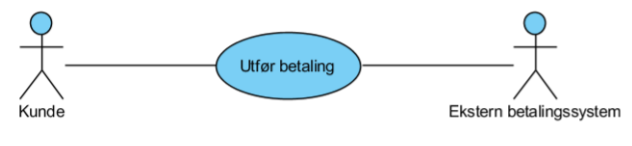
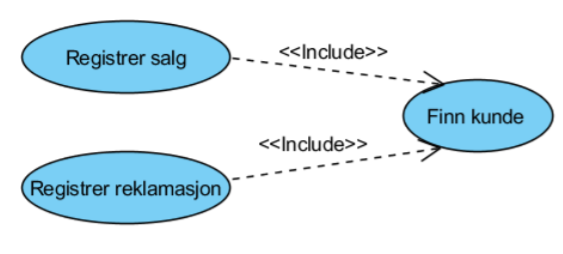
UML

# Use Case (Brukstilfelle)

Et use case beskriver en typisk og spesifikk brukssituasjon av systemet. Ta med verb og hvem. En aktør er typisk en bruker av et system.



Man kan detaljere diagrammet mer med <<include>> og <<extend>>. I eksemplet er det oppgitt at man må finne en kunde for å registrere et salg.

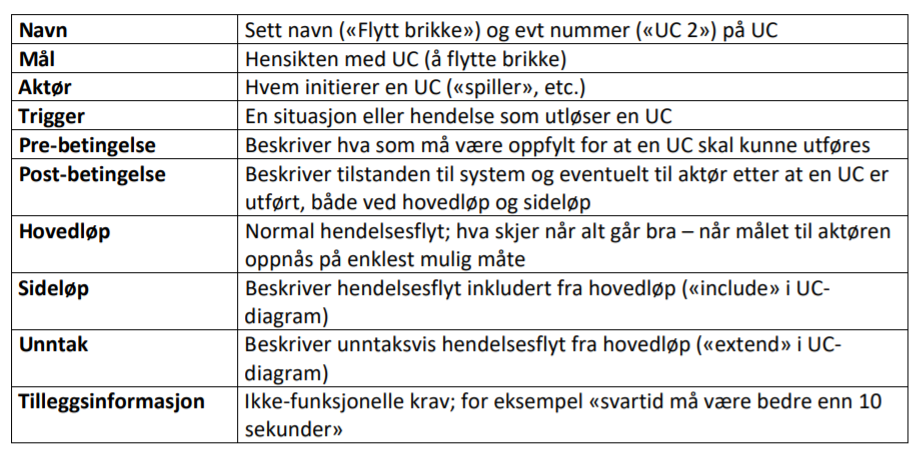


I <<extend>> sier man at man har et tilleggs-case som utvider base-caset. Merk at pilen går andre veien.

*Systemgrense*: en boks rundt komponentene i et system, slik at man at kan skille mellom flere systemer.

**Tekstlig beskrivelse**

Man beskriver hvert use case for å beskrive kravene til systemet. Eks. på mal:

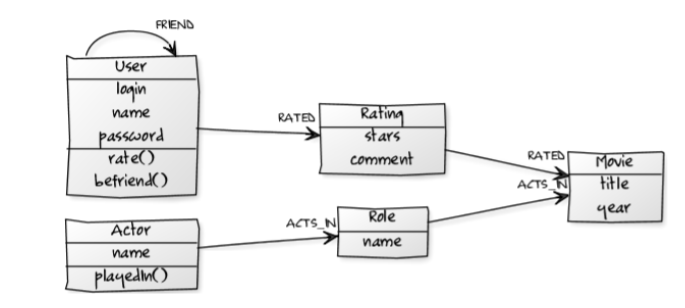


# Domenemodellen

Vi finner **entiteter** i et **problemdomene** og viser forholdet mellom dem:

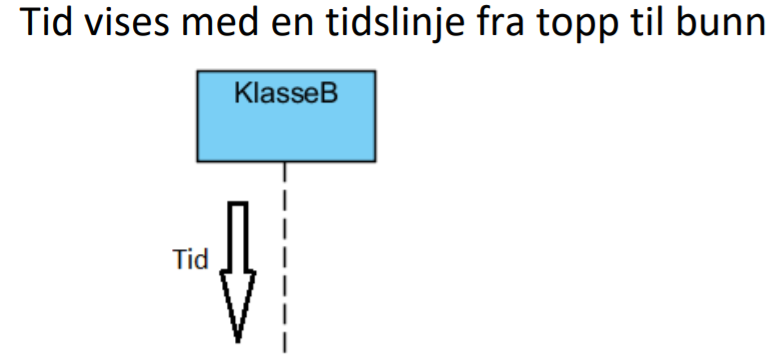
*Entitet*: subjekter, objekter i den virkelige verden. F.eks. en konto eller kunde i et banksystem.

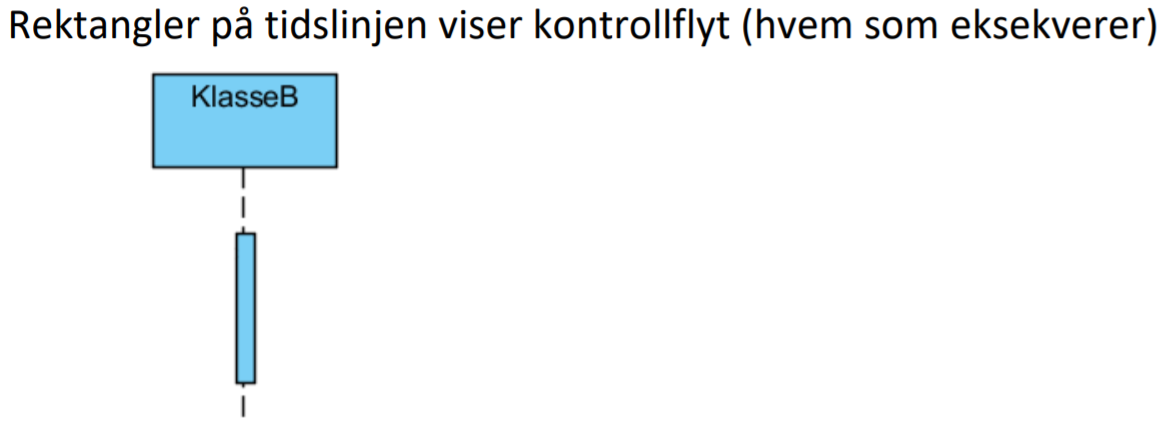
Den skal ikke være detaljert, den skal vise konseptene i et domene på en oversiktlig måte.

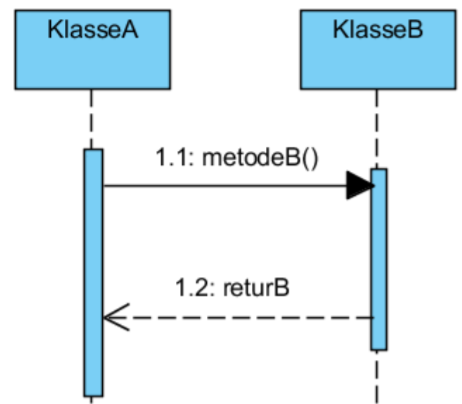


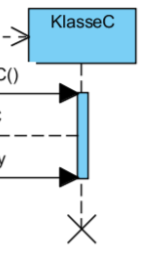
# Sekvensdiagram

Brukes til å illustrere kode i et oversiktlig diagram. Viser kontrollflyten i programmet. Kan også brukes til å illustrere hvordan systemer samarbeider.





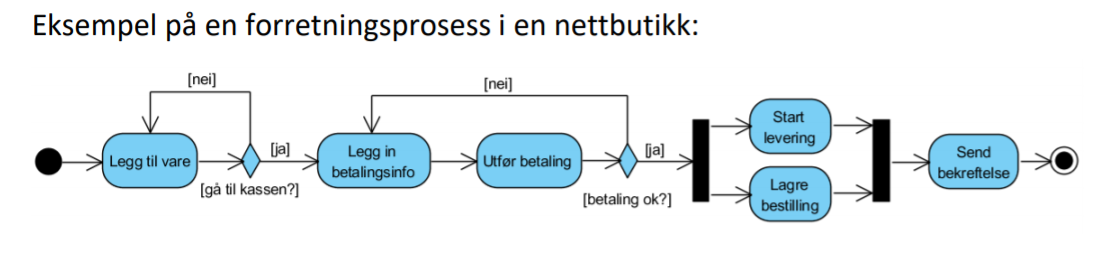
**Meldinger** vises med heltrukken linje og tekst. Hvis dette er på klassenivå vil det være en metode. Returmeldinger vises med en stiplet linje og tekst, slik som illustrasjonen til høyre.

**Destruksjon av objekter**

Hvis et objekt skal «destrueres» kan man representere dette med et kryss på tidslinjen.

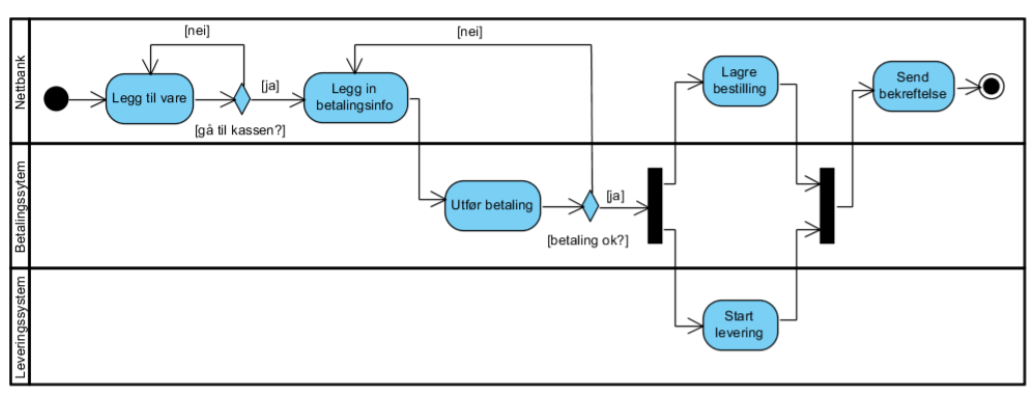
# Aktivitetsdiagram

Skal vise aktiviteter i en prosess. Hvilken rekkefølge de må utføres i, hvilke som kan gjøres i parallell og eventuelle betingelser for at de skal utføres. Parallelle aktiviteter representeres med en «stolpe», hvor det «splittes» og «joines».



**Swim Lanes**

Brukes for å representere hvilket system som utføre hvilken aktivitet i diagrammet. Et mer komplisert aktivitetsdiagram enn ovenfor.



# Tilstandsmaskin

For å illustrere hvilke tilstander et objekt kan være i og hva som gjør at de endrer tilstand. Et slikt objekt er typisk en entitet i et domene. Vi kan for eksempel se på **status** til en bestilling i en **nettbutikk**.

**Komponenter**

