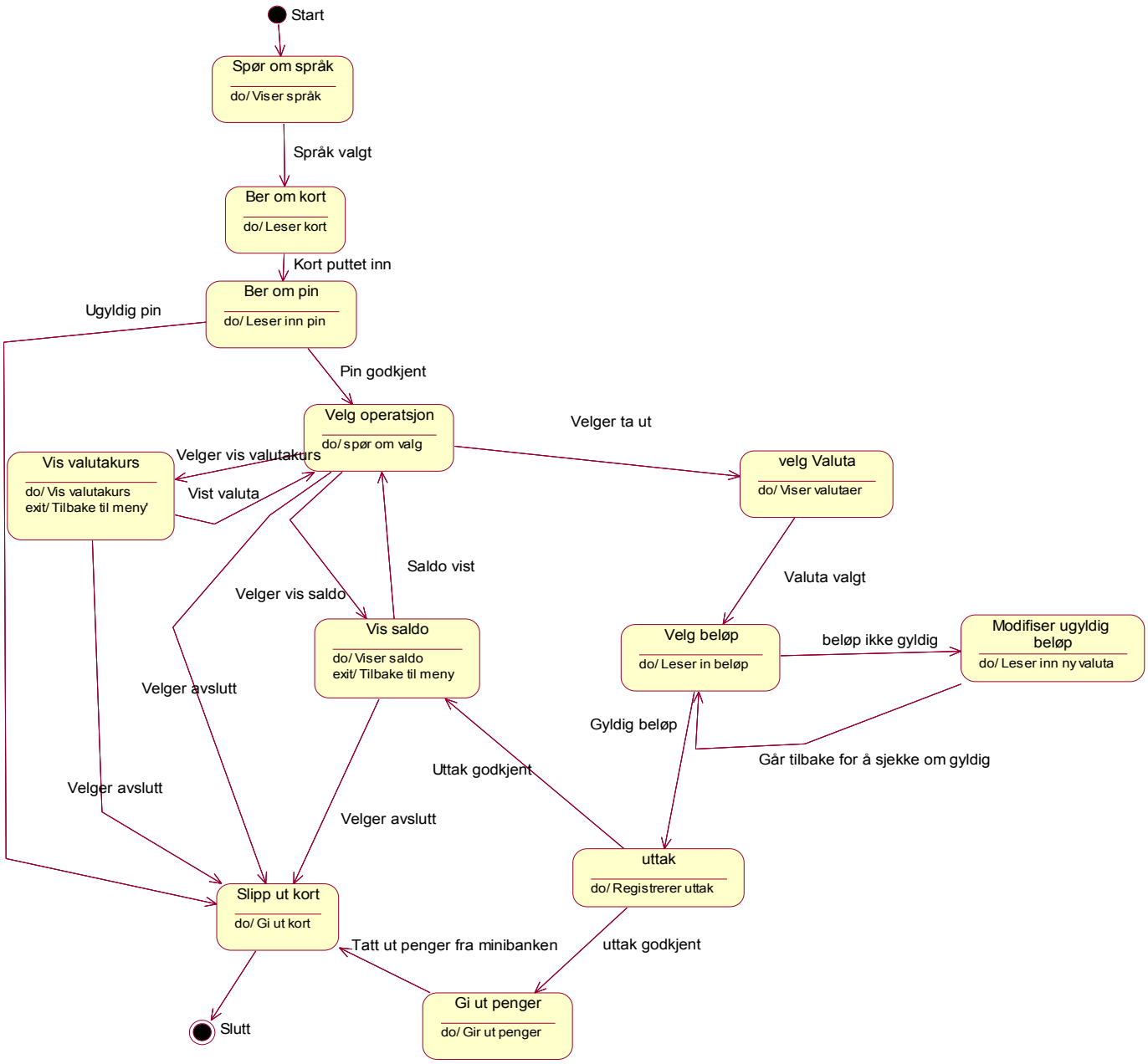
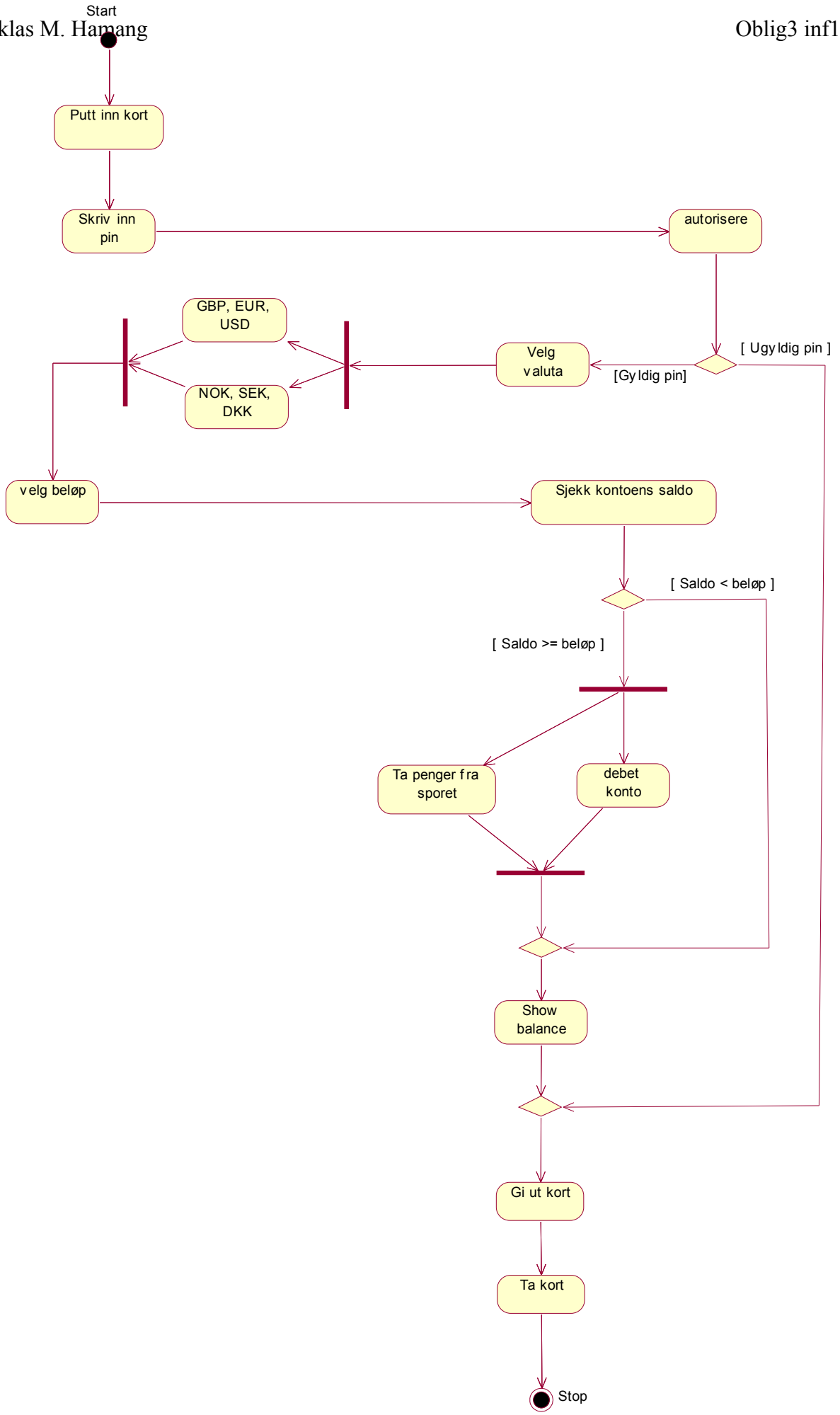
Oppgave 1a)



Oppgave 1b)


Start
Putt inn kort
Skriv inn pin
autorisere
GBP, EUR, USD
NOK, SEK, DKK
Velg valuta
[Gyldig pin]
[Ugyldig pin]
velg beløp
Sjekk kontoens saldo
[Saldo < beløp]
[Saldo >= beløp]
Ta penger fra sporet
debet konto
Show balance
Gi ut kort
Ta kort
Stop

Oppgave 1c)

karakteristisk for disse to diagrammene er:
– Aktivitets diagram kan representere hendelsesflyten i et use case grafisk og kan representere et sted mot programkode. Diagrammet beskriver tilstanden av aktiviteter ved å vise sekvensen av aktiviteter som er/blir utført.
– Tilstandsdiagram beskriver de mulige tilstandene til et objekt som hendelser inntreffer, det viser også overgangene mellom de tilstandene vi er interessert i.

Aktivitetsdiagaram brukes når man vil se sammenhengen mellom tingene og rekkefølgen tingene må gjøres. Den kan også for å designe applikasjoner med flere parallelle tråder eller komplekse algoritmer på et et lavere abstraksjonsnivå.

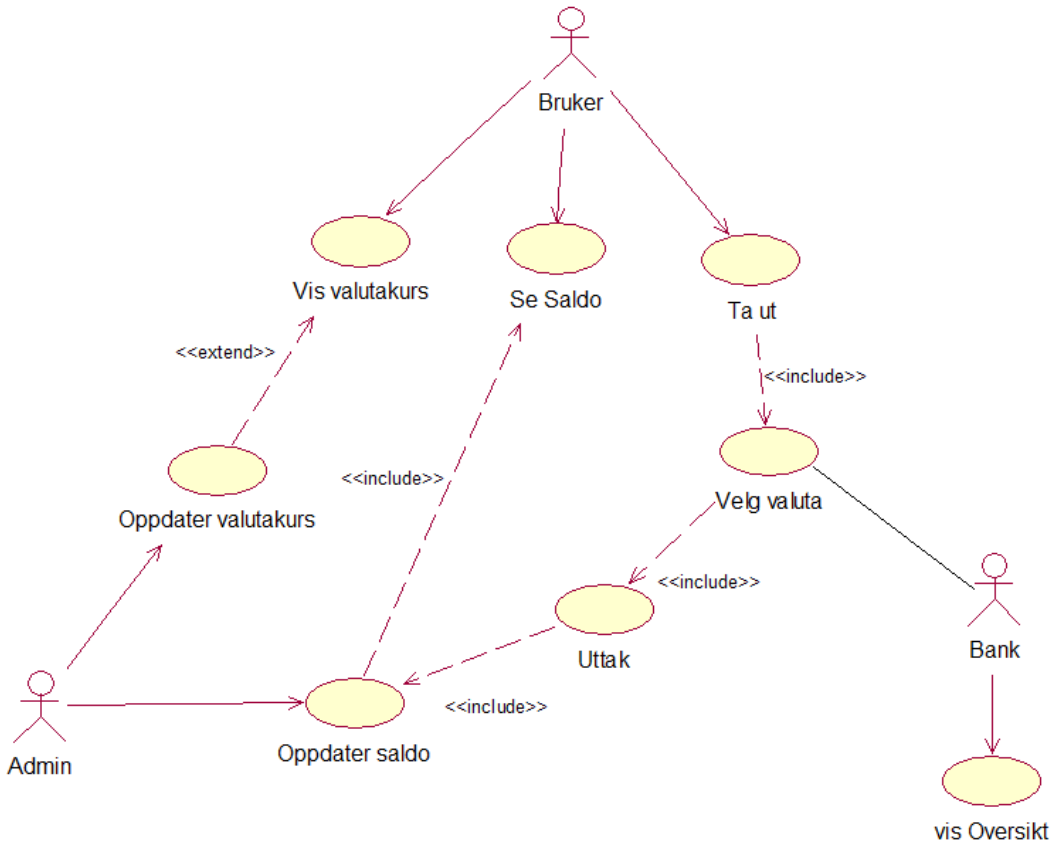
Tilstandsdiagram:
Dette brukes når vi ønsker å demonstrere virkemåten til et objekt gjennom mange bruksmåter av systemet. Den brukes også når man vil ha en samlet over avhangigheten mellom tilstandene og reaksjonene.

Oppgave 2a)
Funksjonelle krav:

1. Systemet må kunne vise oversikt over valutakursene.
2. Systemet må kunne vise saldoen til en konto.
3. Systemets admin må kunne oppdatere saldoer.
4. Systemet må kunne ta ut kontanter for den valgte valutaen.
5. Systemet må kunne ha en oversikt over alle konto/kreditt info som er registrert i banken.
6. Systemets admin må kunne oppdatere valutakursen.

Use case diagram på neste side.

Use case diagram:



2b)

Use case: Ta ut kontanter fra minibank.
Aktør: kunde
Pre betingelse: ingen
trigger: kunden setter inn kort i mini banken.

Hoved flyt:

1. Kunden putter kortet sitt(navn, fDato, saldo, kontonummer), velger språk og tast inn pinkode.
2. Systemet sjekker om de gitte opplysningen fra kundens kort med databasen til banken og den koden kunden skriver inn.
3. En tilbakemelding blir gitt til kunden når pinkoden er godkjent.
4. Systemet sjekker saldoen til kunden.
5. Systemet viser meny så kunden kan velge en av tre alternativer (1. valutakurs, 2. uttakk, 3. saldo).
6. Hvis valg 1: gir opp en liste over valutakursen som er lagt er i systemet.
7. Hvis valg 2: Systemet henter en liste over de tilgjnglige kursene som er på systemet.
8. Kunden blir opplyst om at uttak grensen til NOK, SEK, DKK er 10000NOK for alle disse valutaene og i disse beløpene 400, 800, 100,1200 og 1400. kunden kan også velge et eget beløp som er delelig med 200. hvis det gjelder GBP, USD eller EUR er beløpet det samme men i beløpene 40, 80, 100, 120 og 140 eller egenbestemt som er delelig på 20.
9. kunden velger en kurs og det beløpet som han/hun ønsker.
10. Systemet sjekker saldoen til kunden og beregner om kunden kan ta ut beløpet eller ikke.
11. Kunden får tilbakemelding om dette beløpet blir godkjent eller ikke.
12. Kunden får pengene og kvittering med før og etter saldo

Alternativ flyt 1, Stage 1: kunden putter inn kortet men det funker ikke
A1.1 kunden prøver et annet kort
A1.2 use case avsluttes.

Alternativ flyt 2 steg 3: pinkode ikke godkjent.
A2.1 prøv igjen
A2.2 use case avsluttes.

Alternativ flyt 3, steg 9: beløpet som kunden har valgt er for stort
A3.1 Systemet lar kunden prøve igjen. Use case fortsetter til steg 10.
A3.2 Use cases avslutter.

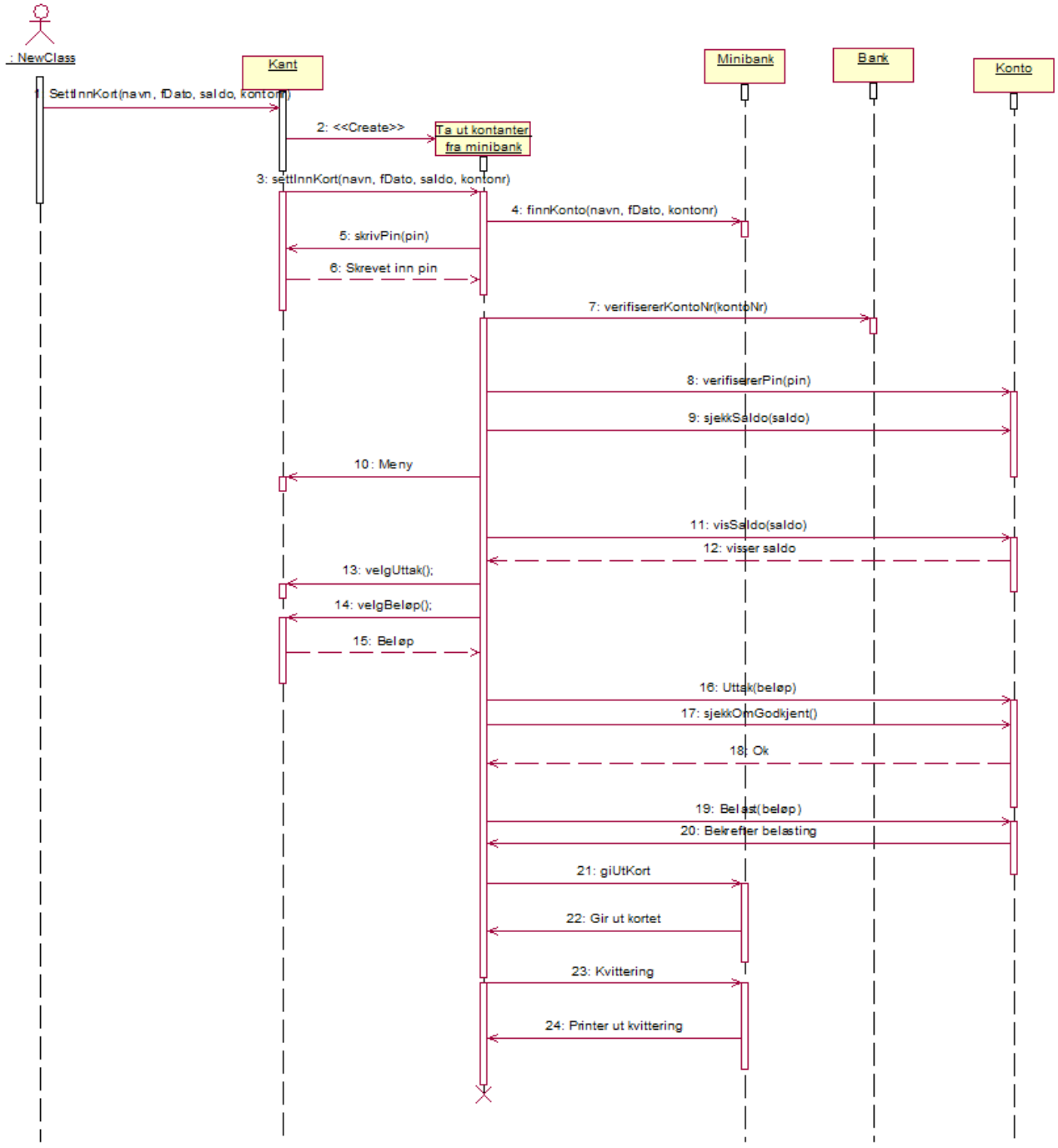
Alternativ flyt 4, steg 12: kunden får ikke ut pangene.
A4.1 ringer banken.

Post betingelse:

Bankens database er registrert i minibank-systemet at kunden (navn, fDato, saldo, kontonummer) har tatt ut penger fra kontoen sin. Kunden blir informert at uttaket var vellykket og blir gitt en kvittering.

Sekvensdiagram på neste side

Sekvensdiagram:



Oppgave 3.

1. Produktkrav: Innsjekking skal være på under 24 timer til avgang.
2. Produktkrav: Kommunikasjon og innloggingssystemet skal ha en sikkerhet mot eksterne eller interne innbrudd.
3. Produkt- eller organisasjons krav: Systemet skal ikke bruke alt for lang tid med innsjekking og billetstatus osv.
4. Produktkrav: systemer skal ha en opptid på 98%
5. organisasjons krav: Systemet skal ikke mangle nødvendige info.

Evaluering av disse kravene:

1. Systemet valideres ved testing på tid.
2. Systemet tester sikkerheten mot andre innbrudd.
3. Systemet valideres ved testing på tid.
4. systemet blir testet en vis periode for å sjekke om den er stabil og brukervennlig
5. Systemet valideres på testing for informasjonsbehandling slik at det ikke mangler noe

Oppgave 4 a)
Tre lags logisk arkitektur:
Presentasjon:
sender infoen fra mobilens nettleser til forretnings logikk.

Forretnings logikk:
Henter frem den dataen som skal vises i presentasjon.

Data:
Databasen finner frem dataen fra logikk laget, lagrer den og fører den tilbake til presentasjon.

Oppgave 4b)
Firelags fysisk arkitektur

Web leser:
mobilens nettleser sender dataen til en web server.

Web server:
samler opp data og sender den til applikasjons serveren, viser frem den dataen som er lagret i data server.

Applikasjonen:
klargjør og beregner dataen som trenges til videre-overføring. Oppdaterer dataen og sender den til web server.

Data server:
henter og lagrer dataen fra applikasjon, sender den tilbake til applikasjonen.

Oppgave 5)
Enhets testing: Enhets testing er for å teste de individuelle komponentene isolert fra hverandre. De delene som kan testes er både metode og objekt klassene. Vi kan teste metoder som «sjekk billett status», «sjekk passasjer info», «logg inn» osv. (sekundær operasjon) eksempelt kan vi teste ulike verdier til teste attributter og teste objekter i alle mulige stadier. I både normale operasjoner og abnormale operasjoner, som vil si at vi kan sette inn forskjellige input å se om systemet kræsjer eller ikke. Men hovedpunktet med enhets testing er som tidligere nevnt å teste individuelle deler av systemet isolert. Og for å teste forskjellige operasjoner med forskjellige input separat.

Integrasjons test: Når det kommer til integrasjons testing er det snakk om å teste om de forskjellige komponentene i systemet vårt virker ordentlig. Vi kan se for oss at systemet vært har tre forskjellige komponenter. Vi vil teste disse komponentene hver for se. Vi har NorFly som er en software komponent som inneholder objekter og metoder som har ansvar for informasjonen om deres passasjerer, ruter osv. Så er det salskapet NorStat hvor systemet står for ekstern informasjon som f.eks. Visum, valuta informasjon, vær- og vindforholdene. Den siste komponenten kommer fra I-Nor som er det grafiske berørings skjerm systemet til vårt system. Vi kan teste hver av de komponentene som har blitt nevnt for å se om de virker som det definerte grensesnittet og dette grensesnittet oppfører seg til samsvar for de spesifikasjonene til dette systemet. Derfor med å teste om de ulike komponentene integrere hovedsystemet vårt(Norfly billett terminal system).

System testing: System testing betyr at vi tester alle komponentene og integrer kontrollerer samspillet mellom dem. For å kunne utføre systemtesting, må vi teste helle systemet vårt som et hel. Vi må teste de ulike komponentene som er nevnt ovenfor for å se om alle jobber sammen relativt til funksjonalitets kravene våre.

Akseptanse test: akseptansetesting betyr at kunden tester systemet for å determinere om systemet er akseptabelt og kan intalleres. Men til spørsmålet om hvilken del av systemet som skal akseptansetestes? Må jeg igjen si hele systemet, det vil si summen av alle våre komponenter blir satt sammen for å bli kontrollert. Et forslag til NorFly kan være å sette sammen et 'lag' som ikke var en del av utviklingen, som kan teste systemet som en 'bruker'. Hele poenget med denne testen er for å se om systemet er klar for bruk eller ikke.

Oppgave 6)

	Beskrivelse	sannhetsverdier							
Betingelser	Terminalen skrur seg ikke på	Y	Y	Y	Y	N	N	N	N
	Terminalens varslingslampe lyser	Y	Y	N	N	Y	Y	N	N
	Innlogging ved hjelp av navn og fødselsnummer fungerer ikke	Y	N	Y	N	Y	N	Y	N
Beslutninger	Kontroller strømkabel	X	X	X	X				
	Kontroller nettkabelen	X	X			X	X	X	
	Undersøk om nødvendige programvare/firmwarre er installert	X	X	X	X	X			
	Kontroller sikkerheten(innlogging osv).	X		X		X		X	
	Undersøk om bankens nettserver kjører			X				X	