

Da er de1. Identifikasjon

Laget når: 29.02.2024

Av forfattere: s374951, s364571, s374922, s374907

Sist oppdatert når: 15.03.2024

Av forfatter: s364571

Hva testes her: Enhetstesting, Integrasjonstesting, Systemtesting

Versjonsnummer av denne planen: 2.0

2. Innledning

Testen skal verifisere at eksisterende funksjonalitet i applikasjonen fungerer. Gi et overblikk over systemet som skal testes.

Systemet som skal testet er en nettbank applikasjon som ble nedlastet i starten av prosjektet. Alle testene laget av gruppen skal testes opp mot nettbank-applikasjonen og dens funksjonaliteter. Når nettbank kjøres gir den oss først en innloggingsportal hvor man har valget å logge inn som bruker eller admin. Innloggingsinformasjonen for begge disse brukerne ligger i nettbank-koden. Når man er logget inn som bruker gir nettbank en oversikt over saldo i første fane. Videre har brukeren tilgang til en rekke funksjonaliteter som foreksempel: registrere batalinger, utføre betalinger, finne transaksjoner, og kunne endre kunde informasjonen. Videre hadde admin brukeren en rekke funksjonaliteter også. Admin brukeren kunne endre og slette brukere, kontoer samt registrere nye. Alt dette skulle testes i de tre ulike test-typene innad i gruppen

3. Testobjekter

Testobjekt i denne testplanen defineres som de programmer som skal gå i drift. Dvs. alle filer, API'er etc.

Beskriv de funksjoner og egenskaper som skal testes og evt. som ikke skal testes. Er det noen prioritering av testene?

- Enhetstesting

Testobjekter

IntelliJ.

I enhetstestene skal vi lage kode som skal inject-e kode inn i en "kopiert" form av repository. Denne skal rettes mot alle sidene hvor originalkoden tar imot en input, som er vår inject, og forventer et svar av originalkoden. Om testen får forventa svar fra originalkoden så har inject-en gått igjennom og testen er godkjent.

Funksjoner og egenspaker som skal og ikke skal testes

Det som skal testes:

- Kjernefunksjonalitet: Hver enhet ble utført riktig.
- Feilhåndtering: håndtere feilsituasjoner som f.eks. feil input

Det som ikke skal testes:

- Interaksjon med andre enheter.

Testprioriteringer

Høy prioritering:

Kjernefunksjonalitet

Middels prioritering:

Feilhåndtering

- Integrasjonstesting

Testobjekter

SoapUi.

Vi skal bruke SoapUi til å opprette og kjøre automatiserte tester for å sikre oss at webtjenestene for nettbanken fungerer som forventet. Våre automatiserte tester innebærer blant annet: å sende forespørsler og validere svar fra nettbanken. Verktøyet SoapUi gir oss muligheten til å sjekke autentisering, håndtere eventuelle feil som kan oppstå og utføre tester fra forskjellige brukere. Samlet sett, sikrer vi påliteligheten til nettbanken med integrasjonstesting.

Funksjoner og egenspaker som skal og ikke skal testes

Det som skal testes:

- Sørge for at data flyter korrekt mellom ulike komponenter og systemer.
 - Sjekke om kommunikasjon mellom systemene fungerer som forventet
- Dataoverføring og feilhåndtering
 - Sjekke hvordan systemet håndterer uventede feilsituasjoner

Det som ikke skal testes:

- Enhetstesting skal ikke utføres igjen
- Systemtesting. Det er ikke meningen å sjekke systemfunksjonalitet
- Brukergrensesnitt

Testprioriteringer

Høy prioritering:

Sørge for at data flyter korrekt mellom ulike komponenter og systemer.

Middels prioritering:

Dataoverføring og feilhåndtering

- Systemtesting

testobjekter

SeleniumIDE.

Vi skal bruke SelendiumIDE til å kjøre igjennom alle funksjonalitetene i systemet. SeleniumIDE tar også hele testen opp på video. Vi skal fysisk gå igjennom alle sidene og ”kjøre” enhver funksjon i systemet som en bruker i form av både kunde og Admin av systemet.

Funksjoner og egenspaker som skal og ikke skal testes

Det som skal testes:

- Logg inn som admin
- Som admin:
 - Endring og sletting av konto
 - Endring og sletting av kunde
 - registrering av kunde og konto
- Logg inn som kunde
- Som kunde:
 - Registrering av betaling
 - Utfør betaling
 - Se på transaksjoner
 - Og all kunde info og endring av den infoen

De vi ikke skal teste var:

- Saldo siden som kunde
- Logg ut som admin eller kunde

Testprioriteringer

Høy prioritering:

Registrer betaling og utfør betaling for kunde, viktig siden kunden skal kunne ha full tilgang til all innafor deres "krav"

Endre/registrer kunde og endre/registrere konto, Admin skal ha tilgang og mulighet til å kunne endre all infoen og eller registrere ny info om enhver kunde og konto

Middels prioritering:

Innlogging av både kunde og Admin, det er ganske viktig sikkerhets messi at dette funker, men det var ikke hovedfokuset i denne oppgaven, det var mer de andre funksjonalitetene i systemet.

4. Fremgangsmåte

Beskriv hvordan testingen skal foregå. Hvilke data skal benyttes. Hvordan automatiseres testene?

- **Enhetstesting**

For testing av enhetstesting i nettbank var målet å teste at individuelle enheter fungerer som forventet. Derfor tenkte gruppen at det første vi måtte gjøre var å teste funksjonaliteter, klasser eller metoder i koden. Derfor startet vi med å analysere koden og tolke koden i programmet IntelliJ. Vi startet med å lage en ny fil under mappen "test" hvor vi endre opp med å kjøre testene opp mot koden. Dataen vi trengte for utføringen av testen finner vi i koden og brukte den for å lage testene. For automatisering av testene tenkte gruppen å testene for hver enhet separat noe som gjorde at vi kunne kjøre flere tester etter hverandre i "test"-filen.

- **Integrasjonstesting**

For Integrasjonstesting i nettbank var målet å teste at enheter samarbeider riktig når de er satt sammen. For disse testene brukte vi programmet SoapUI. Gruppen startet med å lage et nytt prosjekt. Deretter lagde vi en TestSuite og til slutt lagde vi ulike TestCases for de ulike testene som skulle kjøres. Totalt tenkte gruppen at tre ulike TestCases for AdminKontoTest, AdminKundeTest, BankTest var nokk. Gruppen brukte videoer som var lagt ut som hjelp samtidig som vi arbeidet med å lage nye tester. Koden kjørte via IntelliJ mens integrasjonstestene ble laget og kjørt. Ettersom at hele TestCasen kunne kjøres var det en fin måte for å sjekke om testene samhandles samtidig som det automatisk kjørte en rekke tester. Når alle testene innenfor TestCasen ble ferdig skrevet, tenkte vi å kjøre hele TestCasen og deretter starte på neste TestCase.

- **Systemtesting**

For systemtest i nettbank var målet å teste at hele systemet fungerer som forventet. I denne delen var fokus mer på sluttproduktet eksempelvis at endringer blir lagret og at programmet fungerer. Google chrome utvidelsen Selenium IDE brukes for systemtestene. Gruppen lagde en ny Selenium fil og lagde filer og filnavn til de ulike testene vi tenkte å lage. Deretter analyserte vi hva som skulle testes og hvordan vi skulle utføre testene. Etter en test blir laget kjørte vi den først alene og så

sammen med andre for å sjekke om resultatet endres. Til slutt ble testene automatisert av Selenium for nettbankapplikasjonen.

4.1 Overordnet teststrategi for

- Enhetstest

Den overordnede strategien for enhetstesting er å fokusere på å validere at de individuelle komponent av programvaren opererer som forventet. Dette innebærer å skrive tester i programvaren som isolerer funksjon, metode, eller klasse, og sikre at de returnerer de forventede resultatene. Returnerer ikke de forventede resultatene kan det være feil i koden.

- Integrasjonstest

Den overordnede strategien for integrasjonstest er å bekrefte at flere individuelle komponenter kan jobbe sammen for å oppfylle det ønskede resultat. Denne typen test tar fatt flere aspekter enn enhetstesting samt trengs det egen programvare for denne testtypen. Det overordnede strategien er å bruke programmet SoapUI til utføringen av testene mens gruppen sammen går gjennom og finner hva som må prioritieres i testingen.

- Systemtest

Den overordnede strategien for systemtest er å validere at hele systemet fungerer som forventet. Denne testingen kontrollerer at alle komponenter og moduler av programvaren fungerer sammen i sin helhet. Den overordnede strategien var å gå gjennom hele systemet og teste funksjonalitetene. Etter dette starter vi på dokumentasjonen av hvilke tester som fungerer og hvilke som ikke fungerer.

4.2 Prosedyre for planlegging og gjennomføring av test

Hvordan skal testene spesifiseres og hvordan skal de kjøres?

Hva er kriteriene for at testene kan kjøres (oppsett av miljø, database etc)?

Hva er kriteriene for at testene er kjørt?

Spesifisering og kjøring av tester

Testene skal spesifiseres ved å definere klare og konkrete testscenarier med forventede resultater. Disse scenariene skal være basert på systemkravene og bruksscenariene for systemet.

Testene skal kjøres ved å gjennomføre de definerte testscenariene mot systemet. Dette kan gjøres manuelt, eller ved hjelp av automatiserte testverktøy, avhengig av naturen til systemet og testene.

Kriterier for kjøring av tester

- Det nødvendige testmiljøet er satt opp, inkludert nødvendig maskinvare, programvare og nettverksinfrastruktur.
- Nødvendige testdata er tilgjengelige i databasen, og databasen er korrekt konfigurert.

Kriterier for fullførte tester

- Alle definerte testscenarier er gjennomført.
- Eventuelle avvik fra de forventede resultatene er dokumentert og rapportert.

4.3 Kriterier for grundighet i testen

På enhetstest-nivå bør det dokumenteres graden av dekning av kodelinjer og betingelser ("code coverage").

For enhetstesting sjekker vi om individuelle enheter fungerer som forventet.

Ved integrasjonstest bør alle interne grensesnitt være utført (API'er).

For integrasjonstesting sjekker vi om enheter som er satt sammen fungerer på riktig måte

For systemtest bør det dokumenteres hvorvidt krav i spesifikasjonen / brukerhistorier er testet.

Videre tester vi helheten av systemet og hvorvidt den oppfyller ønskede forventninger

5. Godkjenningskriterier

Et eksempel på et godkjenningskriteriene kan være:

Testen ansees godkjent når 95% av de planlagte tester er utført og det ikke finnes noen åpne A og B feil.

Ved rapportering av feil, skal feil og avvik kategoriseres. Dette vil bli brukt som underlag for godkjennings- og underkjenningskriterier av testene. Følgende kategorier kan f.eks benyttes for feil og avvik:

Kategori A: Feil i en eller flere funksjoner som forårsaker stopp. For eksempel maskin/system/funksjon må startes på nytt.

Kategori B: Feil har alvorlige konsekvenser i systemet eller tilstøtende systemer, for eksempel feil i databaser, alvorlige feil i rapporter slik at disse tolkes feil etc.

Kategori C: Feil med mindre alvorlige konsekvenser, for eksempel mindre grad av ustabilitet, mindre funksjons-/egenskapsmangler etc.

Kategori D: Feil uten alvorlige konsekvenser, for eksempel layout-/trykkfeil i skjermbilder, rapporter eller dokumentasjon som ikke har konsekvenser for forståelsen av disse.

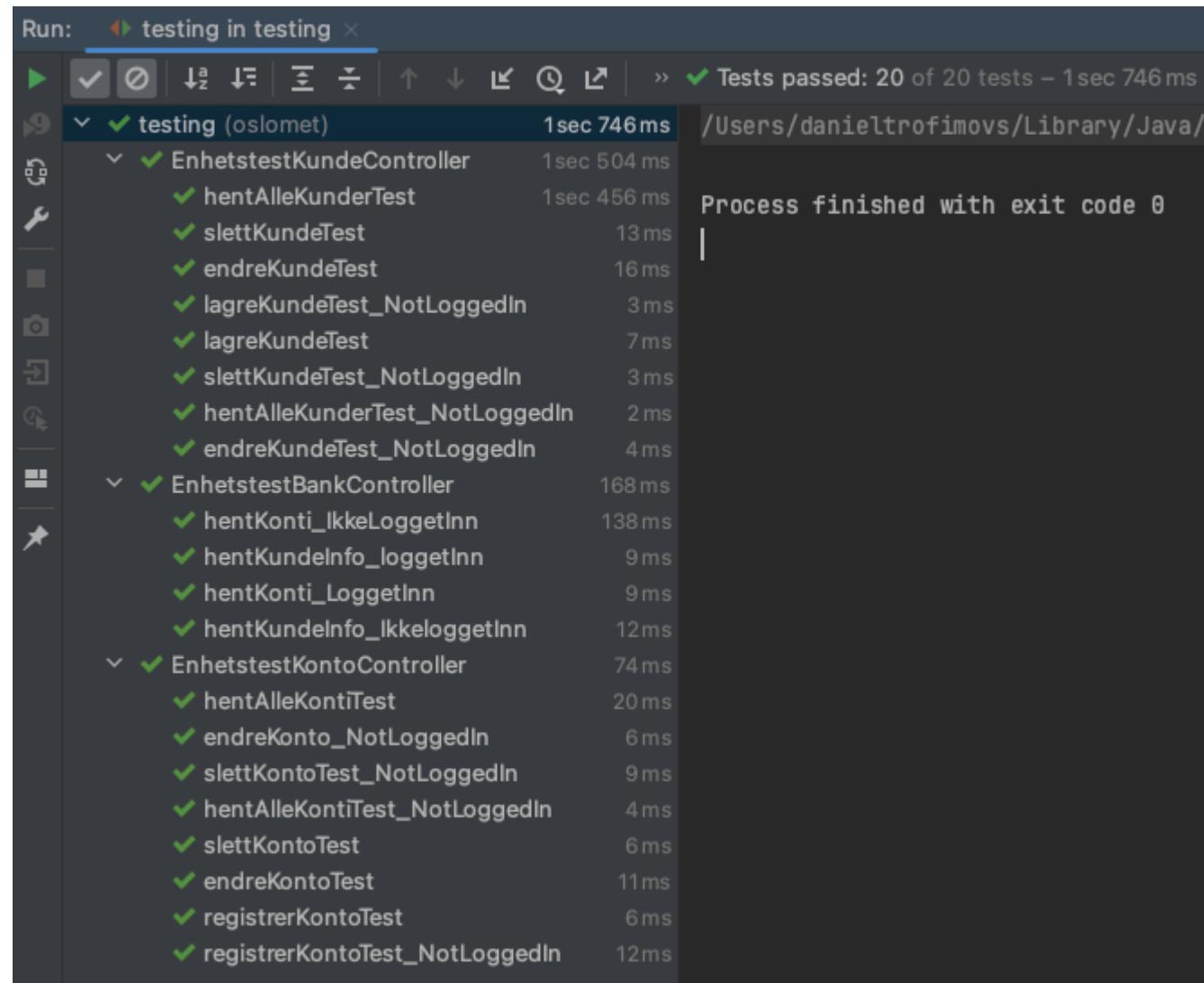
6. Testmiljø

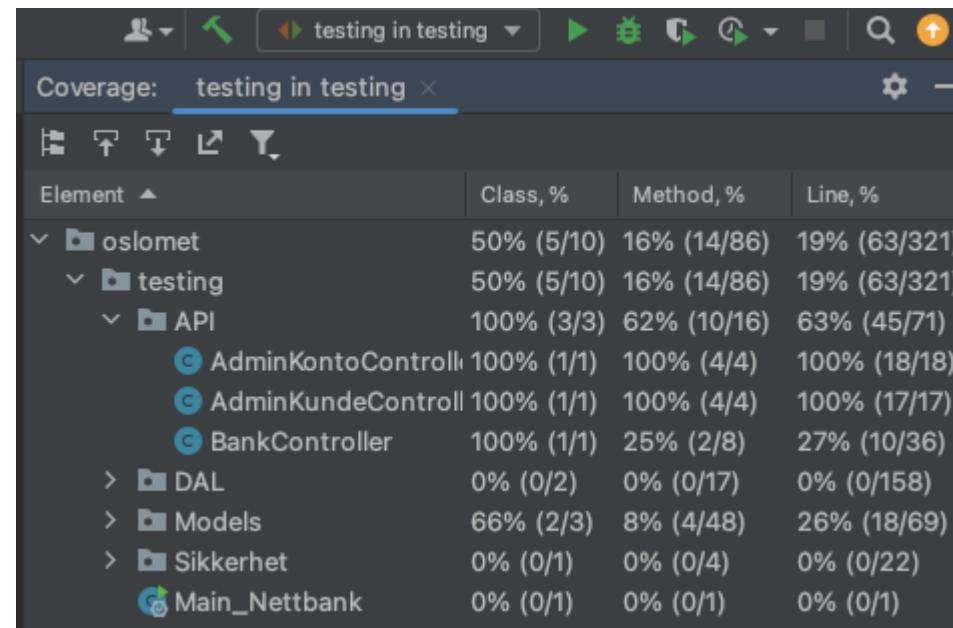
Beskriv de ulike testmiljøene og verktøy benyttet for de tre testtypene.

En MacBook koblet til OsloMet-nettverket kjørte alle testene. For utføringen av testtypene ble ulike verktøy brukt. For enhetstestin ble kodeprogrammet IntelliJ IDE brukt. For integrasjonstesting og dokumentasjon av testene ble SoapUI og Excel brukt. Til sist ble google chrome utvidelsen Selenium IDE brukt for systemtesting med MTM-Azure(Microsoft Test Management) brukt til dokumentasjon.

7. Resultatdokumenter

- **Sluttrapport for test**
- **Testplan** (dette dokument)
- **Testmanuskript og testtilfeller** (Test Scenarier/ Test Cases, både automatiske og evt. manuelle scripts)
- **Testlogger, feilrapporter**
- **Beskrivelse av test input og output data, filer, databaser**
- **Test programmer** (test data generatorer, stubber, etc.)





Coverage: testing in testing

Element	Class, %	Method, %	Line, %
oslomet	50% (5/10)	16% (14/86)	19% (63/321)
testing	50% (5/10)	16% (14/86)	19% (63/321)
API	100% (3/3)	62% (10/16)	63% (45/71)
AdminKontoController	100% (1/1)	100% (4/4)	100% (18/18)
AdminKundeController	100% (1/1)	100% (4/4)	100% (17/17)
BankController	100% (1/1)	25% (2/8)	27% (10/36)
DAL	0% (0/2)	0% (0/17)	0% (0/158)
Models	66% (2/3)	8% (4/48)	26% (18/69)
Sikkerhet	0% (0/1)	0% (0/4)	0% (0/22)
Main_Nettbank	0% (0/1)	0% (0/1)	0% (0/1)

```
1 package oslomet.testing;
2
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.mockito.InjectMocks;
6 import org.mockito.Mock;
7 import org.mockito.junit.MockitoJUnitRunner;
8 import oslomet.testing.API.BankController;
9 import oslomet.testing.DAL.BankRepository;
10 import oslomet.testing.Models.Konto;
11 import oslomet.testing.Models.Kunde;
12 import oslomet.testing.Sikkerhet.Sikkerhet;
13
14 import java.util.ArrayList;
15 import java.util.List;
16
17 import static org.junit.jupiter.api.Assertions.
18     assertEquals;
19 import static org.junit.jupiter.api.Assertions.
20     assertNull;
21 import static org.mockito.ArgumentMatchers.anyString;
22 import static org.mockito.Mockito.when;
23
24 @RunWith(MockitoJUnitRunner.class)
25 public class EnhetstestBankController {
26
27     @InjectMocks
28     // denne skal testes
29     private BankController bankController;
30
31     @Mock
32     // denne skal Mock'es
33     private BankRepository repository;
34
35     @Mock
36     // denne skal Mock'es
37     private Sikkerhet sjekk;
38
39     @Test
40     public void hentKundeInfo_loggetInn() {
```

```
40      // arrange
41      Kunde enKunde = new Kunde("01010110523",
42          "Lene", "Jensen", "Askerveien 22", "
43          3270",
44          "Asker", "22224444", "HeiHei");
45
46      when(sjekk.loggetInn()).thenReturn("01010110523");
47
48      when(repository.hentKundeInfo(anyString())).thenReturn(enKunde);
49
50      // act
51      Kunde resultat = bankController.hentKundeInfo();
52
53      // assert
54      assertEquals(enKunde, resultat);
55
56  @Test
57  public void hentKundeInfo_IkkeloggetInn() {
58
59      // arrange
60      when(sjekk.loggetInn()).thenReturn(null);
61
62      //act
63      Kunde resultat = bankController.hentKundeInfo();
64
65      // assert
66      assertNull(resultat);
67  }
68
69  @Test
70  public void hentKonti_LoggetInn()  {
71      // arrange
72      List<Konto> konti = new ArrayList<>();
73      Konto kontol1 = new Konto("105010123456", "
74          01010110523",
75          720, "Lønnskonto", "NOK", null);
```

```
75      Konto konto2 = new Konto("105010123456", "12345678901", 1000, "Lønnskonto", "NOK", null);  
76      konti.add(konto1);  
77      konti.add(konto2);  
78  
79      when(sjekk.loggetInn()).thenReturn("01010110523");  
80  
81      when(repository.hentKonti(anyString())).  
thenReturn(konti);  
82  
83      // act  
84      List<Konto> resultat = bankController.  
hentKonti();  
85  
86      // assert  
87      assertEquals(konti, resultat);  
88  }  
89  
90  
91  @Test  
92  public void hentKonti_IkkeLoggetInn() {  
93      // arrange  
94  
95      when(sjekk.loggetInn()).thenReturn(null);  
96  
97      // act  
98      List<Konto> resultat = bankController.  
hentKonti();  
99  
100     // assert  
101     assertNull(resultat);  
102  }  
103 }  
104  
105
```

```
1 package oslomet.testing;
2
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.mockito.InjectMocks;
6 import org.mockito.Mock;
7 import org.mockito.junit.MockitoJUnitRunner;
8 import oslomet.testing.API.AdminKontoController;
9 import oslomet.testing.DAL.AdminRepository;
10 import oslomet.testing.Models.Konto;
11 import oslomet.testing.Sikkerhet.Sikkerhet;
12
13 import java.util.ArrayList;
14 import java.util.List;
15
16 import static org.junit.Assert.assertEquals;
17 import static org.junit.Assert.assertNull;
18 import static org.mockito.Mockito.*;
19
20
21 @RunWith(MockitoJUnitRunner.class)
22 public class EnhetstestKontoController {
23
24     @InjectMocks
25     // denne skal testes
26     private AdminKontoController kontoController;
27
28     @Mock
29     // denne skal Mock'es
30     private AdminRepository repository;
31
32     @Mock
33     // denne skal Mock'es
34     private Sikkerhet sjekk;
35
36     @Test
37     public void hentAlleKontiTest() {
38
39         // arrange
40
41         //Tar eksisterende listen over kontoer. Lager
```

```
41  en kopi av denne og kaller den for expectedList
42      List<Konto> expectedList = new ArrayList<>();
43      //Sjekker og logger inn med personnummeret
44      //01010110523
45      when(sjekk.loggetInn()).thenReturn("01010110523");
46      //Henter alle kontoene til det gitte
47      //personnummeret og putter det inn i expectedList
48      when(repository.hentAlleKonti()).thenReturn(
49          expectedList);
50
51      // act
52      List<Konto> result = kontoController.
53      hentAlleKonti();
54
55      // assert
56      assertEquals(expectedList, result);
57  }
58
59  @Test public void hentAlleKontiTest_NotLoggedIn
60  () {
61
62      //Arrange
63      //Gir ikke adgang hvis bruker ikke finnes i
64      //systemet
65      when(sjekk.loggetInn()).thenReturn(null);
66
67      //Act
68      List<Konto> result = kontoController.
69      hentAlleKonti();
70
71      //Assert
72      assertNull(result);
73  }
74
75
76  @Test
77  public void registrerKontoTest()  {
78      // arrange
79      //lager ny konto
80      Konto konto = new Konto();
```

```
74          //Sjekker om kunde er logget inn
75          when(sjekk.loggetInn()).thenReturn("01010110523");
76
77          //Registrerer konto
78          when(repository.registrerKonto(konto)).
79          thenReturn("Seksess, en ny konto har blitt
80          registrert");
81          // act
82          String retur = kontoController.
83          registrerKonto(konto);
84
85          // assert
86          //returnerer den nye kontoen
87          assertEquals("Seksess, en ny konto har blitt
88          registrert", retur);
89
90
91      @Test public void registrerKontoTest_NotLoggedIn
92      () {
93          // Arrange
94          Konto konto = new Konto();
95          when(sjekk.loggetInn()).thenReturn(null);
96          // Act
97          String result = kontoController.
98          registrerKonto(konto);
99          //Assert
100         assertEquals("Ikke innlogget", result);
101
102
103     @Test public void endreKontoTest(){
104         //Arrange
105         Konto konto = new Konto();
106         when(sjekk.loggetInn()).thenReturn("01010110523");
107         when(repository.endreKonto(konto)).
108         thenReturn("Endre konto");
109
110         //Act
111         String result = kontoController.endreKonto(
112         konto);
```

```
105
106      //Assert
107      assertEquals("Endre konto", result);
108  }
109
110 @Test public void endreKonto_NotLoggedIn() {
111     //Arrange
112     Konto konto = new Konto();
113     when(sjekk.loggetInn()).thenReturn(null);
114
115     //Act
116     String result = kontoController.endreKonto(
117         konto);
118
119     //Assert
120     assertEquals("Ikke innlogget", result);
121 }
122
123 @Test public void slettKontoTest(){
124     //Arrange
125     Konto kontonummer = new Konto();
126     when(sjekk.loggetInn()).thenReturn(
127         "01010110523");
128     when(repository.slettKonto(String.valueOf(
129         kontonummer))).thenReturn("Slettet konto");
130
131     //Act
132     String result = kontoController.slettKonto(
133         String.valueOf(kontonummer));
134
135     //Assert
136     assertEquals("Slettet konto", result);
137 }
138
139 @Test public void slettKontoTest_NotLoggedIn(){
140     //Arrange
141     Konto kontonummer = new Konto();
142     when(sjekk.loggetInn()).thenReturn(null);
143
144     //Act
145     String result = kontoController.slettKonto(
```

```
141 String.valueOf(kontonummer));  
142  
143     //Assert  
144     assertEquals("Ikke innlogget", result);  
145 }  
146 }  
147  
148  
149
```

```
1 package oslomet.testing;
2
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.mockito.InjectMocks;
6 import org.mockito.Mock;
7 import org.mockito.junit.MockitoJUnitRunner;
8 import oslomet.testing.API.AdminKontoController;
9 import oslomet.testing.API.AdminKundeController;
10 import oslomet.testing.DAL.AdminRepository;
11 import oslomet.testing.Models.Konto;
12 import oslomet.testing.Models.Kunde;
13 import oslomet.testing.Sikkerhet.Sikkerhet;
14
15 import java.util.ArrayList;
16 import java.util.List;
17
18 import static org.junit.Assert.assertEquals;
19 import static org.junit.Assert.assertNull;
20 import static org.mockito.ArgumentMatchers.any;
21 import static org.mockito.Mockito.when;
22
23 @RunWith(MockitoJUnitRunner.class)
24 public class EnhetstestKundeController {
25     @InjectMocks
26     // denne skal testes
27     private AdminKundeController kundeController;
28
29     @Mock
30     // denne skal Mock'es
31     private AdminRepository repository;
32
33     @Mock
34     // denne skal Mock'es
35     private Sikkerhet sjekk;
36
37     @Test
38     public void hentAlleKunderTest() {
39         //Arrange
40         List<Kunde> expectedList = new ArrayList<>();
41         when(sjekk.loggetInn()).thenReturn("
```

```
41 01010110523");
42     when(repository.hentAlleKunder()).thenReturn(
43         expectedList);
44     //Act
45     List<Kunde> result = kundeController.hentAlle
46     ();
47     //Assert
48     assertEquals(expectedList, result);
49 }
50
51 @Test
52 public void hentAlleKunderTest_NotLoggedIn() {
53     //Arrange
54     List<Kunde> expectedList = new ArrayList<>();
55     when(sjekk.loggetInn()).thenReturn(null);
56
57     //Act
58     List<Kunde> result = kundeController.hentAlle
59     ();
60     //Assert
61     assertEquals(null, result);
62 }
63
64 @Test
65 public void lagreKundeTest() {
66     // arrange
67     //lager ny kunde
68     Kunde innKunde = new Kunde();
69     //Sjekker om kunde er logget inn
70     when(sjekk.loggetInn()).thenReturn(
71         "01010110523");
72     //Registrarer kunde
73     when(repository.registrerKunde(any(Kunde.
74         class))).thenReturn("Kunde lagret");
75     // act
76     String retur = kundeController.lagreKunde(
77         innKunde);
```

```
76
77         // assert
78         //returnerer den nye kunden
79         assertEquals("Kunde lagret", retur);
80     }
81
82     @Test
83     public void lagreKundeTest_NotLoggedIn() {
84         Kunde innKunde = new Kunde();
85         when(sjekk.loggetInn()).thenReturn(null);
86
87         String retur = kundeController.lagreKunde(
88             innKunde);
89         assertEquals("Ikke logget inn", retur);
90     }
91
92     @Test
93     public void endreKundeTest() {
94         Kunde innKunde = new Kunde();
95         when(sjekk.loggetInn()).thenReturn(
96             "01010110523");
97         when(repository.endreKundeInfo(any(Kunde.
98             class))).thenReturn("Kunde endret");
99
100        String retur = kundeController.endre(
101            innKunde);
102
103        assertEquals("Kunde endret", retur);
104    }
105
106    @Test
107    public void endreKundeTest_NotLoggedIn() {
108        Kunde innKunde = new Kunde();
109        when(sjekk.loggetInn()).thenReturn(null);
110
111        String retur = kundeController.endre(
112            innKunde);
113
114        assertEquals("Ikke logget inn", retur);
115    }
116
```

```
112
113     @Test
114     public void slettKundeTest() {
115         when(sjekk.loggetInn()).thenReturn("01010110523");
116         when(repository.slettKunde("01010110523")).thenReturn("Kunde slettet");
117
118         String retur = kundeController.slett("01010110523");
119
120         assertEquals("Kunde slettet", retur);
121     }
122
123     @Test
124     public void slettKundeTest_NotLoggedIn() {
125         when(sjekk.loggetInn()).thenReturn(null);
126
127         String retur = kundeController.slett("01010110523");
128
129         assertEquals("Ikke logget inn", retur);
130     }
131 }
132
```

Integrasjonstester av Nettbank

Change Log	Person	Date
Laget dokument	Danils Trofimovs s374922	12-Mar-24
Lagt inn informasjon	Danils Trofimovs s374923	12-Mar-24

Test Results	Number
Total Test Cases Passed	40
Total Test Cases Failed	0

Test Summary

Total Passed 8
 Total Failed 0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
	ADMIN KONTO				
AdminKontoTest	Logger inn som Admin	Logget inn	bruker, passord	Logget inn	Pass
AdminKontoTest	Hent alle kontoer	Oversikt over kontoer	adminKonto/hentAlle	Oversikt over kontoer (JSON)	Pass
AdminKontoTest	> Antall Kontoer	3	\$	3	Pass
AdminKontoTest	> Konto Type Konto1	Lønnskonto	\$.[0].type	Lønnskonto	Pass
AdminKontoTest	> Konto Type Konto2	Sparekonto	\$.[1].type	Sparekonto	Pass
AdminKontoTest	> Konto Type Konto3	Brukskonto	\$.[2].type	Brukskonto	Pass
AdminKontoTest	Registrer ny konto	OK	"personnummer": "01010110523", "kontonummer": "105010123458", "saldo": 0.0, "type": "Sparekonto", "valuta": "NOK", "transaksjoner": null	OK	Pass
AdminKontoTest	Endre konto	OK	"personnummer": "01010110523", "kontonummer": "105010123458", "saldo": 0.0, "type": "Brukskonto", "valuta": "NOK", "transaksjoner": null	OK	Pass

Test Summary

Total Passed 8
 Total Failed 0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
	ADMIN KUNDE				
AdminKundeTest	Logger inn som Admin	Logget inn	bruker, passord	Logget inn	Pass
AdminKundeTest	Hent alle kunder	Oversikt over kunder	adminKunde/hentAlle	Oversikt over kunder (JSON)	Pass
AdminKundeTest	> Antall Kunder	2	\$	2	Pass
AdminKundeTest	> Kunde Fornavn Kunde1	Lene	\$.[0].fornavn	Lene	Pass
AdminKundeTest	> Kunde Fornavn Kunde2	Per	\$.[1].type	Per	Pass
AdminKundeTest	Registrer ny kunde	OK	"personnummer": "12345678900", "fornavn": "Daniel", "etternavn": "Fryjordet", "adresse": "Osloveien 6", "postnr": "1262", "poststed": "Oslo", "telefonnr": "96874556", "passord": "HeiHei"	OK	Pass

AdminKundeTest	Endre kunde	OK	<pre>"personnummer": "12345678900", "fornavn": "Henrik", "etternavn": "Fryjordet", "adresse": "Osloveien 6", "postnr": "1262", "poststed": "Oslo", "telefonnr": "96874556", "passord": "HeiHei"</pre>	OK	Pass
AdminKundeTest	Slett kunde	OK	personnummer: 12345678900	OK	Pass

Test Summary

Total Passed	24
Total Failed	0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
KUNDE					
BankTest	Logge Inn Kunde	OK	personnummer: 01010110523	OK	Pass
BankTest	Hente Transaksjoner				Pass
BankTest	> Antall Transaksjoner		1 \$		1 Pass
BankTest	> Sjekker kontonummer	105010123456	\$.kontonummer	105010123456	Pass
BankTest	> Sjekker personnummer	1010110523	\$.personnummer	1010110523	Pass
BankTest	Hente Kontoer til Kunden				Pass
BankTest	> Antall Kontoer		3 \$		3 Pass
BankTest	> Kontonummer1	105020123456	\$.[0].kontonummer	105020123456	Pass
BankTest	> Kontonummer1	22334412345	\$.[0].kontonummer	22334412345	Pass
BankTest	> Kontonummer2	105020123456	\$.[1].kontonummer	105020123456	Pass

BankTest	Registrere Ny Betaling	OK	"txID": 10, "fraTilKontonummer": "20102012345", "belop": 1, "dato": "2015-03-15", "melding": "Hallo", "avventer": "1", "kontonummer": "105010123456"	OK	Pass
BankTest	Hente Eksisterende Betalinger				Pass
BankTest	> Antall Betalinger		4 \$		4 Pass
BankTest	> txID1		1 \$.[0].txID		1 Pass
BankTest	> txID2		2 \$.[1].txID		2 Pass
BankTest	Utfør Betaling		txID: 10		Pass
BankTest	> Antall Betalinger		3 \$		3 Pass
BankTest	> Til Kontonummer1	20102012345	\$.[0].fraTilKontonummer	20102012345	Pass
BankTest	> Til Kontonummer2	20102012347	\$.[2].fraTilKontonummer	20102012347	Pass
BankTest	Hente Kunde Informasjon				Pass
BankTest	> Antall		1 \$		1 Pass
BankTest	> Personnummer	01010110523	\$.personnummer	01010110523	Pass
BankTest	> Fornavn	Lene	\$.fornavn	Lene	Pass
			fornavn: "Lene", etternavn: "Jensen", adresse: "Askerveien 22", postnr: "3270", poststed: "Asker", telefonnr: "22224444", passord: "HeiHei"		
BankTest	Endre Kunde Informasjon	OK		OK	Pass

The image shows a software interface for managing test cases, likely a test runner or a test management tool. On the left, a tree view displays a hierarchy of test cases and their steps. On the right, a results summary window shows the status of each test case.

Test Case Hierarchy (Left):

- AdminKontoTest (Passed)
 - Test Steps (5)
 - LoggInnAdmin
 - hentKonti
 - registrerKonto
 - endreKonto
 - slettKonto
 - Load Tests (0)
 - Security Tests (0)- AdminKundeTest (Passed)
 - Test Steps (5)
 - loggInnAdmin2
 - hentAlle
 - lagreKunde
 - endreKunde
 - slettKunde
 - Load Tests (0)
 - Security Tests (0)
- BankTest (Passed)
 - Test Steps (9)
 - LoggInnKunde
 - hentTransaksjoner
 - hentKonti
 - hentSaldi
 - registrerBetaling
 - hentBetalinger
 - utforBetaling
 - hentKundeInfo
 - endreKundeInfo

Test Results Summary (Right):

The results summary window is titled "KundeTestCases" and shows the following test cases and their statuses:

- AdminKontoTest: FINISHED
- AdminKundeTest: FINISHED
- BankTest: FINISHED

Below the results, there are tabs for "Description", "Properties", "Setup Script", and "TearDown Script".

Executing	▷	▷	○	⌚	▼
http://localhost:8080					
	Command			Target	Value
X AdminEndreKonto	1	✓ open		/	
X AdminEndreKunde	2	✓ set window size		1166x1032	
X AdminRegistrereKonto	3	✓ click		linkText=Logg inn Admin	
X AdminRegistrereKunde	4	✓ click		id=bruker	
✓ KundeEndreInfo	5	✓ type		id=bruker	Admin
X KundeRegistrerBetaling	6	✓ click		id=passord	
✓ KundeSaldo	7	✓ type		id=passord	Admin
✓ KundeUtførOgSjekkBetaling	8	✓ click		id=logginn	
	9	✓ click		linkText=Endre konto	
	10	✓ click		id=saldo1	
	11	✓ type		id=saldo1	0
	12	✓ click		linkText=Endre	
	13	✓ assert value		id=saldo1	0
	14	✓ click		id=saldo1	
	15	✓ choose ok on next confirmation			
	16	✓ click		css=tr:nth-child(2) .btn-danger	
	17	✓ assert not text		id=kontonummer2	105020123456
	18	X assert value		id=saldo1	0

X AdminEndreKonto	http://localhost:8080		
X AdminEndreKunde	Command	Target	Value
X AdminRegistrereKonto	1 ✓ open	/	
X AdminRegistrereKunde	2 ✓ set window size	1425x1032	
✓ KundeEndreInfo	3 ✓ click	linkText=Logg inn Admin	
X KundeRegistrerBetaling	4 ✓ click	id=bruker	
✓ KundeSaldo	5 ✓ type	id=bruker	Admin
✓ KundeUtførOgSjekkBetalung	6 ✓ click	id=passord	
	7 ✓ type	id=passord	Admin
	8 ✓ click	id=loggInn	
	9 ✓ click	id=fornavn1	
	10 ✓ type	id=fornavn1	Lene1
	11 ✓ click	linkText=Endre	
	12 ✓ assert value	id=fornavn1	Lene1
	13 ✓ click	id=fornavn1	
	14 ✓ choose ok on next confirmation	css=tr:nth-child(2) .btn-danger	
	15 ✓ click		
	16 X assert value	id=fornavn1	Lene1

X AdminEndreKonto	http://localhost:8080		
X AdminEndreKunde	Command	Target	Value
X AdminRegistrereKonto	1 ✓ open	/	
X AdminRegistrereKunde	2 ✓ set window size	1425x1032	
✓ KundeEndreInfo	3 ✓ click	linkText=Logg inn Admin	
X KundeRegistrerBetaling	4 ✓ click	id=bruker	
✓ KundeSaldo	5 ✓ type	id=bruker	Admin
✓ KundeUtførOgSjekkBetaling	6 ✓ click	id=passord	
	7 ✓ type	id=passord	Admin
	8 ✓ click	id=loggInn	
	9 ✓ click	linkText=Register konto	
	10 ✓ click	id=kontoNr	
	11 ✓ type	id=kontoNr	11223344556
	12 ✓ click	id=type	
	13 ✓ type	id=type	Brukskonto
	14 ✓ click	id=valuta	
	15 ✓ type	id=valuta	NOK
	16 ✓ click	id=personnummer	
	17 ✓ type	id=personnummer	01010110523
	18 ✓ click	id=regKonto	
	19 ✓ click	linkText=Endre konto	
	20 X assert element present	id=kontonummer4	11223344556

X AdminEndreKonto	http://localhost:8080		
X AdminEndreKunde	Command	Target	Value
X AdminRegistrereKonto	8 ✓ click	id=loggInn	
X AdminRegistrereKunde	9 ✓ click	linkText=Registrer kunde	
✓ KundeEndreInfo	10 ✓ click	id=personnummer	
X KundeRegistrerBetaling	11 ✓ type	id=personnummer	98765432109
✓ KundeSaldo	12 ✓ click	id=fornavn	
✓ KundeUtførOgSjekkBetaling	13 ✓ type	id=fornavn	Hans
	14 ✓ click	id=etternavn	
	15 ✓ type	id=etternavn	Hansen
	16 ✓ click	id=adresse	
	17 ✓ type	id=adresse	Osloveien 6
	18 ✓ click	id=postnr	
	19 ✓ type	id=postnr	0162
	20 ✓ click	id=poststed	
	21 ✓ type	id=poststed	Oslo
	22 ✓ click	id=telefonnr	
	23 ✓ type	id=telefonnr	12345678
	24 ✓ click	id=password	
	25 ✓ type	id=password	HeiHei
	26 ✓ click	id=regKunde	
	27 ✓ click	linkText=Endre kunde	
	28 X assert element present	id=personnummer3	98765432109

X AdminEndreKonto	http://localhost:8080		
X AdminEndreKunde	Command	Target	Value
X AdminRegistrereKonto	1 ✓ open	/	
X AdminRegistrereKunde	2 ✓ set window size	974x1032	
✓ KundeEndreInfo	3 ✓ click	css=html	
X KundeRegistrerBetalung	4 ✓ click	linkText=Logg inn	
✓ KundeSaldo	5 ✓ click	id=personNr	
✓ KundeUtførOgSjekkBetalung	6 ✓ type	id=personNr	01010110523
	7 ✓ click	id=password	
	8 ✓ type	id=password	HeiHei
	9 ✓ click	id=loggInn	
	10 ✓ click	linkText=Kunde-info	
	11 ✓ click	id=fornavn	
	12 ✓ type	id=fornavn	Lene1
	13 ✓ click	id=endre	
	14 ✓ assert value	id=fornavn	Lene1
	15 ✓ click	linkText=Kunde-info	
	16 ✓ click	id=fornavn	
	17 ✓ type	id=fornavn	Lene
	18 ✓ click	id=endre	
	19 ✓ assert value	id=fornavn	Lene

Executing	▶	▶	▶	▶
	http://localhost:8080			Value
	Command		Target	
X AdminEndreKonto	1	✓ open	/registerBetaling.html	
X AdminEndreKunde	2	✓ set window size	1127x1032	
X AdminRegistrereKonto	3	✓ click	id=tilKonto	
✓ KundeEndreInfo	4	✓ click	id=tilKonto	
X KundeRegistrerBetaling	5	✓ double click	id=tilKonto	
✓ KundeSaldo	6	✓ type	id=tilKonto	105010123456
✓ KundeUtførOgSjekkBetaling	7	✓ click	id=belop	
	8	✓ type	id=belop	3500
	9	✓ click	id=regBetaling	
	10	✓ click	linkText=Transaksjoner	
	11	✓ click	id=fraDato	
	12	✓ type	id=fraDato	2012-12-12
	13	✓ click	id=tilDato	
	14	✓ type	id=tilDato	2024-12-12
	15	✓ click	id=kontoNr	
	16	✓ select	id=kontoNr	label=105010123456
	17	✓ click	id=sok	
	18	X assert text	css=tr:nth-child(1) > td:nth-child(3)	3500

X AdminEndreKonto	http://localhost:8080		
X AdminEndreKunde	Command	Target	Value
X AdminRegistrereKonto	1 ✓ open	/	
X AdminRegistrereKunde	2 ✓ set window size	1918x1030	
✓ KundeEndreInfo	3 ✓ click	linkText=Logg inn	
X KundeRegistrerBetaling	4 ✓ click	id=personNr	
✓ KundeSaldo	5 ✓ type	id=personNr	01010110523
✓ KundeUtførOgSjekkBetaling	6 ✓ click	id=password	
	7 ✓ type	id=password	HeiHei
	8 ✓ click	id=loggInn	
	9 ✓ assert text	css=tr:nth-child(1) > td:nth-child(1)	22334412345

Executing	▶	▶	○	🕒
	http://localhost:8080			
	Command	Target	Value	
X AdminEndreKonto	1 ✓ open	/		
X AdminEndreKunde	2 ✓ set window size	1255x1032		
X AdminRegistrereKonto	3 ✓ click	linkText=Logg inn		
✓ KundeEndreInfo	4 ✓ click	id=personNr		
X KundeRegistrerBetalung	5 ✓ type	id=personNr		01010110523
✓ KundeSaldo	6 ✓ click	id=passord		
✓ KundeUtførOgSjekkBetalung	7 ✓ type	id=passord		HeiHei
	8 ✓ click	id=loggInn		
	9 ✓ click	linkText=Utfør betalinger		
	10 ✓ click	css=tr:nth-child(1) .btn		
	11 ✓ click	linkText=Transaksjoner		
	12 ✓ click	id=fraDato		
	13 ✓ type	id=fraDato		2012-12-12
	14 ✓ click	id=tilDato		
	15 ✓ type	id=tilDato		2015-03-20
	16 ✓ click	id=kontoNr		
	17 ✓ select	id=kontoNr		label=105010123456
	18 ✓ click	id=sok		
	19 ✓ assert text	css=tr:nth-child(1) > td:nth-child(4)		Fjordkraft
	20 ✓ mouse over	id=sok		
	21 ✓ mouse out	id=sok		

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [13](#): 1 : Som kunde ønsker jeg å kunne endre på min informasjon

Test cases (1)

Test case [23](#): Endre informasjon til kunde

PROPERTIES

Test Case Id:	23
Assigned To:	Danils Trofimovs
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Gå til localhost:8080	Få opp innlogging portalen til nettbanken	
2	Logg inn	Oversikt over saldo	
3	Trykk på "Kunde-Info" knapp	Få opp side med din informasjon	
4	Trykk på inputfeltet for å endre informasjon	Ny informasjon om kunde blir skrevet inn	
5	Trykk på "Endre info" knapp	Informasjon endret	

LINKS

ID	WorkItem Type	Link type	Title
about:blank			

[1](#) User Story

Tests

Som kunde ønsker jeg å kunne endre på min informasjon

[24](#) Bug

Tests

Endre informasjon til kunde Failed Knappen "Endre info" Fungerer ikke

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Danils Trofimovs	Windows 10	Danils Trofimovs	Tuesday, March 12, 2024 0:00:00.000	0:00:00.000	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [14](#): 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner

Test cases (1)

Test case [26](#): Oversikt over kundens transaksjoner

PROPERTIES

Test Case Id:	26
Assigned To:	Danils Trofimovs
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Logg inn som kunde på localhost:8080	Du kommer inn på kundens nettbank	
2	Trykk på "Transaksjoner i navigasjonsbaren"	Kommer til side med oversikt over transaksjoner	
3	Velg konto f.eks. "105020123456" og velg tidsperiode (til- og fra dato)	Input blir lagret i feltene	
4	Trykk på knappen "Søk"	Oversikt over utførte transaksjoner for valgt konto over bestemt tidsperiode	

LINKS

ID	WorkItemType	Link type	Title
2	User Story	Tests	Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Danils Trofimovs	Windows 10	Danils Trofimovs	Tuesday, March 12, 2024 0:00:00.000	0:00:00.000	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [15](#): 3 : Som kunde ønsker jeg å se saldo

Test cases (1)

Test case [25](#): Kunde Saldo

PROPERTIES

Test Case Id:	25
Assigned To:	Sjur Svardal
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	Åpner innloggingsportalen vinduet	
2	Kunden skriver inn sitt personnummer og passord	Innloggingen sender deg rett til saldo oversiktssiden	

LINKS

ID	WorkItemType	Link type	Title
3	User Story	Tests	Som kunde ønsker jeg å se saldo

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Sjur Svardal	Windows 10	Sjur Svardal	Tuesday, March 12, 2024 0:00:00.000	0:00:00.000	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [16](#): 4 : Som kunde ønsker jeg å kunne registrere betaling

Test cases (1)

Test case [28](#): Registrere betaling som kunde

PROPERTIES

Test Case Id:	28
Assigned To:	Danils Trofimovs
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Logg inn som kunde på localhost:8080	Du kommer inn på kundens nettbank	
2	Trykk på "Register betaling" i navigasjonsbaren	Du kommer inn på siden for å registrere betaling	
3	Velg hvilken konto du vil betale fra, hvilken konto du vil betale til, betalingsdato, beløp, melding og trykk på "Registrer betaling"	Da skal betalingen registreres og være tilgjengelig på transaksjoner	

LINKS

ID	WorkItemType	Link type	Title

[4](#) User Story

Tests

Som kunde ønsker jeg å kunne registrere betaling

[30](#) Bug

Tests

Knappen for å registrere betaling fungerer ikke. Betaling blir ikke registrert, error 415

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Failed	Danils Trofimovs	Windows 10	Danils Trofimovs	Tuesday, March 12, 2024 0:00:00.000	0:00:00.000	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [17](#): 5 : Som kunde ønsker jeg å utføre betalinger

Test cases (1)

Test case [29](#): Kunde utføre betaling

PROPERTIES

Test Case Id:	29
Assigned To:	Sjur Svardal
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	kommer inn til innloggingssiden	
2	Logg deg inn som kunde med personnummer og passord	Kommer til saldo oversikten	
3	Trykk deretter på "utfør betalinger" fanen	Kommer til betalings side med oversikt over planlagte betalinger	
4	Trykk på knappen "utfør betaling"	Betalingen går gjennom og den fjernes	

LINKS

ID	WorkItemType	Link type	Title

5 User Story

Tests

Som kunde ønsker jeg å utføre betalinger

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Sjur Svardal	Windows 10	Sjur Svardal	Tuesday, March 12, 2024 0:00:00.000	0	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [18](#): 6 : Som admin ønsker jeg å kunne endre/slette kunder

Test cases (1)

Test case [27](#): Admin endre/slette kunder

PROPERTIES

Test Case Id:	27
Assigned To:	Henrik Fryjordet Wiik
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	kommer til innloggings siden	
2	Trykk på "logg in admin" knappen i høyre hjørne	kommer til Admin innloggings siden	
3	Logg deg inn som admin	kommer til "endre kunde" siden på nettsiden	
4	Redigerer på en av fletene og trykker på "endre" knappen til høyre for feltet	informasjonen i feltet er endret	
5	Trykk på "slett" knappen til høyre	hele raden med info om kunden skal forsvinne	

LINKS

ID	WorkItemType	Link type	Title
36	Bug	Tests	Fungerer ikke å endre kunde informasjon som Admin, error 415
6	User Story	Tests	Som admin ønsker jeg å kunne endre/slette kunder

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Failed	Henrik Fryjordet Wiik	Windows 10	Danils Trofimovs	Tuesday, March 12, 2024 0:00:00.000	0	0

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [19](#): 7 : Som admin ønsker jeg å registrere kunde

Test cases (1)

Test case [31](#): Admin registrer kunde

PROPERTIES

Test Case Id:	31
Assigned To:	Henrik Fryjordet Wiik
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	kommer til innloggings siden	
2	Trykker på "logg inn admin" knappen	kommer til Admin innloggings siden	
3	Logger deg inn som Admin	kommer inn på "endre kunde" siden av nettsiden	
4	Trykker på "registrer kunde" i meny baren	kommer til "registrer kunde" siden	
5	Legger til info om kunden og trykker på "registrer kunde" knappen	den nye kunden skal være registrert	

LINKS

ID	WorkItemType	Link type	Title
7	User Story	Tests	Som admin ønsker jeg å registrere kunde
37	Bug	Tests	Admin registrer kunde Failed/ Knappen fungerer ikke

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Failed	Henrik Fryjordet Wiik	Windows 10	Sjur Svardal	Tuesday, March 12, 2024 0:00:00.000	0	

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [20](#): 8 : Som admin ønsker jeg å se oversikt over mine kunder

Test cases (1)

Test case [33](#): Admin Oversikt over kunder

PROPERTIES

Test Case Id:	33
Assigned To:	Sjur Svardal
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	Kommer til innloggings siden	
2	Trykk på knappen "logg inn Admin"	Kommer til admin innloggingssiden	
3	Logg inn som admin med brukernavn og passord	Kommer til oversiktssiden med alle kundene	

LINKS

ID	WorkItemType	Link type	Title
8	User Story	Tests	Som admin ønsker jeg å se oversikt over mine kunder

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Sjur Svardal	Windows 10	Sjur Svardal	Tuesday, March 12, 2024 0:00:00.000	0	

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønsker jeg å endre/slette kontoer (ID: [22](#))

Test suite [21](#): 9 : Som Admin ønsker jeg å registrere konto

Test cases (1)

Test case [32](#): Registrere konto som Admin

PROPERTIES

Test Case Id:	32
Assigned To:	Danils Trofimovs
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Gå inn på localhost:8080	Få opp nettbank nettside	
2	Trykk på Logg inn Admin	Tar deg med til innlogging for admin	
3	Skriv inn brukernavn "Admin" og passord "Admin"	Logger deg inn som admin bruker	
4	Trykk på "Register konto" i navigasjonsbaren	Tar deg med til side for å registrere konto	
5	Tast inn kontonummer, type, valuta, personnummer og trykk på "Registrer konto"	Lager ny konto for denne personen	

LINKS

ID	WorkItemType	Link type	Title
35	Bug	Tests	Knappen registrer konto fungerer ikke. Konto blir ikke registrert, error 415
9	User Story	Tests	Som Admin ønsker jeg å registrere konto

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Failed	Danils Trofimovs	Windows 10	Danils Trofimovs	Tuesday, March 12, 2024 0:00:00.000	0	

Test plan [11](#): SystemTest

Suite hierarchy

Static suite: SystemTest (ID: [12](#))

- Requirement-based suite: 1 : Som kunde ønsker jeg å kunne endre på min informasjon (ID: [13](#))
- Requirement-based suite: 2 : Som kunde ønsker jeg å kunne få oversikt over mine transaksjoner (ID: [14](#))
- Requirement-based suite: 3 : Som kunde ønsker jeg å se saldo (ID: [15](#))
- Requirement-based suite: 4 : Som kunde ønsker jeg å kunne registrere betaling (ID: [16](#))
- Requirement-based suite: 5 : Som kunde ønsker jeg å utføre betalinger (ID: [17](#))
- Requirement-based suite: 6 : Som admin ønsker jeg å kunne endre/slette kunder (ID: [18](#))
- Requirement-based suite: 7 : Som admin ønsker jeg å registrere kunde (ID: [19](#))
- Requirement-based suite: 8 : Som admin ønsker jeg å se oversikt over mine kunder (ID: [20](#))
- Requirement-based suite: 9 : Som Admin ønsker jeg å registrere konto (ID: [21](#))
- Requirement-based suite: 10 : Som admin ønskog jeg å endre/slette kontoer (ID: [22](#))

Test suite [22](#): 10 : Som admin ønskog jeg å endre/slette kontoer

Test cases (1)

Test case [34](#): Admin endre/slette konto

PROPERTIES

Test Case Id:	34
Assigned To:	Sjur Svardal
State:	Design

STEPS

#	Action	Expected value	Attachments
1	Åpne localhost:8080	Åpner innloggings siden	
2	Trykk på knappen "logg inn Admin"	Kommer til innloggingssiden til admin	
3	Skriv inn brukernavn og passord	Kommer til oversiktssiden over kundene	
4	Trykk på knappen "Endre Konto"	Kommer til konto oversiktssiden	
5	Trykk på felte som ønskes å endres og trypp på "Endre" knappen	Verdien du har endret blir stående	
6	Trypp på "Slett" knappen til den ønskede kontoen	Kontoen blir slettet og vil forsvinne	

LINKS

ID	WorkItemType	Link type	Title
10	User Story	Tests	Som admin ønskег jeg å endre/slette kontoer

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration	Build number
Passed	Sjur Svardal	Windows 10	Sjur Svardal	Tuesday, March 12, 2024 0:00:00.000		0

Mal for sluttrapport fra test

Sluttrapporten skal sammenfatte resultatene fra testingen og gi en evaluering av disse. Dette er et meget viktig dokument som vil gi beslutningstakere en indikasjon på kvaliteten til systemet.

1. Identifikasjon

Oppgi:

- Hvilke tester gjelder denne sluttrapporten?
- Enhetstesting
- Integrasjonstesting
- Systemtesting
- Referanse til alt testmateriale (testplan, testdata, testlogg, feillogg etc.), slik at en kan finne det igjen.
- Testplan-Oblig1
- Navn og versjonsnummer for denne sluttrapporten
- Sluttrapport-Oblig1 versjon 1
- Forfatter
- S374951
- S364571
- S374922
- s374907
- Dato
- 15.03.24

2. Sammendrag

Oppsummer evalueringen av det produkt som er testet.

Identifiser også testmiljøet som er brukt.

Dette dokumentet dekker resultatene av testingen av Nettbanken og gir en evaluering av systemets kvalitet. Sluttrapporten omfatter identifikasjon av testtypene, avvik fra testplanen, evaluering av testingens grundighet, identifisering av gjenstående aktiviteter, avvik i produktet, vurdering av produktet som er testet, og en oppsummering av aktiviteter og lærdommer fra testingen. Webapplikasjonen Nettbank ble testet med enhetstester implementert i originalkoden ved hjelp av IntelliJ, integrasjonstester ved hjelp av SoapUI og systemtester ved hjelp av Selenium IDE. Etter å ha utført disse testene, kom vi borti flere avvik i systemet. Vi kom med en delvis løsning på disse avvikene, og gjenstår med to resterende avvik. Vurderingen av produktet indikerer risikoen ved å sette det i drift med og uten implementering av løsninger. Til slutt gir dokumentet en oppsummering av testens aktiviteter og lærdommer. Det nevner

utfordringer i testspesifikasjon og utførelse, verktøyene som ble brukt, og hva som kunne vært gjort annerledes. Erfaringene fra enhetstesting, integrasjonstesting og systemtesting med verktøy som IntelliJ IDE, SoapUI og Selenium IDE, og deres unike fordeler er diskutert. Denne kombinasjonen av testnivåer og verktøy har gitt verdifull innsikt i kvaliteten og påliteligheten til programvaren. Til tross for at vi ikke fikk noen tydelige feil på enhetstestene og integrasjonstestene (før implementering av vår løsning), vil vi fremheve at det fremdeles kan være feil.

3. Avvik i forhold til testplan

Beskriv alle avvik mellom den utførte test og testplanen, testdesign og annen testdokumentasjon, og gi en begrunnelse.

Under hele prosessens varighet var gruppen målrettet med å følge malen vi lagde i starten av prosjektet. Testplanen ble fulgt med få unntak. Underveis i prosessen dokumenterte også gruppen testene jevnlig. Vår hypotese var at alle testene skulle gå igjennom uten noen feil, men det gjorde de ikke siden det var feil i original koden. Testene kom derfor tilbake som feil og vi måtte bruke til å analysere om det var testene eller koden som var feil.

4. Vurdering av testens grundighet

Evaluer testingens grundighet mot de grundighetskrav som er beskrevet i din testplan. Beskriv eventuelle avvik. Beskriv hva som er for lite testet og hvorfor.

Vi mener at testene vi har utført, virker omfattende og grundig utført. Vi har dekket, etter vår mening, alle funksjonene til nettbanken, både fra kundens perspektiv og admins perspektiv. Testene mener vi derfor dekker store deler av systemets handlinger, men anerkjenner fortsatt at det kan være feil ettersom fravær fra feil betyr ikke at feilene ikke eksisterer.

5. Gjenstående aktiviteter/oppgaver

Beskriv hvilke testaktiviteter/oppgaver som eventuelt gjenstår og som må følges opp videre, f.eks. i senere tester. Det kan være aksjonspunkter som andre må ta tak i.

Noen aspekter av nettbank-applikasjonen er komplette likevel gjenstår det deler for at systemet er komplett og kan brukes. Disse delene blir listet i *del 6.2* og baserer seg på mangel på endringer til en funksjonalitet. Feilene som ble funnet i systemtestene, må fikses slik at systemet fungerer som det burde. Etter at feilene blir rettet må systemet testes igjen for å sikre seg at funksjonalitetene fungerer og samhandler med hverandre. Aksjonspunkter er listet i *del 6 punkt 2*.

6. Avvik i produktet

- Gi et sammendrag av testresultatene.

Integrasjonstestene ble skrevet utifra javakoden, koden er det eneste vi har på å se hva informasjon som skal bli sendt via http. Dette fører til at vi skrev integrasjonstester der vi sender informasjon via bodyen. Når vi da gjorde systemtestene, fungerte ikke nettsiden og vi fikk en 415 unsupported media type error. Spørsmålet da er om javakoden er feil, eller måten vi sender http post requesten på er feil, altså javascripten.

- Identifiser alle avvik som ble funnet. Alle avvikene under kategoriserer vi som B.

Admin kunne ikke endre på konto til kunde ettersom knappen “Endre” ga feilmeldingen error 415 unsupported media type.

Admin kunne ikke registrere ny konto til kunde ettersom knappen “Registrer konto” ga feilmeldingen error 415 unsupported media type.

Admin kunne ikke registrere en ny kunde ettersom knappen “Registrer kunde” ga feilmeldingen error 415 unsupported media type.

Admin kunne ikke endre på informasjon til kunde ettersom knappen “Endre” ga feilmeldingen error 415 unsupported media type.

Kunde kunne ikke registrere betaling ettersom knappen “Registrer betaling” ga feilmeldingen error 415 unsupported media type.

- Oppsummer deres løsning.

415 Unsupported media type betyr at http requesten var ikke det backenden forventet. Når vi leser java-koden ser vi at den spør om en body, der body-en er en JSON. Når vi går tilbake til javascript koden ser vi at vi poster til /endre med bruk av \$.post. Altså vi bruker jQuery sin post funksjon. Hvis vi heller bruker \$.ajax med content-type application/json. Der sender vi javascript objektet i bodyen til ajax funksjonen. Dette kan vi teste fordi i integrasjonstesten våres fungerer /endre når vi gir riktig http request.

Enklere forklart, dersom vi fjerner request body fra controllerne vil koden “fungere” som den skal. Når du har gjort dette vil du se at integrasjonstestene som ble laget på forhånd vil feile. Det er også viktig å få med seg at dette ikke løser alle avvikene i produktet. Avvik som registrere betaling og registrere ny kunde eksisterer fortsatt. Registrere betaling går videre, men blir ikke utført. Ny kunde kan ikke registreres med personnummer. Dersom du fyller ut ny kunde uten personnummer, vil det fungere.

Med denne løsningen vil de gjenstående uløste avvik, kategoriseres som C.

- Identifiser alle gjenstående uløste avvik.

Registrere ny betaling (som faktisk går gjennom) som kunde.

Registrere ny kunde (med personnummer) som admin.

7. Vurdering av produktet som er testet

Gi en totalvurdering av hvert testobjekt med dets eventuelle begrensninger. Dette skal baseres på testresultatene og på godkjenningskriteriene som tidligere er planlagt brukt (se testplan).

Gi en indikasjon av risikoen med å sette produktet i drift.

Dersom vi ikke implementerer vår løsning:

Vi ville ikke ha settet produktet i drift ennå ettersom Admin ikke har tilgang til å gjøre noen funksjoner på nettsiden. I tillegg har ikke kundene muligheten til å registrere betalinger, noe som vil være problematisk. Produktet vil kreve mye jobb for den som har tilgang på databasen, ettersom de manuelt må endre på ting.

Dersom vi implementerer vår løsning:

Vi ville vurdert å sette produktet i drift, dersom vi tester systemet på nytt og det kommer løsning på de gjenstående avvikene fortløpende.

8. Sammendrag av aktiviteter og lærdom

Gi en grov beskrivelse av hele testingens forløp. Vekten her er å trekke ut lærdom av det testprosjektet som nå avsluttes.

Beskriv f. eks.:

- Utfordringer i testspesifikasjoner og utførelse
- Anvendte verktøy og erfaringer fra disse
- Hva burde vært gjort annerledes

Erfaringen med å utføre enhetstesting, integrasjonstesting og systemtesting har gitt oss verdifull innsikt i kvaliteten og påliteligheten til programvare. Enhetstesting ble gjennomført med IntelliJ IDE. Det tillot oss å isolere og teste individuelle komponenter av koden for å sikre korrekt utfall. Ved å utføre testene i IntelliJ IDE har det gitt oss en dypere forståelse for hvordan man kan teste sine egne prosjekter fremover og hvordan det kan påvirke hvordan man skriver egen kode.

For integrasjonstesting brukte vi SoapUI, et verktøy som lar oss teste samspillet mellom ulike komponenter. SoapUI har gitt en oversiktlig måte å skape og kjøre integrasjonstester på. Men noen utfordringer som ble overkommet og kjennskapet til programmet økte ble det lettere å

sette seg inn i hvordan testene utførtes og programmet fungerte. Ved å teste større deler av programmet ble konteksten av testene mer synlig og lettere å relatere til. Utføringen av integrasjonstestene var nyttige ettersom det ga oss verdifull kunnskap i hvordan programmet sine komponenter snakket sammen og samhandlet.

For systemtesting benyttet vi Selenium IDE, som muliggjør automatisert testing av webapplikasjoner. I oppgavens kontekst betyde det at vi fikk testet nettsiden nettbank i sin helhet. Hoved fokuset var på hvordan funksjonalitetene fungerte og om nettsiden vi testet mot var velfungerende. Systemtesting ga oss kunnskap om hvordan et godt system skal oppføre seg samtidig som det viste oss hvor lett det var å stole på et uferdig system.

Hver av disse testnivåene og programmene har sine unike fordeler. Mens enhetstesting fokuserer på isolerte komponenter, gir integrasjonstesting et helhetsbilde av systemkomponentenes samspill, og systemtesting sørger for at hele programvaren fungerer som en helhet. IntelliJ IDE, SoapUI og Selenium IDE skiller seg ved å tilby spesifikke funksjonaliteter for deres respektive testnivåer. IntelliJ legger vekt på enkel enhetstestintegrering i utviklingsmiljøet, SoapUI gir omfattende støtte for API-testing og simulering, mens Selenium IDE fokuserer på automatisert webapplikasjonstesting. Denne kombinasjonen av testnivåer og verktøy har forsterket vår evne til å forstå og teste programvare.