

UKE 10

Use case og kravhåndtering

UML modellering I

IN1030 - Gruppe 1 & 3D

Plan for timen

- Prosjektledelse og Risikohåndtering
- UML-diagram & krav
- UML diagram i IN1030
- Use Case diagram + Sekvensdiagram

Prosjektledelse

Hvilke 6 tema er viktige innenfor prosjektledelse?

1. Forretningsplan: Hvorfor?

2. Organisasjon: Hvem?

3. Kvalitet: Hva og hvor bra?

4. Plan: Hvordan? Hvor mye? Når?

5. Risiko: Hva hvis?

6. Endring: Hvilke endringer skaper systemet?

F.O.K.P.R.E

Risikohåndtering

Knagger å henge risiko på:

- Prosjektrisiko påvirker planlegging og ressurser.
- Produkttrisiko påvirker kvaliteten eller ytelsen til programvare under utvikling.
- Forretningsrisiko påvirker organisasjonen som utvikler eller kjøper programvaren.

Vurderinger for risikoanalyse:

Sannsynlighet: *svært lav - lav - moderat - høy - svært høy*

Konsekvens: *ubetydelig - mindre alvorlig - alvorlig - katastrofal*

Risikomatrise

Risiko	Sannsynlighet	Konsekvens	Tiltak
Ansatte blir syke	Høy	Alvorlig	Ha tilgang til vikarbyrå
Intern konflikt	Svært høy	Katastrofal	Teambuilding HMS-oppfølging

Hva er UML?

Unified Modeling Language (UML) er en industristandard for datarelatert modellering, forvaltet av et internasjonalt konsortium kalt [Object Management Group](#) (OMG). ([Wikipedia](#)).

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?**

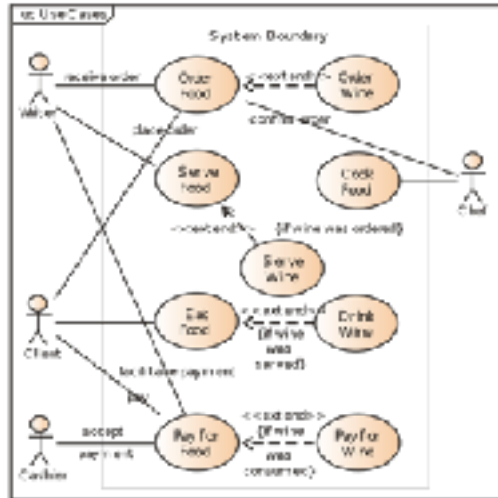
Fordi

- vi skal identifisere krav,**
- vi skal være enige om hvordan endelig løsning skal bli,**
- vi skal kunne vedlikeholde systemet,**
- vi skal kunne lage et godt system design,**
- vi ønsker at andre skal kunne ta over vedlikehold av systemet.**

**Hvilke 4 diagram skal dere
kunne?**

Hvilke diagram er dette?

Use Case-diagram:



1.

Aktivitetsdiagram:

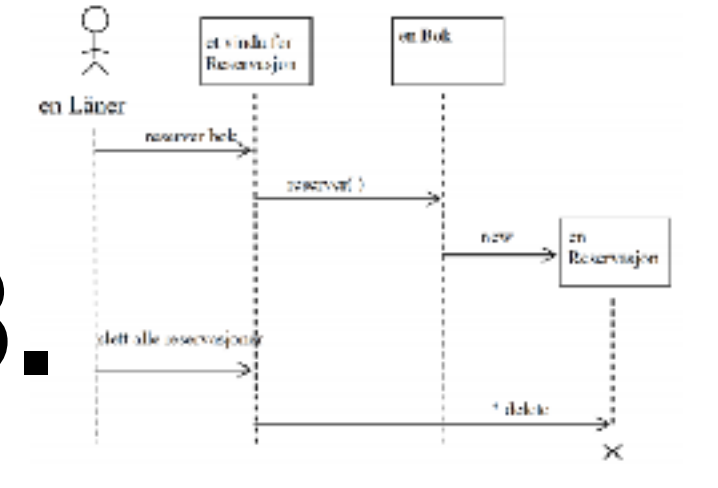


2.

Hvilke diagram er dette?

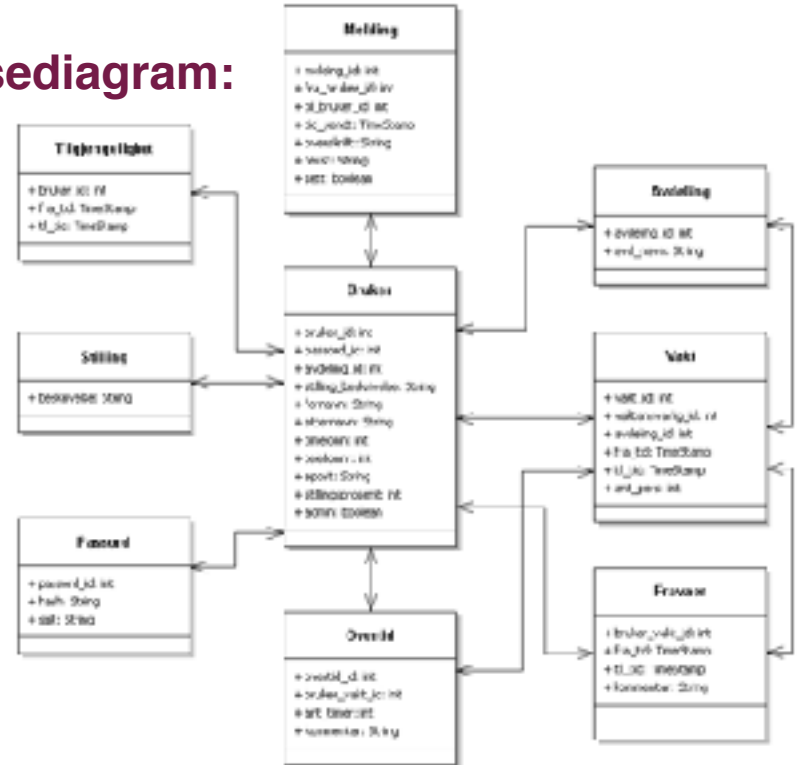
Sekvensdiagram:
m:

3.

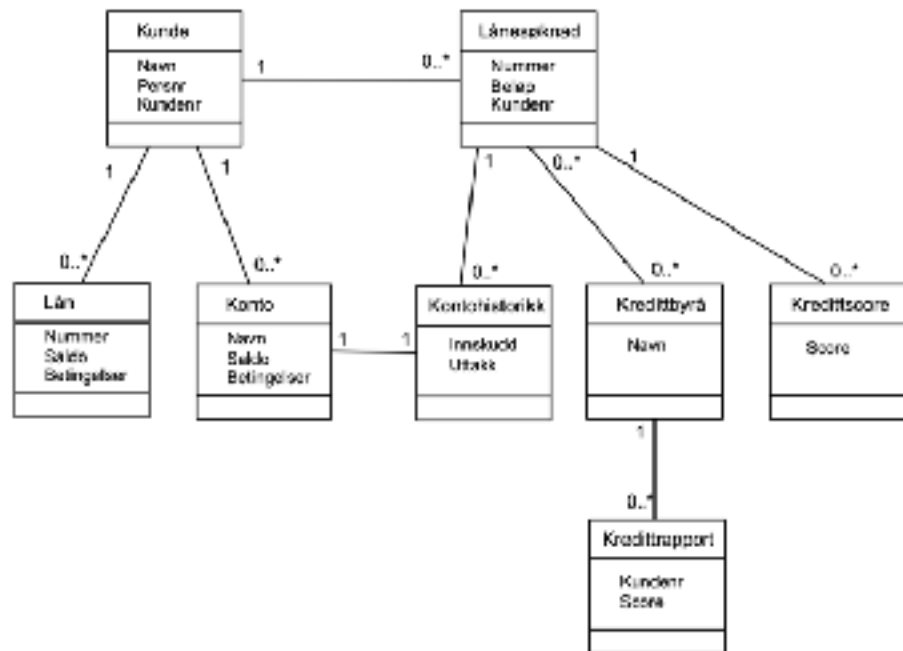


Klassediagram:

4.



Domenemodell - klassediagram uten metoder



- UML klassediagrammer uten metoder
- Domenemodellen viser objekter i problemdomenet.
- Hensikten med domønemodellen er å forstå objektene og få en oversikt over terminologi.
- Domenemodellen er nyttig i forbindelse med use case modellering fordi:
 - Domenemodellen viser informasjonen om objekter i use casene.
 - Den er et viktig verktøy for å sjekke at use casene er beskrevet med riktig detaljeringsnivå.

Kort om “våre” modeller

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

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

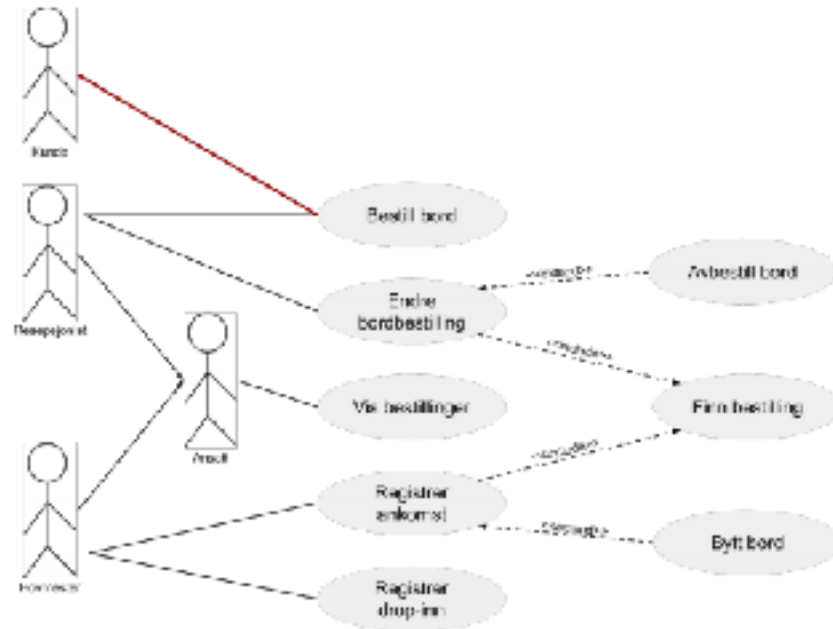
Klassediagram: Viser struktur: objektklasser av et system, deres attributter og metoder, og assosiasjonene mellom klassene.

Aktivitetsdiagram: Viser aktivitetsflyten i en prosess eller dataprosessering.

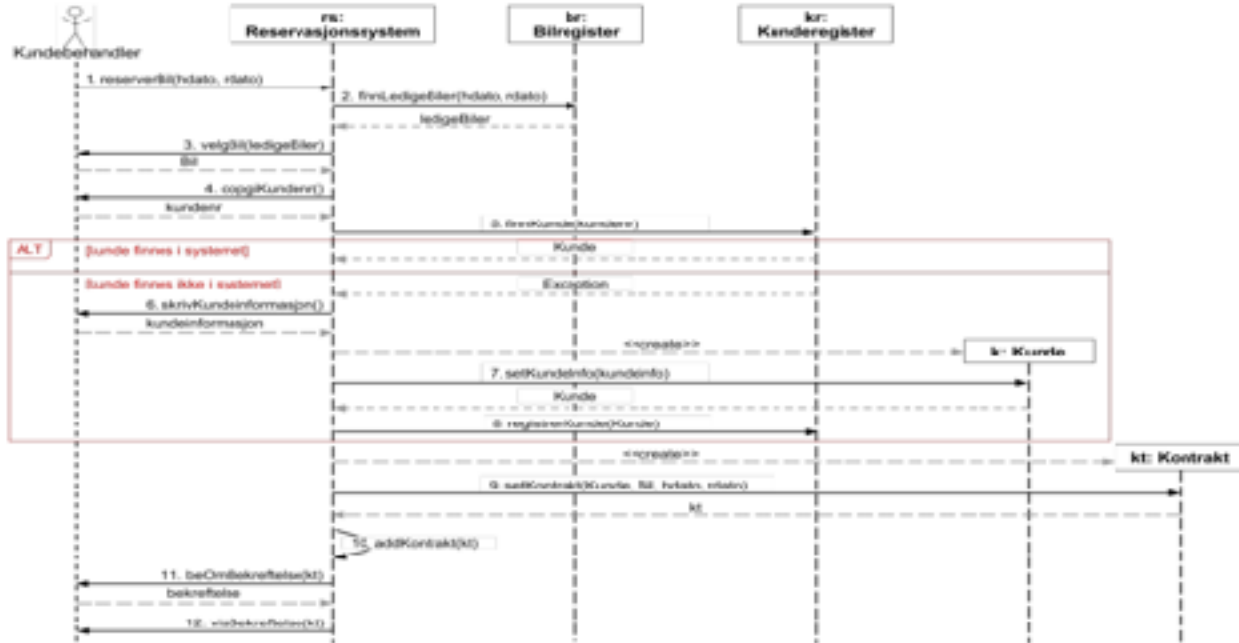
***OBS: Det er viktig at alle modeller for et og samme system samsvarer !
En metode som blir brukt i et sekvensdiagram må også være med i et klassediagram.***

Interaksjonsmodeller

Use Case diagram



Sekvensdiagram



Aktør vs interessent...

Aktør: aktiv rolle, kommuniserer med systemet.

Interessent: kommuniserer ikke nødvendigvis med systemet.

Primær- vs sekundæraktør...

Primæraktør: eget mål i kommunikasjonen med systemet.

Sekundæraktør: trengs for at primær aktøren skal nå målet, kommuniserer også aktivt med systemet.

Use Case diagram

Hvilke mål har primær aktøren? → Boble

Hvem hjelper til med å nå primær aktørs mål? → Sekundær aktør

Fra brukerhistorie til Use Case diagram:

Format: Som [rolle] ønsker jeg [funksjon] for å oppnå [nytteverdi]

Som kunde ønsker jeg å vite totalkostnaden på billetten.

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:

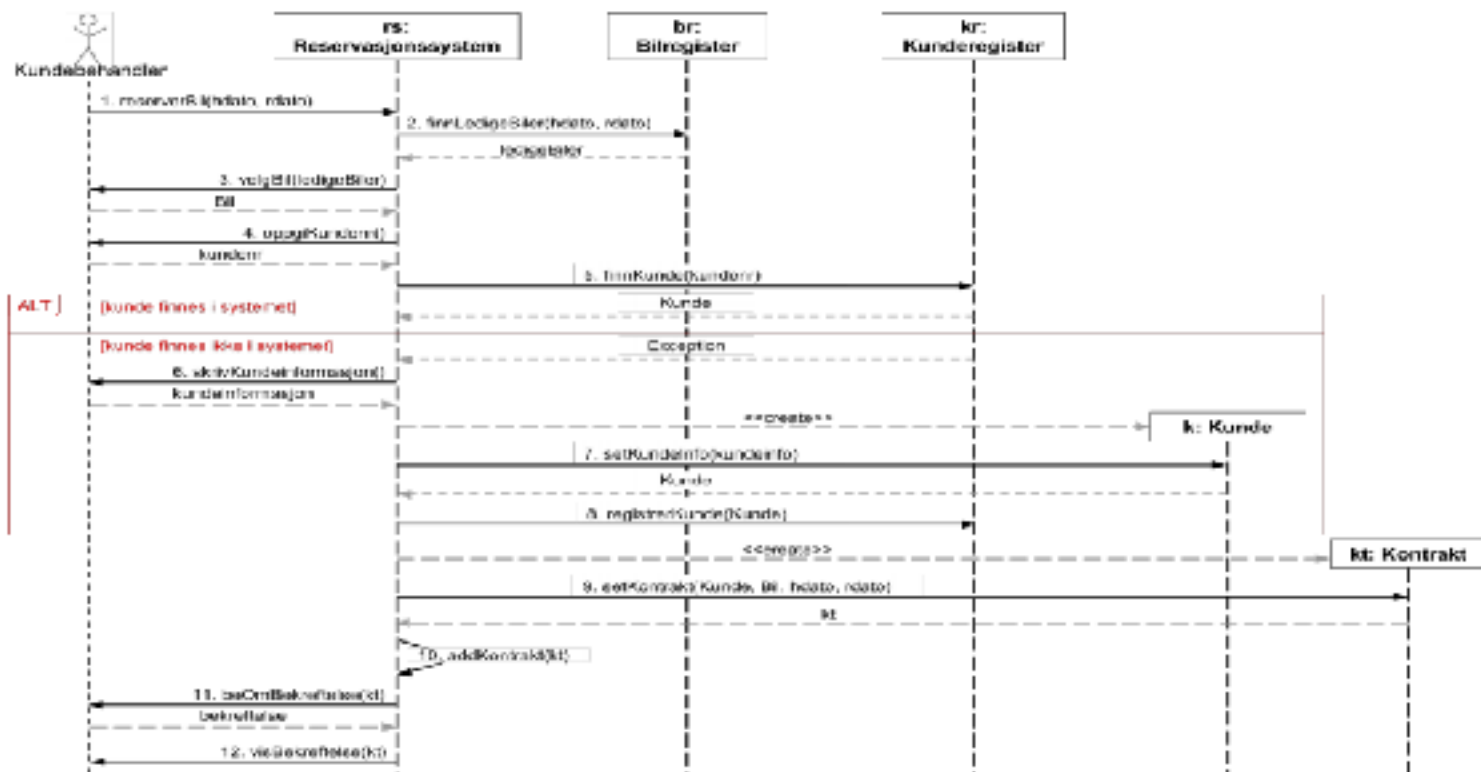
1. Kundebehandler velger tidsintervall (hentdato og returdato)
2. Systemet returnerer en liste over tilgjengelige biler innenfor de spesifiserte datoene
3. Kundebehandler velger én av bilene.
4. Systemet ber om kundenr og finner kunden i systemet
5. Systemet bekrefter at bilen er reservert for den gitte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

4.1 Kunden finnes ikke

4.2 Systemet oppretter ny kunde og returnerer til steg 3



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.

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 de ulike kundene)/Kontrakt (objekt for utleiekontrakt)*

Hvem skal interagere med systemet? – *Kundebehandler*

Aktør: **Kundebehandler.**

Klasser/Objekter: **Reservasjonssystem, Bilregister, Kunderegister, Kunde, Kontrakt.**

Tips til modellering av sekvensdiagram

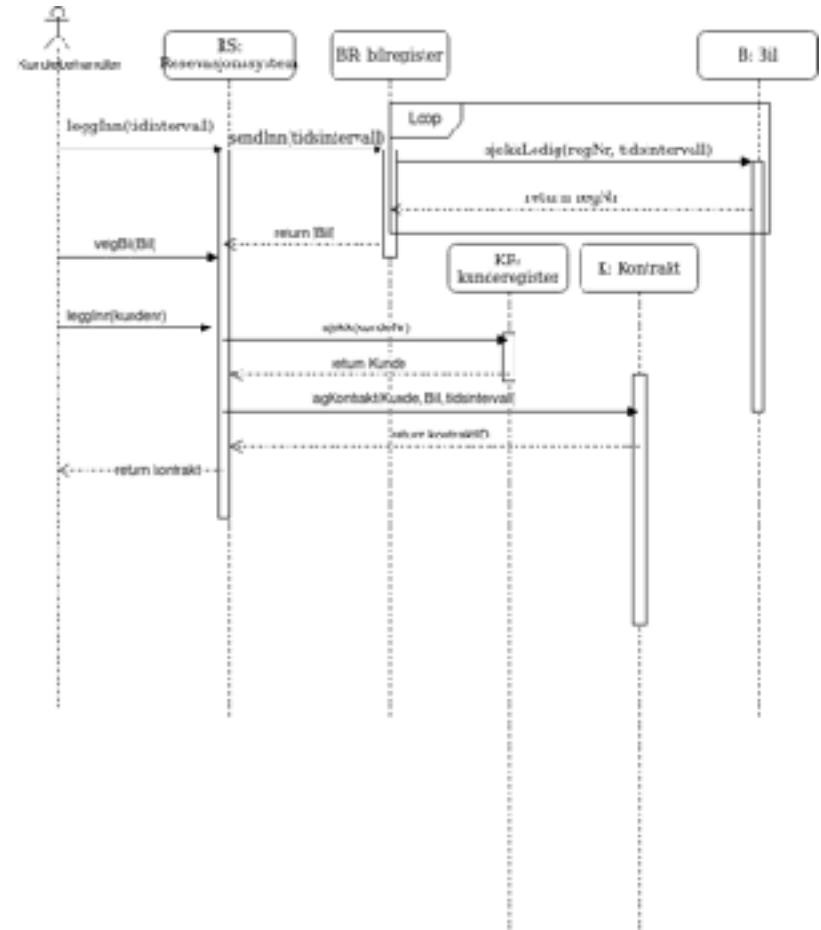
2: Lag et tenkt, tekstlig oppsett basert på hovedflyt:

- **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 **Reservasjonssystem**.
- **Reservasjonssystem** ber om kundenr.
- **Kundebehandler** gir kundenr.
- **Reservasjonssystem** 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

SPØRSMÅL 5a

Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Finn aktører for systemet.**

SPØRSMÅL 5b

Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Finn use cases for systemet.**

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

Når gjester ankommer, blir de **plassert** ved sitt bord av **hovmesteren**, og **bestillingen markeres** med **“ankommet”**. Hvis gjestene plasseres ved et annet bord, **registreres dette bordbyttet**. **Tidspunktet** et gitt bort må være **ledig** igjen kan også **registreres**. Kunder kan **endre bestilling** / avbestille på forhånd.

Det er mulig å få bord **uten** å ha **bestilt** på **forhånd**, gitt at **ledige bord** finnes. Når gjester får bord uten å ha bestilt dette, markeres **tidspunkt**, **bord** og **antall** i systemet, men ikke navn og telefonnummer.

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

SPØRSMÅL 5a

Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Finn aktører for systemet.**

Svar:

- **Resepsjonist:** motta samtaler og håndtere bestillinger
- **Hovmester:** plassere gjester
- Videre kan man *generalisere* disse to til “**Ansatt**”, med samme mål: se oversikt over bestillinger/ plasseringer.

SPØRSMÅL 5b

Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Finn use cases for systemet.**

Analyse:

Hva skal aktørene kunne gjøre? Hvilke resultater vil aktøren oppnå?

Identifiser de viktigste oppgavene (se etter verb)

- Skape / Lagre / Endre / Lese / Slette

Notasjon for use case-funksjoner:

Figur: Oval

Merkelapp: Navn på use case → verbfrase



SPØRSMÅL 5c

Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Lag et use case diagram for systemet.**

SPØRSMÅL 5c

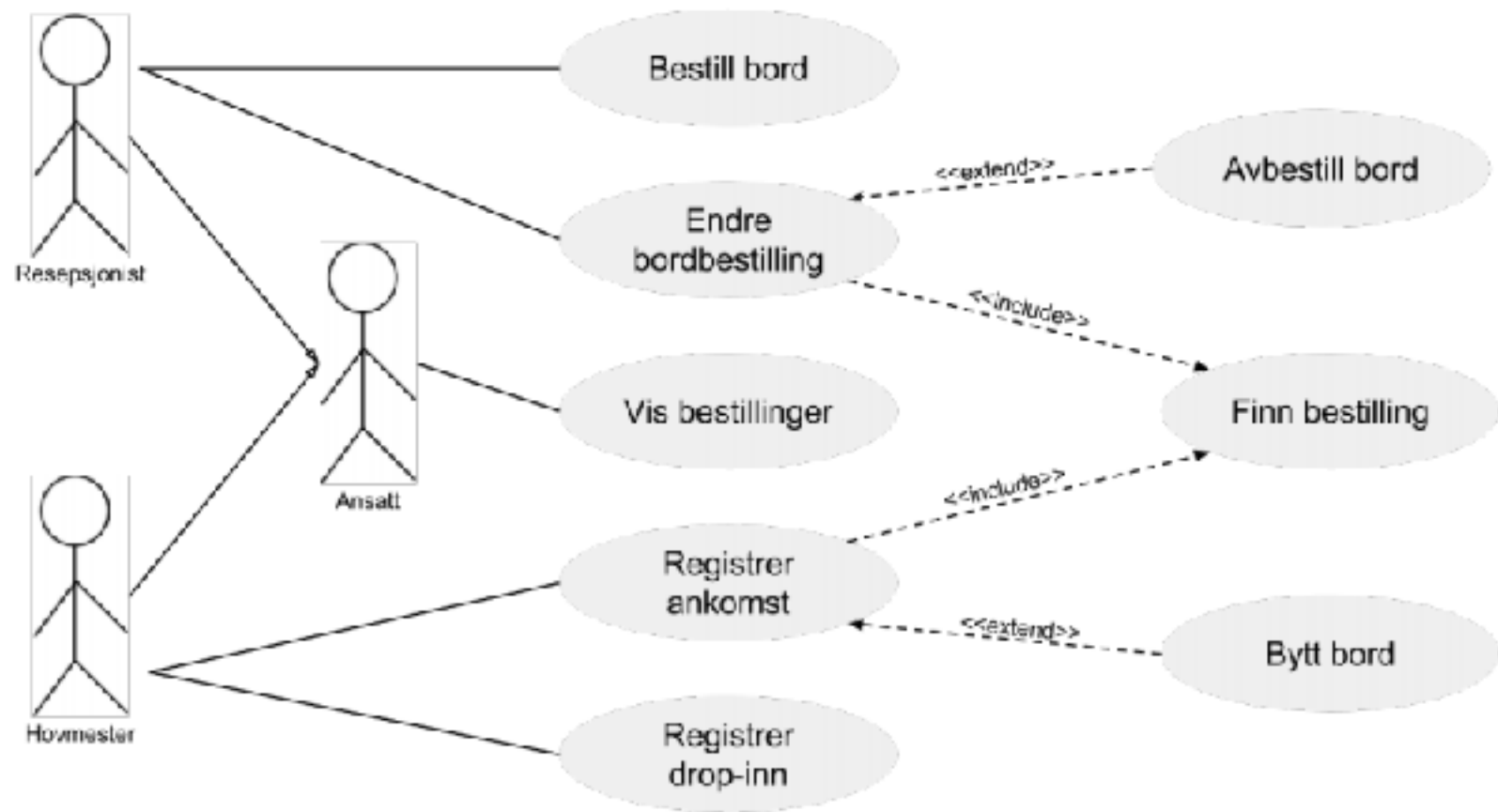
Spørsmål: Analyser beskrivelsen av et system som skal håndtere bord og bordbestillinger på en restaurant. **Lag et use case diagram for systemet.**

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.







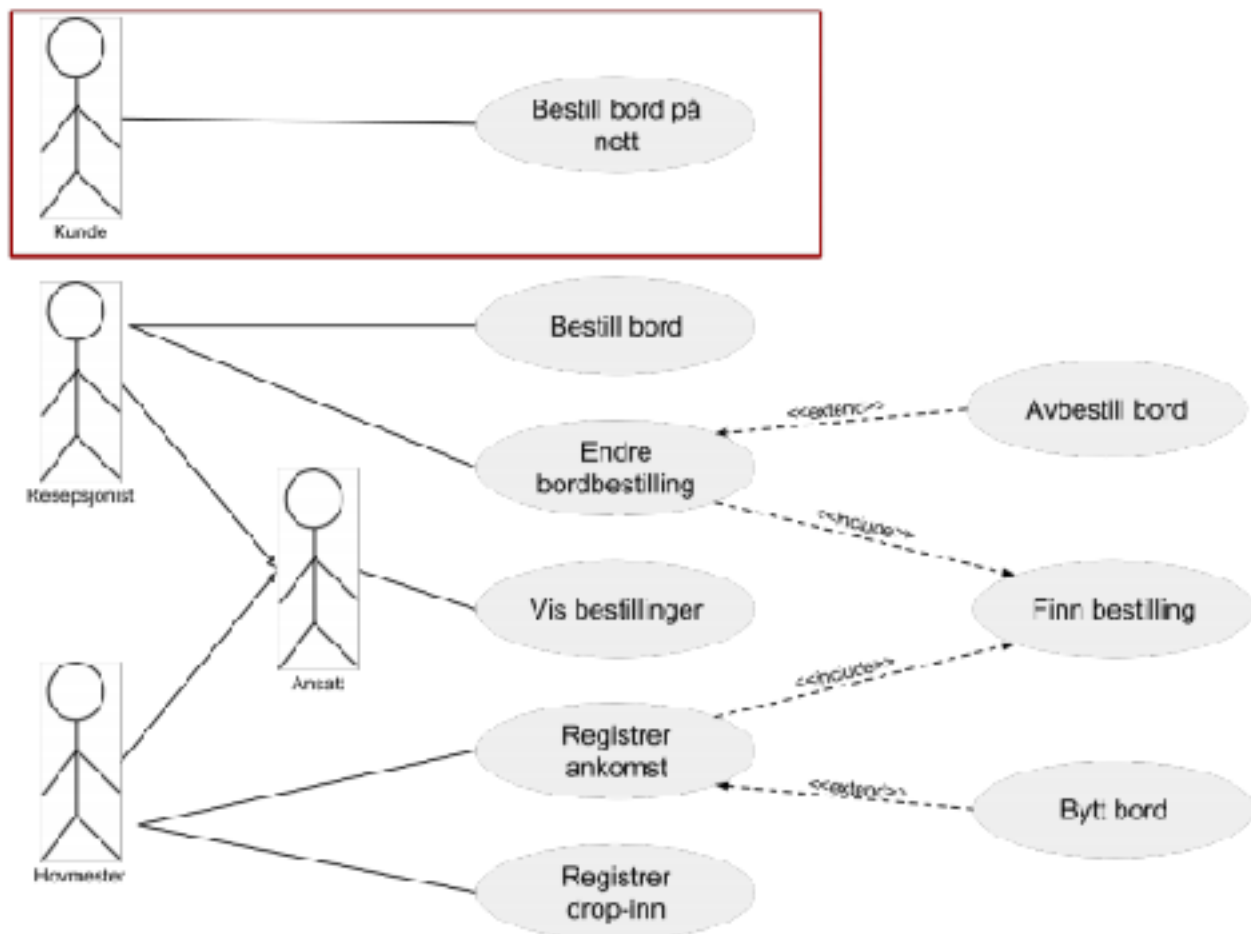
SPØRSMÅL 5d

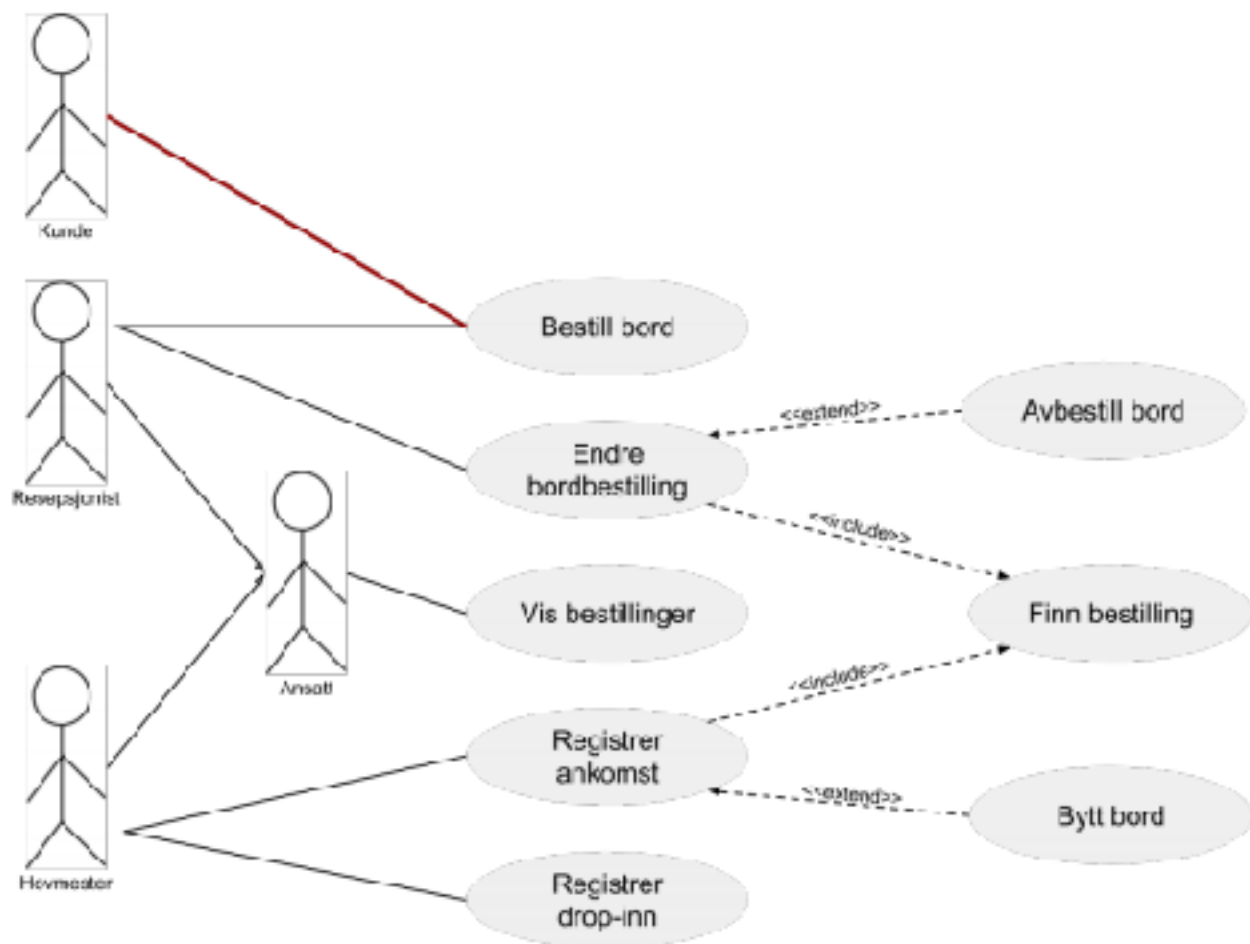
Spørsmål: Anta at en kunde kan bestille bord på nett i tillegg til over telefon. Gjør eventuelle modifikasjoner i løsningene i oppgave a, b og c.

Analyse:

Ny aktør **Kunde**, med mål: **Bestille bord på nett**

- Nytt use case?
 - Skal man opprette et nytt use case “Bestill bord på nett?”
 - Skal man beholde “Bestill bord”?
- Oppdater use case-diagrammet





Andre fine videoer:

[Aktivitetsdiagram](#)

[Sekvensdiagram](#)

[Use Case-diagram](#)

Denne uken

Modellering I: Use case og kravhåndtering

Neste uke

Modellering II: Objektorientert design.

Frist: Oblig 4, 13/4 kl. 23.59

