

# Use Case- & Aktivitetsdiagram

Repetisjonstime 16/5 2022

# Plan for timen

- Interessenter/aktører
- Use Case
- Tekstlig beskrivelse
- Aktivitetsdiagram
- Plass til spørsmål!
- Løsning av eksamensoppgaver
- Løsningsforslag

**Menti for spørsmål:**

<https://www.menti.com/eo8obmgfb4>  
eller **4403 5823**

**Still gjerne spørsmål løpende her om du ønsker å være anonym :)**

## Oppgave 4.3 - Kravspesifikasjon for systemet

- Skriv minst **seks brukerhistorier** for systemet til Festklar. Nevn minst **tre forskjellige aktører**. Sett brukerhistoriene opp i en prioritert liste basert på hva du anser som viktigst for sluttproduktets funksjonalitet.
- Sett opp en liste over minst fire ikke-funksjonelle krav som dere ønsker å stille til systemet. Del opp kravene i produktkrav, organisatoriske krav og eksterne krav, og få med minst ett krav av hver type.
- Forklar hvordan hver av de ikke-funksjonelle kravene du presenterte i oppgave b kan testes.

## Oppgave 4.4 - Use case for systemet og modellering av use case «Lei plagg»

- Tegn et **use case-diagram som inkluderer alle nødvendige use case som trengs for å oppfylle kravene du spesifiserte i form av brukerhistorier i oppgave 4.3 a. Ta med alle involverte aktører (både primære og sekundære)**.
- Lag en **tekstlig beskrivelse for use case «Lei plagg»** ved bruk av nettsiden til Festklar.no. **Ha med aktører, eventuelle pre- og postbetingelser, hovedflyt og minst ett alternativ flyt.**
- Lag et sekvensdiagram for use case «Lei plagg» slik det er beskrevet i oppgave b. Ta med alternativ flyt.
- Lag et klassediagram som reflekterer use case i sekvensdiagrammet i oppgave c. Ta med assosiasjoner med multiplisitet mellom klassene, og metoder og attributter til hver klasse. Skriv gjerne egne forutsetninger.
- Modeller et aktivitetsdiagram for use case «Lei plagg».**

# Interessenter og aktører

**Aktør:** aktiv rolle, kommuniserer med systemet.

- *En aktør er en samlebetegnelse for å angi alle grupper av personer som anvender systemet, eller øvrige systemer som blir anvendt av systemet*
- *En aktør representerer en rolle som feks et menneske eller et annet system når det kommuniserer med dette systemet*
- *En aktør kommuniserer med systemet via ett eller flere use case*

**Interessent:** kommuniserer ikke nødvendigvis med systemet.

- *En betegnelse for alle personer, grupper eller organer, som påvirkes av eller påvirker systemets utvikling/bruk. Både direkte og indirekte*
- *En interessent kan være en person, gruppe, organisasjon eller institusjon*
- *For en interessent har utforming av en løsning eller utfallet av en hendelse betydning*

# Primær- vs sekundæraktør

**Primæraktør:** eget **mål** i kommunikasjonen med systemet.

**Sekundæraktør:** trengs for at primæraktøren skal nå målet, kommuniserer også aktivt med systemet.

# Hvordan finner vi aktørene?

## *Spørsmål man kan stille for å finne aktører:*

- Hvem skal bruke systemet?
- Hvem skal administrere systemet?
- Hvem
  - Tilbyr informasjon til systemet?
  - Bruker informasjonen fra systemet?
  - Fjerner informasjon fra systemet?
- Hvilke eksterne ressurser skal systemet bruke?
- Hvilke andre systemer skal kommunisere med dette systemet?

# Brukerhistorier

*Vi kjenner dem fra smidig utvikling hvor funksjonelle krav må kommunikeres lett.*

"Som <rolle> ønsker jeg <funksjon> for å oppnå <nytteverdi>"

1. Som kunde ønsker jeg å motta kjøpte billetter som pdf-fil, slik at jeg enkelt kan finne de fram
2. Som kundebehandler ønsker jeg å kunne hente ut oversikt over kundens bestillinger, slik at jeg kan bistå ved spørsmål fra kunden
3. Som kunde ønsker jeg å registrere bankkortet mitt på brukerkontoen slik at jeg kan betale raskere for kjøpene mine
4. Som eier ønsker jeg å se omsetning slik at jeg får en oversikt over inntekter

***Viser tydeligt de funksjonelle kravene.***

***Hver brukerhistorie må kunne omsettes til eget use case.***

# Use Case-diagram

**Menti for spørsmål:**

<https://www.menti.com/eo8obm>

[gfb4](#) eller **4403 5823**

**Still gjerne spørsmål løpende her  
om du ønsker å være anonym :)**

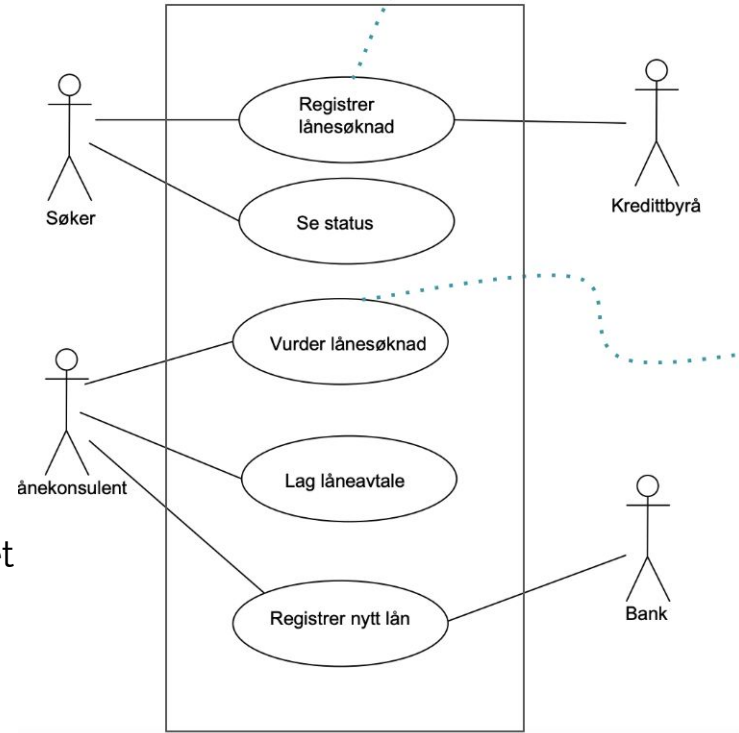


# Use case-diagram

- Interaksjonsperspektivet: en beskrivelse av hvordan systemet oppnår et mål av verdi for en aktør
  - Gjennom et enkelt use case eller gjennom et komplett use case som består av flere hendelsesforløp/flyt
- Et use case beskriver en komplett **funksjonell enhet**: one person - one place - one time. *Vær spesifikk.*
- Et use case **må være testbart**: går det an å gjennomføre i systemet?
  - En måte å teste om de funksjonelle kravene er oppfylt.
- Viser systemets funksjonalitet og samspillet mellom systemet og omgivelsene (brukere, andre systemer, komponenter)
- Bør ha yderligere informasjon som kan supplere diagram. Eks. Tekstlig beskrivelse av use case.

# Notasjon

- "Ramme" som viser hva som er i systemet og hva som er utenfor
- Aktører - tegnes som strekfigurer
  - Primæraktører settes på **venstre** side
  - (Eventuelle) sekundæraktører settes på **høyre** side
- Use case - tegnes som ovaler med "merkelapper" skrevet i verb-fraser
  - Settes gjerne i logisk rekkefølge nedover
- Heltrukne streker mellom aktører og use case (ikke piler)
- Piler med stiplet linje for <<include/extend>> mellom use case (vær obs på retningen på pilen)



# Use Case diagram

Hvilke mål har primæraktøren? → oval

Hvem hjelper til med å nå primæraktørs mål? → Sekundæraktør

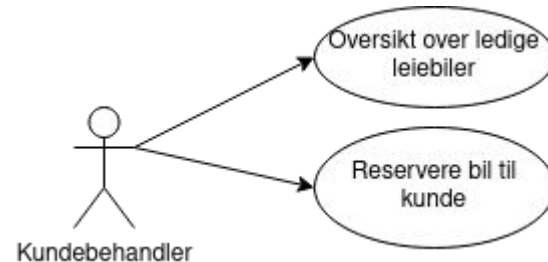
## Fra brukerhistorie til Use Case diagram:

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.

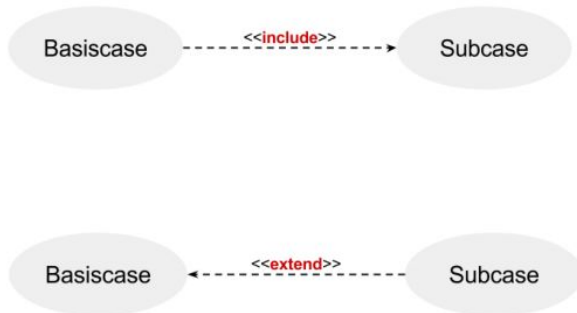
**Obs:** her er ikke med <nytteverdi> - men dere må ha det med på eksamen.



# Use Case-relasjoner

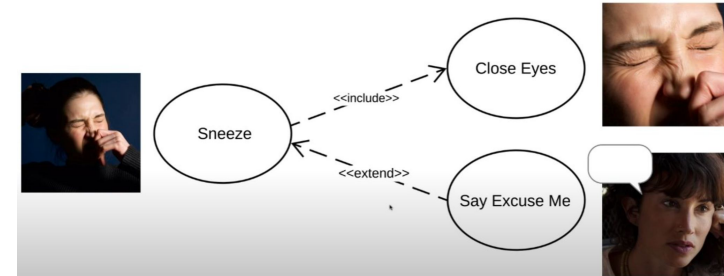
## Include

- Et use case kan være en del av ett eller flere andre use case
- Indikerer at et (sub) use case inneholder nødvendig funksjonalitet for gjennomførelsen av et annet basiscase
- Skjer alltid ved gjennomførelsen av basiscaset.



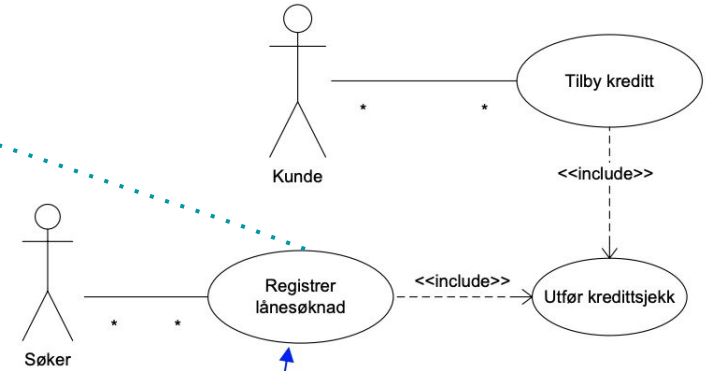
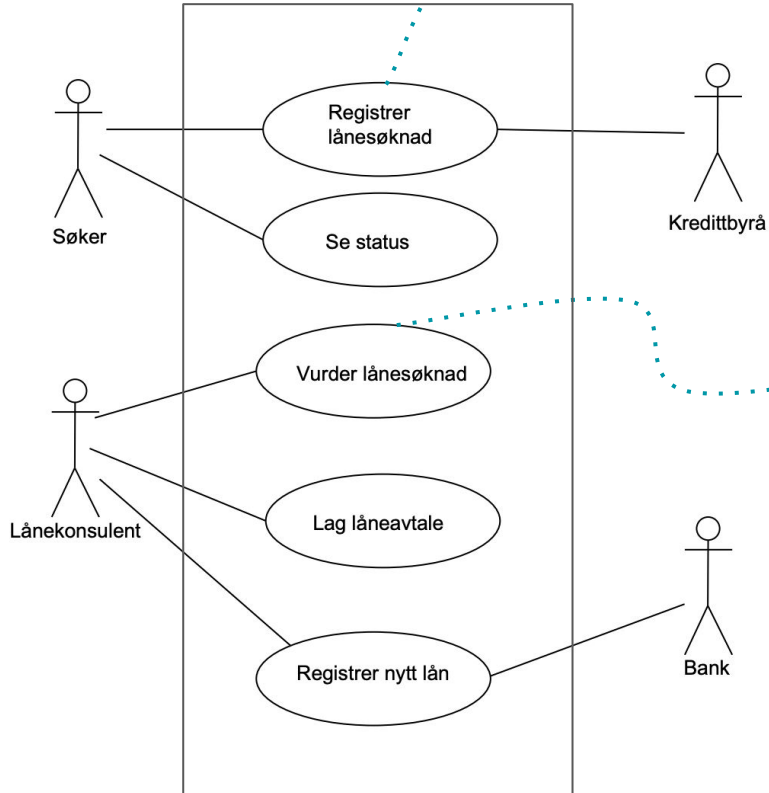
## Extend

- Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter
- Utvider oppførselen/funksjonalitet til et basiscase, som utføres under spesielle omstendigheter

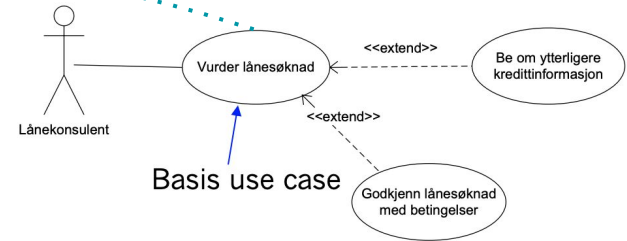


Skjermdump fra: <https://youtu.be/zid-MVo7M-E>

# Eksempel



Basis use case



Basis use case

Extend use case

# 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.

# Alternativ flyt

Hvis noe utenom det «vanlige» skjer, oppstår en alternativ flyt.

Dette fungerer på samme måte som i en tekslig beskrivelse.

**Refererer til hvilket steg** i hovedflyten det alternative flyt går ut fra - og hvilket steg det møter det igjen.

Eks. Du skal logge inn, men har ikke bruker. Da må du inn i et **alternativt løp** for å opprette bruker. Deretter kan du begynne på starten igjen, på use caseet logg inn.

# Betingelser

Fordi use cases er del av et system og er testbare, så er vi interessert i hvilke betingelser som må være til stede både før og etter testen.

## Prebetingelse

- Betingelse(r) som må være til stede i systemet før use case kan begynne.
  - Eksempelvis: pantmaskinen som er tilbake til start og klar til å mottake pant

## Postbetingelse

- Betingelse som må være tilstede når use case slutter.
  - Eksempelvis: Kunde får kvittering eller lodd fra pantmaskinen



# Aktivitetsdiagram

**Menti for spørsmål:**

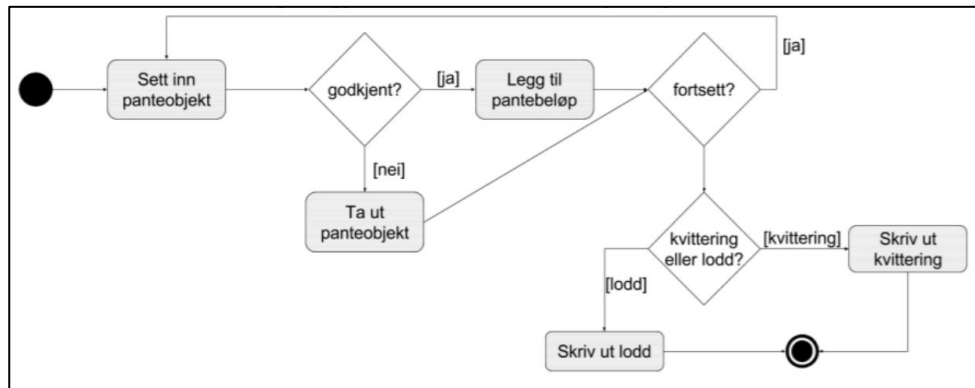
<https://www.menti.com/eo8obm>

[gfb4](#) eller **4403 5823**

**Still gjerne spørsmål løpende her  
om du ønsker å være anonym :)**

# Aktivitetsdiagram

- Grafisk representasjon av **arbeidsflyt**
- **Aktiviteter** og tilhørende **handlinger** (actions)
- Viser overordnet **kontrollflyt** i systemet mellom aktiviteter
- Beskriver hvordan mulige **utfall** av en **aktivitet påvirker flyten**
- Viser hvilke **aktiviteter** som kan utføres **parallelt**



Løsningsforslag, ukesoppgaver  
uke 12

# Aktivitetsdiagram

## Karakteristikk:

- Aktivitetsdiagrammer er datadrevet
- Vi modellerer aktivitetene som må gjøres for å fullføre en gitt prosess
- Diagrammet viser hvordan et system går fra en aktivitet til en annen
- Det viser hvilken input fra bruker som leder til ulike utfall og aktiviteter
- Systemet reagerer på input
- Output påvirker aktivitetsflyt
- Det viser avhengighetsforhold mellom aktiviteter
- Hvilke aktiviteter som kan kjøres i parallell

## Det er nyttig å lage aktivitetsdiagram:

- For å se flyten til en gitt aktivitet.
- For å finne prosesser som kan kjøres parallelt fordi de er uavhengige av hverandre.
- For å få forstå arbeidsrutiner bedre.

# Notasjon

**Start:** angir hvor flyten starter

Full sirkel

**Slutt:** angir hvor flyten ender

Full sirkel med ring rundt

**Aktiviteter:** angir ulike aktiviteter som inngår i arbeidsflyten

- Representeres med navngitte, avrundede rektangler
- Kan være fysisk (“godkjenn søknad”) eller elektronisk (“vis kjøpshistorikk”)

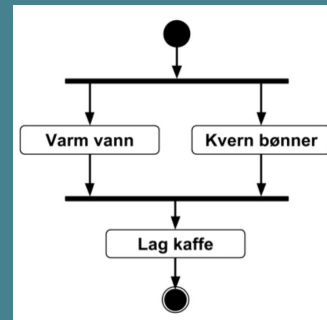
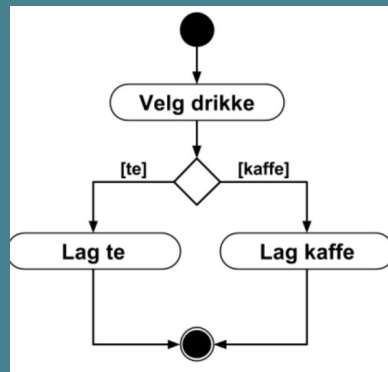
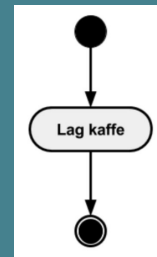
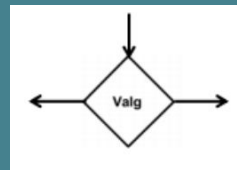
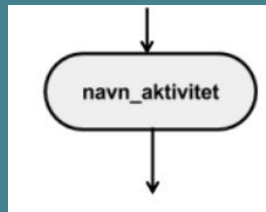
**Valg:** angir at man står ovenfor et valg (decision)

Eksempel: IF, IF-ELSE, CASE, osv.

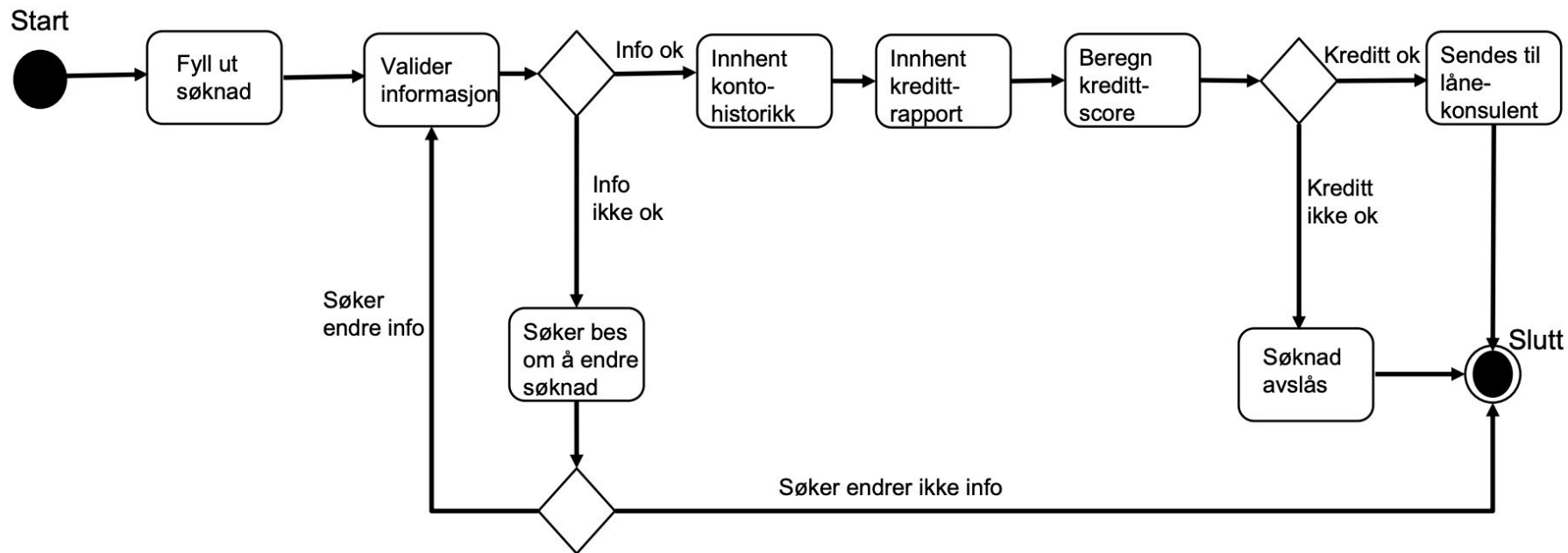
Valgdiamant

**Blokkeringer (bar):**

- Representerer start (split) og slutt (join) for parallelle prosesser
- Viser hvilke prosesser man må vente på, før man kan gå videre



# Aktivitetsdiagram



# Menti

**Menti for spørsmål:**

<https://www.menti.com/eo8obm>

gfb4 eller **4403 5823**

# Oppgave

- A) Forsøk å løs/lag tankekart over potensiell løsning på eksamensoppgaven (du trenger ikke å bli helt ferdig!). *15 min*
- B) Gå sammen med sidemann - tal om løsningene deres. Hvordan har dere hver løst oppgaven? *15 min*
- C) Gjennomgang og diskusjon i plenum

**Menti for spørsmål:**

<https://www.menti.com/eo8obmgfb4>

eller **4403 5823**

**Still gjerne spørsmål løpende her om du ønsker å være anonym :)**

## Oppgave 4.3 - Kravspesifikasjon for systemet

- Skriv minst seks brukerhistorier for systemet til Festklar. Nevn minst tre forskjellige aktører. Sett brukerhistoriene opp i en prioritert liste basert på hva du anser som viktigst for sluttproduktets funksjonalitet.
- Sett opp en liste over minst fire ikke-funksjonelle krav som dere ønsker å stille til systemet. Del opp kravene i produktkrav, organisatoriske krav og eksterne krav, og få med minst ett krav av hver type.
- Forklar hvordan hver av de ikke-funksjonelle kravene du presenterte i oppgave b kan testes.

## Oppgave 4.4 - Use case for systemet og modellering av use case «Lei plagg»

- Tegn et use case-diagram som inkluderer alle nødvendige use case som trengs for å oppfylle kravene du spesifiserte i form av brukerhistorier i oppgave 4.3 a. Ta med alle involverte aktører (både primære og sekundære).
- Lag en tekstlig beskrivelse for use case «Lei plagg» ved bruk av nettsiden til Festklar.no. Ha med aktører, eventuelle pre- og postbetingelser, hovedflyt og minst ett alternativ flyt.
- Lag et sekvensdiagram for use case «Lei plagg» slik det er beskrevet i oppgave b. Ta med alternativ flyt.
- Lag et klassediagram som reflekterer use case i sekvensdiagrammet i oppgave c. Ta med assosiasjoner med multiplisitet mellom klassene, og metoder og attributter til hver klasse. Skriv gjerne egne forutsetninger.
- Modeller et aktivitetsdiagram for use case «Lei plagg».



## Oppgave 4.3 - Kravspesifikasjon for systemet

- Skriv minst **seks brukerhistorier** for systemet til Festklar. Nevn minst **tre forskjellige aktører**. Sett brukerhistoriene opp i en prioritert liste basert på hva du anser som viktigst for sluttproduktets funksjonalitet.
- Sett opp en liste over minst fire ikke-funksjonelle krav som dere ønsker å stille til systemet. Del opp kravene i produktkrav, organisatoriske krav og eksterne krav, og få med minst ett krav av hver type.
- Forklar hvordan hver av de ikke-funksjonelle kravene du presenterte i oppgave b kan testes.

## Oppgave 4.4 - Use case for systemet og modellering av use case «Lei plagg»

- Tegn et **use case-diagram som inkluderer alle nødvendige use case som trengs for å oppfylle kravene du spesifiserte i form av brukerhistorier i oppgave 4.3 a. Ta med alle involverte aktører (både primære og sekundære).**
- Lag en **tekstlig beskrivelse for use case «Lei plagg»** ved bruk av nettsiden til Festklar.no. **Ha med aktører, eventuelle pre- og postbetingelser, hovedflyt og minst ett alternativ flyt.**
- Lag et sekvensdiagram for use case «Lei plagg» slik det er beskrevet i oppgave b. Ta med alternativ flyt.
- Lag et klassediagram som reflekterer use case i sekvensdiagrammet i oppgave c. Ta med assosiasjoner med multiplisitet mellom klassene, og metoder og attributter til hver klasse. Skriv gjerne egne forutsetninger.
- Modeller et aktivitetsdiagram for use case «Lei plagg».**

# (vores lille) Løsningsforslag

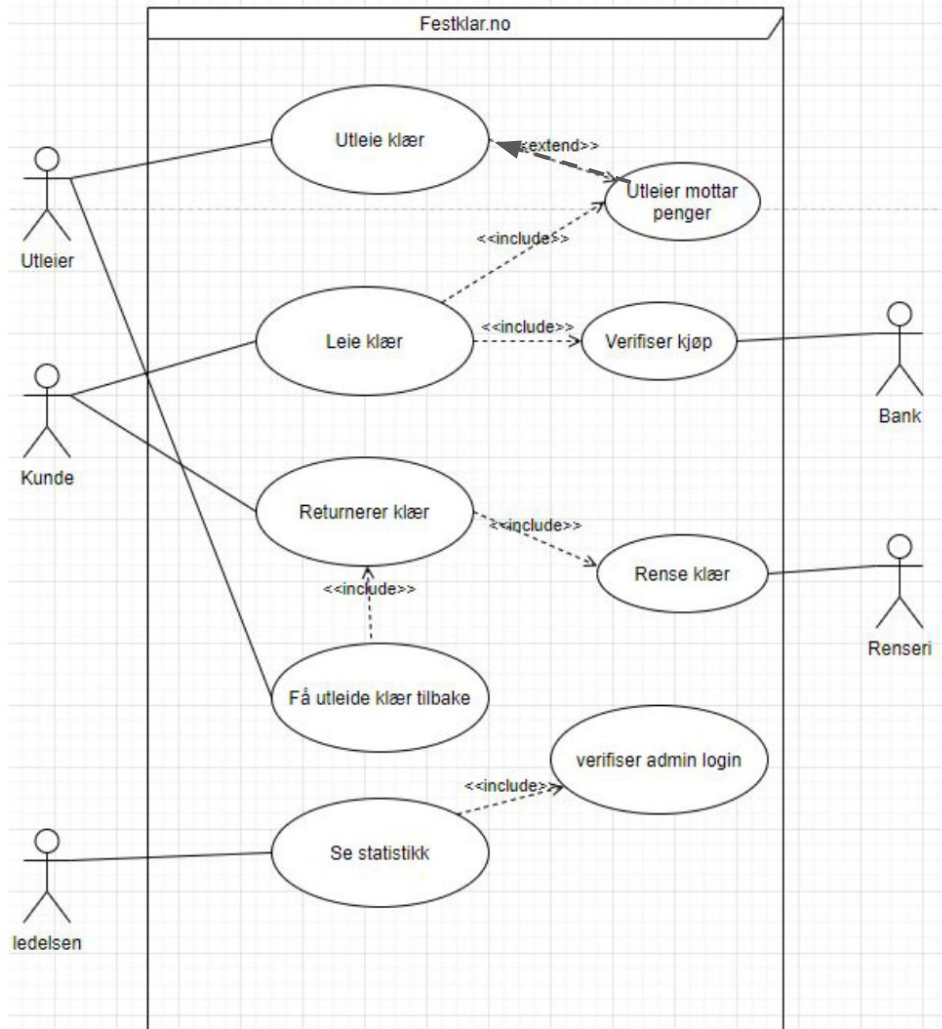
4.3.a) (Husk å skrive det antall som blir spurt om)

Som leier/kunde ønsker jeg å leie klær for å kunne ha noe fint på meg på fest

Som utleier ønsker å leie ut klær for å tjene litt ekstra penge

# Use Case diagram

4.4.a)



# Tekstlig beskrivelse

4.4.b)

**Case:** Lei plagg

**Aktør(er):** Kunde

**Prebetingelser:** Ledige plagg i systemet.

**Postbetingelser:** Kunde får leiet plagg.

*(Bare 2 alternative flyt med her)*

**Alternative flyt:**

A1.1: Feil login

A1.2: Tilbake til startside

A2.1: Systemet meddeler at kunder ikke finnes i systemet

A2.2: Kunde oppretter bruker

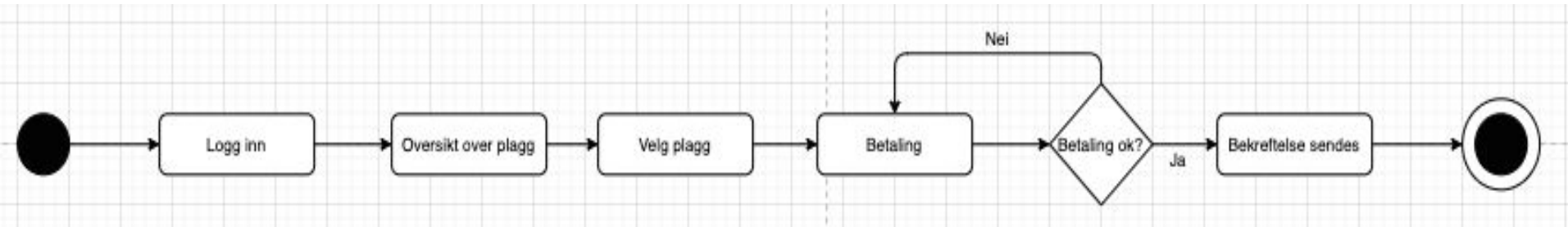
A2.3: Tilbake til startside

**Hovedflyt:**

1. Kunde logger seg inn i system
2. System viser oversikt av ledige plagg
3. Kunde klikker inn på et plagg
4. System viser detaljer om plagg
5. Kunde bekrefter og går til betaling
6. System behandler betalingsinfo
7. System registrerer reservasjon/kjøp
8. System sender bekreftelse til kunden
9. Transaksjon avsluttes og system går tilbake til startsidan.

# Aktivitetsdiagram

4.4.e)



# Takk for i dag!

Spørsmål?

Send meg gjerne en epost på *[nhmoller@uio.no](mailto:nhmoller@uio.no)*

# Lykke til på eksamen!