
Sekvens og klassediagram

Eksamensforberedelse

IN1030

Plan for dagen

- UML
- Sekvensdiagram
- Klassediagram

Hva er UML?

- UML (Unified Modeling Language) - er et standardisert språk for å representere struktur og oppførsel av et system.
- Blir brukt innen programvareutvikling for å modellere og dokumentere designet av et system.

**UNIFIED
MODELING
LANGUAGE™**



Hvorfor lager vi UML diagram?

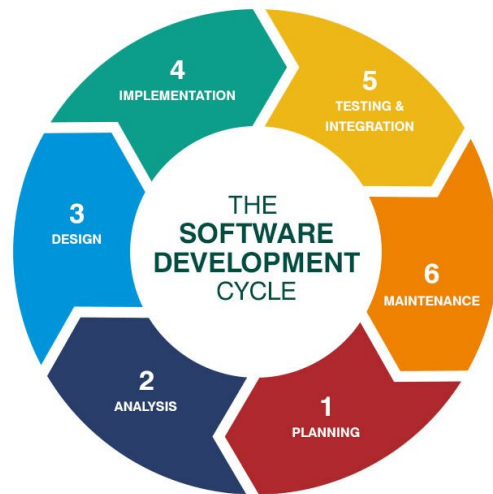
Hvorfor lager vi UML diagram?

- For å identifisere krav
- For å være enige om hvordan endelig løsning skal bli
- For å kunne vedlikeholde systemet
- For at andre skal kunne ta over vedlikehold av systemet
- For at vi skal kunne lage et godt systemdesign



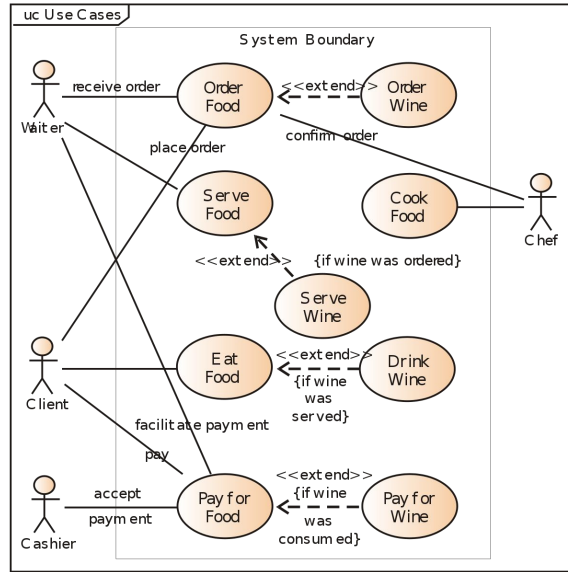
Når lager vi UML diagram?

- I kravanalysen
 - ◆ For å få en felles forståelse av systemet
- I designprosessen
 - ◆ For å beskrive et tenkt system
- Etter implementasjon av systemet
 - ◆ For å dokumentere eksisterende system



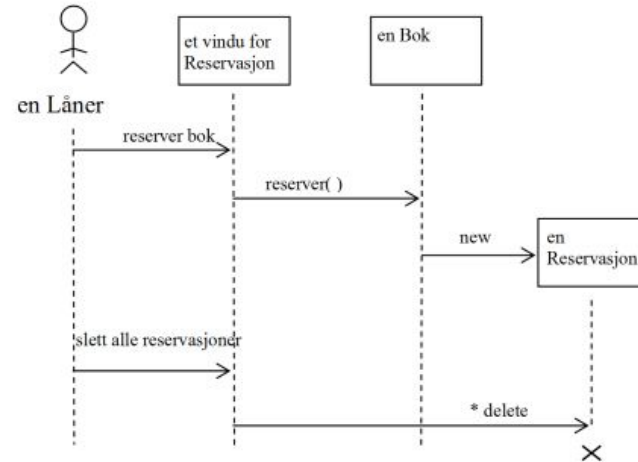
Hvilke 4 diagram skal dere
kunne?

Use case diagram



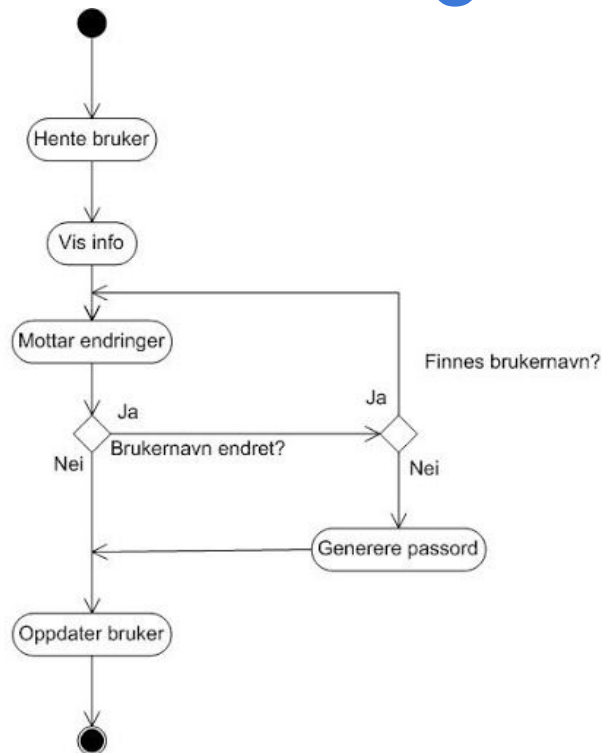
Viser interaksjon mellom et system og omgivelsene (aktører). Tar utgangspunkt i primæraktørs mål og hvordan sekundæraktører assisterer dette målet gjennom systemet.

Sekvensdiagram



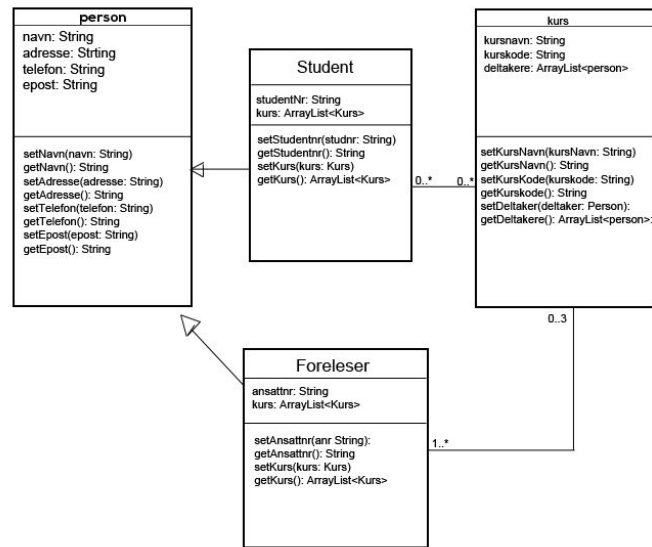
Viser interaksjon og informasjonsflyten mellom aktørene og systemet og systemkomponentene i form av objektklasser.

Aktivitetsdiagram



Viser aktivitetsflyten i en prosess.

Klassediagram



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

Spørsmål?

Interessent vs aktør

→ Interessent

- ◆ En som generelt blir påvirket av systemets implementasjon.

→ Aktør

- ◆ En som har en aktiv rolle med systemet og faktisk anvender det.

→ Hva er forskjellen på aktør og interessent?

→ Vil en aktør alltid være en interessent?

→ Vil en interessent alltid være en aktør?



Primær vs sekundær aktør

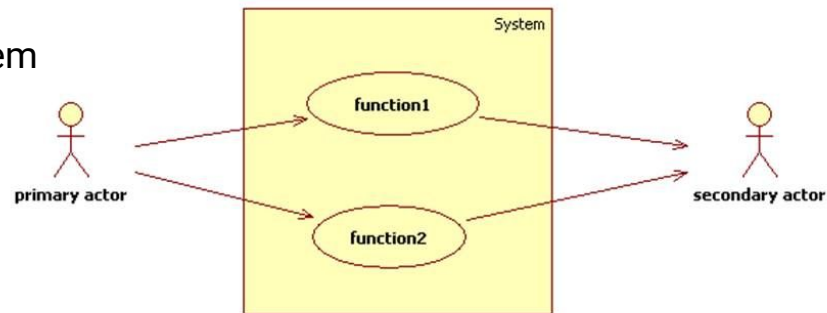
→ Primær aktør:

- ◆ Har et eget mål i kommunikasjon med systemet
 - Eksempel: bruker av banksystem

→ Sekundær aktør

- ◆ Trengs for at primæraktøren skal kunne nå målet sitt - kommuniserer også med systemet
 - Eksempel: banktjenesten sin kundeservice

- ❖ Aktører trenger ikke være en person - kan være et system

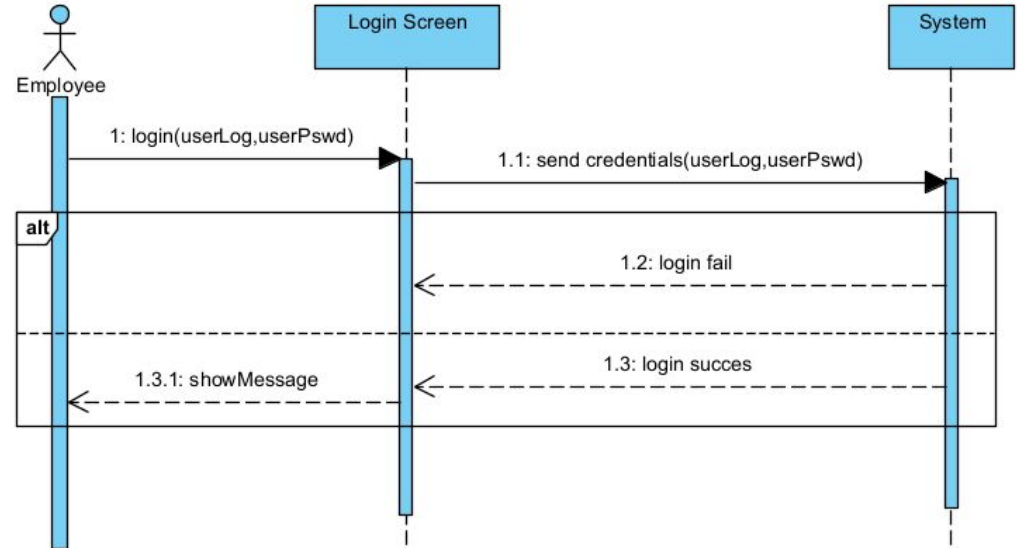


Sekvensdiagram

Sekvensdiagram

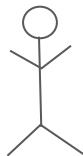
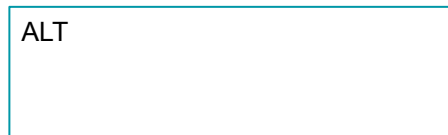
- Brukes for å se hvordan objekter og komponenter interagerer med hverandre over tid
- Lages gjerne når systemet skal designes og før implementering - etter kravspec
- Teknisk avbildning av interaksjonen

Aktøren interagerer med systemet og vi ser hvilke metodekall som gjøres i de ulike klassene.



Sekvensdiagram notasjon

- **Metodekall** (med parametre): heltrukket pil
- **Returverdi**: stiplet pil
- **Alternativ flyt**: settes i en "boks"
- **Aktører**: Strekmennesker
- **Klasser/Objekter**: Bokser
- **Lifeline**: Vertikale bokser



Tips til modellering av sekvensdiagram

- **Identifiser de ulike aktørene/objektene.**
- **Lag et tenkt, tekstlig oppsett basert på hovedflyt.**
 - Jo mer detaljert, desto enklere blir det å modellere
- **Modeller steg for steg, basert på stegene i hovedflyten.**
- **Inkluder alternativ flyt etter at du har laget en modell for hovedflyten.**

Navn: Kjøp billett som eksisterende kunde

Primæraktør: Kunde

Sekundæraktør: -

Prebetingelse: Kunden har internett

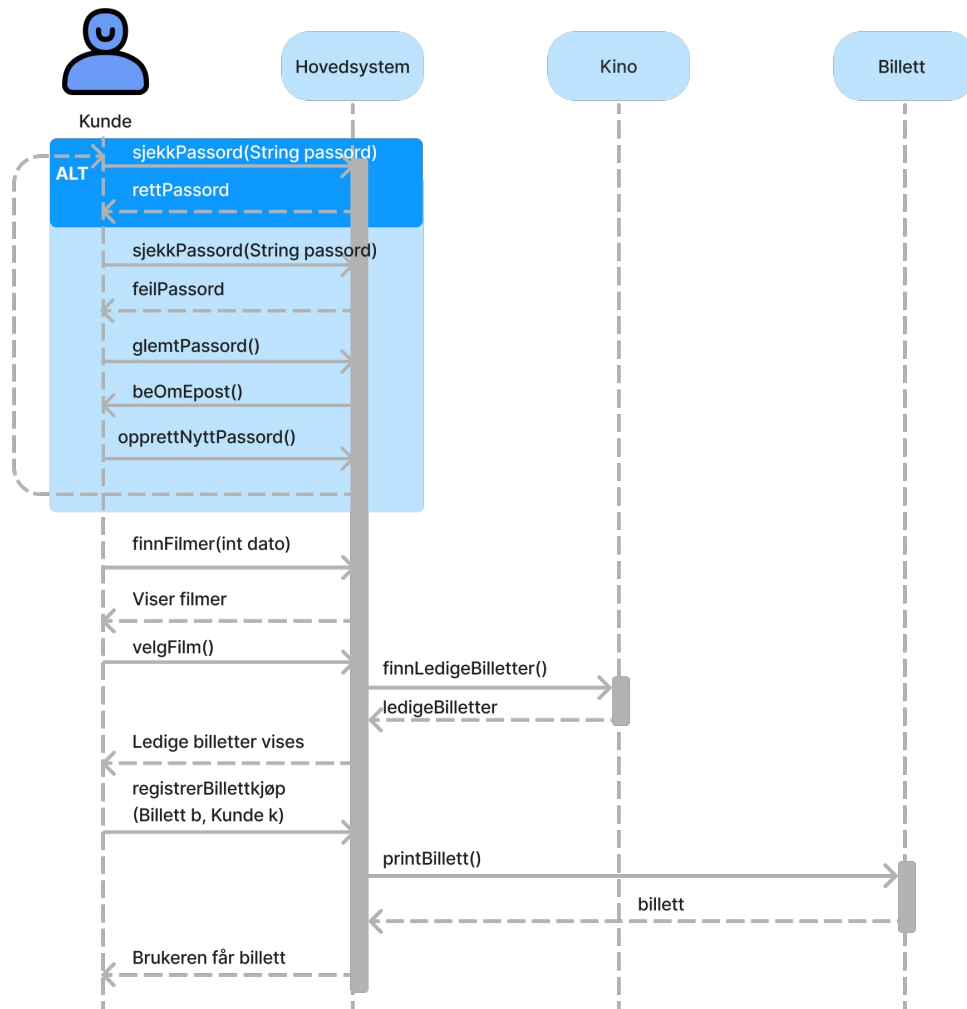
Postbetingelse: Kunde får billett

Hovedflyt:

1. Kunde åpner nettsiden og kommer til innloggingssiden
2. Kunde skriver inn riktig brukernavn og passord og trykker "Logg inn"
3. Systemet slipper inn kunde på sin kundeprofil
4. Kunde går inn på filmoversikt
5. Systemet returnerer en liste over tilgjengelige filmer
6. Kunde trykker på ønsket film
7. Systemet returnerer en liste over tilgjengelige billetter
8. Kunde velger ønsket billett og kjøper med forhåndsinnlagt betalingsinformasjon på sin konto
9. Billetten blir lagt til i kundes liste over billetter og tatt ut av kinoens og hovedsystemets liste over ledige billetter
10. Billett blir lagt inn som PDF i kundes billettoversikt

Alternativ flyt punkt 2:

- 2.1: Kunde husker ikke passord
- 2.2: Kunde trykker på "Glemt passord"
- 2.3: Systemet ber om e-post fra kunde
- 2.4: Kunde får tilsendt link for å opprette nytt passord
- 2.5: Kunde lager nytt passord og logger inn, og returnerer til [steg 3](#)



Eksamensoppgave

Krav, modellering og sikkerhet

Følgende krav ble vurdert som sentrale etter at brukerundersøkelsen var blitt gjennomført (Oppgave 1):

K1: Audio fra en konsultasjon mellom lege og pasient skal tas opp og deretter transkriberes på en lokal maskin.

K2: Lege skal kunne starte og stoppe et opptak fra en konsultasjon.

K3: Transkripsjon skal starte automatisk etter at et opptak er stoppet.

K4: Oppsummering av konsultasjon skal starte automatisk etter at transkripsjon er ferdigstilt.

K5: En lege skal i samråd med pasient kunne redigere og eventuelt forkaste oppsummeringen før denne eventuelt lagres i pasientjournalen.

K6: Transkripsjonen skal permanent slettes senest 24 timer etter endt konsultasjon.

Med “transkripsjon” menes at lyd lagret på et passende filformat (f.eks. .wav eller .mp3) oversettes til maskinlesbar tekst (f.eks. .txt).

Sekvensdiagram

I denne oppgaven kan du svare med digital håndtegning. Bruk eget skisseark (utdelt). Se instruksjon for utfylling av skisseark i lenken under oppgavelinjen.

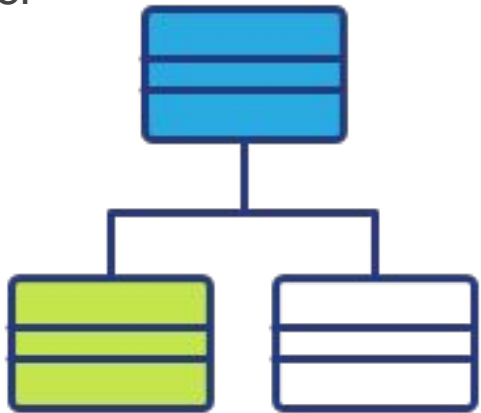
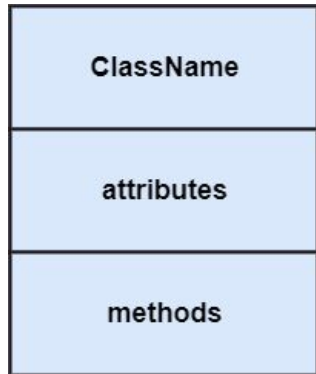
d) Modelér kravene K1–K5 i et sekvensdiagram. Anta at ingen av de tekniske komponentene som inngår i løsningen, feiler og at pasienten har samtykket til opptak.

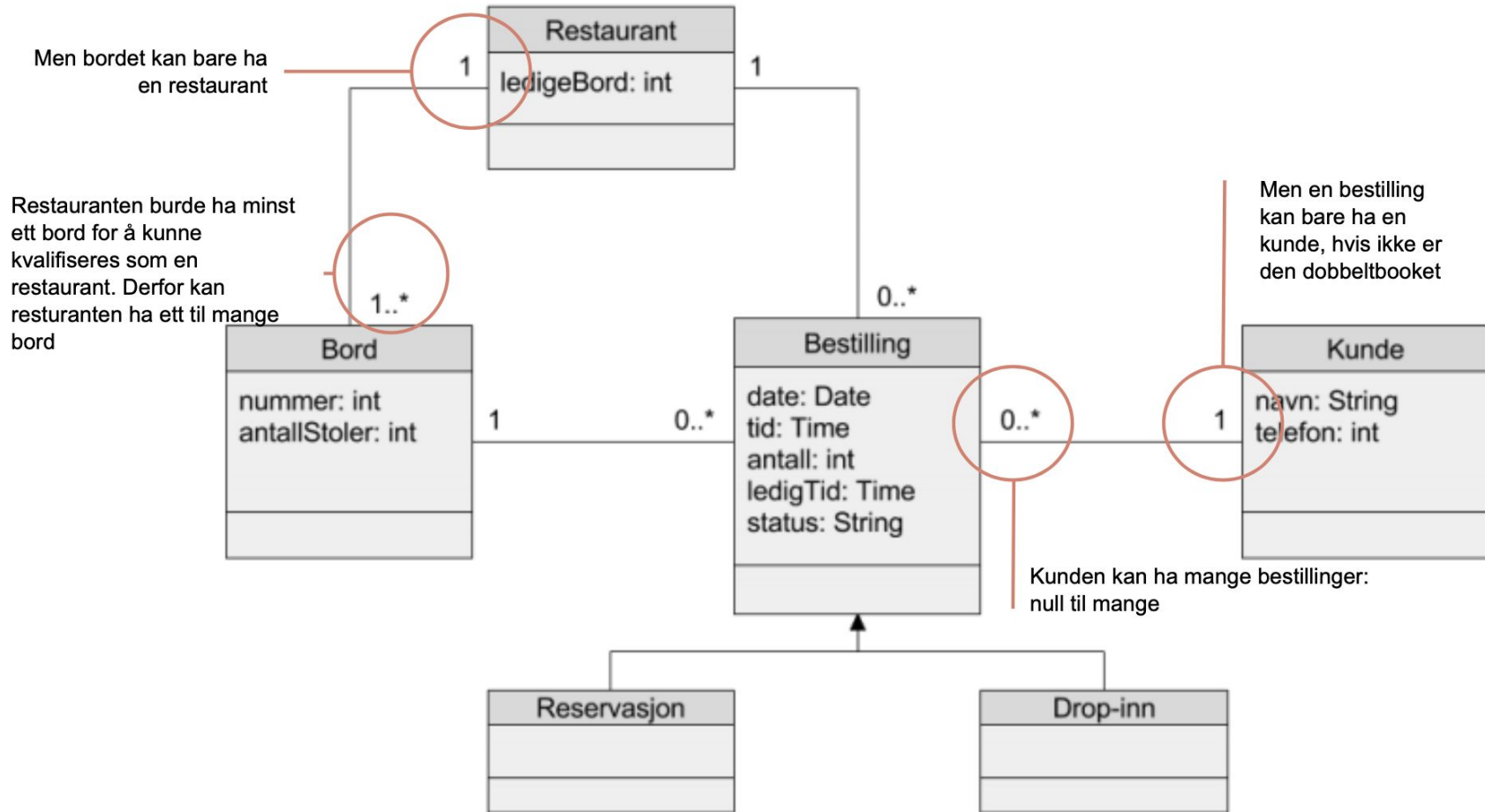
Tips: fokusér på lege som aktør (det er legen som er innlogget i systemet). I tillegg kan du bruke en klasse Konsultasjon med metodene `start_opptak()` og `stopp_opptak()`. Hvis nødvendig, gjør egne antagelser, men gjør i så fall rede for disse.

Klassediagram

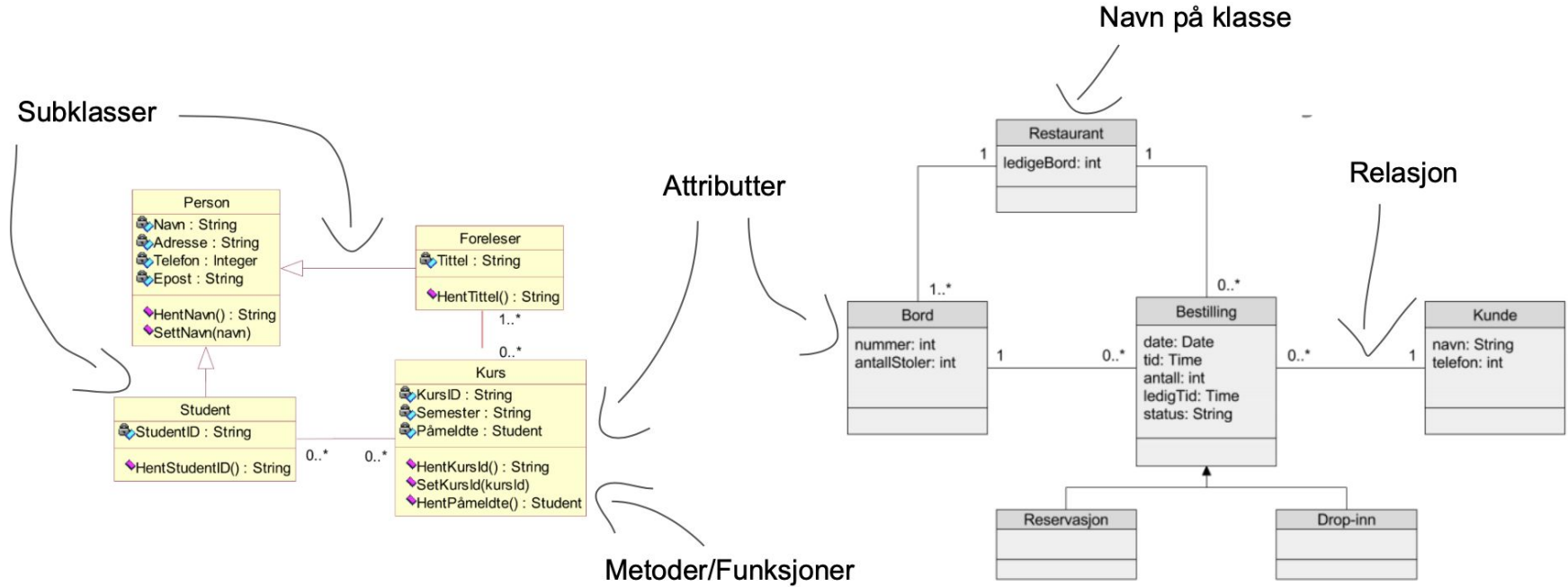
Klassediagram

- Strukturmodell som viser strukturen i et system
- Viser objektklassene i systemet og assosiasjonene mellom disse klassene
- Viktig å lage for å få oversikt over et system
- **Assosiasjon**: en link mellom klasser som indikerer at det er en relasjon mellom dem
 - ◆ 1 (nøyaktig én)
 - ◆ 0 .. 1 (null eller én)
 - ◆ * eller 0..* (null eller mer)
 - ◆ 1 ..* (én eller mer)

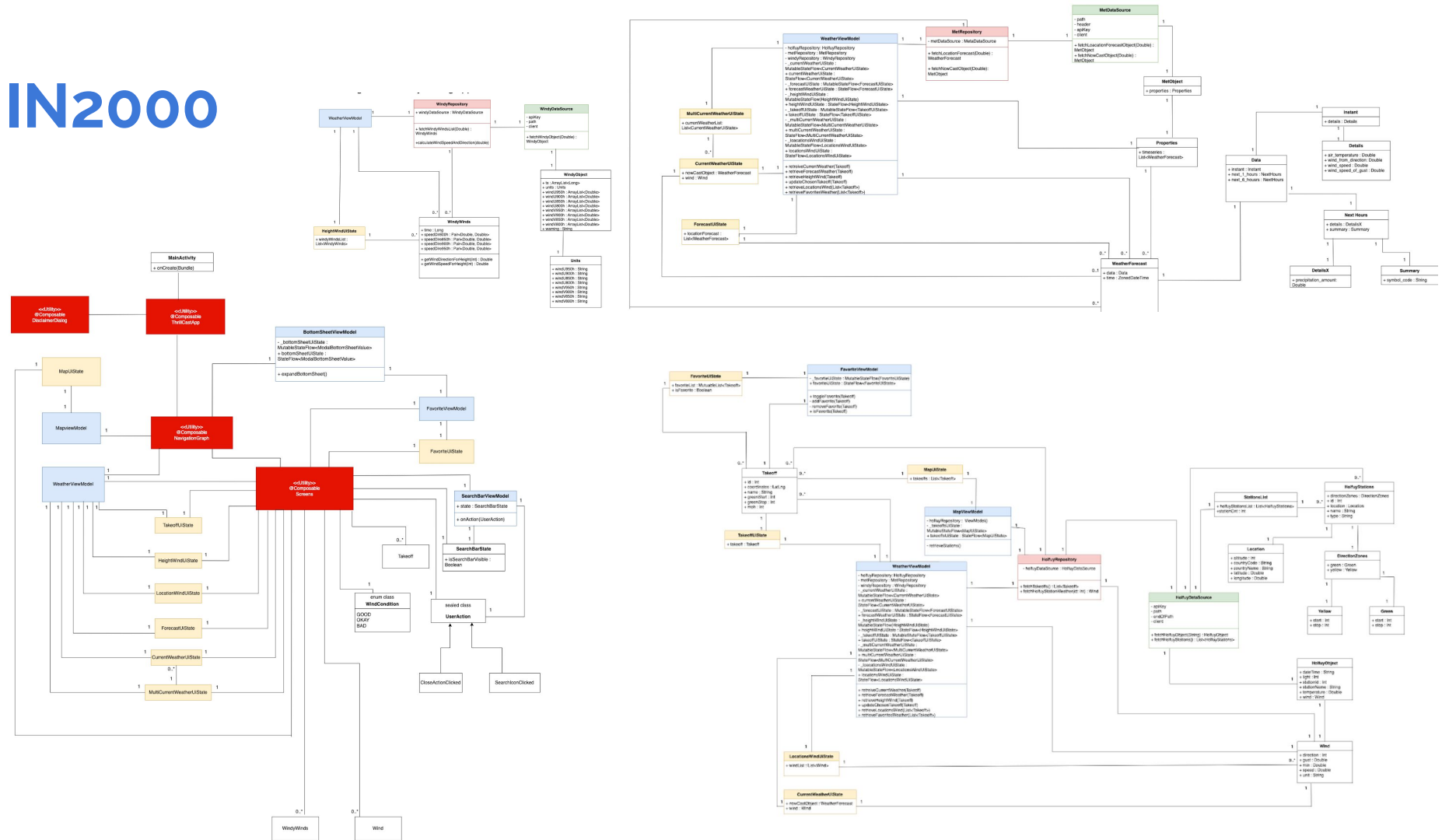




Notasjon



IN2000



Noen spørsmål?

Hva er hovedforskjellen mellom et klassediagram og et sekvensdiagram?

- **A.** Et klassediagram viser tidssekvensen av hendelser, mens et sekvensdiagram viser strukturen og relasjonene mellom klasser.
- **B.** Et klassediagram viser strukturen og relasjonene mellom klasser, mens et sekvensdiagram viser tidssekvensen av hendelser.
- **C.** Et klassediagram viser interaksjoner mellom brukere (aktører) og systemet, mens et sekvensdiagram viser de interne relasjonene mellom klasser.
- **D.** Et klassediagram viser brukerinteraksjoner, mens et sekvensdiagram viser databaseoppsett.

Når i en utviklingsprosess brukes et klassediagram?

- A. Under kravinnhentingsfasen for å forstå brukerkravene.
- B. Under designfasen for å modellere systemets statiske struktur og relasjoner mellom klasser.
- C. Under implementasjonsfasen for å teste systemet.
- D. Etter systemet er ferdig, for å dokumentere systemets struktur og støtte eventuell gjenbruk.

Når i en utviklingsprosess brukes et sekvensdiagram?

- A. Under kravinnhentingsfasen for å dokumentere brukerkravene.
- B. Under designfasen for å vise interaksjonene mellom objekter over tid.
- C. Under implementasjonsfasen som en referanse for å skrive koden.
- D. Under leveringsfasen for å pakke og distribuere programmet.

Noen spørsmål?

Noen fine videoer å se før dere modellerer

- Sekvensdiagram
- Klassediagram

Spørsmål?

Ikke nøl med å sende oss en mail:)

erikaesk@uio.no

sandefo@uio.no
