

Systemutvikling

Obligatorisk oppgave 2

Modellering av krav

GRUPPE 27

Adrian Westlund – s198571

Hallbjørn Storruste – s165519

Oppgave 1

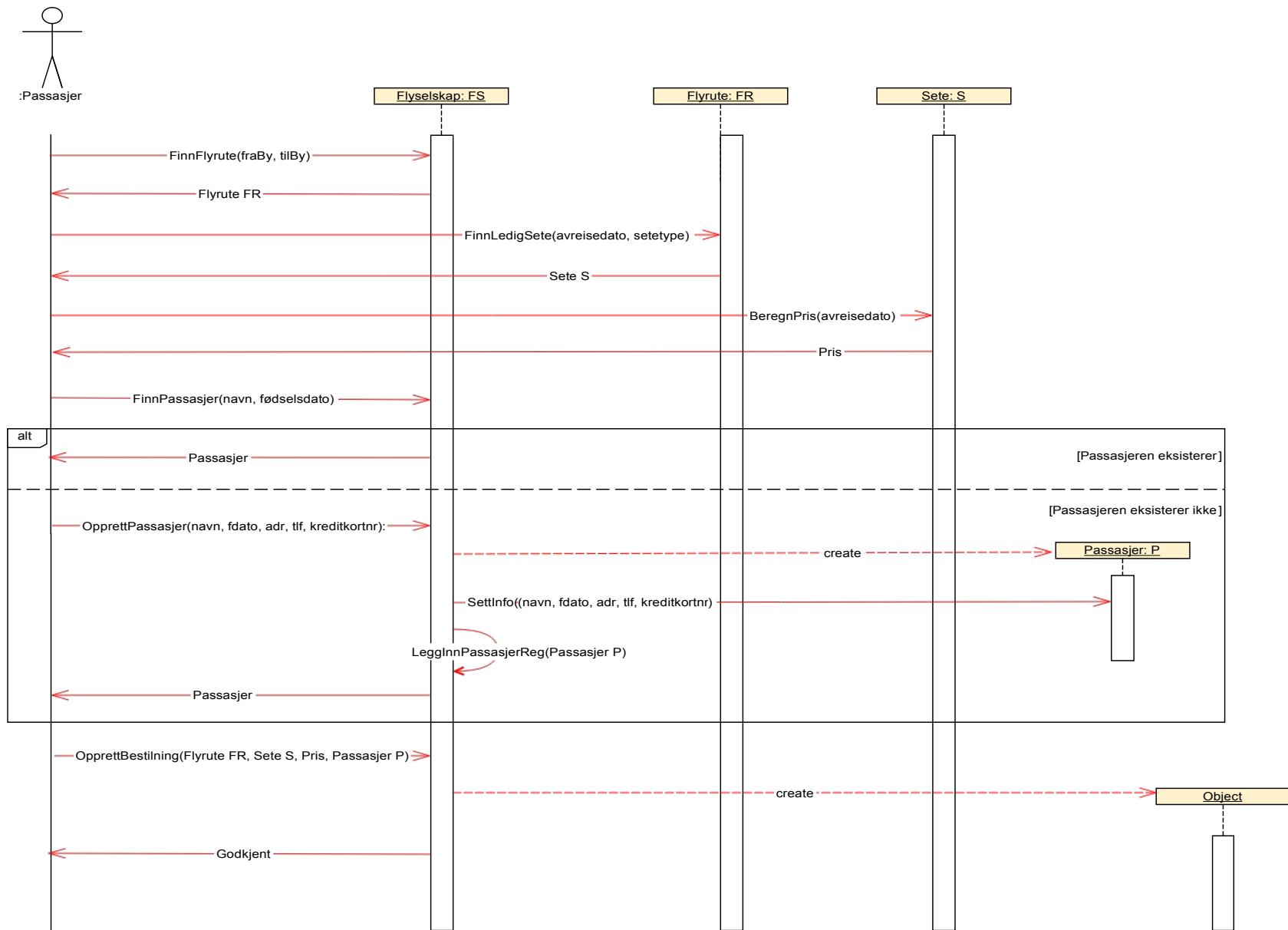
Vi har gjort følgende utvidelse av klassene:

Flyselskap har fått et passasjerregister(hashmap<passasjer>) og to nye metoder leggInnPassasjerReg(Passasjer P) returnerer en boolean og opprettBestilling(flyrute FR, Sete S, Pris, Passasjer P) returnerer en boolean.

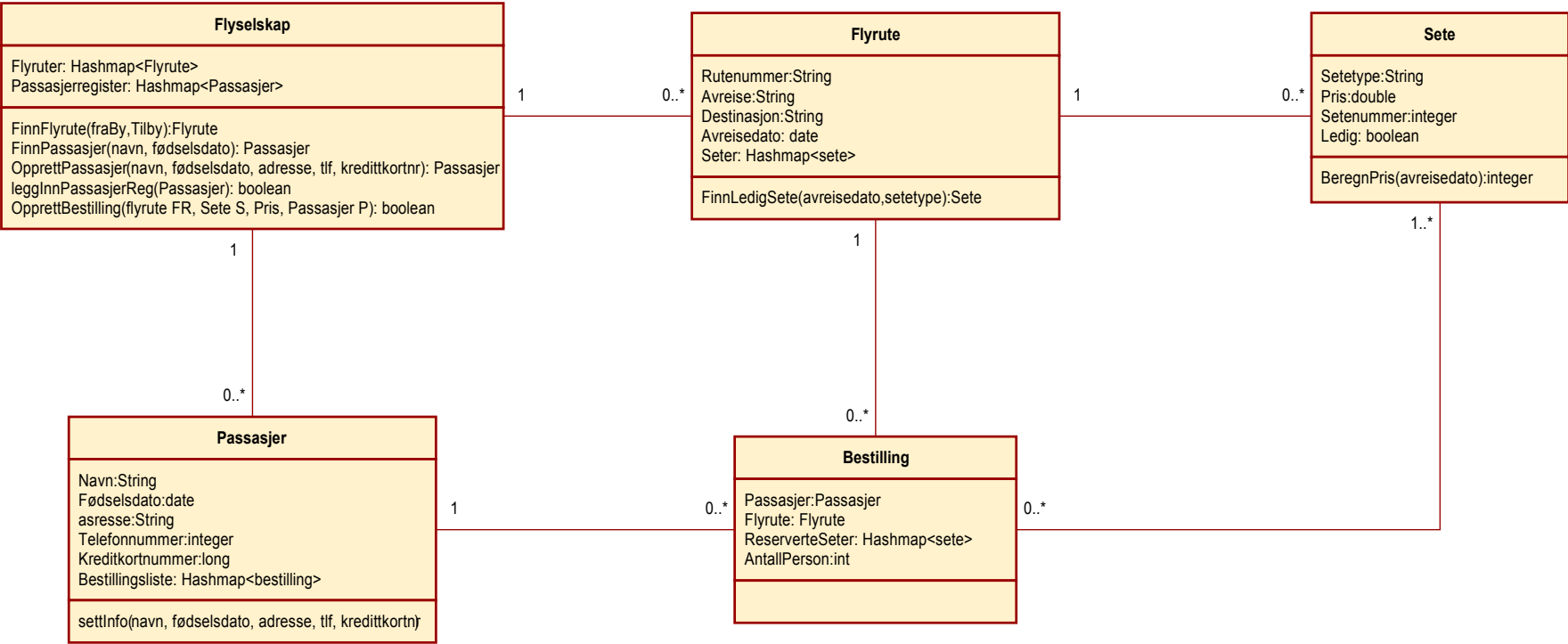
Sete har fått en attributt som heter Ledig, dette er en boolean, for å kunne se om setet er ledig.

Passasjer har fått en metode som heter settInfo(navn, fødselsdato, adresse, tlf, kredittkortnr) for å sette all informasjon etter at man har opprettet objektet.

Bruksmønster: «Bestill flyreise»

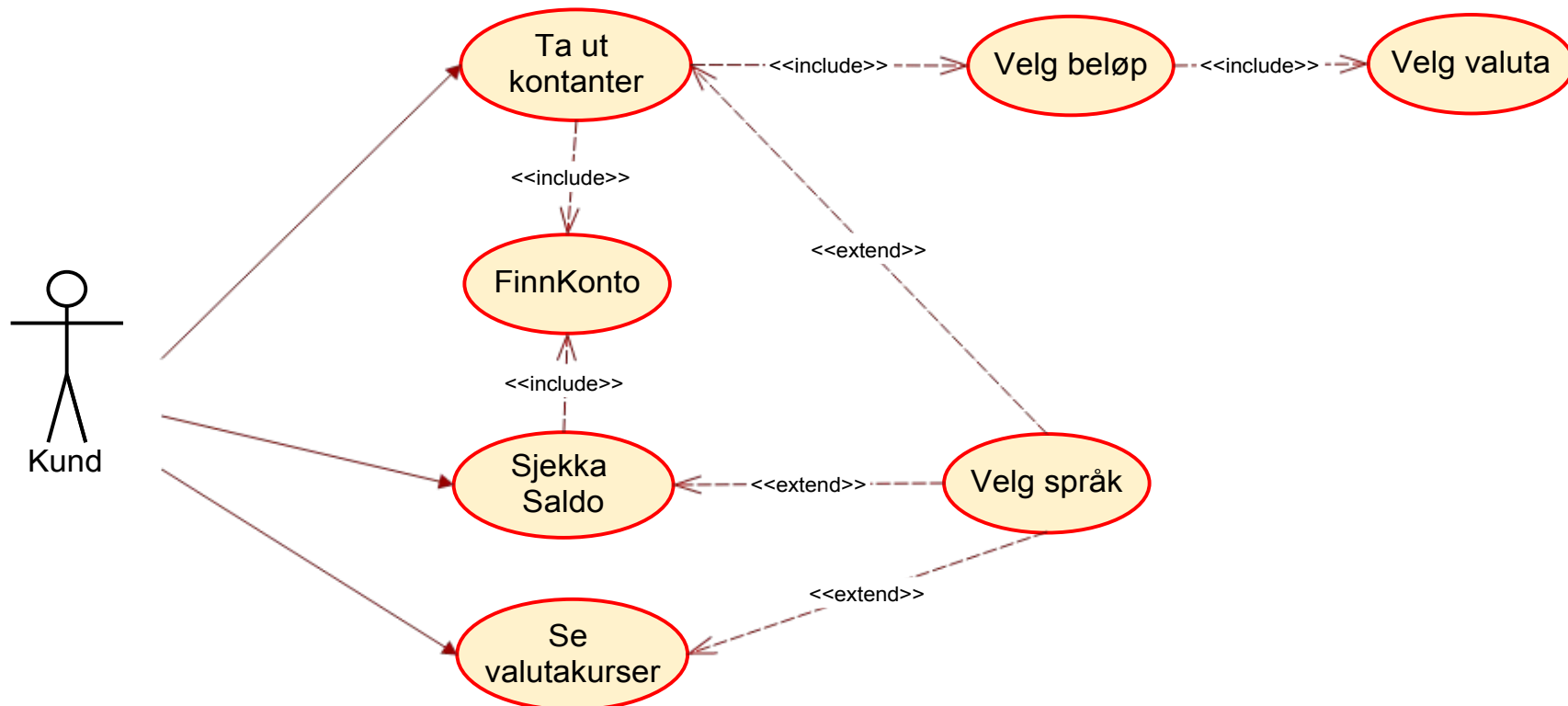


Oppgave 2



Oppgave 3

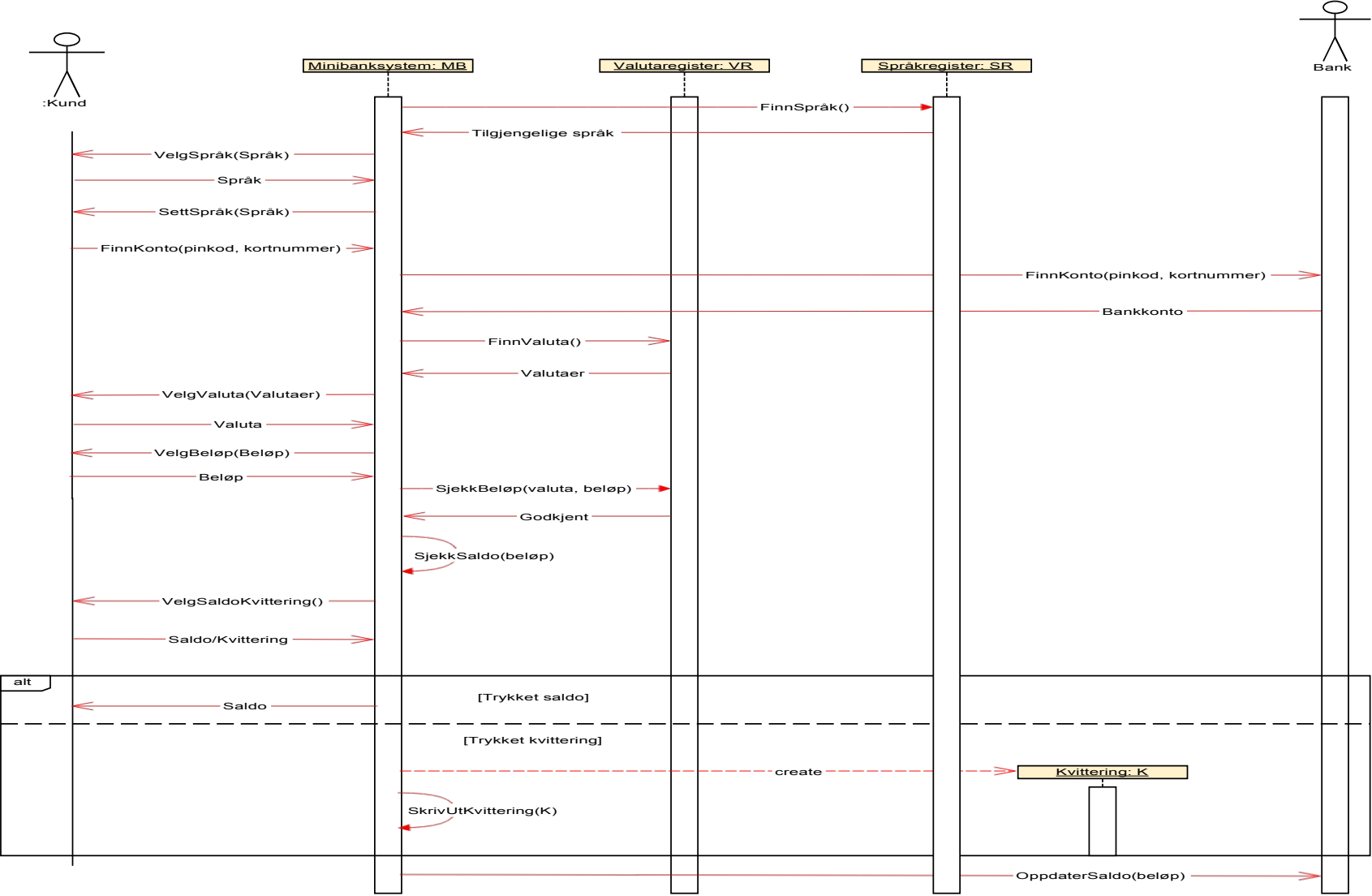
a) Vi tenker at for å sjekke saldo og for å ta ut kontanter må man finne en konto. Skal man ta ut kontanter må man velge beløp, men for å velge beløp må man først velge valuta. For å se valutakurser behøver man ikke å finne konto, dvs man behøver ikke sette inn bankkort. Ved alle tilfeller er det mulig å velge språk.



b)

Kunden symboliserer interfaceet. I sekvensdiagrammet har vi delt systemet inn i tre klasser: Minibanksystem som representerer kontrolleren i systemet (jamfør MVC). Valutaregister inneholder alle tilgjengelige valutaer, samt hvilke gyldige beløp for de ulike valutaene. Språkregister inneholder en liste over alle tilgjengelige språk. Bank er en sekundær aktør, et eksternt system som vårt system kommuniserer med. Vi har tatt med en alternative flyt hvor man enten har valgt saldo på skjerm eller skrive ut en kvittering. Ved sistnevnte alternativ blir det opprettet ett objekt av typen kvittering, som så blir skrevet ut.

Bruksmønster: «Ta ut kontanter»



Oppgave 4

a)

1. Det er kun de ansatte og VIP-passasjerer som skal kunne oppgradere setetype selv.
2. Det skal bare gå å oppgradere sete hvis det finnes ledige business seter.
3. Det skal kun være lov å sjekke inn når det er mindre enn 24 timer til avgang.
4. Systemet skal utvikles med JDK 8.
5. Systemet skal følge forskrift om universell utforming.

b)

1. Det er kun de ansatte og VIP-passasjerer som skal kunne oppgradere setetype selv.

- For å kunne oppgradere sete må man være innlogget. For å kunne logge inn må man ha en bruker. Med en ansatt-bruker kan man få tilgang og oppgradere en passasjer sitt sete. En VIP-passasjer har en egen bruker og kan kun oppgradere sitt eget sete.

2. Det skal bare gå å oppgradere sete hvis det finnes ledige business seter.

- Når man skal oppgradere et sete skal systemet kontrollere om det finnes ledige business seter. Om oppgraderingen er vellykket skal systemet redusere antall ledige business seter.

3. Det skal kun være lov å sjekke inn når det er mindre enn 24 timer til avgang.

- Når man forsøker å sjekke inn skal systemet sjekk hvor lang tid det er igjen til avgang. Hvis det er mer enn 24 timer igjen skal passasjeren få beskjed om dette og innsjekking skal ikke gjennomføres.

4. Systemet skal utvikles med JDK 8.

- Alle utviklere skal bruke java 8 sitt klassebibliotek. Alle maskiner som brukes må også ha java 8 installert.

5. Systemet skal følge forskrift om universell utforming.

- Utviklere må ta hensyn til forskrift om universell utforming av IKT-løsninger.

c)

PRODUKTKRAV

1. Det er kun de ansatte og VIP-passasjerer som skal kunne oppgradere setetype.
2. Det skal bare gå å oppgradere sete hvis det finnes ledige business seter.
3. Det skal kun være lov å sjekke inn når det er mindre enn 24 timer til avgang.

ORGANISATORISKE KRAV

4. Systemet skal utvikles med JDK 8.

EKSTERNE KRAV

5. Systemet skal følge forskrift om universell utforming.