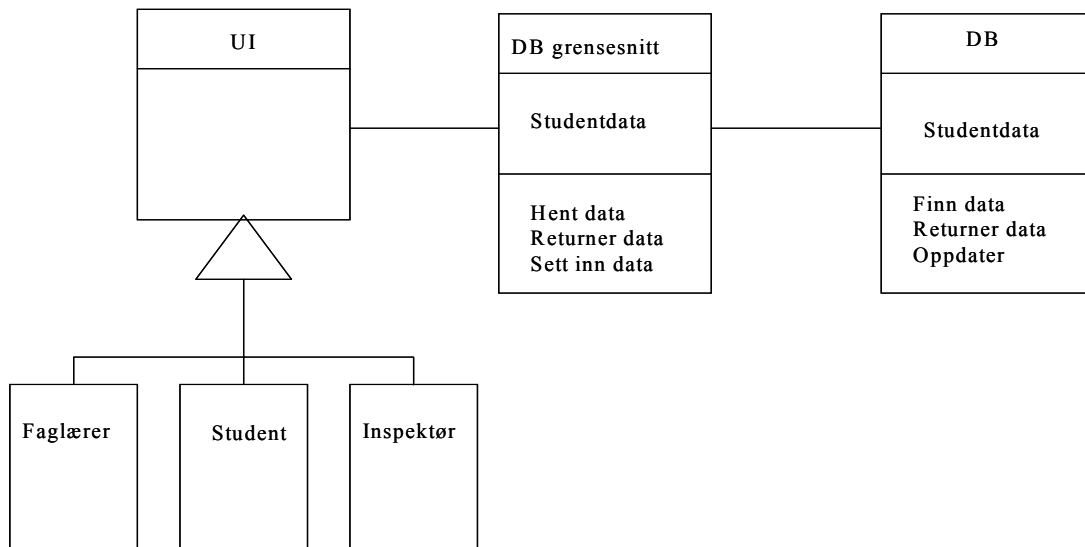
Figur 3: Faglærer legger inn resultatene for et fag.



Figur 4 : En student sjekker alle sine karakterer for et semester

Kommentarer til konseptuelt klasediagram:

Siden studentene kna gjøre sine egne antakelser vil det være mulig å komme fram til mange forskjellige klasediagrammer. Det som er vist her er minimum.



Figur 5: Konseptuelt klasediagram