

Uke 10

IN1030 – Gruppe 6

Dagens plan:

- Oblig 4
- Repetisjon
 - UML modellering
 - Use Case diagram
 - Aktivitetsdiagram
 - Sekvensdiagram
 - Klassediagram
- Ukesoppgaver

Min mailadresse er mysc@uio.no

Mitt navn (til kontakt gjennom teams) er **My Schultheiss**



Oblig 4

Formålet med oppgaven er å oppnå forståelse for foranalyse og kravhåndtering, samt forskjellen på funksjonelle og ikke-funksjonelle krav. Oppgaven skal gi innsikt i hva en prosessmodell og en utviklingsprosess er, samt hva som karakteriserer disse. Oppgaven gir også trening i modellering av use case-diagram.

Innleveringsfrist 13 april



UML

Når bruker vi UML?

I kravanalysen

I designprosessen

Etter implementasjon

Når vi vil:

..ha felles forståelse

..beskrive tenkt system

...dokumentere eksisterende system

Hvorfor lager vi UML diagrammer?

For å

- Identifisere krav
- Ha en felles forståelse av tenkt løsning
- Kunne vedlikeholde systemet
- Kunne lage et godt systemdesign
- Sørge for at andre kan ta over vedlikehold av systemet

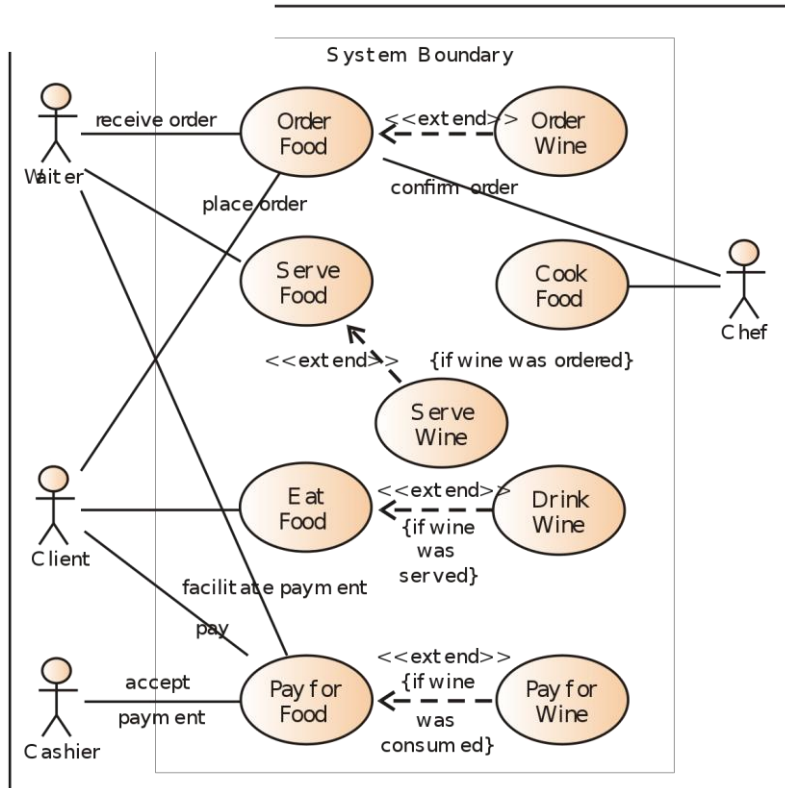


UML-diagrammer – de fire vi
fokuserer på

Hvilke diagrammer er dette?

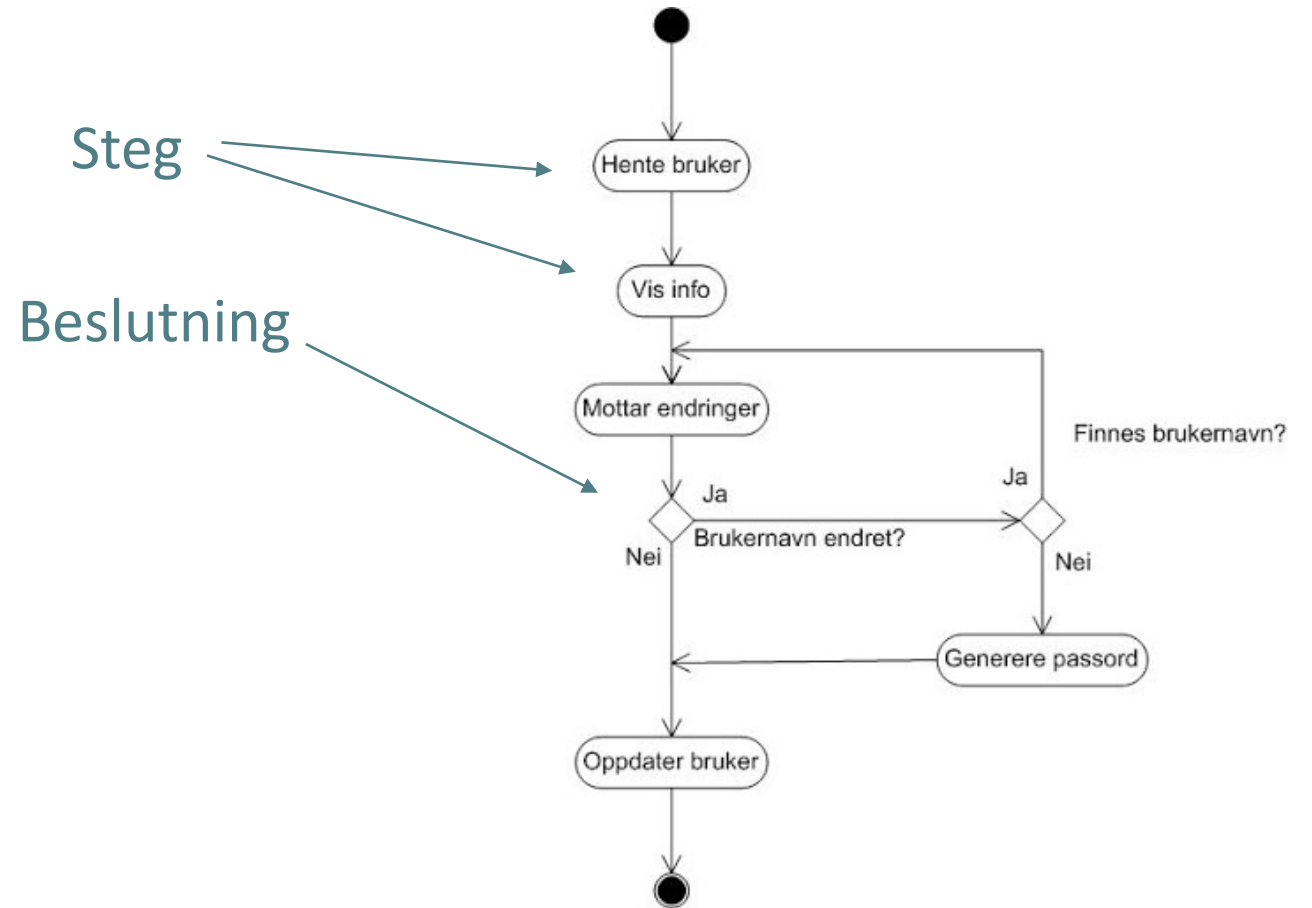
Use Case-diagram

Viser interaksjon mellom et system og omgivelsene. Tar utgangspunkt i primæraktørs mål og hvordan sekundæraktører assisterer dette målet gjennom systemet.



Aktivitetsdiagram

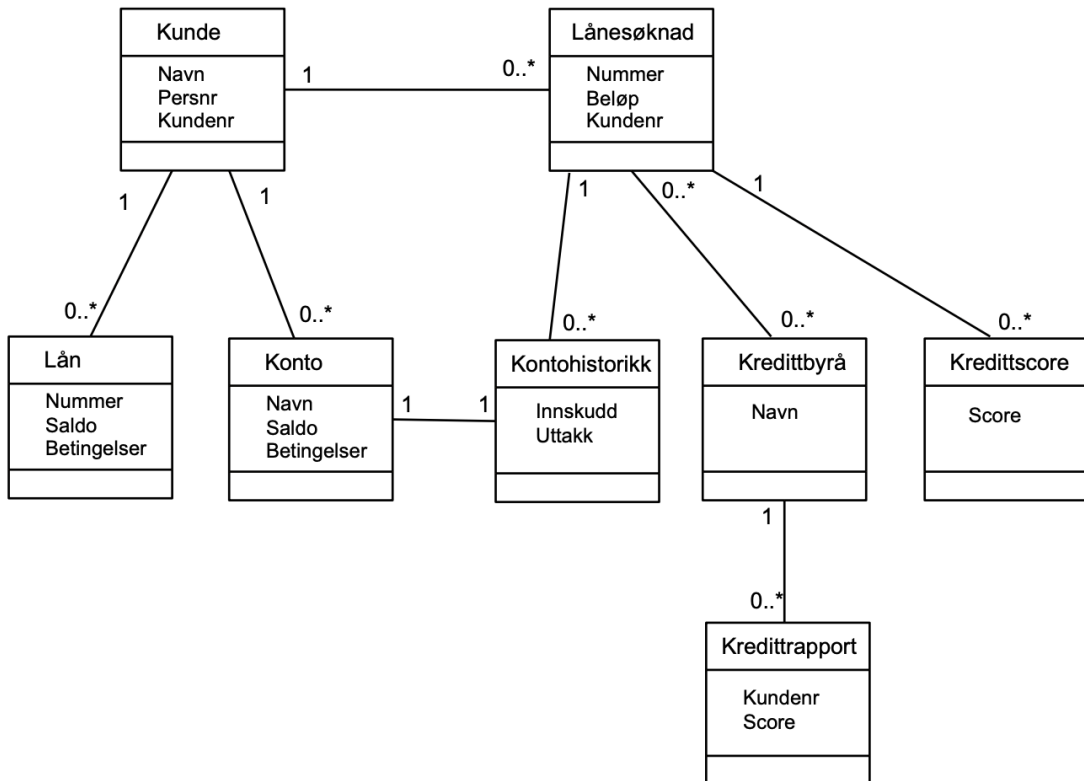
Viser aktivitetsflyten i en prosess eller dataprosessering.



Hvilke diagrammer er dette?

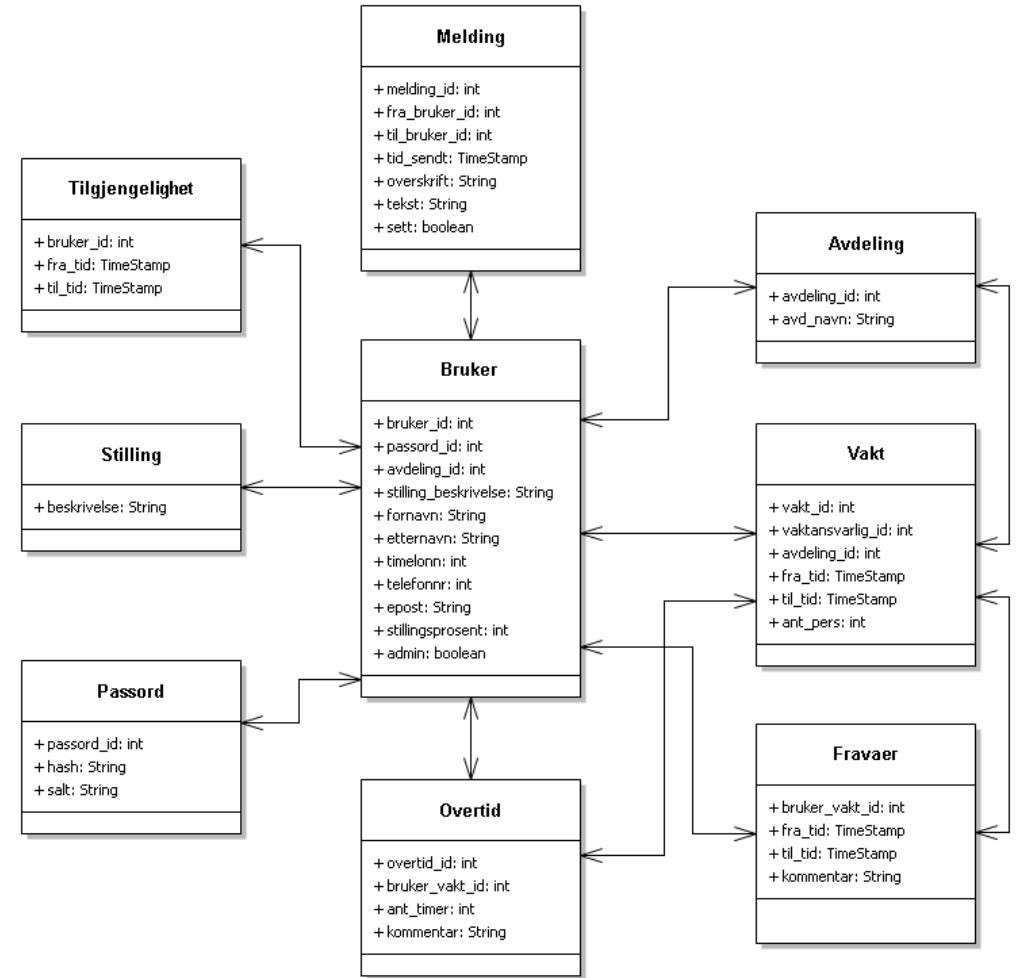
Domenemodell

Viser struktur: objektklasser av et system.
Hensikten er å skape en oversikt og forståelse
over objekter og terminologi.



Klassediagram

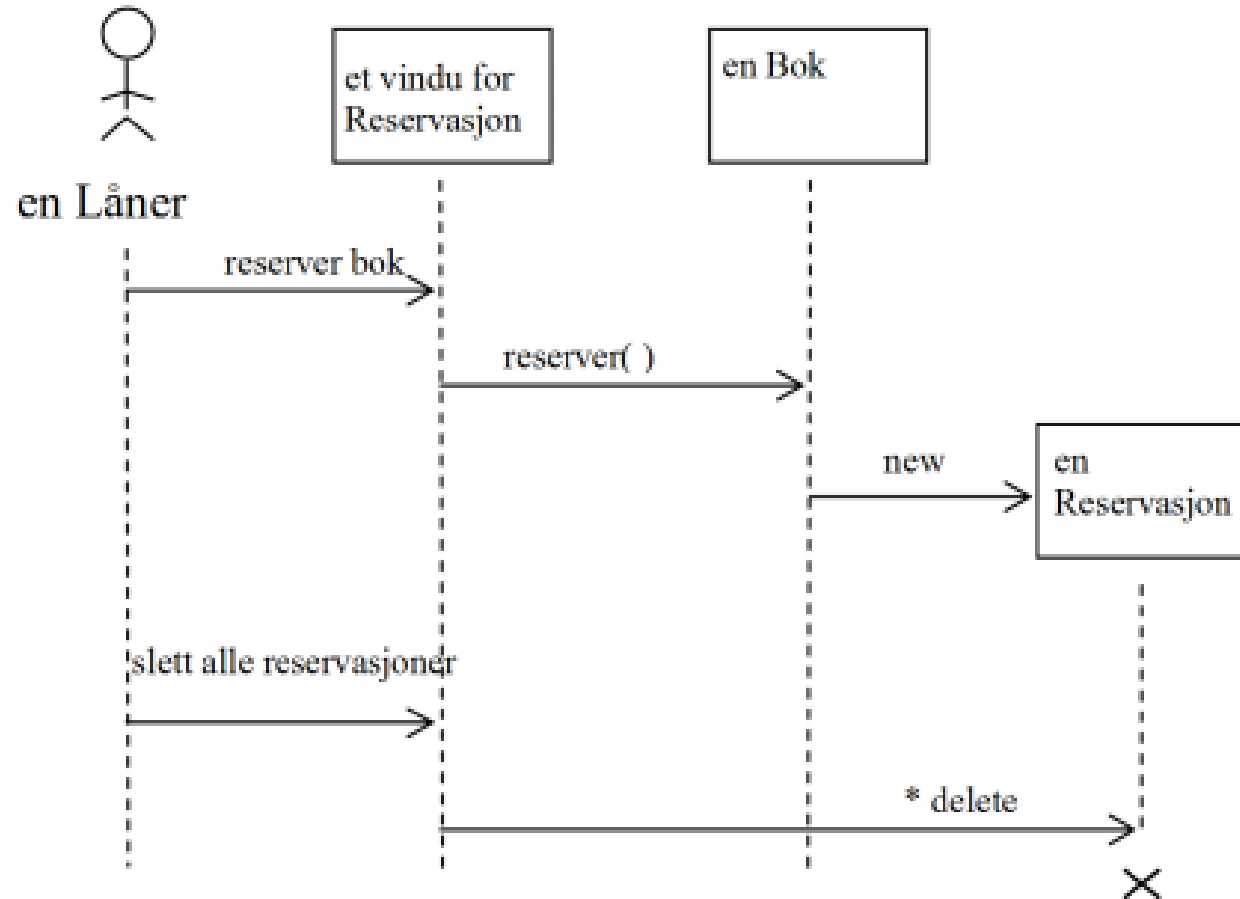
Viser struktur: objektklasser av et system, deres attributter og metoder, og assosiasjonene mellom klassene.



Hvilket diagram er dette?

Sekvensdiagram

Viser interaksjon og informasjonsflyten mellom aktørene og systemet og systemkomponentene i form av objektklasser.

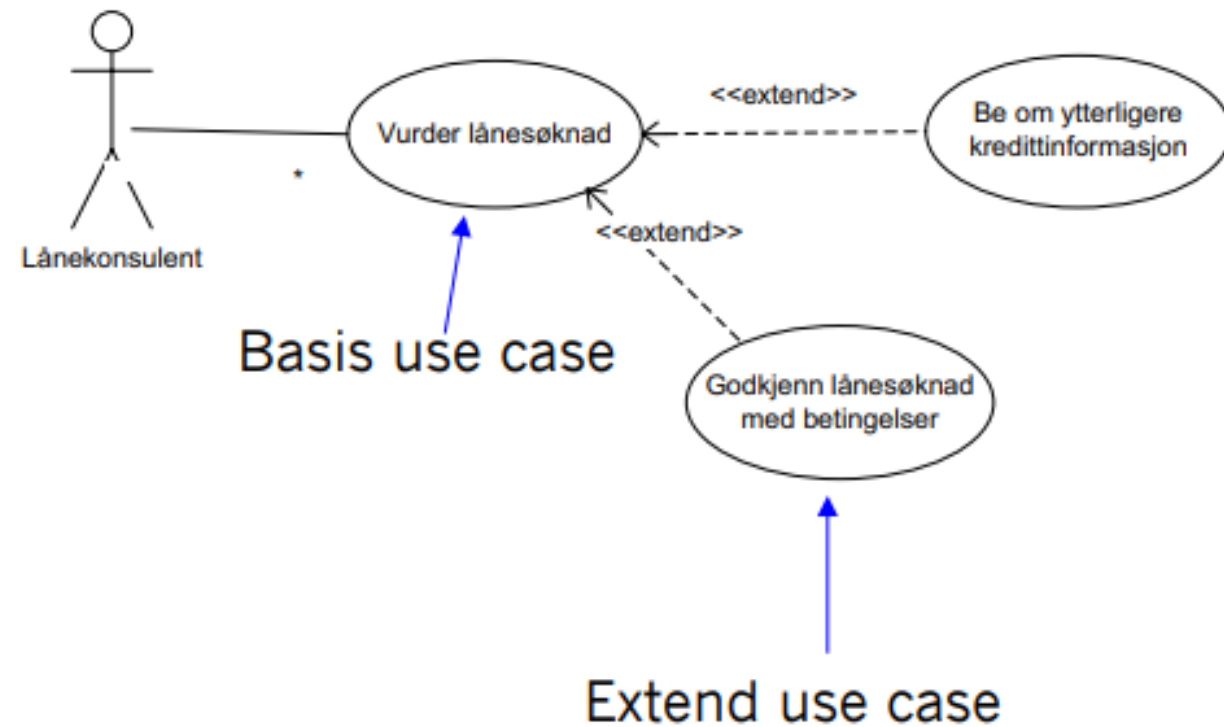
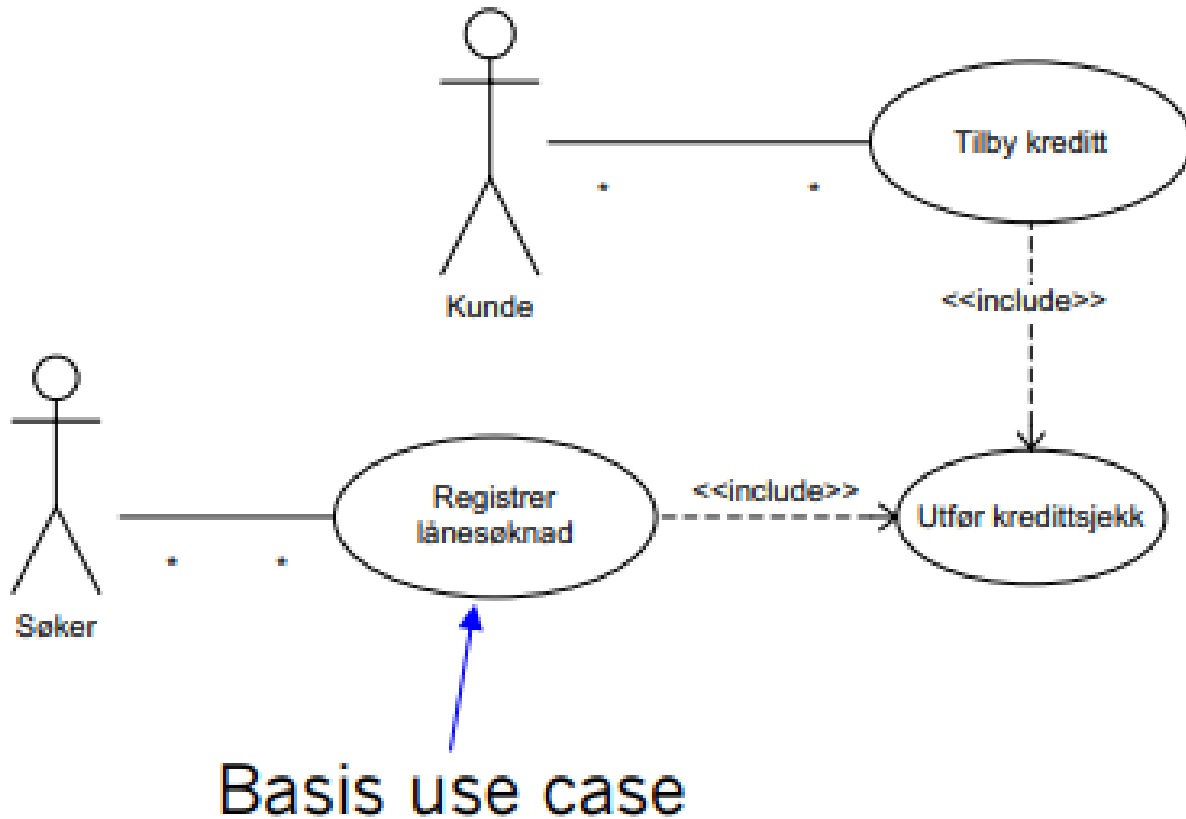




Interaksjonsmodeller

Use case diagram og sekvensdiagram

- Hjelper oss å definere brukerkrav
 - Use case – interaksjon mellom eksterne aktører og system
 - Sekvensdiagram – interaksjon mellom de ulike systemene/komponentene





Aktør vs. interessent

Primær- vs. sekundæraktør

Primæraktør: Har eget/egne mål i kommunikasjon med systemet

Sekundæraktør: Trengs for at primæraktøren skal nå sitt/sine mål, kommuniserer også aktivt med systemet

Use case diagram

Hvilke mål har primæraktøren? → **Boble**

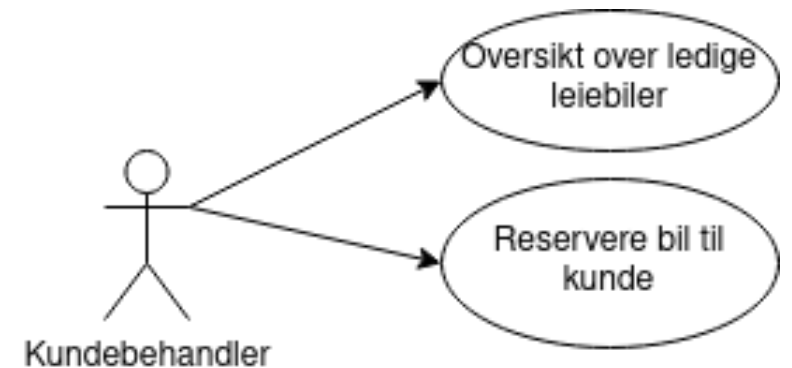
Hvem hjelper til med å nå primæraktørs mål? →
Sekundær aktør

Fra brukerhistorie til Use Case diagram:

Som kunde ønsker jeg å vite totalkostnaden på billeien.

Som kundebehandler ønsker jeg å se hvilke biler jeg kan leie ut til kunden.

Som kundebehandler ønsker jeg å reservere bil til kunden.



Tekstlig beskrivelse av use case

Navn: Reserver bil

Primæraktør: Kundebehandler

Sekundæraktør: -

Prebetingelse: Ingen

Postbetingelse: Leiekontrakt for spesifisert bil og kunde med gitte utleiedatoer er opprettet

Hovedflyt:

1. Kundebehandler velger tidsintervall (hentedato og returdato)
2. Systemet returnerer en liste over tilgjengelige biler innenfor de spesifiserte datoene
3. Kundebehandler velger én av bilene.
4. Systemet ber om kundenr. og finner kunden i systemet
5. Systemet bekrefter at bilen er reservert for den gitte perioden

Alternativ flyt punkt 2:

- 2.1: Det finnes ingen tilgjengelige biler i valgt tidsintervall.
- 2.2. Systemet opplyser om at det ikke er tilgjengelige biler innenfor oppgitt tidsintervall.
- 2.3. Kundebehandler oppgir et nytt tidsintervall (steg 1) eller avslutter bruksmønsteret.

Tips til modellering av sekvensdiagram

1. Identifiser de ulike aktørene/objektene.
2. Lag et tenkt, tekstlig oppsett basert på hovedflyt.
3. Modeller steg for steg, basert på stegene i hovedflyten.
4. Inkluder alternativ flyt etter at du har laget en modell for hovedflyten.

Tekstlig beskrivelse av use case

Navn: Reserver bil

Primæraktør: Kundebehandler

Sekundæraktør: -

Prebetingelse: Ingen

Postbetingelse: Leiekontrakt for spesifisert bil og kunde med gitte utleiedatoer er opprettet

Hovedflyt:

1. Kundebehandler velger tidsintervall (hentedato og returdato)
2. Systemet returnerer en liste over tilgjengelige biler innenfor de spesifiserte datoene
3. Kundebehandler velger én av bilene.
4. Systemet ber om kundenr. og finner kunden i systemet
5. Systemet bekrefter at bilen er reservert for den gitte perioden

Alternativ flyt punkt 2:

- 2.1: Det finnes ingen tilgjengelige biler i valgt tidsintervall.
- 2.2. Systemet opplyser om at det ikke er tilgjengelige biler innenfor oppgitt tidsintervall.
- 2.3. Kundebehandler oppgir et nytt tidsintervall (steg 1) eller avslutter bruksmønsteret.

Tips til modellering av sekvensdiagram

1. Identifiser de ulike aktørene/objektene.

- **Hvilket system er det snakk om?**
 - Reservasjonssystem for biler
- **Har vi eventuelle undersystemer?**
 - Bilregister/kunderegister (avhengig av hvordan systemet er implementert)
- **Har vi eventuelle objekter?**
 - Kunde (objekter for ulike kunder), kontrakt (objekter for utleiekontrakt)
- **Hvem skal interagere med systemet?**
 - Kundebehandler
- **Aktør:** Kundebehandler
- **Klasser/objekter:** Reservasjonssystem, bilregister, kunderegister, kunde, kontrakt

Tips til modellering av sekvensdiagram

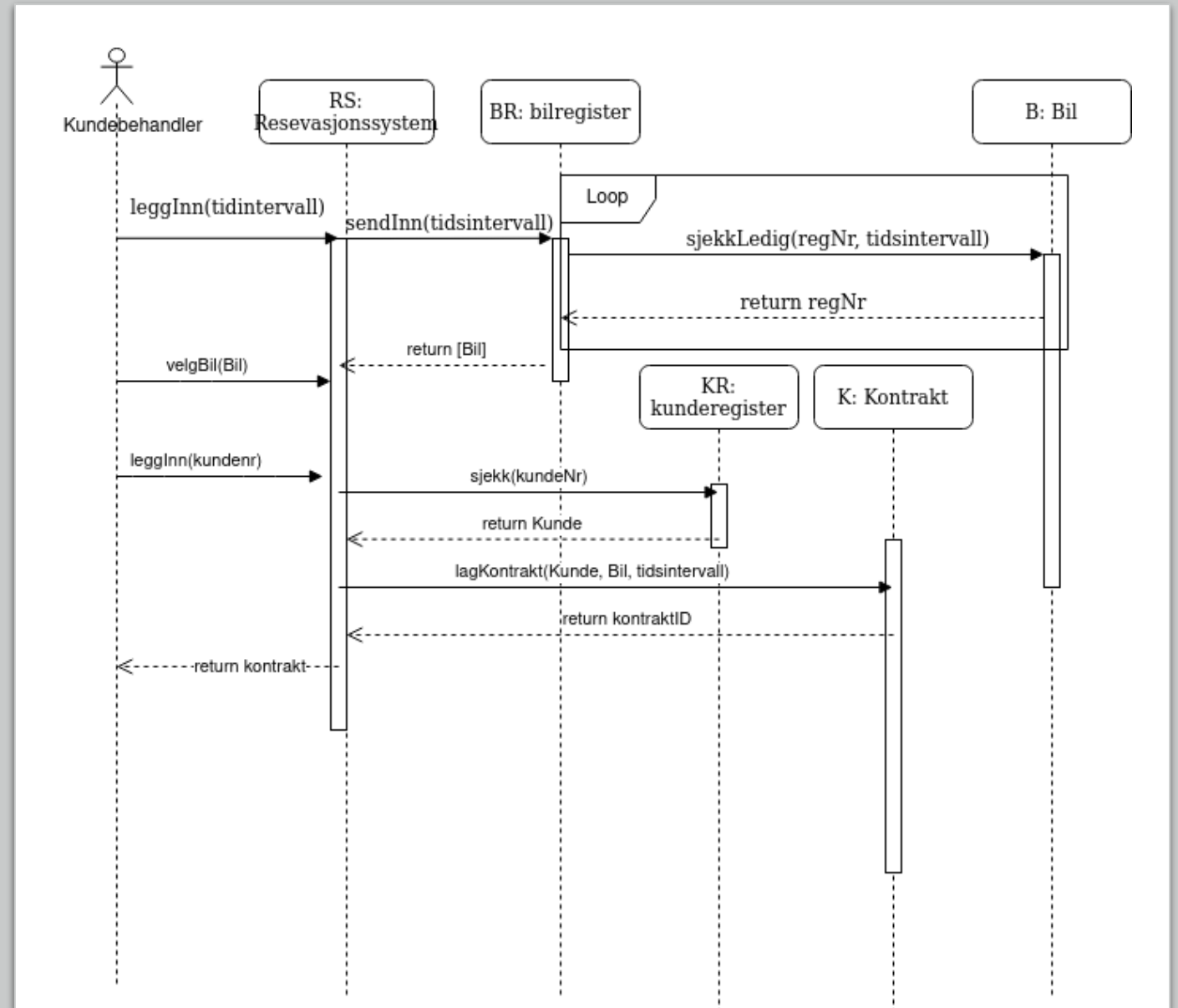
2. Lag et tenkt, tekstlig oppsett basert på hovedflyt:

- **Aktør (kundebehandler)** til venstre
- Interagerer med **Reservasjonssystemet** når hen velger tidsintervall.
- Da interagerer **Reservasjonssystemet** med **Bilregister** og henter ut tilgjengelige biler.
- «Viser» tilgjengelige biler til kunde.
- **Kundebehandler** (basert på «ekstern» interaksjon med kunde kanskje) velger bil og «sender» dette til **Reservasjonssystemet**.
- **Reservasjonssystemet** ber om kundenr.
- **Kundebehandler** gir kundenr.
- **Reservasjonssystemet** finner kunde i **Kunderegister**.
- *Kundeobjekt på plass eller være med i Sekvensdiagrammet? Må det registeres som alternativ flyt?*
- **Kontrakt** blir laget med metode fra **Kundeobjekt**

Sekvensdiagram

Hovedflyt:

1. Kundebehandler velger tidsintervall (hentedato og returdato)
2. Systemet returnerer en liste over tilgjengelige biler innenfor de spesifiserte datoene
3. Kundebehandler velger én av bilene.
4. Systemet ber om kundenr og finner kunden i systemet
5. Systemet bekrefter at bilen er reservert for den gitte perioden





Ukesoppgaver

1.

Hva er et use case, og hvorfor er det nyttig å lage dem?

- En slags historie om hvordan systemet oppnår et mål av verdi for en aktør

2.

Hva skiller aktører fra interessenter?

- Hvilken rolle/tilknytning de har til bruken og utviklingen av systemet

3.

Hva er en aktør i et use case diagram, og hva er forskjellen på en primær- og sekundær aktør?

- Aktører er brukere av systemet eller systemer som brukes av eller bruker systemet
- Interessenter kan være brukere av systemet, men påvirker også og/eller blir påvirket av systemets kravspesifikasjon og utvikling

5.

Anta følgende beskrivelse av et system som skal håndtere bord og bordbestillinger på en restaurant:

Systemet skal støtte bordreservasjoner og bordplassering i en restaurant. Kunder kontakter restauranten for å bestille eller avbestille bord. En resepsjonist mottar samtalene. Bestillinger legges inn for et bestemt bord sammen med antall personer. For hver bestilling registreres en kontaktperson med navn og telefonnummer.

Når gjester ankommer, blir de plassert ved sitt bord av hovmesteren, og deres bestilling markeres med “ankommet”. Hvis gjestene plasseres ved et annet bord enn det som var registrert med bestillingen, så registreres bordbyttet i bestillingen. Tidspunktet da et gitt bord må være ledig igjen kan også registreres. Kunder kan endre bestilling eller avbestille bord på forhånd.

Det er selvfølgelig mulig å spise uten å ha bestilt på forhånd hvis det er ledige bord. Når gjester får bord uten å ha bestilt dette, markeres det i systemet med tidspunkt, bord og antall, men uten navn og telefonnummer.

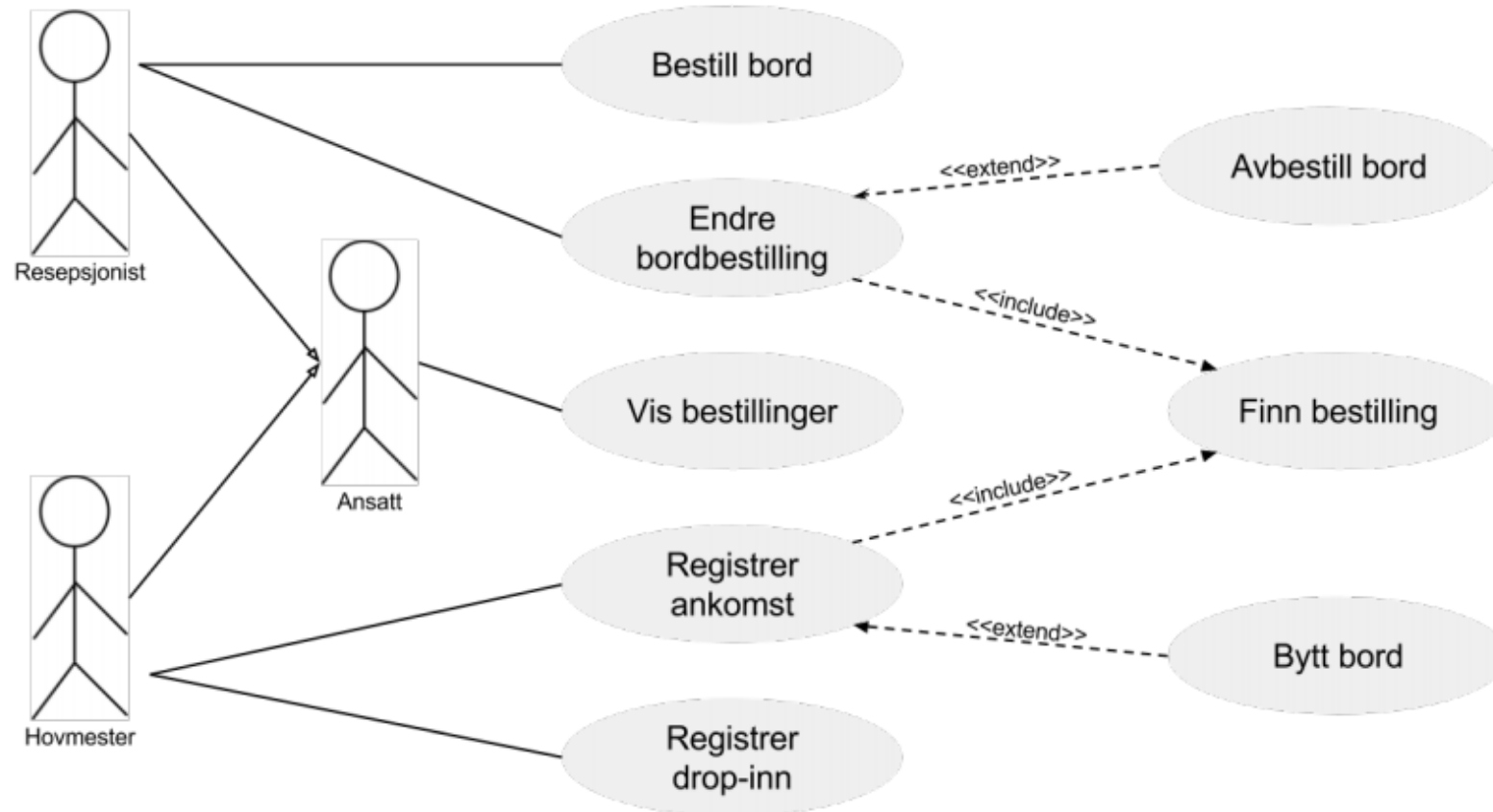
Når nye bestillinger registreres i systemet, eller eksisterende bestillinger endres, skal skjermbildet umiddelbart oppdateres, slik at de ansatte på restauranten alltid har oppdatert informasjon tilgjengelig.

- a) Finn aktører for systemet.
- b) Finn use cases for systemet
- c) Lag et use case diagram for systemet

“include-relasjonen”: Indikerer at et (sub) use case inneholder nødvendig funksjonalitet for gjennomførelsen av et annet basiscase.

“extend-relasjonen”: Utvider oppførselen / funksjonalitet til et basiscase, som utføres under spesielle omstendigheter.

5.



“include-relasjonen”: Indikerer at et (sub) use case inneholder nødvendig funksjonalitet for gjennomførelsen av et annet basiscase.

“extend-relasjonen”: Utvider oppførselen / funksjonalitet til et basiscase, som utføres under spesielle omstendigheter.

6.

- a. Hva er en tekstlig beskrivelse av et use case?
- b. Lag én tekstlig beskrivelse av et av use casene fra 5b. Ta med eventuelle pre- og postbetingelser og få med minst én alternativ flyt

7.

Hva er et sekvensdiagram?

UC1: Find a product

Level: User

Main actors: Customer

Main scenario:

1. Customer chooses a category of the product.
2. System presents a list of products belonging to the category.
3. Customer chooses the product.
4. System displays detailed information about the product.

Alternative scenarios:

- 1.A. Customer would like to search for the product.
 - 1.A.1. Customer searches for the product [UC2].

Extensions:

- 2.A. Customer would like to add some products to favorites.
 - 2.A.1. Customer chooses products and add them to favorites [UC3].

8.

Hvorfor er det nyttig å benytte sekvensdiagrammer?

6.

- a. Hva er en tekstlig beskrivelse av et use case?
- b. Lag én tekstlig beskrivelse av et av use casene fra 5b. Ta med eventuelle pre- og postbetingelser og få med minst én alternativ flyt

Navn: Bestill bord

Aktører: Resepsjonist

Prebetingelse: ingen

Postbetingelser (mål): Bord er reservert på kunde

Hovedflyt:

1. Kundebehandler ber systemet om å finne et ledig bord på en gitt tid og dato
2. Systemet finner ledige bord
3. Kundebehandler registrerer kundens navn, adresse og telefonnummer
4. Systemet lagrer kundens navn, adresse, og telefonnummer
5. Systemet ber om bekreftelse på reservasjon om bord
6. Kundebehandler bekrefter reservasjon av bord
7. Systemet registrerer reservering av bord på kunde

Alternativ flyt:

2.1. Systemet finner ingen ledige bord

2.2. Returnerer til hovedflyt, steg 1.

Modellering av alternativ flyt – Identifiser avvik i hovedflyten; der hovedflyten har alternativ. Hva skjer i slike tilfeller? Hvordan responderer systemet?

Alternativ til alternativ flyt i oppgaven:

Bruker ber systemet om å finne et ledig bord til en gitt tid og dato:

Systemet returnerer ledig bord → INGEN LEDIGE BORD

Bruker registrerer kundens navn og telefonnummer:

Systemet lagrer kundens navn og telefonnummer → UGYLDIG
NAVN/TELEFONNUMMER

Systemet ber om bekreftelse på reservasjon av bord

Bruker bekrefter reservasjon av bord → BRUKER AVKREFTER RESERVASJON

9.

Et bilutleiefirma ønsker et informasjonssystem som kundebehandlerne kan benytte for utleie av biler. Under er det et forslag til en tekstlig beskrivelse av bruksmønsteret Reserver bil.

Lag et sekvensdiagram for hovedflyten og én av de alternative flytene i den tekstlige beskrivelse.

Tekstlig beskrivelse av use case 'Reserver bil'

Navn: Reserver bil

Aktør: Kundebehandler

Prebetingelse: Ingen

Postbetingelse: Leiekontrakt for spesifisert bil og kunde med gitte utleiedatoer er opprettet

Hovedflyt:

1. Kundebehandler velger tidsintervall (hentedato og returdato)
2. Systemet returnerer en liste over tilgjengelige biler innenfor de spesifiserte datoene
3. Kundebehandler velger én av bilene.
4. Systemet ber om kundenr og finner kunden i systemet
5. Systemet bekrefter at bilen er reservert for den gitte perioden

Alternativ flyt 2.1: Det finnes ingen tilgjengelige biler i valgt tidsintervall.
2.2. Systemet opplyser om at det ikke er tilgjengelige biler innenfor oppgitt tidsintervall.
2.3. Kundebehandler oppgir et nytt tidsintervall (steg 1) eller avslutter bruksmønsteret.

Alternativ flyt 4.1: Kunden finnes ikke.
4.2. Systemet oppretter ny kunde og returnerer til steg 3

Analyse av den tekstlige beskrivelsen:

1. Identifiser de aktuelle objektene/aktørene:
 - Hvilket system er det snakk om?
 - Finnes det undersystemer? Hva med objekter?
 - Hvem skal interagere med systemet?
2. Oppsett (rekkefølgen gis av flyten):
 - Aktører plasseres til venstre i sekvensdiagrammet
 - Hva interagerer aktører med?
3. Modeller hendelsesforløpet ved å følge den tekstlige beskrivelsen
 - Rekkefølgen for hoved- og alternativ flyt bestemmer rekkefølgen i diagrammet
 - Hvert steg i den tekstlige beskrivelsen er tilnærmet lik en pil i diagrammet
 - Data som sendes mellom objektene reflekteres i diagrammet
 - Metodekall (med parametre) = Heltrukket pil
 - Returverdier = stiplet pil

