

Del 1

Oppgave 1

Explain, in your own words, the seven testing principles. Use examples to show their purpose. Do you have objections to any of the principles?

De 7 test prinsippene:

1. Testing viser om det er defekt i systemet. Testing kan vise at det finnes defekter, men den kan ikke bevise at det er ingen defekter. Testing reduserer sannsynligheten for at det finnes gjenværende defekter i softwaren. Selv om ingen defekter blir funnet, vil dette ikke nødvendigvis bety at det ikke lenger finnes defekter.
2. Fullstendig test er umulig. Det er ikke mulig å teste alt, vi ser på risikoer og prioriteringer for å sette fokus på testingen.
3. Tidlig testing: Vi bør starte med testaktivitetene tidlig for å finne defektene så tidlig som mulig i utviklingsprosessen, og det skal også fokuseres på godt definerte mål.
4. Klyngedannelse av defekt: Alle defektene ligger i en klynge. Hvis man finner én defekt, så vil det si at det er enklere å finne de andre defektene, fordi de mest sannsynlig ligger på samme sted. Et mindre antall moduler inneholder vanligvis mesteparten av defektene som blir funnet i testen før frigivelsen, eller er ansvarlig for de fleste defektene i drift.
5. "Pesticide-paradoks": Hvis de samme testene gjentas om og om igjen, så vil de etter hvert ikke finne nye feil. For å unngå en slik situasjon må testtilfellene gjennomgås regelmessig, og det må utarbeides nye tester for å teste forskjellige deler av en programvare.
6. Test er avhengig av konteksten: Testing utføres på ulike måter i ulike kontekster.
7. Feiltagelse vedrørende "fravær av feil": Det hjelper lite å rette på feil hvis systemet ikke imøtekommer brukernes behov og krav.

Oppgave 2

2.1 Use the V-model to explain how testing can be integrated in the life cycle of a software development project. What are the benefits of using the V-model with regards to the quality of the system, as well as the time and resources used in the development process?

V-modellen gir oss en oversikt over hvilken rekkefølge test oppgavene må gjøres, slik at man kan enkelt utnytte denne modellen for å unngå sløsing av tid og ressurs. Ved bruk av denne modellen vil testing bli prioritert helt i begynnelsen og dette vil hjelpe med å finne

defekter i tidlig stadie slik at dette ikke senere påvirker softwaren i en større grad. En av hovedfordelene av å ta i bruk V-modellen er at man får testet ganske tidlig og testingen blir gjennomført flere ganger i modellen slik at man så ofte som mulig leter etter defekter. Ettersom disse defektene blir tatt hånd om med en gang vil dette føre til at man sparer mye mer tid i motsetning til hvis man har testing senere i modellen og finner defektene senere enn man ville. Man trenger dermed ikke bruke like mye tid og ressurser på å ta hånd om defekter.

2.2 Is it possible to use the V-model in agile programming development projects? If so, explain how.

Det er mulig å ta i bruk V-modellen i agile programmerings utviklings prosjekter, men kompleksiteten av disse prosjektene fører til at det blir ugunstig å ta i bruk. Det er mer aktuelt å bruke V-modellen når man har et prosjekt hvor kravene er allerede forhåndsbestemt og godt definert og ikke dynamisk som i agile programmerings utviklings prosjekter.

2.3 Explain to what extent the V-model ensures the seven test principles from Exercise 1 are safeguarded.

V-modellen prioriterer testing i høy grad i modellen sin og oppfyller kravene til de syv test prinsippene ettersom man må teste flere ganger underveis, som betyr at selv om man ikke finner feil på første test, så vil man gjennomføre neste test og muligens finne feil der. Modellen setter høy fokus på testing og starter tidlig for å finne defekter og er mest gunstig ved prosjekter der kravene er godt definert. Det er også ulike tester som skal tas innen modellen dermed oppfyller den "pesticide-paradoks"-ets krav.

2.4 What is the difference between a test level and a test type? Explain the four different.

i) Test levels

Unit test: Her setter man fokus på spesifikke deler av softwaren for å bestemme om hver av delene fungerer. Hovedmålet her er å se om funksjonene fungerer slik som det har blitt definert i kravene.

Integration testing: Integrasjonstesting fokuserer på å teste enhetene i grupper, disse testene leter etter defekter mellom moduler eller funksjoner. Dette tester også hvor kompatibel enhetene er sammen.

System testing: I systemtesting så tester man softwaren som en helhet. Målet er å evaluere om systemet har oppnådd alle kravene og for å se om den oppfyller kvalitetsstandardene.

Acceptance testing: Tester om softwaren er klar for utgivelse. Det kan for eksempel være endringer underveis som kan bli misforstått og ikke stemmer akkurat overens med kravene brukeren har satt.

ii) Test types

Funksjonell testing: en type testing som bekrefter at hver enkelt funksjon av programvaren fungerer i samsvar med kravspesifikasjonen. Det er vanlig og skille mellom to typer teknikker altså black box og white box testing, men denne typen testing omfatter hovedsakelig black box testing. Dette innebærer at man ikke fokuserer på kildekoden til programvaren/produktet, men heller hvordan brukergrensesnittet vil fungere under en test.

Ikke-funksjonell testing: kvaliteten av komponentene eller systemet testes. Det er vanlig å referere ikke-funksjonell testing til aspekter av systemet eller komponenter som ikke er forbundet med en handlig bruker gjør eller en funksjon som for eksempel sikkerheten av systemet eller skalerbarheten. Eksempel på ikke-funksjonell kan være "Hvor mange kan logge på systemet på en gang?"

Automasjons testing: Det er en type test der testeren setter opp skripter og bruker det på andre software for å teste produktet. Denne prosessen involverer automasjon av en manuell prosess. Automasjons testing er brukt for å repetere kjøring av testing scenarioer som blir gjennomført manuelt, raskt og repeterende.

Oppgave 3

3.1 Explain, in your own words, the difference between static and dynamic testing techniques.

Statisk testing: Man vurderer koden uten å kjøre den. Det inneholder vurderinger og gir oversikt over hvordan de utføres. Hovedmålet med statisk testing er å forbedre kvaliteten på programvareprodukter.

Dynamisk testing: Testingen blir utført ved å kjøre testobjektet på en datamaskin. Eksempler på dynamisk testing er enhetstesting, systemtesting og integrasjonstesting.

3.2 Explain the different steps in a formal review.

Planlegging: den første fasen av formal review er planlegging. De følgende punktene under definerer planleggingsfasen av formal review:

1. Definere review kriteriene med forfatter tilstede
2. Velge de riktige ansatte/personene for prosjektet
3. Tildeler roller for personer i teamet
4. Definere entry/exit
5. Velge hvilke deler av dokumentet som skal vurderes
6. Sjekke entry-kriteriene

Kick-off: er en kort introduksjon til reviewet. Målet med kick-off møte er at alle på teamet er i samme bølgelengde som innebærer at alle på teamet forstår målet med denne reviewen.

Hovedmålet med kick-off møte er følgende:

1. Fordele dokumenter
2. Forklare mål, prosess og dokumenter for teamet

Forberedelse: denne fasen av formal review jobber deltakerne i teamet individuelt med dokumentet og skriver ned defekter, spørsmål og kommentarer til reviewet ut ifra deres forståelse av dokumentet. Hovedmålet for forberedelsesfasen er følgende:

1. Forberedelse for review-møte ved vurdere dokumentene
2. Notere de potensiell defekter, spørsmål og kommentarer

Review-møte: defekter, spørsmål og kommentarer diskuteres.

1. **Logging:** testleder fokuserer på å skrive så mange defekter som mulig og alle defekter skal kategoriseres i følgende kategorier
 - a. Critical
 - b. Major
 - c. Minor

2. Diskusjon

3. Avgjørelse

Re-work: Basert på defektene som har blitt funnet under forberedelsen og review møtet, så vil forfatteren forbedre dokumentet steg for steg.

1. I denne fase retter vi på defekter som har blitt funnet under de foregående fasene. Ikke alle defekter må fikses på, hvilke defekter som skal rettes på avgjøres av forfatteren.
2. Loggføre de defektene som har blitt fikset på

Follow-up: i denne fasen må moderatoren sørge for at jobben(det som utføres i foregående faser)har blitt gjort på en tilfredsstillende måte. Moderatoren har ikke ansvar for å sjekke at alle rettelsene er gjort i detalj.

1. Sjekke at defektene har blitt loggført(adressed)

2. Samle metrikker
3. Sjekke exit-kriterier

3.3 A walkthrough and an inspection are two different types of reviews. Explain the difference between them; including their purpose and the roles of the participants.

Walkthrough: Forfatter av dokumentet guider teamet gjennom dokumentet og deler hans/hennes tanker for å oppnå felles forståelse, samtidig samle teamets tilbakemeldinger.

Hensikt:

- Presentere dokumentet for stakeholdere både for de innenfor og utenfor software disiplin, for å kunne få mest mulig informasjon om temaet som dekkes i dokumentet.
- Forklare og evaluere dokumentets innhold.
- Å få en felles forståelse av dokumentet
- Undersøke og diskutere validiteten av en gitt løsning

Deltakernes roller:

- **Forfatter:** leder møtet
- **Scribe:** skriver referat fra møtet. (Forfatter kan også være en scribe)
- Testleder, stakeholdere, testere

Inspection: ledes av en moderator med erfaring/kunnskap. Dokumentet blir forberedt og sjekket grundig av testerne før møtet. Det trengs to personer med lik kunnskap til å undersøke et produkt. Defektene som er funnet blir dokumentert i en logging liste eller issue log.

Hensikt:

- Hjelper forfatteren til å oppnå god kvalitet av dokumentet
- Fjerner feil effektivt og så tidlig så mulig
- Forbereder produktets kvalitet
- Oppretter god forståelse ved å utveksle informasjon
- Lærer av defektene som er funnet og unngår at samme feil skal oppstå igjen.

Deltakernes roller:

- **Moderator:** leder møtet
- **Scribe/forfatter:** skriver referat
- **Peers:** undersøker produktet

Oppgave 4

4.1 Why are specification-based testing techniques also called Black-box techniques?

Det er fordi i spesifikasjon basert testings teknikker så ser de på software som en black-box med inputs og outputs.

Testere har ikke noe kunnskap av hvordan systemet eller komponentene er strukturert inne i boksen. I black-box testing er testeren konsentrert på hva softwaren gjør og ikke på hva den ikke gjør.

4.2 Why are structure-based testing techniques also called White-box techniques?

What is usually the purpose of these techniques?

Det er fordi i struktur-basert testings teknikker i likhet med white-box teknikker krever kunnskap av hvordan softwaren er implementert og hvordan den fungerer. I white-box testing vil testeren konsentrere seg på hvordan softwaren gjør det. For eksempel, en strukturell teknikk fokuserer på loops i softwaren.

Hensikten med struktur-baserte teknikker er test coverage measurement og strukturelle test case design.

4.3 When do we use experience-based techniques? What are the benefits and drawbacks related to these techniques?

I erfaringsbasert teknikk så er kunnskap, ferdigheter og bakgrunn som er viktig for test forhold og testtilfeller. Erfaringene av både tekniske og forretningsfolk er nødvendig, fordi det bringer ulike perspektiver for test analyse og design prosessen. På grunn av at man har tidligere erfaringer med lignende systemer, kan føre til at man får en idé på hva som kan ha gått galt og det er veldig nyttig for testing.

Dette kan være den eneste typen av teknikk som blir brukt for lav-risiko systemer, men denne tilnærmingen kan være særlig nyttig under ekstreme tidspress.

4.4 Would you say that some testing techniques are better than others? What is your opinion on the use of the different techniques?

Ja, hvis man tenker på hvilke del eller hva det er man tester. Slik som for å teste hvordan hele programmet fungerer funksjonelt vil man gjerne bruke systemtesting ovenfor unit testing ettersom unit testing bare fokuserer på deler av programmet og ikke på helheten. Dermed er det mer gunstig å bruke systemtesting ettersom man får det resultatet man søker etter. Det er derfor viktig å vite hvilke metoder det finnes og forskjellen slik at man vet hvilke som passer best til det man ønsker å se etter i programmet. Det er viktig å bruke forskjellige teknikker for å få ulike resultater og analysere ulike deler av programmet.

Del 2

Oppgave 5

5.1 Suppose a valid date is entered. Determine in this case the different

i) Equivalence partitions

Partitions	Zodiac Sign	Period
1	Aries	March 21 - April 19
2	Taurus	April 20 - May 20
3	Gemini	May 21 - June 20
4	Cancer	June 21 - July 22
5	Leo	July 23 - August 22
6	Virgo	August 23 - September 22
7	Libra	September 23 - October 22
8	Scorpio	October 23 - November 21
9	Sagittarius	November 22 - December 21
10	Capricorn	December 22 - January 19
11	Aquarius	January 20 - February 18
12	Pisces	February 19 - March 20

ii) the boundary values

Partitions	Zodiac Sign	Period	Valid Dates	Invalid Dates	Valid month	Invalid month
1	Aries	March 21 - April 19	21 - 19	Lower than 21 and higher than 19	3 and 4	Lower than 3 and higher than 4
2	Taurus	April 20 - May 20	20 - 20	Lower than 20 and higher than 20	4 and 5	Lower than 4 and higher than 5
3	Gemini	May 21 - June 20	21 - 20	Lower than 21 and higher than 20	5 and 6	Lower than 4 and higher than 6
4	Cancer	June 21 - July 22	21 - 22	Lower than 21 and higher than 22	6 and 7	Lower than 6 and higher than 7
5	Leo	July 23 - August 22	23 - 22	Lower than 23 and higher than 22	7 and 8	Lower than 7 and higher than 8
6	Virgo	August 23 - September 22	23 - 22	Lower than 23 and higher than 22	8 and 9	Lower than 8 and higher than 9
7	Libra	September 23 - October 22	23 - 22	Lower than 23 and higher than 22	9 and 10	Lower than 9 and higher than 10
8	Scorpio	October 23 - November 21	23 - 21	Lower than 23 and higher 21	10 and 11	Lower than 10 and higher than 11
9	Sagittarius	November 22 - December 21	22 - 21	Lower than 22 and higher than 21	11 and 12	Lower than 11 and higher than 12
10	Capricorn	December 22 - January 19	22 - 19	Lower than 22 and higher 19	12 and 1	Lower than 12 and higher than 1
11	Aquarius	January 20 - February 18	20 - 18	Lower than 20 and higher 18	1 and 2	Lower than 1 and higher than 2

12	Pisces	February 19 - March 20	19 - 20	Lower than 19 and higher than 20	2 and 3	Lower than 2 and higher than 3
-----------	---------------	------------------------	---------	----------------------------------	---------	--------------------------------

5.3 Which equivalence partitions and boundary values should you add if you don't know whether or not the user has entered a valid date?

i) Equivalence partitions

Date	Month
1 - 30	April, June, September, November
1 - 31	January, March, May, July, August, October, December
1 - 29	February

Partitions	Zodiac Sign	Period
1	Aries	March 21 - April 19
2	Taurus	April 20 - May 20
3	Gemini	May 21 - June 20
4	Cancer	June 21 - July 22
5	Leo	July 23 - August 22
6	Virgo	August 23 - September 22
7	Libra	September 23 - October 22
8	Scorpio	October 23 - November 21
9	Sagittarius	November 22 - December 21
10	Capricorn	December 22 - January 19
11	Aquarius	January 20 - February 18
12	Pisces	February 19 - March 20

ii) the boundary values

Date	Month	Boundary values
1 - 30	April, June, September, November	0 og 31
1 - 31	January, March, May, July, August, October, December	0 og 32
1 - 29	February	0 og 30

5.4 There are at least three defects/incidents in this program (there may be more). Can you find them? Consider the degree of severity of the failures. In which order will you prioritize correcting the defects?

Defect 1: Når man taster inn alt annet enn tall så burde det ha kommet opp en informativ melding om hva som var feil med inputen, men det gjorde det ikke, og derfor er dette en defect.

Defect 2: 21.12 skal i utgangspunktet gi en valid dato, men den er satt opp til å gi invalid dato.

Defect 3: Hvis man skriver måneden eller dagen du skal ha med et vilkårlig antall nuller vil det ikke påvirke resultatet. For eksempel kan man skrive "0000000001" og det vil fortsatt tilsvare januar.

Defekt 2 er det største defektet her ettersom folk med den bursdagsdatoen vil stadig få feil når vi forsøker å taste inn informasjonen. Dette burde være det første man skal fikse blant defektene ettersom dette er noe som vil skje flest ganger. Deretter er det defekt 1 som kommer etter og burde fikses der man får tilbakemelding på hva som gikk galt for det kan hende at folk taster inn feil ved et uhell og når man ikke får tilbakemelding kan det hende de ikke skjønner hva som gikk galt. Defekt 3 er mindre alvorlig og kan tas til sist ettersom det ikke skader noe ved programmet.

5.5 Use the template below to write an incident report about the defects/incidents you have discovered.

Test incident report identifier

ID: 01

Summary

The program tested for defects were the HoroscopeX.jar. The test gave three defects.

Incident description

Incident	Inputs	Expected Results	Anomalous Results
1	Symbols Letters No inputs	Warning or error about what the problem was with the input	No warnings or error
2	Date: 21.12	Correct horoscope of the date	Warning of incorrect date
3	Arbitrary number of zeros before the correct date	Warning of too many numbers or too many zeros	Correct horoscope despite the large number. The zeros before the number is ignored.

Impact

The potential impact to the system from the different incidents:

Incident 1 - Major, severe problem but possible to work around

Incident 2 - Mission critical, the application will not function

Incident 3 - Minor, does not impact the functionality or usability of the process

The order in which the incidents are to be addressed

Incident 1 - Immediate, must be fixed as soon as possible

Incident 2 - Immediate, must be fixed as soon as possible

Incident 3 - Deferred, the defect can be left in if necessary due to time or costs

Oppgave 6

6.1 Set up a decision table with the different business rules, including the conditions and related price. There should be one rule for each combination of conditions.

Business rules:

1. The “rush hour” is defined as the period from 07:30 until 09:30, and later from 15:30 until 17:30.
2. Electric vehicles are not charged outside the “rush hour” period, but are charged a fee of 30 NOK during peak times.
3. Petrol- and diesel vehicles are required to pay 50 NOK outside the “rush hour” period, and get an additional 100 percent increase in the price during peak times.

Conditions	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
C1: Rush hour?	Y	Y	Y	Y	N	N	N	N	N	Y	N	Y	N	Y	Y	N
C2: Outside rush hour?	Y	Y	Y	N	N	N	N	Y	Y	N	Y	N	Y	N	Y	N
C3: Electric car?	Y	Y	N	N	N	N	Y	Y	Y	N	N	Y	N	Y	N	Y
C4: Diesel or petrol?	Y	N	N	N	N	Y	Y	Y	N	Y	Y	N	N	Y	Y	N
Action																
Pay 30 NOK	X	X	X			X	X	X				V		X	X	X
Pay 50 NOK	X	X	X			X	X	X			V			X	X	X
Pay 100 NOK	X	X	x			X	X	X		V				X	X	x

6.2 Reduce the number of rules by simplifying (rationalizing) the decision table, without losing any of the system functionality.

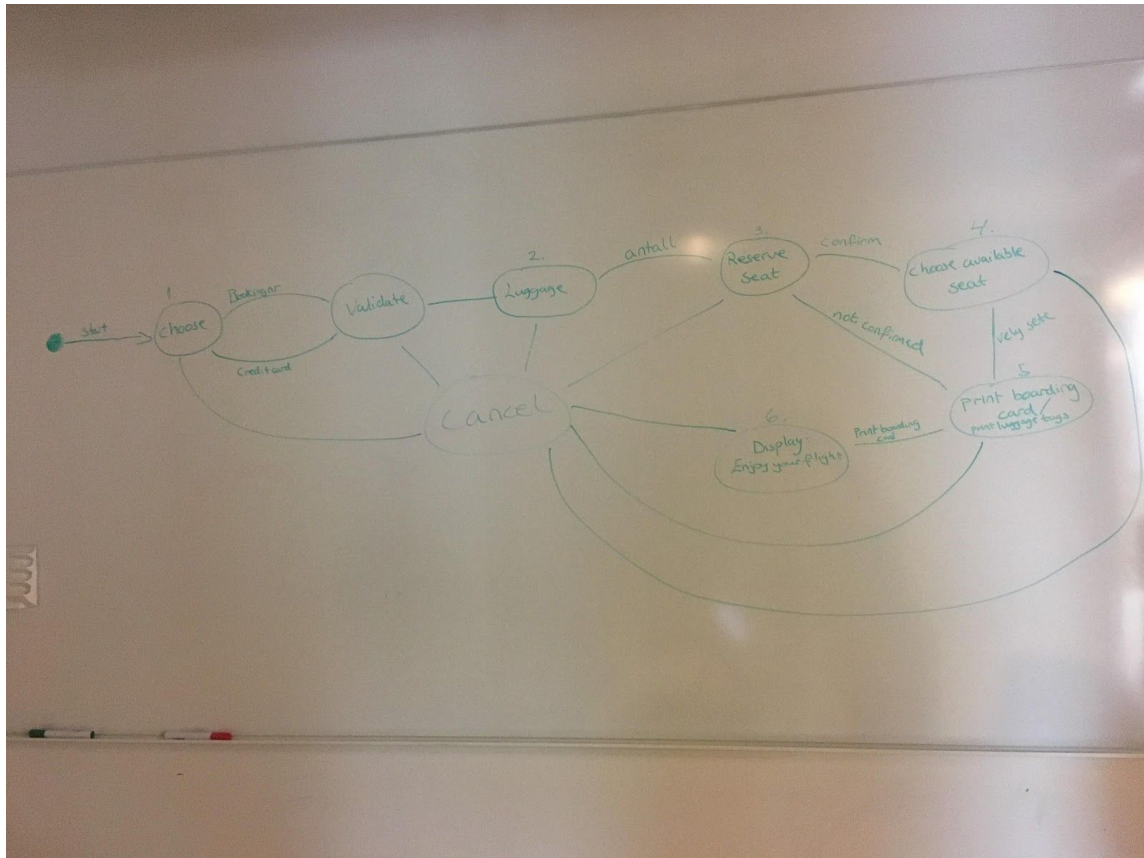
Conditions	R1	R2	R3	R4
C1: Rush hour?	Y	Y	N	N
C2: Electric car?	Y	N	Y	N
Action				
Pay 30 NOK	V			
Pay 50 NOK				V
Pay 100 NOK		V		

Vi antar at hvis det ikke er rush hour, så er det vanlig tid og tar derfor ikke med dette i decision table. Hvis det ikke er elektrisk bil, så antar vi at det er diesel og petrol.

Del 3

Oppgave 7

7.1 Draw a state transition diagram that shows how the machine works.



7.2 Set up a state transition table showing every legal and illegal (impossible) state.

Current state	Input	Action	Next state	Incorrect
S1	Valid: Booking/credit card	-	S2	S1
S2	Nr.: Luggage	Continue/cancel	S3, A1	-
S3	Reserve/change seat	Confirm/cancel	S4, S5, A1	-
S3		Continue/cancel	S5	-
S4	Choose available seat	Continue/cancel	S5, A1	-
S5	-	-	S6, A1	-
S6	-	-	S1	-

7.3 How do you measure

i) State coverage?

- State coverage kan måles ved å teste hver og en tilstand, dette viser oppførselen til hver og en tilstanden. Det kan også være enklere å finne defekter hvis man tester tilstandene.

ii) Transition coverage?

- Transition coverage kan måles ved å teste hvordan overgangen mellom de ulike tilstandene er og hvordan de oppdateres fra den ene tilstanden til den andre. Må dekke alle

7.4 Write down a sequence of states to achieve 100 percent transition coverage. What is the state coverage of the same sequence? Is there any relationship between transition coverage and state coverage?

For å kunne få en 100% transition coverage så må alle overgangene mellom tilstandene i testingen av system være dekket, derfor bør man ha sekvenser som går gjennom overgangene. Men for å få 100% state coverage så vil man dermed fokusere på å dekke alle de ulike tilstandene. De er ulike ved at i en state coverage så kan man gå gjennom alle tilstandene uten å gå gjennom alle overgangene.

100% transition coverage er alltid 100% state coverage, men ikke andre veien.

Oppgave 8

In this exercise, you shall use case testing to test the check-in machine.

8.1 What kind of scenarios will be the main focus in your use cases?

1. Choose check-in method
 - 1.1 Check-in with booking nr.
 - 1.1.1 Enter booking nr.
 - 1.1.2 Write nr. and click continue
 - 1.2 Check-in with credit card
 - 1.2.1 Enter credit card nr.
 - 1.2.2 Write nr. and click continue
2. Put in the amount of luggage items you want to check in.
 - 2.1 Click "Continue"
3. Seat arrangements
 - 3.1 Reserve a seat
 - 3.1.1 Choose an available seat

3.1.2 Click "Continue"

3.2 Change an existing seat reservation

4. The boarding pass will be automatically printed out along with the luggage tags.
5. The machine wishes you an enjoyable flight

5.1 Return to 1

Alternative scenario:

On every scenario step, there's an alternative scenario where you can cancel:

? Cancel and return to 1

1.1.3 The booking number is incorrect, return to 1

1.2.3 The card is rejected, return to 1

8.2 Write a testing procedure to test the check-in machine.

Test-1-Successful-Check-In

1. Choose to check in with booking number, type in number and continue.
2. Put in an amount of luggage and continue.
3. Change an existing seat reservation and continue.
4. (Automatic: print out of boarding pass and luggage tags, and receive message)
5. End.

Test-2-Cancel-During-Check-In

1. Choose to check in with credit card and type in number and continue.
2. Put in an amount of luggage and continue.
3. Reserve a seat, but cancel the check in.
4. Return to start.
5. End.

Test-3-Alternative-Check-In

1. Choose to check in with booking number.
2. Error, incorrect booking number, return to start.
3. End.