

# Sluttrapport for testing av TestingBank

Versjon: 1.0

Dato: 2020-12-03

Forfattere: Jørgen Huseby, Felix Emhjellen, Elin H. Hegsvold, Zakariya Ibrahim og Jostein J.  
Hauge

# Innhold:

<b>1. Identifikasjon</b>	<b>3</b>
<b>2. Sammendrag</b>	<b>4</b>
2.1. Enhetstest	4
2.2. Integrasjonstest	4
2.3. Systemtest	4
<b>3. Avvik i forhold til testplanen</b>	<b>5</b>
<b>4. Vurdering av testens grundighet</b>	<b>5</b>
4.1. Enhetstest	5
4.2. Integrasjonstest	5
4.3. Systemtest	5
<b>5. Gjenstående aktiviteter/oppgaver</b>	<b>6</b>
<b>6. Avvik i produktet</b>	<b>6</b>
6.1. Enhetstest:	6
6.1.1. Identifiserte avvik	6
6.1.2. Løsning på avvik	6
6.1.3. Gjenstående avvik	6
6.2. Integrasjonstest:	7
6.2.1. Identifiserte avvik	7
6.2.2. Løsning på avvik	7
6.2.3. Gjenstående avvik	7
6.3. Systemtest:	8
6.3.1. Identifiserte avvik	8
6.3.2. Løsning på avvik	8
6.3.3. Gjenstående avvik	8
<b>7. Vurdering av produktet som er testet</b>	<b>8</b>
7.1 Sikkerhet/Passord	8
7.2 Funksjonalitet	9
7.3 Kundeinfo/Personvern	9
7.4 Kodekvalitet	9
<b>8. Sammendrag av aktiviteter og lærdom</b>	<b>10</b>
<b>9. Referanser</b>	<b>10</b>
<b>10. Vedlegg</b>	<b>12</b>
10.1. Testplan	12
10.2. Testdata	19

10.2.1. Database	19
10.2.2. Stubs	21
10.2.2.1 AdminDBStub	21
10.2.2.2 BankDBstub	25
10.2.3 Initiering av databasen (initDB.php)	29
10.3. Testlogg / feillogg	30
10.3.1. Enhetstest	30
Bank funksjoner	32
Admin funksjoner	47
10.3.2. Integrasjonstest	59
10.3.2.1 Oversikt i excel	59
Uten database-tilkobling	59
Bank funksjoner	60
Admin funksjoner	63
10.3.2.2 Skjermbilder fra SoapUI	66
Uten database manuell-test	66
Bank funksjoner	67
Admin funksjoner	93
10.3.3. Systemtest	121
10.3.3.1. Katalon Recorder	121
Bank funksjoner	122
Admin funksjoner	125
10.3.3.2. MTM-Rapport	138
Bank funksjoner	138
Admin funksjoner	146

# 1. Identifikasjon

Denne sluttrapporten omfatter enhetstest, integrasjonstest og systemtest av applikasjonen TestingBank. Testplan for de nevnte testene er vedlagt under punkt 10.1. Testdataene brukt under integrasjonstest og systemtest er vedlagt i form av bilder av databasen i phpMyAdmin under punkt 10.2.1. Under enhetstesten ble det derimot brukt stub-filer og disse er vedlagt under punkt 10.2.2. Her skal det også nevnes at det i forbindelse med registrering av kunder, kontoer og transaksjoner under integrasjonstesten, ble opprettet flere entiteter enn de som foreligger i databasen ved initiering. Disse er dokumentert i testloggen til integrasjonstesten, se punkt 10.3.2. Øvrige testlogger og feillogger er å finne under punkt 10.3.

## 2. Sammendrag

### 2.1. Enhetstest

Enhetstesting ble gjennomført i Coolbeans/Openbeans ved hjelp av test-rammeverket PHPUnit. Under denne testen var det BLL-laget av TestingBank som ble testet. Siden denne typen test ikke skal kjøres mot selve databasen, ble det lagt til to Stub-filer i DAL-laget som i stedet returnerer hardkodede data. Disse to ble gitt navnene; adminDatabaseStub, som serverer adminLogikk, og bankDatabaseStub.php som serverer bankLogikk (kunde delen av banken), med data. Det ble deretter laget egne test-filer for hver metode i stub'ene. Etter at alle testene var ferdigkrevet, viste det seg at 89.02% av dem var bestått. Feil i regex'ene til sjekkLoggInn-metoden ble så fikset for at prosentandelen skulle komme opp til 100%. Se punkt 6.1 for mer informasjon om denne feilen. Testene hadde en total code coverage på 14.97%, men av filene som har noe betydning i denne testen, hadde bankLogikk 88.46% og adminLogikk 95%. 5% av hver av disse er dependency injection, mens de resterende 6.54% av bankLogikk er endreKundeInfo-metoden. Siden denne metoden også ligger i adminLogikk, ble den kun testet der. Med andre ord kan vi si at alle metodene per d.d. tilfredsstiller ønsket oppførsel og enhetstesten er derfor godkjent.

### 2.2. Integrasjonstest

Integrasjonstesten ble gjennomført i testverktøyet SoapUI fra SmartBear. I denne testen var det API-laget av TestingBank som ble testet, altså samspillet mellom metodene og opp mot databasen. Testene ble systematisk planlagt i et google spreadsheet-dokument før de ble lagt inn i SoapUI. Dette dokumentet ligger vedlagt under punkt 10.3.2 i vedleggslisten. Totalt ble det gjennomført 64 test caser og 90% av disse ble godkjent. 6 av de 7 test-casene som feiler, gjør dette fordi API'et returnerer "ok" uavhengig av hva som skjer, og den siste feilen skyldes at det ikke returneres noe feilmelding når bruker prøver å logge inn når databasen er nede. Alle disse feilene faller inn under kategori B og er årsaken til at integrasjonstesten underkjennes. Vi oppdaget også en skrivefeil i adminLogikk, men denne ble rettet under testingen. Se punkt 6.2 for mer informasjon om disse feilene.

### 2.3. Systemtest

Systemtesten ble planlagt og spesifisert ved hjelp av Microsoft Test Manager (MTM) og spilt inn av Katalon Recorder utvidelsen for Chrome. Under denne testen var det systemet i sin helhet som ble testet, og det ble derfor laget en rekke brukerhistorier som testene ble basert på. Alle brukerhistoriene finnes i vedleggslisten under punkt 10.3.3. Gjennomføringen av systemtesten gir et resultat på at 100% av testene er gjennomført og godkjente. De feilene vi dog har plukket opp manuelt i MTM, er at regEx'ene ikke stopper brukeren i å legge data inn i databasen. Brukeren vil bli møtt med en rød validerings-feilmelding, men som sagt vil inputen fortsatt påvirke databasen. Utdypende forklaring av feilene og løsninger finnes under punkt 6.3.

# **3. Avvik i forhold til testplanen**

Siden selve testplanen ble skrevet etter testene ble gjennomført i dette prosjektet, er det naturligvis svært få avvik mellom den utførte test og testplanen. Det ble likevel planlagt før integrasjonstesten at denne testen skulle planlegges ferdig før testene skulle spesifiseres. Under gjennomføringen ble det lagt inn tester i SoapUI før alle testene var ferdig planlagte, noe som resulterte i at flere av testene måtte legges inn på nytt. Bortsett fra dette, ble alt utført etter testplanen.

# **4. Vurdering av testens grundighet**

## **4.1. Enhetstest**

En enhetstest, også kjent som komponenttesting er en essensiell del av testing, der vi tester én og én metode hver for seg, for feil/endringer som kan forekomme. Vi har skrevet en kode for å teste en annen kode, og utført en komplett enhetstest som ga oss utslag i code coverage for bankLogikk på 88,46% og adminLogikk på 95%. Denne totalen gir oss en indikasjon på grundigheten i henhold til graden av dekning av kodelinjer og betingelser. I henhold til kriteriene for grundighet i testplanen som er en code coverage på 95%, har vi utført enhetstesten svært grundig ettersom de manglende prosentene på bankLogikk skyldes at en metode ligger dobbelt og testes derfor kun på den ene siden. Videre har koden blitt nøye gjennomgått for å påse at stubbene gjenspeiler alle utfall, samt at grenseverdier og RegEx'er er nøye testet.

## **4.2. Integrasjonstest**

I motsetning til enhetstesten der én og én metode skulle testes, testet vi i integrasjonstesten samspillet mellom de kallende metodene. Vi utførte kall og test mot alle API'ene, med unntak av apiAdminHeader.php og apiBankHeader.php, der vi totalt endte opp med 64 test caser. Spesifikasjon av alle test casene er vedlagt under punkt 10.3.2, der dokumenterte vi feil løpende gjennom en sammendrags-tabell som ga utslag for antall bestårte og antall feil på de testede API-lagene. Vi testet også hva som ville forekomme når databasen feiler. Da det ikke var trivielt å stoppe databasen fra SoapUI (Krattebøl, 2020, s.16), unngikk vi en helautomatisert test, og kjørte integrasjonstesten en del med databasen oppe og en med databasen nede. Vi har også testet mot grenseverdiene i de metodene som henter objekter fra databasen, og kan dermed konkludere med at graden av grundighet i denne testen er svært høy i henhold til kriteriene i testplanen.

## **4.3. Systemtest**

Under systemtesten gjorde vi tester for å dekke hele systemet, alt fra brukergrensesnittet til databasen og tilbake. Her ble systemets funksjoner testet mot brukerhistoriene vi hadde

laget, for å sjekke hva systemet gjorde og hvilket resultater vi fikk ut. Dette testet vi ned til minste detalj, der vi testet mot alle input felt, også i forhold til regex'ene. For å kontrollsjekke at alle spesifikasjonene var utført testet vi hvert inputfelt i isolasjon ved Assert mot alle input feltene for seg. I henhold til kriteriene i testplanen der vi skulle teste hvorvidt krav i spesifikasjonen/brukerhistoriene var testet, er den i høyeste grad utført både manuelt og automatisert.

## 5. Gjenstående aktiviteter/oppgaver

Under integrasjonstesting ble det oppdaget at metoden hentKundeInfo i bankDatabase.php har funksjonalitet for å sjekke om personnummeret til kunden eksisterer i databasen, og hvis ikke returnerer den en feilmelding. Her er det personnummeret i session, altså personnummeret som det logges inn med, som blir brukt. Siden man ikke kan logge inn uten et personnummer som ligger i databasen, var det ikke mulig å fremprovosere denne feilmeldingen. Det er derfor ikke testet om denne funksjonaliteten fungerer. Det kan diskuteres om funksjonaliteten i det hele tatt er nødvendig, siden man ikke får logget inn uten at personnummeret eksisterer i databasen. Dersom den av sikkerhetsmessige årsaker skal være der bør den testes før produktet lanseres.

## 6. Avvik i produktet

### 6.1. Enhetstest:

#### 6.1.1. Identifiserte avvik

I enhetstesten fant vi kun regex feil som gjaldt passord og brukernavn i innloggings-metoden. Denne feilen faller under kategori C.

#### 6.1.2. Løsning på avvik

Endringer som ble gjort i koden angår validering av passord og brukernavn. Vi valgte å gjøre dette da eneste kravet til passordet var at det måtte inneholde minst to tegn. Vi endret valideringen til å matche et sterkt passord krav, noe som tilsier at den må inneholde en stor bokstav, minst et tall, minst et symbol og minst 6 tegn.

#### 6.1.3. Gjenstående avvik

Etter rettet feil på regex, gjenstår det ingen uløste avvik knyttet til enhetstesten.

## 6.2. Integrasjonstest:

### 6.2.1. Identifiserte avvik

Under integrasjonstesten var det 7 test-caser som feilet, noe som utgjør 10% av de totale test-caser. 6 av de 7 test-casene som feiler, feiler fordi API'et returnerer "ok" uavhengig av hva som skjer. Denne feilen faller inn under kategori B da den ikke stopper programmet, men har en alvorlig konsekvens for systemet ettersom feilene hverken blir fanget opp, gjort noe med eller kommunisert tilbake til brukeren. Den siste feilen skyldes at det ikke returneres noe feilmelding når bruker prøver å logge inn når databasen er nede.

Det ble også funnet en skrivefeil i registerKonto-funksjonen i adminDatabase.php som gjorde at API'et ikke fikk tak i denne funksjonen og dermed ikke fikk noen respons på sitt kall.

### 6.2.2. Løsning på avvik

Forslag på løsningen på avvikene som gikk på at API'et returnerer "ok" uavhengig av hva som skjer, er å endre denne hardkodede responsen til å returnere egnede feilmeldinger. Det ble ikke gjort noen endringer i koden for å fikse disse feilene, da det ville hatt videre konsekvenser i systemet.

Forslag på løsning på manglende respons på database hadde vært å lagt inn en test for om databasen er tilgjengelig. Dersom databasen ikke er tilgjengelig foreslår vi å returnere en feilmelding til brukeren, og evt. igangsette tiltak eller reservaløsninger.

Når det gjelder skrivefeilen valgte vi å rette opp i denne da det var en rask og enkel fiks, samt at det gjorde det mulig å videre teste metoden. se fig.6.2.2.1

The screenshot shows a code diff interface comparing two versions of the file TestingBank/DAL/adminDatabase.php. The left column shows line numbers 105, 106, 107, 108, 109, 110, and 111. The right column shows the corresponding line numbers from the previous version. The top line is annotated with @@ -105,7 +105,7 @@ class AdminDB. Lines 105, 106, and 107 show closing brace symbols. Line 108 is highlighted in red and contains a minus sign (-) followed by the original function name registerKonto(\$konto). Line 108 is also highlighted in green and contains a plus sign (+) followed by the corrected function name registerKonto(\$konto). Lines 109, 110, and 111 show the continuation of the function definition with SQL query assignment and result retrieval logic.

TestingBank/DAL/adminDatabase.php		@@ -105,7 +105,7 @@ class AdminDB
	105	105 } 106 } 107 }
	108	- function registerKonto(\$konto) + function registerKonto(\$konto)
	109	{ 110 \$sql = "Select * from Kunde Where Personnummer = '\$konto->personnummer'"; 111 \$resultat = \$this->db->query(\$sql);

fig.6.2.2.1 retting av skrivefeil

### 6.2.3. Gjenstående avvik

Etter rettet skrivefeil gjenstår fremdeles de 6 avvikene som gikk på at API'et returnerte "ok" uansett hva som skjer, samt mangelen på respons dersom databasen var nede.

## 6.3. Systemtest:

### 6.3.1. Identifiserte avvik

Under den automatiserte systemtesten i Katalon ble det ikke funnet noen avvik. Når systemtesten ble gått gjennom manuelt i MTM ble det likevel funnet 2 avvik av kategori B. Disse gjelder funksjonene for opprettelse av ny kunde og opprettelse av ny konto på admin-siden. Avvikene skyldes at man får gjennomført opprettelsen selv om man fyller inn ugyldig data og selv om regex'ene på klientsiden reagerer med røde valideringsmeldinger.

Her bør det da også nevnes at det mangler validering av mange av input feltene. Mer om dette under punkt 7.2, vurdering av produktets funksjonalitet.

### 6.3.2. Løsning på avvik

Løsningen på nevnte avvik er å deaktivere registrer-knappen inntil valideringen godkjenner alle feltene. På den måten vil det være umulig å registrere ugyldig data. Det ble ikke foretatt noen endringer i kode for å rette denne feilen.

### 6.3.3. Gjenstående avvik

Ettersom ingen av avvikene er løst gjenstår alle nevnte avvik.

## 7. Vurdering av produktet som er testet

Etter gjennomført testing har vi gjort en vurdering av produktet, og funnet ut at produktet ikke burde settes i drift før flere sentrale feil er rettet opp. Dette skyldes at integrasjonstesten og systemtesten er underkjent, fordi de begge har B-feil og dermed ikke kan godkjennes på bakgrunn av godkjenningskriteriene i testplanen. Programmet kommer til kort under spesielt disse punktene; sikkerhet, funksjonalitet, personvern og kodekvalitet. Under kan du lese mer om hvert punkt.

### 7.1 Sikkerhet/Passord

Sikkerheten er ikke god nok med tanke på passordkryptering. Det er ikke brukt noen form for hashing eller salt på passordene, og det ligger ikke kryptert i databasen. Hashing og salting av passord er en måte å øke sikkerheten for brukerne sine (Arias, 2018). For en bankapplikasjon som denne, der sikkerheten er alfa og omega, er det muligens også nødvendig med sterkere kryptering for å holde systemet fritt for sikkerhetsbrudd. Før testingen ble satt i gang var det heller ikke noen RegEx'en som sørget for at brukerne lager sterke passord. Dette ble rettet til følgende regex:

`"/^(?=.*[a-z])(?=.*[A-Z])(?=.*.*\d)(?=.*.*[@$!%*?&])[A-Za-z\d@$!%*?&]{6,20}$/"`

## 7.2 Funksjonalitet

Når det kommer til funksjonaliteten, er det både mangler og ting som ikke gjør som man forventer at det skal. Gjennom testingen ble det blant annet funnet en rekke manglende RegEx'er på klientsiden. Viktigheten av RegEx er i stor grad at det ikke skal være mulig å legge inn informasjon som ikke er gyldig. Det var eksempelvis mangel på RegEx på telefonnummer i adminEndreKunde-funksjonen, noe som gjør at man kan skrive inn et telefonnummer med for få eller for mange tall. Det finnes også eksempler på funksjoner der det ligger RegEx'er, men hvor man får gjennomført oppgaven sin selv om RegEx'en melder om feil i input. Programmet har også gjennomgående mangler når det gjelder tilbakemeldinger til brukeren når programmet ikke kan utføre forventede handlinger. Valideringsmeldinger vises på de input-feltene som har RegEx, men det skulle gjerne vært mer informative meldinger som forklarer problemet og eventuelle løsninger brukeren kan foreta seg.

## 7.3 Kundeinfo/Personvern

Dette produktet utgjør en stor risiko for forbrukerne, da det er en rekke krav som bør oppfylles før det kan settes i bruk. Blant dem mange er at produktet er nødt til å følge eksisterende lovverk og grunnleggende person- og forbrukerverns lov. Per nå så er det ingen krav til godkjenning av vilkår fra brukeren side. Bruker har også mulighet til å utføre funksjonalitet som admin ville gjort i lignende programmer, slik som endring av saldo uten krav om godkjenning fra regEx eller banksystemet, noe som hadde vært ypperlig for brukeren men ikke ideelt og noe kostbart for banken. For at programmet skal kunne settes i drift må de nevnte punktene være oppfylt, og en rekke valideringer må implementeres for å få rettigheter til utføre en endring. Ved å ha innfridd de nevnte punktene er vi overbevist om at det også vil føre til trygghet for brukeren, noe som er elementært for at de skal agere med programmet.

## 7.4 Kodekvalitet

Kodekvaliteten er så som så. Til tider virker den god og strukturert, men det helhetlig inntrykket er at den er noe rotete og sluvete. Dette sier vi basert på følgende feil funnet i kode: RegistrerKunde som ligger i både adminDatabase og bankDatabase når den kun blir brukt på adminsiden. Denne typen feil er mer overflødig kode (ubrukt kode) enn det er en feil. En annen feil vi kom over var skriveauen i AdminRegistrerKonto-funksjonen, som er nevnt under punkt 6.2.

## 8. Sammendrag av aktiviteter og lærdom

Testingens forløp har tatt oss gjennom et forminsket men reelt innblikk i en testprosess av et ”ferdig” utviklet program. Vi har gått gjennom planlegging-, gjennomføring- og evalueringfasen slik at vi forstår rammene som ligger til grunn når man tester programvare. Vi har støtt på flere problemer som har gitt oss gode pekepinner på hvordan man egentlig burde gjennomført et testprosjekt, og dermed lært mye og gitt oss erfaring. Videre kommer noen konkrete temaer og teorier som vi har tilegnet oss kunnskap om.

Når det kommer til det tekniske, har vi lært å anvende testverktøyene phpUnit, SoapUI og katalon, samt planleggingsverktøyene Microsoft Test Manager og Google Spreadsheets. PhpUnit og Katalon tok vi lett, og selv om det fantes lite hjelp på nett løste vi ganske raskt de problemene vi støtte på. SoapUI var mer tidkrevende og programmet virket en smule utdatert i den forstand at det krasjet om man åpnet for mange vinduer og kunne ikke brukes med versjonshåndtering. MTM følte noe tungvint i starten og det var varierende hvor stabil TFS oppførte seg. Google Spreadsheets er et verktøy vi hadde kjennskap til fra før av, men ble overrasket over hvor lett og oversiktlig det var å planlegge tester i det.

Planlegging og loggføring har gjennomgående vært vektlagt under prosjektet og erfaringen kommer godt med i videre arbeid med testing. Det som kunne vært gjort annerledes er at testplanen burde vært laget før testene ble gjennomført. Hadde det vært gjort kunne vi f.eks. benyttet oss av godkjenningskriteriene på en skikkelig måte gjennom hele testfasen, og da loggført resultatene fortløpende i henhold til disse.

Vi har lært at det er viktig med gode planverktøy slik at selve testingen byr på minst mulig stopp og går smertefritt. Det å ha strukturerte testdata er også viktig for å forenkle spesifiseringen av testene, og på dette punktet ser vi i ettertid at vi burde ha skrota testdataen i databasen og opprettet helt nye strukturerte data. Vi har også lært at 100% code coverage er ikke nødvendigvis bra, og hvis man oppnår et slikt resultat burde man dobbeltsjekke om man har skrevet testene riktig. Og at det er viktig å sjekke testene grundig for menneskelige feil, da det også forekommer i testfasen.

Arbeidet har også lært oss mye om hvordan det er å jobbe i et større team, noe som kommer godt med videre i yrkeslivet.

## 9. Referanser

- Arias, D. (2018, Mai 3). *Adding Salt to Hashing: A Better Way to Store Passwords*. Retrieved from Auth0:  
<https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/>
- Krattebøl, T. (03. april 2020) *Testing av programvare: Integrasjonstest1* (Forelesning s.16) Norge: Oslo Metropolitan Universitet

- Justis- og beredskapsdepartementet. (2015). Lov om behandling av personopplysninger (personopplysningsloven). Hentet April 12, 2020 fra Lovdata: <http://lovdata.no/dokument/NL/lov/2000-04-14-31>
- OWASP. (u.d.). Password length and complexity. Hentet April 14, 2020 fra: [https://www.owasp.org/index.php/Password\\_length\\_%26\\_complexity](https://www.owasp.org/index.php/Password_length_%26_complexity)

# **10. Vedlegg**

## **10.1. Testplan**

### **Testplan for testing av TestingBank**

Versjon: 1.0

Dato laget: 25.03.2020

Av forfatter: Jostein Hauge, Jørgen Huseby, Felix Emhjellen, Elin H. Hegsvold, Zakariya Ibrahim

Hva testes her: Enhetstest, integrasjonstest og systemtest

Sist oppdatert: 16.04.2020

Av forfatter: Elin H. Hegsvold

**Innhold:**

<b>1. Innledning</b>	<b>3</b>
<b>2. Testobjekter</b>	<b>3</b>
2.1. Enhetstest	3
2.2. Integrasjonstest	3
2.3. Systemtest	3
2.4. Prioritering	3
<b>3. Fremgangsmåte</b>	<b>4</b>
3.1. Overordnet teststrategi for	4
3.1.1. Enhetstest	4
3.1.2. Integrasjonstest	4
3.1.3. Systemtest	4
3.2. Prosedyre for planlegging og gjennomføring av test	4
3.2.1. Enhetstest	4
3.2.2. Integrasjonstest	5
3.2.3. Systemtest	5
3.3. Kriterier for grundighet i testen	5
3.3.1 Enhetstest	5
3.3.2. Integrasjonstest	6
3.3.3. Systemtest	6
<b>4. Godkjenningskriterier</b>	<b>6</b>
4.1. Enhetstest	6
4.2. Integrasjonstest	6
4.3. Systemtest	6
<b>5. Testmiljø</b>	<b>6</b>
5.1. Enhetstest:	6
5.2. Integrasjonstest:	7
5.3. Systemtest:	7
<b>6. Resultatdokumenter</b>	<b>7</b>

## 1. Innledning

Målet for testene som skal gjennomføres er å verifisere at eksisterende funksjonalitet i applikasjonen fungerer. Applikasjonen som skal testes er en PHP-bygd nettbank. Løsningen er lagdelt, med API, BLL og DAL lag, og er knyttet til en SQLite database som pr. dd. kjører på localhost.

Som kunde har man en egen profil hvor man kan se informasjonen som er lagret om en selv og redigere på dette. Kunden kan få oversikt over kontoene sine med opplysninger som saldo og lignende, samt utføre transaksjoner og betalinger. En administrator av nettbanken, typisk en bank-ansatt, kan få oversikt over kunder, kontoer og all informasjonen knyttet til disse. Administratoren kan også opprette nye kunder og kontoer, samt endre informasjonen om disse.

## 2. Testobjekter

### 2.1. Enhetstest

I enhetstesten skal alle metodene i BLL-filene testes, det vil si adminLogikk.php og bankLogikk.php. Alle metodene som ligger i nevnte filer, kaller på metoder i DAL som igjen bruker databasen for å hente ut informasjon. Under enhetstesten skal det brukes stub'er som erstatning for databasene, slik at testen ikke er avhengig av kontakt med databasen for å teste enhetene i programmet.

Her er det verdt å nevne at registrerKunde-metoden ligger i både bankDatabase.php og adminDatabase.php på DAL. Da denne metoden kun er i bruk i admin-delen, og bankLogikk.php ikke har noen funksjon som kaller den i bankDatabase.php, skal den kun testes på admin-siden.

### 2.2. Integrasjonstest

I integrasjonstesten skal alle filene under API-laget testes, med unntak av apiAdminHeader.php og apiBankHeader.php. Filene består av metoder som kaller på riktig metode i BLL, som igjen kaller på den riktige DAL-metoden som deretter henter ut, legger inn, endrer eller sletter informasjon i databasen. Stub'ene blir ikke benyttet i denne testen.

### 2.3. Systemtest

I systemtesten skal alle filene i hele systemet testes.

### 2.4. Prioritering

Det er ingen prioritering i testen. Alt skal testes og det er ingen enkeltdeler som er viktigere enn andre.

### 3. Fremgangsmåte

#### 3.1. Overordnet teststrategi for

##### 3.1.1. Enhetstest

Her er det viktig å opprette enhetstester tidlig da testene automatiseres og både kan og burde kjøres ofte gjennom utvikling av programmet/systemet for å finne konsekvensene av endret kode. PHPUnit skal brukes for å automatisere enhetstestene. Siden testene skal kjøres ofte er det viktig at det tar kort tid. For at det ikke skal brukes tid og ressurser på å hente data fra databasen, skal det derfor lages stub-filer med hardkodet data.

##### 3.1.2. Integrasjonstest

Siden det ved en integrasjonstest er samspillet metodene seg i mellom som skal testes, er det viktig å sette seg godt inn i de ulike metodene og planlegge nøye før testene spesifisieres. Metodene skal også testes opp mot databasen, og det er derfor viktig at databasen har tilstrekkelig test-data tilgjengelig og at man enkelt kan resette databasen til initiell tilstand. For å forenkle denne prosessen skal det lages et php-program (initDB.php) som gjennom et SQL-script sørger for at databasen oppdateres. For effektiviseringens skyld, samt best mulig dekning, bør integrasjonstesten automatiseres. Det skal også kjøres en test mot databasen når den ikke er koblet opp, og denne må gjøres manuelt.

##### 3.1.3. Systemtest

Under en systemtest er det systemet i sin helhet som skal testes, at klientsiden og serversiden fungerer sammen. I likhet med integrasjonstesten er det derfor også her viktig å planlegge godt før testene spesifisieres. Målet med testen er å sikre at brukerens behov er dekket, og det er derfor vanlig å basere systemtesten på en kravspesifikasjon. Siden dette ikke foreligger ved dette prosjektet, skal det i stedet utarbeides brukerhistorier basert på funksjonaliteten i programmet. Også her er det data i databasen som det skal testes mot, og samme programmet som i integrasjonstesten skal også brukes her. Systemtesten skal gjennomføres manuelt gjennom MTM, i tillegg til at den skal automatiseres ved hjelp av Katalon Recorder.

#### 3.2. Prosedyre for planlegging og gjennomføring av test

##### 3.2.1. Enhetstest

For at enhetstestene skal kunne kjøres må PHPUnit være installert i CoolBeans/OpenBeans. For at man skal kunne se hvor mye av koden som er enhetstestet (code coverage) må Xdebug være installert. Videre må det opprettes stub-filer med data som kan imitere databasen før testfilene kan spesifisieres.

Det skal lages egne testfiler for hver enkelt av metodene i BLL-laget som skal testes. Testene skal så kjøres og evt. endres på til de dekker så mye som mulig av koden. Siden testene er uavhengige av hverandre kan de både kjøres samlet og hver for seg. Her vil det være naturlig å kjøre testene enkeltvis når de utvikles og samlet med jevne mellomrom for oversiktens skyld.

Enhetstesten regnes som kjørt når den oppfyller godkjenningskriteriene, se punkt 4.

### **3.2.2. Integrasjonstest**

For at integrasjonstestene skal kunne kjøres må SoapUI være installert og databasen initiert. For at SoapUI skal få tilgang til applikasjonen og databasen under delen av testen med databasen opp, må også serveren startes i XAMPP først. For testen av databasen ikke koblet til, må XAMPP MySQL Database være koblet fra.

Siden det her er viktig å planlegge testingen nøye i forkant, skal det lages et excel-dokument som holder rede på alle API'ene, rekkefølgen de skal testes i og hvilke test-caser som skal gjennomføres på de enkelte API'ene. Om hver test-case skal det planlegges forventet resultat og hva slags test-data som skal benyttes. Når alle API'ene er nøye gjennomgått skal testene spesifiseres i SoapUI. Test-casene skal sorteres i to test-suiter, en for kunde-siden (bank) og en for admin-siden. Hver test-case skal representere ett API, og for hver test-case skal det legges inn ett test-step for hver test mot API'et. Enkelte tester kan i utgangspunktet kjøres enkeltvis, men siden mange av testene er avhengige av andre tester bør hele test-suiten kjøres for hver gang. Dette gjelder da ikke test av database ikke tilgjengelig, som må gjøres for seg selv.

Integrasjonstesten regnes som kjørt når alle test-casene er spesifisert i SoapUI og den oppfyller godkjenningskriteriene, se punkt 4.

### **3.2.3. Systemtest**

For at systemtesten skal kunne kjøres må i første omgang MTM være oppå gå på en Team Foundation Server. Videre må Katalon Recorder installeres i nettleseren, databasen initieres og web-serveren startes i XAMPP slik at Katalon får tilgang til applikasjonen og databasen.

Siden systemtesten spesifiseres ut i fra brukerhistorier, skal disse utarbeides så tidlig som mulig, før de skal planlegges og dokumenteres i MTM. Her skal de sorteres i to features, en for kunde-siden og en for admin-siden. Det skal så planlegges tester ut i fra brukerhistoriene, der hver brukerhistorie er representert som en test-suite. Hver suite skal inneholde en test-case med ulike steg som skal gjennomføres under testingen. Disse stegene skal inneholde en beskrivelse av hva som skal gjøres, forventet resultat, samt et skjermbilde av hvordan resultatet ser ut i applikasjonen. Testene skal så kjøres manuelt og markeres som «passed» eller «failed» avhengig av resultatet. Videre skal testene spilles inn med Katalon Recorder og assertions skal legges inn for å sjekke at applikasjonen responderer med forventet resultat. Til slutt kjører Katalon gjennom testene automatisk og kjøringene som failer markeres med rødt.

Systemtesten regnes som kjørt når alle brukerhistorier er dekket i både MTM og Katalon, og testen oppfyller godkjenningskriteriene, se punkt 4.

## **3.3. Kriterier for grundighet i testen**

### **3.3.1 Enhetstest**

Enhetstesten er grundig gjennomført når code coverage på filene i BLL er 95%. Testen dekker da all kode bortsett fra konstruktøren som kun har dependency injection, en funksjon for å bestemme om det er stub'en eller databasen som skal kalles. På stub-filene skal code coverage være 100%, men det må også sjekkes grundig at stubbene gjenspeiler alle de forskjellige utfallene av metodene i database-filene. Videre må det på metoder som har som funksjon å hente data, testes for grenseverdier. Her skal det minst testes for henting av

ingen, en og flere (minst 2, helst 3) objekter. Også regex'en til sjekkLoggInn-funksjonen må testes for alle de ulike "ulovlige" variantene av personnummer og passord.

### 3.3.2. Integrasjonstest

Integrasjonstesten skal omfatte alle API-filene, med unntak av apiAdminHeader.php og apiBankHeader.php. For at disse skal kvalifiseres som grundig testet må alle mulige utfall av metodene testes. Dette innebærer også her testing av grenseverdier i de metodene som henter objekter fra databasen.

### 3.3.3. Systemtest

For systemtesten skal alle systemets funksjoner (i form av brukerhistorier) testes. Her skal også regex'ene på klientsiden testes. Testene skal kjøres både manuelt og automatisert.

## 4. Godkjenningskriterier

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

Kategori A: Feil som forårsaker stopp. For eksempel maskin/system/funksjon må startes på nytt.

Kategori B: Feil som har alvorlige konsekvenser for systemet eller tilstøtende systemer.

Kategori C: Feil med mindre alvorlige konsekvenser.

Kategori D: Feil uten alvorlige konsekvenser.

### 4.1. Enhetstest

Enhetstesten må være 100% for å godkjennes, men når det gjelder code coverage skal testen godkjennes dersom minst 95% av koden er dekket.

### 4.2. Integrasjonstest

Integrasjonstesten skal godkjennes når man har 95% godkjente testcaser og ingen av de underkjente casene har kategori A eller B feil.

### 4.3. Systemtest

Systemtesten skal, som integrasjonstesten, godkjennes ved 95% der ingen underkjente caser har av kategori A eller B feil.

## 5. Testmiljø

Gjennomgående skal applikasjonen kjøres lokalt på testmaskina ved hjelp av XAMPP og databasen skal administreres i phpMyAdmin. Videre skal det brukes ulike verktøy i de ulike testene, se underpunkter.

### 5.1. Enhetstest:

Enhetstestene skal skrives og kjøres i CoolBeans/OpenBeans ved hjelp av test-rammeverket PHPUnit.

### **5.2. Integrasjonstest:**

Integrasjonstestene skal planlegges i Microsoft Excel og gjennomføres i test-verktøyet SoapUI.

### **5.3. Systemtest:**

Systemtesten skal planlegges i Microsoft Test Manager (MTM) via Team Foundation Server. I MTM skal også testene gjøres manuelt. For å få automatisert testene skal Chrome utvidelsen Katalon Recorder brukes.

## **6. Resultatdokumenter**

For oversikt over testdata brukt under testingen, se vedlegg 10.2 i sluttrapporten. Her ligger bilder av databasen, stubbene og program for å initiere databasen. Testlogger og feillogger ligger vedlagt under 10.3 i samme sluttrapport.

## 10.2. Testdata

### 10.2.1. Database

Oversikt:

Table	Action	Rows	Type	Collation	Size	Overhead
Konto		7	InnoDB	latin1_swedish_ci	16.0 KiB	-
Kunde		4	InnoDB	latin1_swedish_ci	16.0 KiB	-
Poststed		4	InnoDB	latin1_swedish_ci	16.0 KiB	-
Transaksjon		13	InnoDB	latin1_swedish_ci	16.0 KiB	-
4 tables	Sum	28	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Konto:

		Kontonummer	Personnummer	Saldo	Type	Valuta
<input type="checkbox"/>		105010123456	01010110523	720	Lønnskonto	NOK
<input type="checkbox"/>		105020123456	01010110523	100500	Sparekonto	NOK
<input type="checkbox"/>		222223344444	11029700112	9000080	Brukskonto	USD
<input type="checkbox"/>		22334412345	01010110523	10234.5	Brukskonto	NOK
<input type="checkbox"/>		33333662222	11029700112	45	Sparekonto	USD
<input type="checkbox"/>		44444992222	11029700112	200	Lønnskonto	USD
<input type="checkbox"/>		55555224444	12345123456	1	Brukskonto	SEK

Kunde:

		Personnummer	Fornavn	Etternavn	Adresse	Postnr	Telponnr	Passord
<input type="checkbox"/>		01010110523	Lene	Hansen	Osloveien 22	0565	22224444	Passord1!
<input type="checkbox"/>		11029700112	Zak	Ibrahim	Nesveien	1440	12020509	Passord1!
<input type="checkbox"/>		12345123456	Johan	Olsson	Svenskevegen	9999	22223535	Passord1!
<input type="checkbox"/>		12345678901	Per	Hansen	Osloveien 82	1234	12345678	Passord1!

Poststed:

	<input type="checkbox"/>				Postnr	Poststed
	<input type="checkbox"/>				0565	Oslo
	<input type="checkbox"/>				1234	Oslo
	<input type="checkbox"/>				1440	Nesoya
	<input type="checkbox"/>				3270	Asker

Transaksjoner:

	<input type="checkbox"/>				TxID	FraTilKontonummer	Belop	Dato	Melding	Kontonummer	Avventer
	<input type="checkbox"/>				1	20102012345	-100.5	2015-03-15	Meny Storo	105010123456	0
	<input type="checkbox"/>				2	20102012345	400.4	2015-03-20	Innebtaling	105010123456	0
	<input type="checkbox"/>				3	20102012345	-1400.7	2015-03-13	Husleie	55551166677	1
	<input type="checkbox"/>				4	20102012347	-5000.5	2015-03-30	Skatt	105010123456	0
	<input type="checkbox"/>				5	20102012345	345.56	2015-03-13	Test	55551166677	0
	<input type="checkbox"/>				6	345678908	3000	2012-12-12		105010123456	0
	<input type="checkbox"/>				7	234534678	15	2012-12-12	Hei	105010123456	0
	<input type="checkbox"/>				8	1234254365	125	2012-12-12	Hopp	105010123456	0
	<input type="checkbox"/>				9	09090909090	2	2020-02-14	Tapas	55555224444	0
	<input type="checkbox"/>				10	55555224444	9000000	2019-12-24	Takk for i går ;)	22222334444	1
	<input type="checkbox"/>				11	55555224444	-80	2019-12-25	For dårlig samvittighet	22222334444	0
	<input type="checkbox"/>				12	105010123456	-123	2020-01-01	En nyydelig dag	22222334444	1
	<input type="checkbox"/>				15	10101334532	100	2020-06-28	Takk for kinoturen	105010123456	1

## 10.2.2. Stubs

### 10.2.2.1 AdminDBStub

```
include_once '../Model/domeneModell.php';

class AdminDBStub
{
    function hentAlleKunder()
    {
        $alleKunder = array();

        $kunde1 = new kunde();
        $kunde1->personnummer ="01010122344";
        $kunde1->fornavn = "Per";
        $kunde1->etternavn = "Olsen";
        $kunde1->adresse = "Osloveien 82";
        $kunde1->postnr = "0270";
        $kunde1->poststed = "Oslo";
        $kunde1->telefonnr="12345678";
        $kunde1->passord = "HeiHei";

        $alleKunder []=$kunde1;

        $kunde2 = new kunde();
        $kunde2->personnummer ="01010122344";
        $kunde2->fornavn = "Line";
        $kunde2->etternavn = "Jensen";
        $kunde2->adresse = "Askerveien 100";
        $kunde2->postnr = "1379";
        $kunde2->poststed = "Asker";
        $kunde2->telefonnr="92876789";
        $kunde2->passord = "HeiHei";

        $alleKunder []=$kunde2;

        $kunde3 = new kunde();
        $kunde3->personnummer ="02020233455";
        $kunde3->fornavn = "Ole";
        $kunde3->etternavn = "Olsen";
        $kunde3->adresse = "Bærumsviken 23";
        $kunde3->postnr = "1234";
        $kunde3->poststed = "Bærum";
        $kunde3->telefonnr="99889988";
        $kunde3->passord = "HeiHei";

        $alleKunder []=$kunde3;

        return $alleKunder;
    }
}
```

```

function endreKundeInfo ($kunde)
{
    $postnrTest = "0270";

    if ($kunde->postnr != $postnrTest)
    {
        if ($kunde->postnr < 1)
        {
            return "Feil";
        }
    }
    return "OK";
}

function registrerKunde ($kunde)
{
    $postnrTest = "0270";

    if ($kunde->postnr != $postnrTest)
    {
        if ($kunde->postnr < 1)
        {
            return "Feil";
        }
    }

    $kundeTest = new kunde ();
    $kundeTest->personnummer = $kunde->personnummer;

    if ($kundeTest->personnummer != "0")
    {
        return "OK";
    }
    else
    {
        return "Feil";
    }
}

function slettKunde ($personnummer)
{
    $personnummerSlett = "12345678910";

    if ($personnummer == $personnummerSlett)
    {
        return "OK";
    }
    else
    {
        return "Feil";
    }
}

```

```

function registrerKonto ($konto)
{
    if ($konto->personnummer == "12345678910")
    {
        return "OK";
    }
    else
    {
        return "Feil";
    }
}

function endreKonto ($konto)
{
    if ($konto->personnummer == null)
    {
        return "Feil";
    }

    if ($konto->kontonummer == null)
    {
        return "Feil";
    }
    return "OK";
}

function hentAlleKonti ()
{
    $alleKontoer = array();
    $konto1 = new konto ();
    $konto1->personnummer = "12345678910";
    $konto1->kontonummer = "12345";
    $konto1->saldo = "4560";
    $konto1->type = "type";
    $konto1->valuta = "Euro";
    $konto1->transaksjoner = array();
    $alleKontoer[] = $konto1;
    $konto2 = new konto ();
    $konto2->personnummer = "12345678911";
    $konto2->kontonummer = "12346";
    $konto2->saldo = "5560";
    $konto2->type = "type";
    $konto2->valuta = "NOK";
    $konto2->transaksjoner = array();
    $alleKontoer[] = $konto2;
    $konto3 = new konto ();
    $konto3->personnummer = "12345678912";
    $konto3->kontonummer = "12347";
    $konto3->saldo = "6560";
    $konto3->type = "type";
    $konto3->valuta = "SEK";
    $konto3->transaksjoner = array();
    $alleKontoer[] = $konto3;

    return $alleKontoer;
}

```

```
function slettKonto ($konto)
{
    $kontonummerSlett = "12345";

    if ($konto->kontonummer == $kontonummerSlett)
    {
        return "OK";
    }
    else
    {
        return "Feil";
    }
}
```

### 10.2.2.2 BankDBstub

```
class BankDBStub
{
    function hentTransaksjoner($kontoNr,$fraDato,$tilDato)
    {
        date_default_timezone_set("Europe/Oslo");
        $fraDato = strtotime($fraDato);
        $tilDato = strtotime($tilDato);

        if($fraDato>$tilDato)
        {
            return "Fra dato må være større enn tildato";
        }

        $konto = new konto();
        $konto->personnummer="010101234567";
        $konto->kontonummer=$kontoNr;
        $konto->type="Sparekonto";
        $konto->saldo =2300.34;
        $konto->valuta="NOK";

        if($tilDato < strtotime('2015-03-26'))
        {
            return $konto;
        }

        $dato = $fraDato;

        while ($dato<=$tilDato)
        {
            switch($dato)
            {
                case strtotime('2015-03-26') :
                    $transaksjon1 = new transaksjon();
                    $transaksjon1->dato='2015-03-26';
                    $transaksjon1->transaksjonBelop=134.4;
                    $transaksjon1->fraTilKontonummer="22342344556";
                    $transaksjon1->melding="Meny Holtet";
                    $konto->transaksjoner[]=$transaksjon1;
                    break;
                case strtotime('2015-03-27') :
                    $transaksjon2 = new transaksjon();
                    $transaksjon2->dato='2015-03-27';
                    $transaksjon2->transaksjonBelop=-2056.45;
                    $transaksjon2->fraTilKontonummer="114342344556";
                    $transaksjon2->melding="Husleie";
                    $konto->transaksjoner[]=$transaksjon2;
                    break;
                case strtotime('2015-03-29') :
                    $transaksjon3 = new transaksjon();
                    $transaksjon3->dato = '2015-03-29';
                    $transaksjon3->transaksjonBelop=1454.45;
                    $transaksjon3->fraTilKontonummer="114342344511";
                    $transaksjon3->melding="Lekeland";
                    $konto->transaksjoner[]=$transaksjon3;
                    break;
            }
            $dato +=(60*60*24); // en dag i tillegg i sekunder
        }
        return $konto;
    }
}
```

```

function sjekkLoggInn($personnummer, $passord){

    if($passord == "Passord1!" && $personnummer == "12345678911"){
        return "OK";
    }
    else{
        return "Feil";
    }
}

function hentKonti($personnummer){
    $konti = [];

    if($personnummer == "000000000000"){
    }
    elseif ($personnummer == "111111111111"){
        $konti[] = "12121212121";
    }
    elseif ($personnummer == "333333333333") {
        $konti[] = "13131313131";
        $konti[] = "14141414141";
        $konti[] = "15151515151";
    }
    else {
        return "Personnummer finnes ikke";
    }
    return $konti;
}

function hentSaldi($personnummer){
    $saldo = [];

    if($personnummer == "000000000000"){
    }
    elseif ($personnummer == "111111111111"){
        $saldo[] = 10;
    }
    elseif ($personnummer == "333333333333") {
        $saldo[] = 100;
        $saldo[] = 200;
        $saldo[] = 300;
    }
    else {
        return "Personnummer finnes ikke";
    }
    return $saldo;
}

function registrerBetaling($kontoNr, $transaksjon){
    if($transaksjon ==! null && $kontoNr ==! null){
        return "OK";
    }
    else{
        return "Feil";
    }
}

```

```

function hentBetalinger($personnummer){
    if($personnummer == "1" || "2" || "4"){
        $transaksjon1 = new transaksjon();
        $transaksjon1->fraTilKontonummer = "1111111111";
        $transaksjon1->belop = 200;
        $transaksjon1->dato = "2015-03-14";
        $transaksjon1->melding = "dingDing";
        $transaksjon1->avventer = "1";

        $transaksjon2 = new transaksjon();
        $transaksjon2->fraTilKontonummer = "1111112222";
        $transaksjon2->belop = 212;
        $transaksjon2->dato = "2020-05-14";
        $transaksjon2->melding = "hei";
        $transaksjon2->avventer = "1";

        $transaksjon3 = new transaksjon();
        $transaksjon3->fraTilKontonummer = "1122222222";
        $transaksjon3->belop = 122;
        $transaksjon3->dato = "2021-12-14";
        $transaksjon3->melding = "Test";
        $transaksjon3->avventer = "1";

        $konto1 = new konto();
        $konto1->kontonummer = "123";
        $konto1->saldo = "100";
        $konto1->type = "type";
        $konto1->valuta = "NOK";
        $konto1->personnummer = $personnummer;

        $konto1->transaksjoner[] = $transaksjon1;
        $konto1->transaksjoner[] = $transaksjon2;
        $konto1->transaksjoner[] = $transaksjon3;

        $konto2 = new konto();
        $konto2->kontonummer = "123";
        $konto2->saldo = "100";
        $konto2->type = "type";
        $konto2->valuta = "NOK";
        $konto2->personnummer = $personnummer;
        $konto2->transaksjoner = array();

        $konto3 = new konto();
        $konto3->kontonummer = "123";
        $konto3->saldo = "100";
        $konto3->type = "type";
        $konto3->valuta = "NOK";
        $konto3->personnummer = $personnummer;
        $konto3->transaksjoner[] = $transaksjon1;

        if($personnummer == "1"){
            //persNr 1 har alt
            return $konto1->transaksjoner;
        }
    }
}

```

```

        if ($personnummer == "2"){
            //persNr 2 har konto, men ingen transaksjoner
            return $konto2->transaksjoner;
        }

        if($personnummer == "4"){
            //persNr 4 har konto og en transaksjon
            return $konto3->transaksjoner;
        }
    }
    if($personnummer == "3"){
        return "Ingen konto";
    }
    return null;
}

function utforBetaling($TxID){
    if($TxID == "1"){
        return "OK";
    }
    else{
        return "Feil";
    }
}

function hentKundeInfo($personnummer)
{
    $kunde = new kunde();

    $kunde->personnummer = "11111222222";
    $kunde->fornavn = "Roger";
    $kunde->etternavn = "Ruud";
    $kunde->adresse = "Gata 1, Brasil";
    $kunde->telefonnr = "87654321";
    $kunde->passord = "Passord123";
    $kunde->postnr = "2040";

    if($personnummer == "11111222222"){
        return $kunde;
    }
    else {
        return "Feil";
    }
}

```

### 10.2.3 Initiering av databasen (initDB.php)

```
<?php
header('Content-Type: application/json');
$OK = true;
$db = new mysqli("localhost", "root","", "Bank");
if($db->connect_error)
{
    $OK=false;
}
$fil= file_get_contents('Bank.sql');

$res = $db->multi_query($fil);
if($res)
{
    do {
        if ($result = $db->store_result()) {
            if($result == false)
            {
                $OK = false;
            }
            $result->free();
        }
        $db->more_results();
    } while ($db->next_result());
}

else
{
    $OK=false;
}
$db->close();
if($OK)
{
    echo json_encode("OK");
}
else
{
    echo json_encode("Feil");
}
```

## 10.3. Testlogg / feillogg

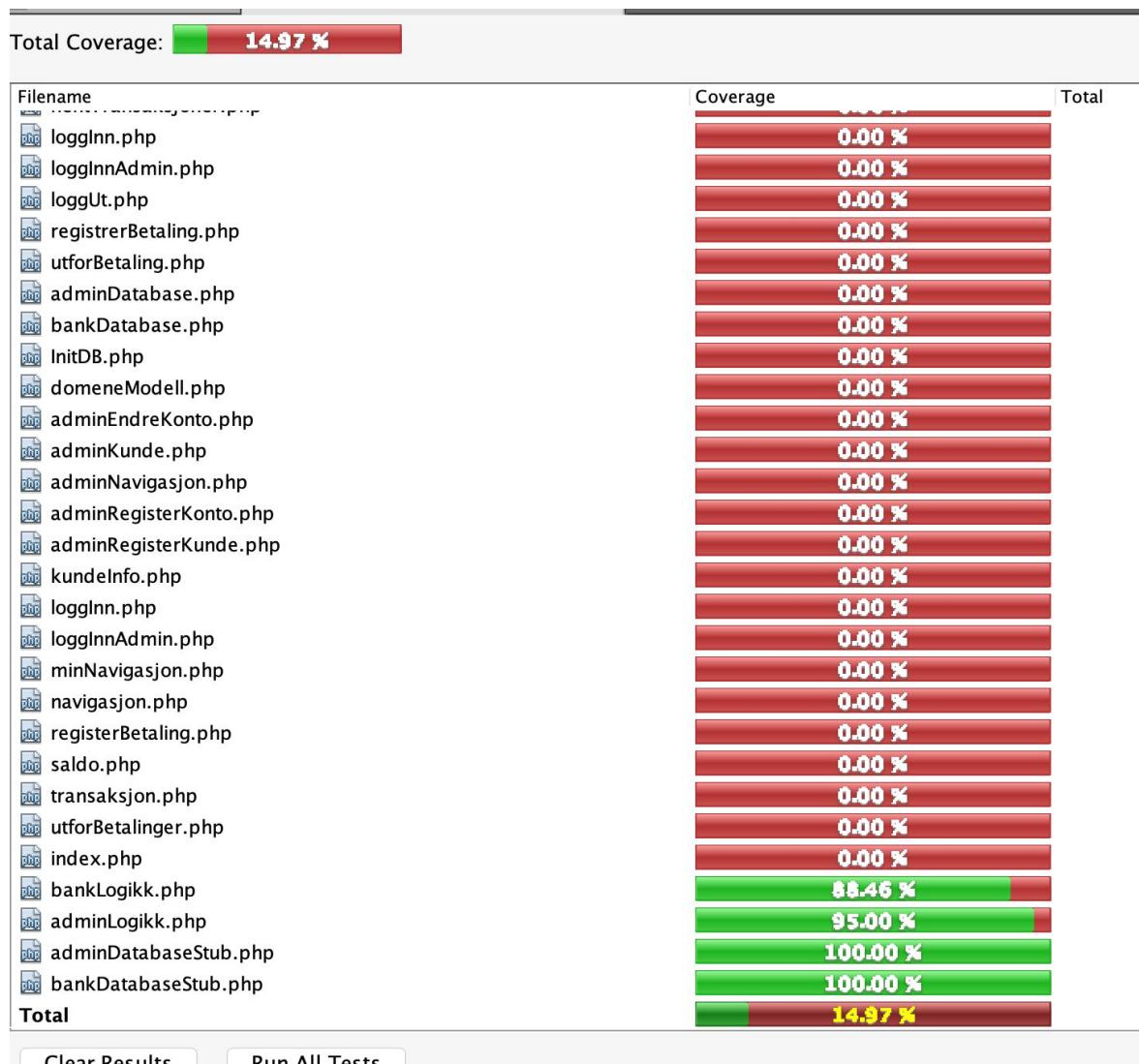
### 10.3.1. Enhetstest

Oversikt over alle enhetstester:

The screenshot shows a test runner interface with a sidebar containing icons for file operations (New, Open, Save, etc.) and navigation (Up, Down, Home). The main area displays a green header bar with the text "Tests passed: 100.00 %". Below this, a message states "All 55 tests passed. (0.124 s)". A list of 18 test cases is shown, each preceded by a green checkmark icon and the text "[PHPUnit] <test name> passed".

Test Case	Status
[PHPUnit] endreKontoTest	passed
[PHPUnit] endreKundeInfoTest	passed
[PHPUnit] hentAlleKontiTest	passed
[PHPUnit] hentAlleKunderTest	passed
[PHPUnit] hentBetalingerTest	passed
[PHPUnit] hentKontiTest	passed
[PHPUnit] hentKundeInfoTest	passed
[PHPUnit] hentSaldiTest	passed
[PHPUnit] hentTransaksjonerTest	passed
[PHPUnit] registrerBetalingTest	passed
[PHPUnit] registrerKontoTest	passed
[PHPUnit] registrerKundeTest	passed
[PHPUnit] sjekkLoggInnTest	passed
[PHPUnit] slettKontoTest	passed
[PHPUnit] slettKundeTest	passed
[PHPUnit] utførBetalingTest	passed

## Oversikt over code coverage:



## Bank funksjoner

### BankLogikk code coverage:

```
<?php
include_once '../DAL/bankDatabase.php';
include_once '../DAL/bankDatabaseStub.php';

class Bank
{
    private $db;
    function __construct($innDb=null)
    {
        if($innDb==null)
        {
            $this->db=new BankDB();
        }
        else
        {
            $this->db=$innDb;
        }
    }

    function hentTransaksjoner($kontoNr, $fraDato, $stilDato)
    {
        $konto= $this->db->hentTransaksjoner($kontoNr, $fraDato, $stilDato);
        return $konto;
    }

    function sjekkloggInn($personnummer,$passord)
    {
        if(!preg_match("/^[\d]{11}$/", $personnummer))
        {
            return "Feil i personnummer";
        }

        if(!preg_match("/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$!%*?&])[A-Za-z\d@$_!%*?&]{6,20}$/", $passord))
        {
            return "Feil i passord";
        }

        $OK = $this->db->sjekkloggInn($personnummer,$passord);
        return $OK;
    }

    function hentKonti($personnummer)
    {
        $konti = $this->db->hentKonti($personnummer);
        return $konti;
    }

    function hentSaldi($personnummer)
    {
        $saldo = $this->db->hentSaldi($personnummer);
        return $saldo;
    }

    function registrerBetaling($kontoNr, $transaksjon)
    {
        $ok = $this->db->registrerBetaling($kontoNr, $transaksjon);
        return $ok;
    }
}
```

```
function hentBetalinger($personnummer)
{
    $betalinger = $this->db->hentBetalinger($personnummer);
    return $betalinger;
}

function utførBetalning($TxID)
{
    $ok = $this->db->utførBetalning($TxID);
    return $ok;
}

function endreKundeInfo($kunde)
{
    $OK= $this->db->endreKundeInfo($kunde);
    return $OK;
}

function hentKundeInfo($personnummer)
{
    $kunde= $this->db->hentKundeInfo($personnummer);
    return $kunde;
}
}
```

## HentTransaksjonerTest:

```
class hentTransaksjonerTest extends PHPUnit\Framework\TestCase {

    public function testdatoFeilTransaksjoner()
    {
        // arrange
        $kontoNr = "10502023523";
        $fraDato = '2015-03-27';
        $tilDato = '2015-03-22';
        $bank=new Bank(new BankDBStub());
        // act
        $konto= $bank->hentTransaksjoner($kontoNr, $fraDato, $tilDato);
        // assert
        $this->assertEquals("Fra dato må være større enn tildato",$konto);
    }

    public function testIngenTransaksjoner()
    {
        // arrange
        $kontoNr = "10502023523";
        $fraDato = '2015-03-20';
        $tilDato = '2015-03-22';
        $bank=new Bank(new BankDBStub());
        // act
        $konto= $bank->hentTransaksjoner($kontoNr, $fraDato, $tilDato);
        // assert
        $this->assertEquals("010101234567",$konto->personnummer);
        $this->assertEquals($kontoNr,$konto->kontonummer);
        $this->assertEquals("Sparekonto",$konto->type);
        $this->assertEquals(2300.34,$konto->saldo);
        $this->assertEquals("NOK",$konto->valuta);
        $tomtArray = array();
        $this->assertEquals($tomtArray,$konto->transaksjoner);
    }
}
```

```

public function testEnTransaksjon()
{
    // arrange
    $kontoNr = "10502023523";
    $fraDato = '2015-03-26';
    $stilDato = '2015-03-26';
    $bank=new Bank(new BankDBStub());
    // act
    $konto= $bank->hentTransaksjoner($kontoNr, $fraDato, $stilDato);
    // assert
    $this->assertEquals("010101234567",$konto->personnummer);
    $this->assertEquals($kontoNr,$konto->kontonummer);
    $this->assertEquals("Sparekonto",$konto->type);
    $this->assertEquals(2300.34,$konto->saldo);
    $this->assertEquals("NOK",$konto->valuta);
    $this->assertEquals('2015-03-26',$konto->transaksjoner[0]->dato);
    $this->assertEquals(134.4,$konto->transaksjoner[0]->transaksjonBelop);
    $this->assertEquals("22342344556",$konto->transaksjoner[0]->fraTilKontonummer);
    $this->assertEquals("Meny Holtet",$konto->transaksjoner[0]->melding);
}
public function testToTransaksjoner()
{
    // arrange
    $kontoNr = "10502023523";
    $fraDato = '2015-03-27';
    $stilDato = '2015-03-30';
    $bank=new Bank(new BankDBStub());
    // act
    $konto= $bank->hentTransaksjoner($kontoNr, $fraDato, $stilDato);
    // assert
    $this->assertEquals("010101234567",$konto->personnummer);
    $this->assertEquals($kontoNr,$konto->kontonummer);
    $this->assertEquals("Sparekonto",$konto->type);
    $this->assertEquals(2300.34,$konto->saldo);
    $this->assertEquals("NOK",$konto->valuta);
    $this->assertEquals('2015-03-27',$konto->transaksjoner[0]->dato);
    $this->assertEquals(-2056.45,$konto->transaksjoner[0]->transaksjonBelop);
    $this->assertEquals("114342344556",$konto->transaksjoner[0]->fraTilKontonummer);
    $this->assertEquals("Husleie",$konto->transaksjoner[0]->melding);
    $this->assertEquals('2015-03-29',$konto->transaksjoner[1]->dato);
    $this->assertEquals(1454.45,$konto->transaksjoner[1]->transaksjonBelop);
    $this->assertEquals("114342344511",$konto->transaksjoner[1]->fraTilKontonummer);
    $this->assertEquals("Lekeland",$konto->transaksjoner[1]->melding);
}

```

```

public function testAlleTransaksjoner()
{
    // arrange
    $kontoNr = "10502023523";
    $fraDato = '2015-03-26';
    $tilDato = '2015-03-30';
    $bank=new Bank(new BankDBStub());
    // act
    $konto= $bank->hentTransaksjoner($kontoNr, $fraDato, $tilDato);
    // assert
    $this->assertEquals("010101234567",$konto->personnummer);
    $this->assertEquals($kontoNr,$konto->kontonummer);
    $this->assertEquals("Sparekonto",$konto->type);
    $this->assertEquals(2300.34,$konto->saldo);
    $this->assertEquals("NOK",$konto->valuta);
    $this->assertEquals('2015-03-26',$konto->transaksjoner[0]->dato);
    $this->assertEquals(134.4,$konto->transaksjoner[0]->transaksjonBelop);
    $this->assertEquals("22342344556",$konto->transaksjoner[0]->fraTilKontonummer);
    $this->assertEquals("Meny Holtet",$konto->transaksjoner[0]->melding);
    $this->assertEquals('2015-03-27',$konto->transaksjoner[1]->dato);
    $this->assertEquals(-2056.45,$konto->transaksjoner[1]->transaksjonBelop);
    $this->assertEquals("114342344556",$konto->transaksjoner[1]->fraTilKontonummer);
    $this->assertEquals("Husleie",$konto->transaksjoner[1]->melding);
    $this->assertEquals('2015-03-29',$konto->transaksjoner[2]->dato);
    $this->assertEquals(1454.45,$konto->transaksjoner[2]->transaksjonBelop);
    $this->assertEquals("114342344511",$konto->transaksjoner[2]->fraTilKontonummer);
    $this->assertEquals("Lekeland",$konto->transaksjoner[2]->melding);
}

```

## SjekkLoggInnTest:

```
class sjekkLoggInnTest extends PHPUnit\Framework\TestCase {

    public function testSjekkLoggInn_OK() {
        // arrange
        $personnummer = "12345678911";
        $passord = "Passord1!";
        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->sjekkLoggInn($personnummer, $passord);
        // assert
        $this->assertEquals("OK", $OK);
    }

    public function testSjekkLoggInn_FeilPersonnummer() {
        // arrange
        $personnummer = "10987654321";
        $passord = "Passord1!";
        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->sjekkLoggInn($personnummer, $passord);
        // assert
        $this->assertEquals("Feil", $OK);
    }

    public function testSjekkLoggInn_FeilPassord() {
        // arrange
        $personnummer = "12345678911";
        $passord = "Passord2!";
        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->sjekkLoggInn($personnummer, $passord);
        // assert
        $this->assertEquals("Feil", $OK);
    }

    // Under ligger tester mot regex
    public function testSjekkLoggInn_ForLangtPersonnummer() {
        // arrange
        $personnummer = "123456789112"; // her er det 12 tegn
        $passord = "Passord1!";
        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->sjekkLoggInn($personnummer, $passord);
        // assert
        $this->assertEquals("Feil i personnummer", $OK);
    }

    public function testSjekkLoggInn_FortPersonnummer() {
        // arrange
        $personnummer = "1234567891"; // her er det 10 tegn
        $passord = "Passord1!";
        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->sjekkLoggInn($personnummer, $passord);
        // assert
        $this->assertEquals("Feil i personnummer", $OK);
    }
}
```

```

public function testSjekkLoggInn_TegnIPersonnummer() {
    // arrange
    $personnummer = "1234-567891";
    $passord = "Passord1!";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i personnummer",$OK);
}

public function testSjekkLoggInn_BokstaverIPersonnummer() {
    // arrange
    $personnummer = "a1234567891";
    $passord = "Passord1!";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i personnummer",$OK);
}

public function testSjekkLoggInn_ForKortPassord() {
    // arrange
    $personnummer = "12345678911";
    $passord = "Pas1!";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i passord",$OK);
}

public function testSjekkLoggInn_ForLangtPassord() {
    // arrange
    $personnummer = "12345678911";
    $passord = "Passoooooooooooooord1!";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i passord",$OK);
}

public function testSjekkLoggInn_ManglerNummerPassord() {
    // arrange
    $personnummer = "12345678911";
    $passord = "Passord!";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i passord",$OK);
}

```

```
public function testSjekkLoggInn_ManglerStorBokstavPassord() {
    // arrange
    $personnummer = "12345678911";
    $passord = "111!!!a";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i passord", $OK);
}

public function testSjekkLoggInn_ManglerTegnPassord() {
    // arrange
    $personnummer = "12345678911";
    $passord = "Passord1";
    $bank = new Bank(new BankDBStub());
    // act
    $OK = $bank->sjekkLoggInn($personnummer, $passord);
    // assert
    $this->assertEquals("Feil i passord", $OK);
}
```

## HentKontiTest:

```
class hentKontiTest extends PHPUnit\Framework\TestCase {

    public function testHentKonti_Ingen() {
        // arrange
        $personnummer = "000000000000";
        $bank = new Bank(new BankDBStub());
        // act
        $konti = $bank->hentKonti($personnummer);
        // assert
        $stomtArray = [];
        $this->assertEquals($stomtArray, $konti);
    }

    public function testHentKonti_En() {
        // arrange
        $personnummer = "111111111111";
        $bank = new Bank(new BankDBStub());
        // act
        $konti = $bank->hentKonti($personnummer);
        // assert
        $this->assertEquals("12121212121", $konti[0]);
    }

    public function testHentKonti_Flere() {
        // arrange
        $personnummer = "33333333333";
        $bank = new Bank(new BankDBStub());
        // act
        $konti = $bank->hentKonti($personnummer);
        // assert
        $this->assertEquals("13131313131", $konti[0]);
        $this->assertEquals("14141414141", $konti[1]);
        $this->assertEquals("15151515151", $konti[2]);
    }

    public function testHentKonti_Feil() {
        // arrange
        $personnummer = "55555555555";
        $bank = new Bank(new BankDBStub());
        // act
        $konti = $bank->hentKonti($personnummer);
        // assert
        $this->assertEquals("Personnummer finnes ikke", $konti);
    }
}
```

## HentSaldiTest:

```
class hentSaldiTest extends PHPUnit\Framework\TestCase {

    public function testHentSaldi_Ingen() {
        // arrange
        $personnummer = "000000000000";
        $bank = new Bank(new BankDBStub());
        // act
        $saldi = $bank->hentSaldi($personnummer);
        // assert
        $stomtArray = [];
        $this->assertEquals($stomtArray, $saldi);
    }

    public function testHentSaldi_En() {
        // arrange
        $personnummer = "111111111111";
        $bank = new Bank(new BankDBStub());
        // act
        $saldi = $bank->hentSaldi($personnummer);
        // assert
        $this->assertEquals(10,$saldi[0]);
    }

    public function testHentSaldi_Flere() {
        // arrange
        $personnummer = "333333333333";
        $bank = new Bank(new BankDBStub());
        // act
        $saldi = $bank->hentSaldi($personnummer);
        // assert
        $this->assertEquals(100,$saldi[0]);
        $this->assertEquals(200,$saldi[1]);
        $this->assertEquals(300,$saldi[2]);
    }

    public function testHentSaldi_Feil() {
        // arrange
        $personnummer = "12345678911";
        $bank = new Bank(new BankDBStub());
        // act
        $saldi = $bank->hentSaldi($personnummer);
        // assert
        $this->assertEquals("Personnummer finnes ikke",$saldi);
    }
}
```

## RegistrerBetalingTest:

```
class registrerBetalingTest extends PHPUnit\Framework\TestCase {

    public function testRegistrerBetaling_OK() {
        // arrange
        $transaksjon = new transaksjon();
        $transaksjon->fraTilKontonummer = "1111111111";
        $transaksjon->belop = 200;
        $transaksjon->dato = "2015-03-14";
        $transaksjon->melding = "dingDing";
        $transaksjon->avventer = "1";

        $kontoNr = "12121212121";

        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->registrerBetaling($kontoNr, $transaksjon);
        // assert
        $this->assertEquals("OK", $OK);
    }

    public function testRegistrerBetaling_Feil() {
        // arrange
        $transaksjon = new transaksjon();

        $kontoNr = "";

        $bank = new Bank(new BankDBStub());
        // act
        $OK = $bank->registrerBetaling($kontoNr, $transaksjon);
        // assert
        $this->assertEquals("Feil", $OK);
    }
}
```

## HentBetalingerTest:

```
class hentBetalingerTest extends PHPUnit\Framework\TestCase {

    public function testHentBetalinger_OK_3Transaksjoner() {
        // arrange
        $bank = new Bank(new BankDBStub());
        $personnummer = "1";
        // act
        $transaksjoner = $bank->hentBetalinger($personnummer);
        // assert
        $this->assertEquals("1111111111", $transaksjoner[0]->fraTilKontonummer);
        $this->assertEquals(200, $transaksjoner[0]->belop);
        $this->assertEquals("2015-03-14", $transaksjoner[0]->dato);
        $this->assertEquals("dingDing", $transaksjoner[0]->melding);
        $this->assertEquals("1", $transaksjoner[0]->avventer);

        $this->assertEquals("1111112222", $transaksjoner[1]->fraTilKontonummer);
        $this->assertEquals(212, $transaksjoner[1]->belop);
        $this->assertEquals("2020-05-14", $transaksjoner[1]->dato);
        $this->assertEquals("hei", $transaksjoner[1]->melding);
        $this->assertEquals("1", $transaksjoner[1]->avventer);

        $this->assertEquals("1122222222", $transaksjoner[2]->fraTilKontonummer);
        $this->assertEquals(122, $transaksjoner[2]->belop);
        $this->assertEquals("2021-12-14", $transaksjoner[2]->dato);
        $this->assertEquals("Test", $transaksjoner[2]->melding);
        $this->assertEquals("1", $transaksjoner[2]->avventer);
    }

    public function testHentBetalinger_OK_1Transaksjoner() {
        // arrange
        $bank = new Bank(new BankDBStub());
        $personnummer = "4";
        // act
        $transaksjoner = $bank->hentBetalinger($personnummer);
        // assert
        $this->assertEquals("1111111111", $transaksjoner[0]->fraTilKontonummer);
        $this->assertEquals(200, $transaksjoner[0]->belop);
        $this->assertEquals("2015-03-14", $transaksjoner[0]->dato);
        $this->assertEquals("dingDing", $transaksjoner[0]->melding);
        $this->assertEquals("1", $transaksjoner[0]->avventer);
    }
}
```

```
public function testHentBetalinger_OK_ingenTransaksjoner(){
    //arrange
    $bank = new Bank(new BankDBStub());
    $personnummer = "2";
    //act
    $transaksjoner = $bank->hentBetalinger($personnummer);
    //assert
    $tomtArray = [];
    $this->assertEquals($transaksjoner,$tomtArray);
}

public function testHentBetalinger_OK_ingenKonto(){
    //arrange
    $bank = new Bank(new BankDBStub());
    $personnummer = "3";
    //act
    $transaksjoner = $bank->hentBetalinger($personnummer);
    //assert
    $this->assertEquals("Ingen konto",$transaksjoner);
}

public function testHentBetalinger_Feil() {
    $bank = new Bank(new BankDBStub());
    $personnummer = "5";
    //act
    $transaksjoner = $bank->hentBetalinger($personnummer);
    //assert
    $this->assertEquals($transaksjoner,null);
}
```

### UtförBetalingTest:

```
class utförBetalingTest extends PHPUnit\Framework\TestCase {

    public function testUtförBetaling_OK() {
        // arrange
        $TxID = "1";
        $bank = new Bank(new BankDBStub());
        // act
        $betaling = $bank->utförBetaling($TxID);
        // assert
        $this->assertEquals("OK", $betaling);

    }
    public function testUtförBetaling_Feil(){
        // arrange
        $TxID = "2";
        $bank = new Bank(new BankDBStub());
        // act
        $betaling = $bank->utförBetaling($TxID);
        // assert
        $this->assertEquals("Feil", $betaling);

    }
}
```

## HentKundeInfo:

```
class hentKundeInfoTest extends PHPUnit\Framework\TestCase {

    public function testHentKundeInfo_OK() {
        // arrange
        $personnummer = "11111222222";
        $bank = new Bank(new BankDBStub());
        // act
        $kunde = $bank->hentKundeInfo($personnummer);
        // assert
        $this->assertEquals("11111222222", $kunde->personnummer);
        $this->assertEquals("Roger", $kunde->fornavn);
        $this->assertEquals("Ruud", $kunde->etternavn);
        $this->assertEquals("Gata 1, Brasil", $kunde->adresse);
        $this->assertEquals("87654321", $kunde->telefonnr);
        $this->assertEquals("Passord123", $kunde->passord);
        $this->assertEquals("2040", $kunde->postnr);
    }

    public function testHentKundeInfo_Feil() {
        // arrange
        $personnummer = "12121212121";
        $bank = new Bank(new BankDBStub());
        // act
        $kunde = $bank->hentKundeInfo($personnummer);
        // assert
        $this->assertEquals("Feil", $kunde);
    }
}
```

## Admin funksjoner

### AdminLogikk code coverage:

```
<?php
include_once '../DAL/adminDatabase.php';
include_once '../DAL/adminDatabaseStub.php';

class Admin
{
    private $db;
    function __construct($innDb=null)
    {
        if($innDb==null)
        {
            $this->db=new AdminDB();
        }
        else
        {
            $this->db=$innDb;
        }
    }

    function hentAlleKunder()
    {
        $kunder= $this->db->hentAlleKunder();
        return $kunder;
    }

    function endreKundeInfo($kunde)
    {
        $OK= $this->db->endreKundeInfo($kunde);
        return $OK;
    }

    function registrerKunde($kunde)
    {
        $OK = $this->db->registrerKunde($kunde);
        return $OK;
    }

    function slettKunde($personnummer)
    {
        $OK = $this->db->slettKunde($personnummer);
        return $OK;
    }

    function registrerKonto($konto)
    {
        $OK = $this->db->registrerKonto($konto);
        return $OK;
    }

    function endreKonto($konto)
    {
        $OK = $this->db->endreKonto($konto);
        return $OK;
    }

    function hentAlleKonti()
    {
        $konti = $this->db->hentAlleKonti();
        return $konti;
    }
}
```

```

        function hentAlleKonti()
    {
        $konti = $this->db->hentAlleKonti();
        return $konti;
    }

    function slettKonto($kontonummer)
    {
        $OK = $this->db->slettKonto($kontonummer);
        return $OK;
    }
}

```

### HentAlleKunderTest:

```

class hentAlleKunderTest extends PHPUnit\Framework\TestCase {
function testHentAlleKunder()
{
    // arrange
    $admin=new Admin(new AdminDBStub());
    // act
    $kunder= $admin->hentAlleKunder();
    // assert
    $this->assertEquals("Per",$kunder[0]->fornavn);
    $this->assertEquals("Olsen",$kunder[0]->etternavn);
    $this->assertEquals("Osloveien 82",$kunder[0]->adresse);
    $this->assertEquals("0270",$kunder[0]->postnr);
    $this->assertEquals("Oslo",$kunder[0]->poststed);
    $this->assertEquals("12345678",$kunder[0]->telefonnr);
    $this->assertEquals("HeiHei",$kunder[0]->passord);

    $this->assertEquals("Line",$kunder[1]->fornavn);
    $this->assertEquals("Jensen",$kunder[1]->etternavn);
    $this->assertEquals("Askerveien 100",$kunder[1]->adresse);
    $this->assertEquals("1379",$kunder[1]->postnr);
    $this->assertEquals("Asker",$kunder[1]->poststed);
    $this->assertEquals("92876789",$kunder[1]->telefonnr);
    $this->assertEquals("HeiHei",$kunder[1]->passord);

    $this->assertEquals("Ole",$kunder[2]->fornavn);
    $this->assertEquals("Olsen",$kunder[2]->etternavn);
    $this->assertEquals("Bærumsveien 23",$kunder[2]->adresse);
    $this->assertEquals("1234",$kunder[2]->postnr);
    $this->assertEquals("Bærum",$kunder[2]->poststed);
    $this->assertEquals("99889988",$kunder[2]->telefonnr);
    $this->assertEquals("HeiHei",$kunder[2]->passord);
}
}

```

### EndreKundeInfoTest:

```
class endreKundeInfoTest extends PHPUnit\Framework\TestCase {  
    function testEndreKundeInfo_OK(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $kunde = new kunde();  
  
        $kunde->fornavn = "Per";  
        $kunde->etternavn = "Hansen";  
        $kunde->adresse = "Osloveien 82";  
        $kunde->postnr = "0270";  
        $kunde->poststed = "Oslo";  
        $kunde->telefonnr = "12345678";  
        $kunde->epost = "HeiHei";  
        // act  
        $OK = $admin->endreKundeInfo($kunde);  
        // assert  
        $this->assertEquals("OK",$OK);  
    }  
    function testEndreKundeInfo_OK_postAdd(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $kunde = new kunde();  
  
        $kunde->fornavn = "Per";  
        $kunde->etternavn = "Hansen";  
        $kunde->adresse = "Osloveien 82";  
        $kunde->postnr = "1000";  
        $kunde->poststed = "Oslo";  
        $kunde->telefonnr = "12345678";  
        $kunde->epost = "HeiHei";  
        // act  
        $OK = $admin->endreKundeInfo($kunde);  
        // assert  
        $this->assertEquals("OK",$OK);  
    }  
}
```

```

function testEndreKundeInfo_OK_postAdd(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $kunde = new kunde();

    $kunde->fornavn = "Per";
    $kunde->etternavn ="Hansen";
    $kunde->adresse = "Osloveien 82";
    $kunde->postnr = "1000";
    $kunde->poststed = "Oslo";
    $kunde->telefonnr = "12345678";
    $kunde->epost = "HeiHei";
    // act
    $OK = $admin->endreKundeInfo($kunde);
    //assert
    $this->assertEquals("OK",$OK);
}

function testEndreKundeInfo_Feil(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $kunde = new kunde();

    $kunde->fornavn = "Per";
    $kunde->etternavn ="Hansen";
    $kunde->adresse = "Osloveien 82";
    $kunde->postnr = "0";
    $kunde->poststed = "Oslo";
    $kunde->telefonnr = "12345678";
    $kunde->epost = "HeiHei";
    // act
    $Feil = $admin->endreKundeInfo($kunde);
    //assert
    $this->assertEquals("Feil",$Feil);
}
}

```

### RegistrerKundeTest:

```
class registrerKundeTest extends PHPUnit\Framework\TestCase {  
    function testRegistrerKunde_OK(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $kunde = new kunde();  
  
        $kunde->personnummer= "12345678910";  
        $kunde->fornavn = "Per";  
        $kunde->etternavn ="Hansen";  
        $kunde->adresse = "Osloveien 82";  
        $kunde->postnr = "0270";  
        $kunde->poststed = "Oslo";  
        $kunde->telefonnr = "12345678";  
        $kunde->epost = "HeiHei";  
        // act  
        $OK = $admin->registrerKunde($kunde);  
        //assert  
        $this->assertEquals("OK",$OK);  
    }  
    function testRegistrerKunde_OK_postAdd(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $kunde = new kunde();  
  
        $kunde->personnummer= "12345678910";  
        $kunde->fornavn = "Per";  
        $kunde->etternavn ="Hansen";  
        $kunde->adresse = "Osloveien 82";  
        $kunde->postnr = "1000";  
        $kunde->poststed = "Oslo";  
        $kunde->telefonnr = "12345678";  
        $kunde->epost = "HeiHei";  
        // act  
        $OK = $admin->registrerKunde($kunde);  
        //assert  
        $this->assertEquals("OK",$OK);  
    }  
}
```

```

function testRegistrerKunde_Feil_postnr(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $kunde = new kunde();

    $kunde->personnummer= "12345678910";
    $kunde->fornavn = "Per";
    $kunde->etternavn ="Hansen";
    $kunde->adresse = "Osloveien 82";
    $kunde->postnr = "0";
    $kunde->poststed = "Oslo";
    $kunde->telefonnr = "12345678";
    $kunde->epost = "HeiHei";
    // act
    $Feil = $admin->registrerKunde($kunde);
    //assert
    $this->assertEquals("Feil",$Feil);
}

function testRegistrerKunde_Feil_personnr(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $kunde = new kunde();

    $kunde->personnummer= "0";
    $kunde->fornavn = "Per";
    $kunde->etternavn ="Hansen";
    $kunde->adresse = "Osloveien 82";
    $kunde->postnr = "0270";
    $kunde->poststed = "Oslo";
    $kunde->telefonnr = "12345678";
    $kunde->epost = "HeiHei";
    // act
    $Feil = $admin->registrerKunde($kunde);
    //assert
    $this->assertEquals("Feil",$Feil);
}
}

```

### SlettKundeTest:

```
class slettKundeTest extends PHPUnit\Framework\TestCase {  
    function testSlettKunde_OK(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $personnummer = "12345678910";  
        // act  
        $OK = $admin->slettKunde($personnummer);  
        //assert  
        $this->assertEquals("OK",$OK);  
    }  
    function testSlettKunde_Feil(){  
        // arrange  
        $admin = new Admin(new AdminDBStub());  
        $personnummer = "12345678911";  
        // act  
        $Feil = $admin->slettKunde($personnummer);  
        //assert  
        $this->assertEquals("Feil",$Feil);  
    }  
}
```

### RegistrerKontoTest:

```
class registrerKontoTest extends PHPUnit\Framework\TestCase {
function testRegistrerKonto_OK(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $konto = new konto();

    $konto->personnummer = "12345678910";
    $konto->kontonummer = "12345678910";
    $konto->saldo= "100";
    $konto->type = "type";
    $konto->valuta = "Nok";
    $konto->transaksjoner = array(); // av transaksjon
    // act
    $OK = $admin->registrerKonto($konto);
    //assert
    $this->assertEquals("OK",$OK);
}

function testRegistrerKonto_Feil(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $konto = new konto();

    $konto->personnummer = "12345678911";
    $konto->kontonummer = "12345";
    $konto->saldo= "100";
    $konto->type = "type";
    $konto->valuta = "Nok";
    $konto->transaksjoner = array(); // av transaksjon
    // act
    $Feil = $admin->registrerKonto($konto);
    //assert
    $this->assertEquals("Feil",$Feil);
}

}
```

### EndreKontoTest:

```
| class endreKontoTest extends PHPUnit\Framework\TestCase {  
|     function testEndreKonto_OK(){  
|         // arrange  
|         $admin = new Admin(new AdminDBStub());  
|         $konto = new konto();  
  
|         $konto->personnummer = "12345678910";  
|         $konto->kontonummer = "12345";  
|         $konto->saldo= "100";  
|         $konto->type = "type";  
|         $konto->valuta = "Nok";  
|         $konto->transaksjoner = array(); // av transaksjon  
|         // act  
|         $OK = $admin->endreKonto($konto);  
|         //assert  
|         $this->assertEquals("OK",$OK);  
|     }  
|     function testEndreKonto_Feil_persNr(){  
|         // arrange  
|         $admin = new Admin(new AdminDBStub());  
|         $konto = new konto();  
  
|         $konto->personnummer = null;  
|         $konto->kontonummer = "12345678910";  
|         $konto->saldo= "100";  
|         $konto->type = "type";  
|         $konto->valuta = "Nok";  
|         $konto->transaksjoner = array(); // av transaksjon  
|         // act  
|         $Feil = $admin->endreKonto($konto);  
|         //assert  
|         $this->assertEquals("Feil",$Feil);  
|     }  
| }
```

```

function testEndreKonto_Feil_persNr(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $konto = new konto();

    $konto->personnummer = null;
    $konto->kontonummer = "12345678910";
    $konto->saldo= "100";
    $konto->type = "type";
    $konto->valuta = "Nok";
    $konto->transaksjoner = array(); // av transaksjon
    // act
    $Feil = $admin->endreKonto($konto);
    //assert
    $this->assertEquals("Feil",$Feil);
}

function testEndreKonto_Feil_kontoNr(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    $konto = new konto();

    $konto->personnummer = "12345678910";
    $konto->kontonummer = null;
    $konto->saldo= "100";
    $konto->type = "type";
    $konto->valuta = "Nok";
    $konto->transaksjoner = array(); // av transaksjon
    // act
    $Feil = $admin->endreKonto($konto);
    //assert
    $this->assertEquals("Feil",$Feil);
}
}

```

## HentAlleKontiTest:

```
class hentAlleKontiTest extends PHPUnit\Framework\TestCase {
function testHentAlleKonti_OK(){
    // arrange
    $admin = new Admin(new AdminDBStub());
    // act
    $konti = $admin->hentAlleKonti();
    //assert
    $this->assertEquals("12345678910",$konti[0]->personnummer);
    $this->assertEquals("12345",$konti[0]->kontonummer);
    $this->assertEquals("4560",$konti[0]->saldo);
    $this->assertEquals("type",$konti[0]->type);
    $this->assertEquals("Euro",$konti[0]->valuta);
    $this->assertEquals(array(),$konti[0]->transaksjoner);

    $this->assertEquals("12345678911",$konti[1]->personnummer);
    $this->assertEquals("12346",$konti[1]->kontonummer);
    $this->assertEquals("5560",$konti[1]->saldo);
    $this->assertEquals("type",$konti[1]->type);
    $this->assertEquals("NOK",$konti[1]->valuta);
    $this->assertEquals(array(),$konti[1]->transaksjoner);

    $this->assertEquals("12345678912",$konti[2]->personnummer);
    $this->assertEquals("12347",$konti[2]->kontonummer);
    $this->assertEquals("6560",$konti[2]->saldo);
    $this->assertEquals("type",$konti[2]->type);
    $this->assertEquals("SEK",$konti[2]->valuta);
    $this->assertEquals(array(),$konti[2]->transaksjoner);
}

}
```

### SlettKontoTest:

```
class slettKontoTest extends PHPUnit\Framework\TestCase {
    function testSlettKonto_OK(){
        // arrange
        $admin = new Admin(new AdminDBStub());
        $konto = new konto();

        $konto->personnummer = "12345678910";
        $konto->kontonummer = "12345";
        $konto->saldo= "100";
        $konto->type = "type";
        $konto->valuta = "NOK";
        $konto->transaksjoner = array(); // av transaksjon
        // act
        $OK = $admin->slettKonto($konto);
        //assert
        $this->assertEquals("OK",$OK);
    }
    function testSlettKonto_Feil(){
        // arrange
        $admin = new Admin(new AdminDBStub());
        $konto = new konto();

        $konto->personnummer = "12345678910";
        $konto->kontonummer = "12346";
        $konto->saldo= "100";
        $konto->type = "type";
        $konto->valuta = "NOK";
        $konto->transaksjoner = array(); // av transaksjon
        // act
        $Feil = $admin->slettKonto($konto);
        //assert
        $this->assertEquals("Feil",$Feil);
    }
}
```

## 10.3.2. Integrasjonstest

### 10.3.2.1 Oversikt i excel

#### Integrasjonstest

Test Results Summary (Auto Calculated)		Number	Percent	Overview		
Total Test Cases Passed		64	90%	Gul = Ikke kritisk feil.		
Total Test Cases Failed		7	10%	Rød = Kritisk feil.		
Total Test Cases Remaining		0	0%	Blå = Skal ikke testes.		
API-rekkefølge:	Type	Status test:	Resultat:	Feil kat.:	Notater	
loggInn.php (uten databasen koblet til)	get	Ferdig	Pass	B	API returnerer ingen feilmelding	
loggInn.php	get	Ferdig	Passed			
hentTransaksjoner.php	get	Ferdig	Passed			
hentKonti.php	get	Ferdig	Passed			
hentSaldi.php	get	Ferdig	Passed			
registrerBetalning.php	post	Ferdig	Passed			
hentBetalinger.php	get	Ferdig	Passed			
utforBetalning.php	get	Ferdig	Passed			
hentKundeInfo	get	Ferdig	obs		Får ikke testet om personnummeret ikke eksisterer	
endreKundeinfo.php	post	Ferdig	Fail	B	API returnerer feil	
loggUt.php	get	Ferdig	Passed			
apiBankHeader.php						
loggInnAdmin.php	get	Ferdig	Passed			
adminEndreKonto.php	post	Ferdig	Fail	B	API returnerer feil	
adminEndreKunde.php	post	Ferdig	Fail	B	API returnerer feil	
adminHentKonti.php	get	Ferdig	Passed			
adminRegistrerKunde.php	post	Ferdig	Passed			
adminRegistrerKonto.php	post	Ferdig	Passed			
adminSlettKonto.php	delete	Ferdig	Passed			
adminSettKunde.php	delete	Ferdig	Fail	B	API returnerer feil	
hentAlleKunder.php	get	Ferdig	Ferdig			
apiAdminHeader.php						

#### Uten database-tilkobling

Test Info		Test Summary	
Test Scenario	Sjekk innlogging uten tilkobling til databasen	Total Passed	0
Test Environment	SoapUI / localhost	Total Failed	1
Test Data Source	<a href="http://localhost/testing/TestingBank/API/loggInn.php">http://localhost/testing/TestingBank/API/loggInn.php</a>	Total Remaining	0
Test Case ID	Test Case Description	Expected Result	Test Data
U-0001	Logg inn med riktig brukernavn/personnummer og passord	Feilmelding	personnummer: 01010110523 passord: Password1!
			Actual Result
			Ingen resultat
			Status
			Fail
			Comments
			Testen feiler fordi den ikke får tilgang til databasen.

## Bank funksjoner

Test Info		Test Summary				
Test Scenario	Sjekk innlogging	Total Passed	2	Actual Result		Status
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/innlogginn.php">http://localhost/testing/TestingBank/API/innlogginn.php</a>	Total Remaining	0			
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
LI-0001	Logg inn med riktig brukernavn/personnummer og passord	Innlogging godkjent ("OK")	personnummer: 01010110523 password: Passord11	Innlogging godkjent ("OK")	Pass	
LI-0002	Logg inn med riktig brukernavn/personnummer men feil passord	Innlogging ikke godkjent ("Feil i innlogging")	personnummer: 01010112345 password: FeilPassord	Innlogging ikke godkjent ("Feil i innlogging")	Pass	

Test Info		Test Summary				
Test Scenario	hente transaksjoner for kunde	Total Passed	4	Actual Result		Status
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/hentTransaksjoner.php">http://localhost/testing/TestingBank/API/hentTransaksjoner.php</a>	Total Remaining	0			
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
HT-0001	Hent ingen transaksjoner	count (0)	KontoNummer: 105010123456 datoFra: 2013-01-01 datoTil: 2013-01-02	count (0)	Pass	
HT-0002	Hente en transaksjon fra konto	count (1)	KontoNummer: 105010123456 datoFra: 2015-03-14 datoTil: 2015-03-16	count (1)	Pass	
HT-0003	Hente to transaksjoner fra konto	count (2)	KontoNummer: 105010123456 datoFra: 2015-03-14 datoTil: 2015-03-21	count (2)	Pass	
HT-0004	Hente alle transaksjoner fra konto	count (6)	KontoNummer: 105010123456 datoFra: "" datoTil: ""	count (6)	Pass	

Test Info		Test Summary				
Test Scenario	hente kontoene til kunde	Total Passed	3	Actual Result		Status
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/hentKonti.php">http://localhost/testing/TestingBank/API/hentKonti.php</a>	Total Remaining	0			
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
HK-0001	Hente kontoer for bruker uten konto	ingen konto returnert [null]	personnummer: 12345678901	ingen konto returnert [null]	pass	
HK-0002	Hente kontoer for bruker med en konto	count (1)	personnummer: 12345123456	count (1)	pass	
HK-0003	Hente kontoer for bruker med tre kontoer	count (3)	personnummer: 01010110523	count (3)	pass	

Test Info		Test Summary				
Test Scenario	hente saldo på kontene til kunde	Total Passed	3	Actual Result		Status
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/hentSaldi.php">http://localhost/testing/TestingBank/API/hentSaldi.php</a>	Total Remaining	0			
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
HS-0001	Hent saldo for ingen konto	Ingen saldo returnert (null)	personnummer: 12345678901	Ingen saldo returnert (null)	Pass	
HS-0002	Hent saldo for en konto	count (1)	personnummer: 12345123456	count (1)	Pass	
HS-0003	Hent saldo for tre kontoer	count (3)	personnummer: 01010110523	count (3)	Pass	

Test Info		Test Summary	
Test Scenario	En kunde skal registrere en betaling	Total Passed	3
Test Environment	SoapUI / localhost	Total Failed	0
Test Data Source	<a href="http://localhost/testing/TestingBank/API/registerBeting.php">http://localhost/testing/TestingBank/API/registerBeting.php</a>	Total Remaining	0
Test Case ID	Test Case Description	Expected Result	Test Data
RB-0001	registrer betaling med riktige opplysninger hentBetalinger	Godkjent("OK") count (1)	tilKonto: 10101334532 dato: 2020-06-28 beløp: 100 melding: "Takk for innskuren!" kontoNr: 103010123456
RB-0002	registrer betaling med feil opplysninger	Ikke godkjent("Feil")	tilKonto: 10101334532 dato: 2020-06-28 beløp: beløftBeløp melding: "Takk for innskuren!" kontoNr: 666666 66 6666
Actual Result	Status	Comments	
Godkjent("OK") count (1)	Pass	(for å vise at betalingen faktisk ble registrert)	
Ikke godkjent("Feil")	Pass		

Test Info		Test Summary	
Test Scenario	Utfører betalinger som avventer	Total Passed	3
Test Environment	SoapUI / localhost	Total Failed	0
Test Data Source	<a href="http://localhost/testing/TestingBank/API/utforBeting.php">http://localhost/testing/TestingBank/API/utforBeting.php</a>	Total Remaining	0
Test Case ID	Test Case Description	Expected Result	Test Data
UB-0001	Sjekk at det ligger tre betaling til avventer på en kunde	Array med tre betaling	TxD = 9
UB-0002	Utfør betaling med riktig opplysninger	Array med to betalinger	TxD = 11
UB-0003	Utfør betaling med feil opplysninger	Array med to betalinger	TxD = 9
Actual Result	Status	Comments	
Array med tre betaling	Pass		
Array med to betalinger	Pass		
Array med to betalinger	Pass		

Test Info		Test Summary	
Test Scenario	En kunde skal se betalingene som avventer	Total Passed	3
Test Environment	SoapUI / localhost	Total Failed	0
Test Data Source	<a href="http://localhost/testing/TestingBank/API/hentBetalinger.php">http://localhost/testing/TestingBank/API/hentBetalinger.php</a>	Total Remaining	0
Test Case ID	Test Case Description	Expected Result	Test Data
HB-0001	Henter betalinger fra person uten betalinger som avventer	returnerer tomt array (null)	personnummer: 12345678901
HB-0002	Henter betalinger fra person med en betaling som avventer	count (1)	personnummer: 12345123456
HB-0003	Henter betalinger fra person med tre betalinger som avventer	count (3)	personnummer: 11029700112
Actual Result	Status	Comments	
returnerer tomt array (null)	Pass		
count (1)	Pass		
count (3)	Pass		

Test Info		Test Summary	
Test Scenario	Hente kunde-informasjon registrert på en kunde	Total Passed	2
Test Environment	SoapUI / localhost	Total Failed	0
Test Data Source	<a href="http://localhost/testing/TestingBank/API/hentKundeInfo.php">http://localhost/testing/TestingBank/API/hentKundeInfo.php</a>	Total Remaining	0
Test Case ID	Test Case Description	Expected Result	Test Data
KI-0001	Informasjon om kunden hentes	Kundeobjekt hentet	personnummer = 010101010523
KI-0002	Postnr ikke registrert i poststed-tabell og informasjon om kunden kan derfor ikke hentes	Feilmelding ("Feil")	personnummer = 12345123456
Actual Result	Status	Comments	
Kundeobjekt hentet	Pass		
Feilmelding ("Feil")	Pass		

Test Info		Test Summary				
Test Scenario	Endre kunde-informasjonen registrert på en kunde	Total Passed	3	Actual Result	Status	Comments
Test Environment	SoapUI / localhost	Total Failed	1			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/endreKundeInfo.php">http://localhost/testing/TestingBank/API/endreKundeInfo.php</a>	Total Remaining	0			
EK-0001	Endre kundeinfo med gyldige opplysninger + eksisterende postnr	Godkjent ("OK")	personnummer: 01010110532 fornavn: Line etternavn: Jensen Hansen adresse: Askerveien 33 postnr: 3270 telefonnr: 33334444 poststed: Asker passord: Passord11	Godkjent ("OK")	Pass	
EK-0002	Endre kundeinfo med gyldige opplysninger + ikke-eksisterende postnr	Godkjent ("OK")	personnummer: 01010110532 fornavn: Len etternavn: Hansen adresse: Osloveien 22 postnr: 0565 telefonnr: 22224444 poststed: Oslo passord: Passord11	Godkjent ("OK")	Pass	
EK-0003	Endre kundeinfo med feil i postnr	Ikke godkjent("Feil")	personnummer: 01010110532 fornavn: Len etternavn: Jensen adresse: Askerveien 22 postnr: 32701 telefonnr: 22224444 poststed: Asker passord: Passord11	Ikke godkjent("Feil")	Pass	
EK-0004	Endre kundeinfo med feil i telefonnr	Ikke godkjent("Feil")	personnummer: 01010110532 fornavn: Len etternavn: Jensen adresse: Askerveien 22 postnr: 3270 telefonnr: 222244445 poststed: Asker passord: Passord11	Godkjent ("OK")	Fail	Faller fordi API returnerer "ok" uansett, selvom det er feil i telefonnummer

Test Info		Test Summary				
Test Scenario	Logge ut fra kundeside	Total Passed	3	Actual Result	Status	Comments
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/logout.php">http://localhost/testing/TestingBank/API/logout.php</a>	Total Remaining	0			
LU-0001	LoggInn	Godkjent ("OK")	personnummer = 12345123456, passord = Passord11	Godkjent ("OK")	Pass	
LU-0002	kopi av hent en saldo	count (1)	personnummer = 12345123456	count (1)	Pass	bare for å vise at det kjører før loggUt
LU-0003	LoggUt	contains empty		contains empty	Pass	

## Admin funksjoner

Test Info		Test Summary			
Test Scenario	En administrator logger inn	Total Passed	2	Total Failed	0
Test Environment	SoapUI / localhost	Total Remaining	0 <th data-cs="2" data-kind="parent">Test Data Source</th> <th data-kind="ghost"></th>	Test Data Source	
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
LIA-0001	Logger inn med riktig brukernavn og passord	Godkjent("Admin")	Brukernavn=Admin Passord=Admin	Godkjent("Admin")	Pass
LIA-0002	Logger inn med feil brukernavn og passord	Feilmelding("Feil i innlogging")	Brukernavn=bruker Passord=feil	Feilmelding("Feil i innlogging")	Pass

Test Info		Test Summary			
Test Scenario	Endre en konto på en kunde	Total Passed	3	Total Failed	2
Test Environment	SoapUI / localhost	Total Remaining	0 <th data-cs="2" data-kind="parent">Test Data Source</th> <th data-kind="ghost"></th>	Test Data Source	
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
AEKO-0001	Endring i personnummer til et som ikke er registrert som kunde	Feilmelding("Feil i personnummer")	personnummer=000000000000 kontonummer=105010123456 type=Lønnskonto saldo:720 valuta=NOK	Feilmelding("Feil i personnummer")	Pass
AEKO-0002	Endring i kontonummer	Feilmelding("Feil i kontonummer")	personnummer=01010110523 kontonummer=000000000000 type=Lønnskonto saldo:720 valuta=NOK	Feilmelding("Feil i kontonummer")	Pass
AEKO-0003	Godkjent endring av konto	Godkjent("OK")	personnummer=01010110523 kontonummer=105010123456 type=Lønnskonto saldo:820 valuta=NOK	Godkjent("OK")	Pass
AEKO-0004	Ugyldig endring av saldo	Feilmelding("Feil")	personnummer=01010110523 kontonummer=105010123456 type=Lønnskonto saldo:test valuta=NOK	Godkjent("OK")	Fail
AEKO-0005	Ugyldig endring av valuta	Feilmelding("Feil")	personnummer=01010110523 kontonummer=105010123456 type=Lønnskonto saldo:820 valuta=Dollar	Godkjent("OK")	Fail

Test Info		Test Summary			
Test Scenario	Endre informasjon om en kunde	Total Passed	2	Total Failed	2
Test Environment	SoapUI / localhost	Total Remaining	0	Test Data Source	
Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status
AEKU-0001	Endre kundeinfo med gyldige opplysninger + ikke-eksisterende postnr	Godkjent ("OK")	Personnummer: 01010110532 Fornavn: Lene Etternavn: Jensen Hansen Adresse: Osloveien 33 PostNr: 0560 telefonnummer: 33334444 poststed: Oslo passord: Passord11	Godkjent ("OK")	Pass
AEKU-0002	Endre kundeinfo med gyldige opplysninger + eksisterende postnr	Godkjent ("OK")	Personnummer: 01010110532 Fornavn: Lene Etternavn: Jensen Adresse: Askerveien 22 PostNr: 3270 telefonnummer: 22224444 poststed: Asker passord: Passord11	Godkjent ("OK")	Pass
AEKU-0003	Endre kundeinfo med feil i postnr	Ikke godkjent("Feil")	Personnummer: 01010110532 fornavn: Lene etternavn: Jensen adresse: Askerveien 22 postnr: 3270 telefon: 22224445 poststed: Asker passord: Passord11	Godkjent ("OK")	Fail
AEKU-0004	Endre kundeinfo med feil i telefonnr	Ikke godkjent("Feil")	Personnummer: 01010110532 fornavn: Lene etternavn: Jensen adresse: Askerveien 22 postnr: 3270 telefon: 22224445 poststed: Asker passord: Passord11	Godkjent ("OK")	Fail

Test Info		Test Summary				
Test Scenario	En administrator henter alle kontoer	Total Passed	8	Actual Result	Status	Comments
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/adminHentKonti.php">http://localhost/testing/TestingBank/API/adminHentKonti.php</a>	Total Remaining	0			
AHK-0001	Hent alle konti	Expected Result	count (7)	Test Data	Actual Result	Status
				kontonummer: - 105010123456 -105020123456 -2233412345 -22222334444	count (7)	Pass
AHK-0002	Kjør adminSlettKonto	Godkjent ("OK")			Godkjent ("OK")	Pass
	Hent tre konti	Expected Result	count (3)	kontonummer:	count (3)	Pass
				-33333662222 -44444992222		(en slett-case pr kontonummer)
AHK-0003	Kjør adminSlettKonto	Godkjent ("OK")	count (1)	kontonummer:	count (1)	Pass
	Hent en konto	Expected Result		-55555224444		(en slett-case pr kontonummer)
AHK-0004	Kjør adminSlettKonto	Godkjent ("OK")			Godkjent ("OK")	Pass
	Hent ingen konto	Expected Result	count (0)		count (0)	Pass
	Kjør initDB	Expected Result				Pass

Test Info		Test Summary				
Test Scenario	En administrator registrerer en ny kunde	Total Passed	4	Actual Result	Status	Comments
Test Environment	SoapUI / localhost	Total Failed	0			
Test Data Source	<a href="http://localhost/testing/TestingBank/API/adminRegistrerKunde.php">http://localhost/testing/TestingBank/API/adminRegistrerKunde.php</a>	Total Remaining	0			
ARKU-0001	Registrere kunde med nytt riktig postNr	Expected Result	Godkjent("OK")	Test Data	Actual Result	Status
				personnummer: 01020312345 fornavn: Hans etternavn: Hansen adresse: Ikarosveien 73 postnr: 2040 poststed: Klofta telefonnr: 22353522 passord: Passord1!	Godkjent("OK")	Pass
ARKU-0002	Registrere kunde med eksisterende postNr	Expected Result	Godkjent("OK")	personnummer: 01030512345 fornavn: Bjørn etternavn: Stjerne adresse: Karl Johan 1 postnr: 2040 poststed: Klofta telefonnr: 22226666 passord: Passord1!	Godkjent("OK")	Pass
ARKU-0003	Registrere kunde med nytt, feil postNr	Expected Result	Feilmelding("Feil")	personnummer: 12121230303 fornavn: Sindre etternavn: Sut adresse: Bobleveien 32 postnr: 123456 poststed: Oslo telefonnr: 11223344 passord: Passord1!	Feilmelding("Feil")	Pass
ARKU-0004	Registrere kunde med eksisterende postNr og feil telefonnummer	Expected Result	Feilmelding("Feil")	personnummer: 06060655555 fornavn: Kjartan etternavn: Kjuus adresse: Nedoverbakken 3 postnr: 1234 poststed: Oslo telefonnr: 123456789 passord: Passord1!	Feilmelding("Feil")	Pass

**Test Info**

Test Scenario En administrator registerer en ny konto på en kunde  
 Test Environment SoapUI / localhost  
 Test Data Source <http://localhost/testing/TestingBank/API/adminRegistrerKonto.php>

**Test Summary**

Total Passed	6
Total Failed	0
Total Remaining	0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
	Kjør adminHentKonti	count (7)	personnummer: 12345678901 kontonummer: 66666778888 type: Brukskonto valuta: NOK	count (7)	Pass	
ARKO-0001	Register konto på kunde	Godkjent("OK")		Godkjent("OK")	Pass	
	Kjør adminHentKonti	count (8)	personnummer: 12345678901 kontonummer: 98765432123 type: Sparekonto valuta: Kroner	count (8)	Pass	
ARKO-0002	Registrer konto på kunde med feil valuta	Feilmelding("Feil")	personnummer: 12345678901 kontonummer: 1234567891234567891011 type: Sparekonto valuta: NOK	Feilmelding("Feil")	Pass	
ARKO-0003	Registrer konto på kunde med feil kontonummer	Feilmelding("Feil")	personnummer: 66664455557 kontonummer: 12345678910 type: Sparekonto	Feilmelding("Feil")	Pass	
ARKO-0004	Registrer konto på kunde som ikke registrert i banken	Feilmelding("Feil i personnummer")	valuta: NOK	Feilmelding("Feil i personnummer")	Pass	

**Test Info**

Test Scenario Sjekk innlogging  
 Test Environment SoapUI / localhost  
 Test Data Source <http://localhost/testing/TestingBank/API/adminSlettKonto.php>

**Test Summary**

Total Passed	2
Total Failed	0
Total Remaining	0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
ASKO-0001	Slett konto med riktig kontonr	Godkjent("OK")	kontonummer: 66666778888	Godkjent("OK")	Pass	
ASKO-0002	Slett konto med feil kontonr	Feilmelding("Feil kontonummer")	kontonummer: 99999887777	Feilmelding("Feil kontonummer")	Pass	

**Test Info**

Test Scenario Admin Slett kunde  
 Test Environment SoapUI / localhost  
 Test Data Source <http://localhost/testing/TestingBank/API/adminSlettKunde.php>

**Test Summary**

Total Passed	1
Total Failed	1
Total Remaining	0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
ASKU-0001	Slett kunde med riktig personnummer	Godkjent("OK")	personnummer: 01030512345	Godkjent("OK")	Pass	
ASKU-0002	Slett kunde med feil personnummer	Feilmelding("Feil")	personnummer: 99999988888	Godkjent("OK")	Fail	Finner fordi API returnerer "ok" uansett hva som skjer

**Test Info**

Test Scenario Admin hent alle kunder  
 Test Environment SoapUI / localhost  
 Test Data Source <http://localhost/testing/TestingBank/API/hentAlleKunder.php>

**Test Summary**

Total Passed	7
Total Failed	0
Total Remaining	0

Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Comments
HAK-0001	Hent alle kunder	count (5)	personnummer: - 01020312345 - 01010110523	count (5)	Pass	
	Kjør adminSlettKunde	Godkjent ("ok")		Godkjent ("ok")	Pass	
HAK-0002	Hent alle kunder når det eksisterer tre kunder	count (3)	personnummer: - 12345123456 - 12345678901	count (3)	Pass	
	Kjør adminSlettKunde	Godkjent ("ok")		Godkjent ("ok")	Pass	
HAK-0003	Hent alle kunder når det eksisterer en kunde	count (1)	personnummer: - 11029700112	count (1)	Pass	
	Kjør adminSlettKunde	Godkjent ("ok")		Godkjent ("ok")	Pass	
HAK-0004	Hent alle kunder når det ikke eksisterer kunder	count (0)		count (0)	Pass	

### 10.3.2.2 Skjermbilder fra SoapUI

Uten database manuell-test

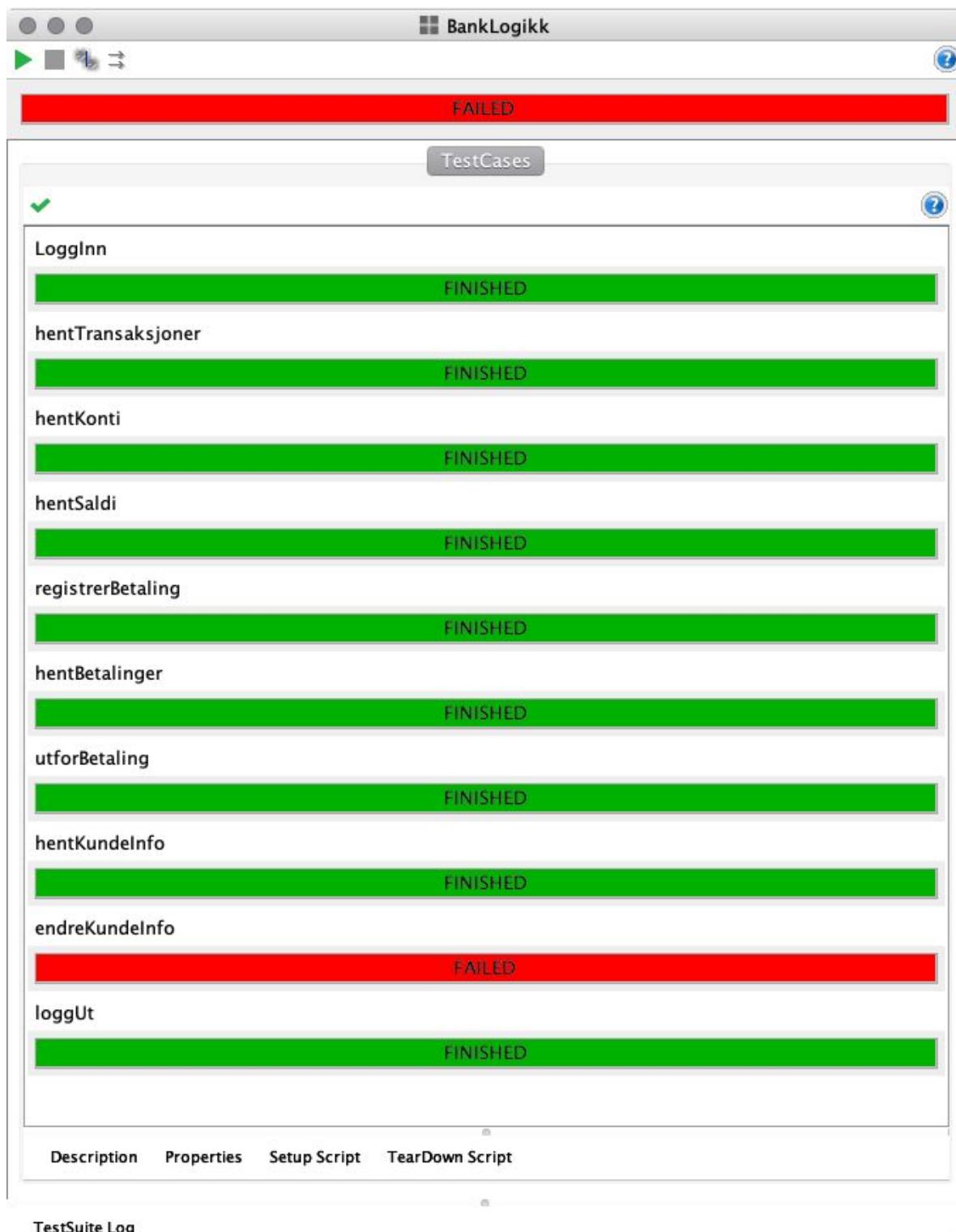
#### Simulering av kjøring uten aksess til database:

The screenshot shows the SoapUI interface with the following details:

- Title Bar:** Shows "Copy of LogInn" with a green checkmark icon.
- Status Bar:** Displays "FAILED".
- Toolbar:** Includes icons for TestSteps, Test On Demand, and various test types like SO, AP, RE, HT, A, JD, BC, etc.
- Test Case Header:** "logInn" is highlighted in orange.
- Test Case Content:** A large empty white area representing the test steps.
- Bottom Navigation:** Buttons for Description, Properties, Setup Script, and TearDown Script.
- Log Area:** Contains:
  - Test started at 2020-04-16 14:29:38.575
  - TestCase failed [Cancelling due to failed test step], time taken = 0
  - Step 1 [logInn] FAILED: took 0 ms**
  - > org.apache.http.conn.HttpHostConnectException: Connection to http://localhost refused
- TestCase Log:** A scrollable log area.

## Bank funksjoner

### Oversikt:



TestSuite Log

## LoggInn:

The screenshot displays two separate SoapUI sessions side-by-side.

**Session 1: InitDB**

- Request URL:** `http://localhost/testing/TestingBank/InitDB.php`
- Method:** GET
- Raw Request Data:** (Empty)
- Response:** Status: OK, Response body: "OK"
- Assertions:** 1 assertion passed (Contains - OK - VALID)

**Session 2: loggInn**

- Request URL:** `http://localhost/testing/TestingBank/API/loggInn.php`
- Method:** GET
- Raw Request Data:** (Empty)
- Request Parameters:**

Name	Value
personnummer	01010110523
passord	Password1!
- Response:** Status: OK, Response body: "OK"
- Assertions:** 1 assertion passed (Contains - OK - VALID)

Screenshot of a REST client interface showing a failed login attempt.

The top status bar shows the session name: **loggInn-FEIL**.

The request details are as follows:

- Method:** GET
- Request URL:** `calhost/testing/TestingBank/API/loggInn.php`
- Raw Request Data:**

Name	Value
personnummer	01010110523
passord	feil
- Response Headers:** Headers (11) | Attachments (0) | Status (200)
- Response Body:** `1 "Feil i innlogging"`

The bottom status bar indicates the session is valid: **loggInn-Feil - VALID**.

Navigation links at the bottom include: Assertions (1), Request Log (0).

## HentTransaksjoner:

The screenshot shows two API requests in SoapUI:

**Request 1: loggInn**

- Method: GET
- Request URL: `calhost/testing/TestingBank/API/loggInn.php`
- Raw Request Data:

Name	Value
personnummer	01010110523
passord	Passord1!
- Response (Raw):

```
1 "OK"
```
- Assertions:
  - Contains - VALID

**Request 2: hentTransaksjoner**

- Method: GET
- Request URL: `ting/TestingBank/API/hentTransaksjoner.php`
- Raw Request Data:

Name	Value
kontoNr	105010123456
fraDato	2013-01-01
tilDato	2013-01-02
- Response (Raw):

```
1 {
  "personnummer": "01010110523",
  "kontonummer": "105010123456",
  "saldo": "720",
  "type": "Lønnskonto",
  "valuta": "NOK",
  "transaksjoner": []
}
```
- Assertions:
  - Contains SALDO - VALID
  - JsonPath Count - VALID

Bottom navigation bar:  
 Assertions (2) Request Log (0)

Method: GET Request URL: ting/TestingBank/API/hentTransaksjoner.php Download Resources

Name	Value
kontoNr	105010123456
fraDato	2015-03-14
tilDato	2015-03-16

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info JMS (0)

```

1 {
2   "personnummer": "01010110523",
3   "kontonummer": "105010123456",
4   "saldo": "720",
5   "type": "Lennskonto",
6   "valuta": "NOK",
7   "transaksjoner": [
8     {
9       "fraTilKontonummer": "20102012345",
10      "transaksjonBelop": "-100.5",
11      "belop": null,
12      "dato": "2015-03-15",
13      "melding": "Meny Storo",
14      "avventer": null
15    }
16  ]
17 }
  
```

Assertions (1) Request Log (0)

TransaksjonerCount - VALID

Method: GET Request URL: ting/TestingBank/API/hentTransaksjoner.php Download Resources

Name	Value
kontoNr	105010123456
fraDato	2015-03-14
tilDato	2015-03-21

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info JMS (0)

```

1 {
2   "personnummer": "01010110523",
3   "kontonummer": "105010123456",
4   "saldo": "720",
5   "type": "Lennskonto",
6   "valuta": "NOK",
7   "transaksjoner": [
8     {
9       "fraTilKontonummer": "20102012345",
10      "transaksjonBelop": "-100.5",
11      "belop": null,
12      "dato": "2015-03-15",
13      "melding": "Meny Storo",
14      "avventer": null
15    },
16    {
17      "fraTilKontonummer": "20102012345",
18      "transaksjonBelop": "400.4",
19      "belop": null,
20      "dato": "2015-03-20",
21      "melding": "Innebalting",
22      "avventer": null
23    }
24  ]
25 }
  
```

JsonPath Count - VALID

Assertions (1) Request Log (0)

The screenshot shows the SoapUI interface with a successful API call result. The request URL is `ting/TestingBank/API/hentTransaksjoner.php`. The response body is displayed in XML format, showing a list of transactions:

```
1<?xml version="1.0"?>
2<transaksjoner>
3    <transasjon>
4        <personnummer>01010110523</personnummer>
5        <kontonummer>105010123456</kontonummer>
6        <saldo>720</saldo>
7        <type>Lennskonto</type>
8        <valuta>NOK</valuta>
9        <transaksjoner>
10            <transasjon>
11                <fraTilkontonummer>20102012345</fraTilkontonummer>
12                <transaksjonBelop>-100.5</transaksjonBelop>
13                <belop>null</belop>
14                <dato>2015-03-15</dato>
15                <melding>Meny Storo</melding>
16                <avventer>null</avventer>
17            </transasjon>
18            <transasjon>
19                <fraTilkontonummer>20102012345</fraTilkontonummer>
20                <transaksjonBelop>400.4</transaksjonBelop>
21                <belop>null</belop>
22                <dato>2015-03-20</dato>
23                <melding>Innebalting</melding>
24                <avventer>null</avventer>
25            </transasjon>
26        </transaksjoner>

```

The status bar at the bottom indicates `Assertions (1) Request Log (0)`.

## HentKonti:

The screenshot displays two separate API requests in the SoapUI application.

**Request 1: loggInnKonto**

- Method:** GET
- Request URL:** `calhost/testing/TestingBank/API/loggInn.php`
- Raw Request Data:**

Name	Value
personnummer	12345678901
passord	Passord1!
- Response:** "OK"
- Validation:** Contains - VALID

**Request 2: hentKontilagenKonto**

- Method:** GET
- Request URL:** `lhost/testing/TestingBank/API/hentKonti.php`
- Raw Request Data:**

Name	Value
personnummer	12345678901
- Response:** <Empty JSON content>
- Validation:** Contains-null - VALID

loggInnEnKonto

Method: GET Request URL: calhost/testing/TestingBank/API/loggInn.php Download Resources

Raw Request

Name Value  
personnummer 12345123456  
passord Passord1!

1 "OK"

Raw XML JSON HTML

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info

+ Contains - VALID

hentEnKonti

Method: GET Request URL: lhost/testing/TestingBank/API/hentKonti.php Download Resources

Raw Request

Name Value  
personnummer 12345123456

1 [{"Kontonummer": "55555224444"}]

Raw XML JSON HTML

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

+ Contains - VALID

**logInn**

Method: GET Request URL: calhost/testing/TestingBank/API/logInn.php Download Resources

Raw Request

Name	Value
personnummer	01010110523
passord	Passord1!

XML JSON HTML Raw

1	"OK"
---	------

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL I

+ Contains - VALID

**hentTreKonti**

Method: GET Request URL: lhost/testing/TestingBank/API/hentKonti.php Download Resources

Raw Request

Name	Value
personnummer	01010110523

XML JSON HTML Raw

1	[
2	{"Kontonummer": "105010123456"},
3	{"Kontonummer": "105020123456"},
4	{"Kontonummer": "22334412345"} ]
5	

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

+ KontonummerCount - VALID

## HentSaldi:

Screenshot of a web testing tool showing two requests: "loggInnIngenKonto" and "hentSaldiIngenKonto".

**loggInnIngenKonto Request:**

- Method: GET
- Request URL: `calhost/testing/TestingBank/API/loggInn.php`
- Raw Request Data:

Name	Value
personnummer	12345678901
passord	Passord1!
- Response:

1	"OK"
---	------
- Validation: Contains - VALID

**hentSaldiIngenKonto Request:**

- Method: GET
- Request URL: `ilhost/testing/TestingBank/API/hentSaldi.php`
- Raw Request Data:

Name	Value
personnummer	12345678901
- Response:

1	<Empty JSON content>
---	----------------------
- Validation: Contains-null - VALID

**loggInnEnKonto**

Method: GET Request URL: [calhost/testing/TestingBank/API/loggInn.php](http://calhost/testing/TestingBank/API/loggInn.php) Download Resources

Name	Value
personnummer	12345123456
passord	Passord1!

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info

**hentSaldoEnKonto**

Method: GET Request URL: [calhost/testing/TestingBank/API/hentSaldi.php](http://calhost/testing/TestingBank/API/hentSaldi.php) Download Resources

Name	Value
personnummer	12345123456

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info JMS (0)

```

1 [ {
2   "Kontonummer": "55555224444",
3   "Personnummer": "12345123456",
4   "Saldo": "3",
5   "Type": "Brukskonto",
6   "Valuta": "SEK"
7 }]
  
```

Contains - VALID

**logInn**

Method: GET Request URL: [calhost/testing/TestingBank/API/logInn.php](http://calhost/testing/TestingBank/API/logInn.php) Download Resources

Name	Value
personnummer	01010110523
passord	Passord1!

Raw Headers Attachments JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info

**Contains - VALID**

**HentSaldoTreKonto**

Method: GET Request URL: [calhost/testing/TestingBank/API/hentSaldi.php](http://calhost/testing/TestingBank/API/hentSaldi.php) Download Resources

Name	Value
personnummer	01010110523

Raw Headers Attachments JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

```

1 [
2   {
3     "Kontonummer": "105010123456",
4     "Personnummer": "01010110523",
5     "Saldo": "720",
6     "Type": "Lønnskonto",
7     "Valuta": "NOK"
8   },
9   {
10    "Kontonummer": "105020123456",
11    "Personnummer": "01010110523",
12    "Saldo": "100500",
13    "Type": "Sparekonto",
14    "Valuta": "NOK"
15  },
16  {
17    "Kontonummer": "22334412345",
18    "Personnummer": "01010110523",
19    "Saldo": "10234.5",
20    "Type": "Brukskonto",
21    "Valuta": "NOK"
22  }
23 ]

```

**SaldoCount - VALID**

## RegistrerBetaling:

logInn

Method: GET Request URL: calhost/testing/TestingBank/API/logInn.php Download F

Raw Request

Name	Value
personnummer	01010110523
passord	Passord1!

1 "OK"

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info

+ Contains - VALID

registerBetalingOk

Method: POST Request URL: testing/TestingBank/API/registerBetaling.php Download Res

Raw Request

Name	Value
tilKonto	10101334532
dato	2020-06-28
belop	100
melding	Takk for kinoturen
kontoNr	105010123456

1 "OK"

Raw XML JSON HTML Headers (11) Attachments (0) SSL Info

+ Contains-OK - VALID

Method: GET Request URL: /testing/TestingBank/API/hentBetaling.php Download Resources

Raw Request

Name	Value
personnummer	01010110523

XML JSON HTML Raw

```
1 {  
2   "TxID": "15",  
3   "FraTilKontonummer": "10101334532",  
4   "Belop": "100",  
5   "Dato": "2020-06-28",  
6   "Melding": "Takk for kinoturen",  
7   "Kontonummer": "105010123456",  
8   "Avventer": "1",  
9   "Personnummer": "01010110523",  
10  "Saldo": "720",  
11  "Type": "Lønnskonto",  
12  "Valuta": "NOK"  
13 }
```

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

JsonPath Count - VALID

Method: POST Request URL: /testing/TestingBank/API/registrerBetaling.php Download Res

Raw Request

Name	Value
tilKonto	10101334532
dato	2020-06-28
belop	tekstlBelop
melding	"Takk for kinoturen"
kontoNr	6666666666

Media Type: application/x-www-form-urlencoded

XML JSON HTML Raw

```
1 "Feil"
```

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info

Contains-Feil - VALID

## HentBetalinger:

Screenshot of a web testing tool showing a successful API call to logInnUtenBetalinger.

Request URL: `calhost/testing/TestingBank/API/logInn.php`

Method: GET

Raw Request Data:

Name	Value
personnummer	12345678901
passord	Passord1!

Response:

1	"OK"

Raw Response Data:

```
1 "OK"
```

Headers (12) Attachments (0) SSL Info

Contains – VALID

Screenshot of a web testing tool showing a successful API call to hentBetalingerUtenBetalinger.

Request URL: `/testing/TestingBank/API/hentBetalinger.php`

Method: GET

Raw Request Data:

Name	Value
personnummer	12345678901

Response:

1	<Empty JSON content>

Raw Response Data:

```
1 <Empty JSON content>
```

Headers (11) Attachments (0) SSL Info JM

Contains-TomtArray – VALID

loggInnEnBetalinger

Method: GET Request URL: /alhost/testing/TestingBank/API/loggInn.php Download Resources

Raw Request

Name	Value
personnummer	01010110523
passord	Passord1!

XML JSON Raw HTML

1 "OK"

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info

hentBetalingerEnBetalinger

Method: GET Request URL: /testing/TestingBank/API/hentBetalinger.php Download Resources

Raw Request

Name	Value
personnummer	01010110523

XML JSON Raw HTML

```
1: {  
2:   "TxID": "15",  
3:   "FraTilKontonummer": "10101334532",  
4:   "Belop": "100",  
5:   "Dato": "2020-06-28",  
6:   "Melding": "Takk for kinoturen",  
7:   "Kontonummer": "105010123456",  
8:   "Avventer": "1",  
9:   "Personnummer": "01010110523",  
10:  "Saldo": "720",  
11:  "Type": "Lønnskonto",  
12:  "Valuta": "NOK"  
13: }
```

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

JsonPath Count - VALID

Method | GET Request URL: /calhost/testing/TestingBank/API/loggInn.php Download Resources

Name	Value
personnummer	11029700112
passord	Passord1!

Raw Headers (11) Attachments (0) SSL Info

```

1 "OK"

```

Contains - VALID

Method | GET Request URL: /testing/TestingBank/API/hentBetaling.php Download Resources

Name	Value
personnummer	11029700112

Raw Headers (11) Attachments (0) SSL Info JMS (0)

```

1 {
2   "TxID": "10",
3   "FraTilKontonummer": "55555224444",
4   "Belop": "900000",
5   "Dato": "2019-12-24",
6   "Melding": "Takk for i går ;)",
7   "Kontonummer": "22222334444",
8   "Avventer": "1",
9   "Personnummer": "11029700112",
10  "Saldo": "900000",
11  "Type": "Brukskonto",
12  "Valuta": "USD"
13 },
14 {
15   "TxID": "11",
16   "FraTilKontonummer": "55555224444",
17   "Belop": "-80",
18   "Dato": "2019-12-25",
19   "Melding": "For dårlig samvittighet",
20   "Kontonummer": "22222334444",
21   "Avventer": "1",
22   "Personnummer": "11029700112",
23   "Saldo": "900000",
24   "Type": "Brukskonto",
25   "Valuta": "USD"
26 },
27 {
28   "TxID": "12",
29   "FraTilKontonummer": "105010123456",
30   "Belop": "-123",
31   "Dato": "2020-01-01",
32   "Melding": "En nyydelyig dag",
33   "Kontonummer": "22222334444",
34   "Avventer": "1",
35   "Personnummer": "11029700112",
36   "Saldo": "900000",
37   "Type": "Brukskonto",
38   "Valuta": "USD"
39 }
40

```

JsonPath Count - VALID

## UtførBetalinger:

**loggInn**

Method: GET Request URL: <calhost/testing/TestingBank/API/loggInn.php> Download

Name	Value
personnummer	11029700112
passord	Passord1!

Raw Request Headers (12) Attachments (0) S

**SjekkBetalinger**

Method: GET Request URL: <st/testing/TestingBank/API/utforBetaling.php> Download Resources

Name	Value
TxID	9

```

1   {
2     "TxID": "10",
3     "FraTilKontonummer": "55555224444",
4     "Belop": "900000",
5     "Dato": "2019-12-24",
6     "Melding": "Takk for i går ;)",
7     "Kontonummer": "22222334444",
8     "Avventer": "1",
9     "Personnummer": "11029700112",
10    "Saldo": "900000",
11    "Type": "Brukskonto",
12    "Valuta": "USD"
13  },
14  {
15    "TxID": "11",
16    "FraTilKontonummer": "55555224444",
17    "Belop": "-80",
18    "Dato": "2019-12-25",
19    "Melding": "For dårlig samvittighet",
20    "Kontonummer": "22222334444",
21    "Avventer": "1",
22    "Personnummer": "11029700112",
23    "Saldo": "900000",
24    "Type": "Brukskonto",
25    "Valuta": "USD"
26  },
27  {
28    "TxID": "12",
29    "FraTilKontonummer": "105010123456",
30    "Belop": "-123",
31    "Dato": "2020-01-01",
32    "Melding": "En nyyydelig dag",
33    "Kontonummer": "22222334444",
34    "Avventer": "1",
35    "Personnummer": "11029700112",
36    "Saldo": "900000",
37    "Type": "Brukskonto",
38    "Valuta": "USD"
39  }
40
41

```

Raw Headers (11) Attachments (0) SSL Info JMS (0)

**JsonPath Count-3 - VALID**

utforBetalingRiktig

Method: GET Request URL: st/testing/TestingBank/API/utforBetaling.php Download Resources

Name	Value
TxID	11

Raw Request Headers (11) Attachments (0) SSL Info JMS (0)

```

1 [
2   {
3     "TxID": "10",
4     "FraTilKontonummer": "55555224444",
5     "Belop": "9000000",
6     "Dato": "2019-12-24",
7     "Melding": "Takk for i går ;)",
8     "Kontonummer": "22222334444",
9     "Avventer": "1",
10    "Personnummer": "11029700112",
11    "Saldo": "9000080",
12    "Type": "Brukskonto",
13    "Valuta": "USD"
14  },
15  {
16    "TxID": "12",
17    "FraTilKontonummer": "105010123456",
18    "Belop": "-123",
19    "Dato": "2020-01-01",
20    "Melding": "En nyydelig dag",
21    "Kontonummer": "22222334444",
22    "Avventer": "1",
23    "Personnummer": "11029700112",
24    "Saldo": "9000080",
25    "Type": "Brukskonto",
26    "Valuta": "USD"
27  }
28 ]

```

JsonPath Count-2 - VALID

utforBetaling-Feil

Method: GET Request URL: st/testing/TestingBank/API/utforBetaling.php Download Resources

Name	Value
TxID	9

Raw Request Headers (11) Attachments (0) SSL Info JMS (0)

```

1 [
2   {
3     "TxID": "10",
4     "FraTilKontonummer": "55555224444",
5     "Belop": "9000000",
6     "Dato": "2019-12-24",
7     "Melding": "Takk for i går ;)",
8     "Kontonummer": "22222334444",
9     "Avventer": "1",
10    "Personnummer": "11029700112",
11    "Saldo": "9000080",
12    "Type": "Brukskonto",
13    "Valuta": "USD"
14  },
15  {
16    "TxID": "12",
17    "FraTilKontonummer": "105010123456",
18    "Belop": "-123",
19    "Dato": "2020-01-01",
20    "Melding": "En nyydelig dag",
21    "Kontonummer": "22222334444",
22    "Avventer": "1",
23    "Personnummer": "11029700112",
24    "Saldo": "9000080",
25    "Type": "Brukskonto",
26    "Valuta": "USD"
27  }
28 ]

```

JsonPath Count-2 - VALID

## HentKundeInfo:

loggInnKunde1

Method: GET Request URL: calhost/testing/TestingBank/API/loggInn.php Download Resources

Raw Request

Name	Value
personnummer	01010110523
passord	Passord1!

XML JSON HTML Raw Headers (12) Attachments (0) SSL Info

+ Contains - VALID

The screenshot shows a SoapUI test step named 'loggInnKunde1'. It's a GET request to 'calhost/testing/TestingBank/API/loggInn.php'. The request payload contains two parameters: 'personnummer' with value '01010110523' and 'passord' with value 'Passord1!'. The response is a single line 'OK' in yellow, indicating a successful login.

hentKundeInfoOK

Method: GET Request URL: /testing/TestingBank/API/hentKundeInfo.php Download Resources

Raw Request

Name	Value
personnummer	01010110523

XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

+ Contains - VALID

The screenshot shows a SoapUI test step named 'hentKundeInfoOK'. It's a GET request to '/testing/TestingBank/API/hentKundeInfo.php'. The request payload contains one parameter: 'personnummer' with value '01010110523'. The response is a JSON object containing customer details:

```
1 {  
2   "personnummer": "01010110523",  
3   "fornavn": "Lene",  
4   "etternavn": "Jensen",  
5   "adresse": "Askerveien 22",  
6   "postnr": "3270",  
7   "poststed": "Asker",  
8   "telefonnr": "22224444",  
9   "passord": "Passord1!"  
10 }
```

The response includes 11 headers and no attachments or SSL info.

**loggInnKunde2**

Method: GET Request URL: /calhost/testing/TestingBank/API/loggInn.php Download R

Name	Value
personnummer	12345123456
passord	Passord1!

Raw Request Headers (11) Attachments (0) SSL In

**hentKundeInfoFeil**

Method: GET Request URL: /testing/TestingBank/API/hentKundeInfo.php Download R

Name	Value
personnummer	12345123456

Raw Request Headers (11) Attachments (0) SSL In

Contains - VALID

Contains-Feil - VALID

## EndreKundelInfo:

Screenshot of a REST client interface showing two requests: "loggInn" and "EndreGyldigInfo".

**Request 1: loggInn (GET)**

Method: GET | Request URL: `calhost/testing/TestingBank/API/loggInn.php` | Download

Raw Request Data:

Name	Value
personnummer	01010110523
passord	Passord1!

Response Data (Raw):

1	"OK"

Validation Status: Contains - VALID

**Request 2: EndreGyldigInfo (POST)**

Method: POST | Request URL: `testing/TestingBank/API/endreKundelInfo.php` | Download

Raw Request Data:

Name	Value
personnummer	01010110532
fornavn	Line
etternavn	Jensen Hansen
adresse	Askerveien 33
postnr	3270
telefonnr	33334444
poststed	Asker
passord	Passord1!

Media Type: application/x-www-form-urlencoded

Response Data (Raw):

1	"OK"

Validation Status: Contains-OK - VALID

EndreGyldig-postnr

Method: POST Request URL: testing/TestingBank/API/endreKundeInfo.php Download Re

Request Raw

Name Value

personnummer	01010110532
fornavn	Lene
etternavn	Hansen
adresse	Osloveien 22
postnr	0565
telefonnr	22224444
poststed	Oslo
passord	Passord1!

Media Type: application/x-www-form-urlencoded

Headers (11) Attachments (0) SSL Info

Raw XML JSON HTML

1 "OK"

Contains-OK - VALID

EndreKundeInfoMedFeilPostnr

Method: POST Request URL: testing/TestingBank/API/endreKundeInfo.php Download Resource

Request Raw

Name Value

personnummer	01010110532
fornavn	Lene
etternavn	Jensen
adresse	Askerveien 22
postnr	32701
telefonnr	22224444
poststed	Oslo
passord	Passord1!

Media Type: application/x-www-form-urlencoded

Headers (11) Attachments (0) SSL Info

Raw XML JSON HTML

1 "Feil"

Contains-Feil - VALID

EndreKundeInfoMedFeilTelefonnr

Method POST Request URL: testing/TestingBank/API/endreKundeInfo.php Download Resources

**Request**

**Raw**

Name	Value
personnummer	01010110532
fornavn	Lene
etternavn	Jensen
adresse	Askerveien 22
postnr	3270
telefonnr	222244445
poststed	Asker

Media Type: application/x-www-form-urlencoded

**Response**

1 "OK"

**Raw** **HTML** **XML**

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (

Contains-Feil - FAILED  
-> Missing token [Feil] in Response

## LoggInn:

The screenshot shows the LoggInn tool interface. The request URL is `calhost/testing/TestingBank/API/loggInn.php`. The method is GET. The request body contains two parameters: `personnummer` with value `12345123456` and `passord` with value `Passord1!`. The response status is "OK".

Name	Value
personnummer	12345123456
passord	Passord1!

Response:

```
1 "OK"
```

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (12) Attachments (0) S

+ ⚙ X ^ v

● Contains – VALID

The screenshot shows the LoggInn tool interface. The request URL is `ilhost/testing/TestingBank/API/hentSaldi.php`. The method is GET. The request body contains one parameter: `personnummer` with value `12345123456`. The response is a JSON object containing account details.

Name	Value
personnummer	12345123456

Response:

```
1 {
  "Kontonummer": "55555224444",
  "Personnummer": "12345123456",
  "Saldo": "1",
  "Type": "Brukskonto",
  "Valuta": "SEK"
}
```

Au... Headers ... Attachments ... JMS Head... JMS Properties... Headers (11) Attachments (0) SSL Info JMS (0)

+ ⚙ X ^ v

● Saldo – VALID

Screenshot of a web browser or API testing tool showing a successful JSON response.

The interface includes:

- Toolbar with icons for back, forward, search, and refresh.
- Header bar with "Logout" button.
- Method dropdown set to "GET".
- Request URL: "localhost/testing/TestingBank/API/logout.php".
- Download Resources button.
- Left panel: "Request" tab selected, showing an empty table for parameters.
- Right panel: "JSON" tab selected, displaying the response content:

```
1 The content you are trying to view cannot be viewed as JSON
```
- Bottom navigation tabs: Headers (8), Attachments (0), SSL Info, JMS (0).
- Status bar at the bottom: "Contains-EMPTY - VALID".

## Admin funksjoner

### AdminLoggInn:

The screenshot shows the SoapUI interface with a successful API call to `InitDB.php`. The request method is GET, and the response body contains the string "OK".

**Request Details:**

- Method: GET
- Request URL: `http://localhost/testing/TestingBank/InitDB.php`
- Raw Request Data (Table):

Name	Value

**Response Body:**

1	"OK"

**Assertions:** Contains-OK – VALID

The screenshot shows the SoapUI interface with a successful login request to `loggInnAdmin.php`. The request method is GET, and the response body contains the string "Admin".

**Request Details:**

- Method: GET
- Request URL: `http://localhost/testing/TestingBank/API/loggInnAdmin.php`
- Raw Request Data (Table):

Name	Value
bruker	Admin
passord	Admin

**Response Body:**

1	"Admin"

**Assertions:** AdminLoggInn-OK – VALID

AdminLogInn-FEIL

Method: GET Request URL: /testing/TestingBank/API/loggInnAdmin.php Download Resources

Request

Name Value  
bruker bruker  
passord feil

Raw XML JSON HTML

1 "Feil i innlogging"

Headers (12) Attachments (0) SSL Info JMS (0)

Online Help

AdminLogInn-FEIL - VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a SoapUI test case titled 'AdminLogInn-FEIL'. The request pane displays a GET request to 'loggInnAdmin.php' with parameters 'bruker' set to 'bruker' and 'passord' set to 'feil'. The response pane shows the JSON result: '1 "Feil i innlogging"'. Below the request pane, tabs for Headers, Attachments, SSL Info, and JMS are visible. The status bar at the bottom indicates '1 : 59'.

### AdminEndreKonto:

LoggInnAdmin

Method: GET Request URL: /testing/TestingBank/API/loggInnAdmin.php Download Resources

Request

Name Value  
bruker Admin  
passord Admin

Raw XML JSON HTML

1 "Admin"

Headers (12) Attachments (0) SSL Info JMS (0)

Online Help

Admin OK - VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a SoapUI test case titled 'LoggInnAdmin'. The request pane displays a GET request to 'loggInnAdmin.php' with parameters 'bruker' set to 'Admin' and 'passord' set to 'Admin'. The response pane shows the JSON result: '1 "Admin"'. Below the request pane, tabs for Headers, Attachments, SSL Info, and JMS are visible. The status bar at the bottom indicates '1 : 59'.

EndrePersonnummer-FEIL

Method: POST Request URL: sting/TestingBank/API/adminEndreKonto.php Download Resources

Request

Name	Value
perso...	0000...
konto...	1050...
type	Lønns...
saldo	720
valuta	NOK

Media Type: application/json

Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-FeilPersonnummer – VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a POST request to 'adminEndreKonto.php'. The request body is a JSON object with a single key-value pair: '1' and the string '"Feil i personnummer"'. The response status is 'VALID'.

EndreKontonummer-Feil

Method: POST Request URL: sting/TestingBank/API/adminEndreKonto.php Download Resources

Request

Name	Value
perso...	0101...
konto...	0000...
type	Lønns...
saldo	720
valuta	NOK

Media Type: application/json

Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-FeilKontonummer – VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a POST request to 'adminEndreKonto.php'. The request body is a JSON object with a single key-value pair: '1' and the string '"Feil i kontonummer"'. The response status is 'VALID'.

GodkjentEndring

Method: POST Request URL: sting/TestingBank/API/adminEndreKonto.php Download Resources

Request Raw

Name	Value
personnummer	01010110523
kontonummer	105010123456
type	Lønnskonto
saldo	820
valuta	NOK

Media Type: application/x-www-form-urlencoded

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains - VALID

Assertions (1) Request Log (0)

This screenshot shows a successful API call in SoapUI. The request pane contains a table with fields like personnummer, kontonummer, type, saldo, and valuta. The response pane shows a single JSON object with the value "OK". The status bar at the bottom right indicates "1 : 59".

EndringAvSaldo

Method: POST Request URL: sting/TestingBank/API/adminEndreKonto.php Download Resources

Request Raw

Name	Value
pers...	010...
kont...	105...
type	Lønn...
saldo	test

Media Type: application/x-www-form-urlencoded

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains - FAILED  
-> Missing token [Feil] in Response

Assertions (1) Request Log (0)

This screenshot shows an failed API call in SoapUI. The request pane is similar to the first one but with a different saldo value. The status bar at the bottom right indicates "1 : 59". A red error message at the bottom states "Contains - FAILED" and "-> Missing token [Feil] in Response".

EndringAvValuta

Method: POST Request URL: sting/TestingBank/API/adminEndreKonto.php Download Resources

**Request**

Name	Value
pers...	010...
kont...	105...
type	Lønn...
saldo	820

Media Type: application/json

**Response**

```
1 "OK"
```

Headers (11) Attachments (0) SSL Info JMS (0)

Contains - FAILED  
-> Missing token [Feil] in Response

Assertions (1) Request Log (0) 1 : 59

## AdminEndreKunde

LoggInnAdmin-OK

Method: GET Request URL: t/testing/TestingBank/API/loggInnAdmin.php Download Resources

**Request**

Name	Value
buker	Admin
passord	Admin

**Response**

```
1 "Admin"
```

Headers (12) Attachments (0) SSL Info JMS (0)

AdminLoggInn-OK - VALID

Assertions (1) Request Log (0) 1 : 59

Method: POST Request URL: sting/TestingBank/API/adminEndreKunde.php Download Resources

Name	Value
pers...	010...
forna...	Lene
etter...	Jens...
adre...	Oslo...
postnr	0560

Media Type: application/json

```

1 "OK"

```

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

Method: POST Request URL: sting/TestingBank/API/adminEndreKunde.php Download Resources

Name	Value
pers...	010...
forna...	Lene
etter...	Jensen
adre...	Aske...
postnr	3270

Media Type: application/json

```

1 <br />
2 <b>Notice</b>: Undefined index: telefonnr in <b>/Applications/XAMPP/xamppf...
3 "OK"

```

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

EndreKundeInfoFeilPostnummer

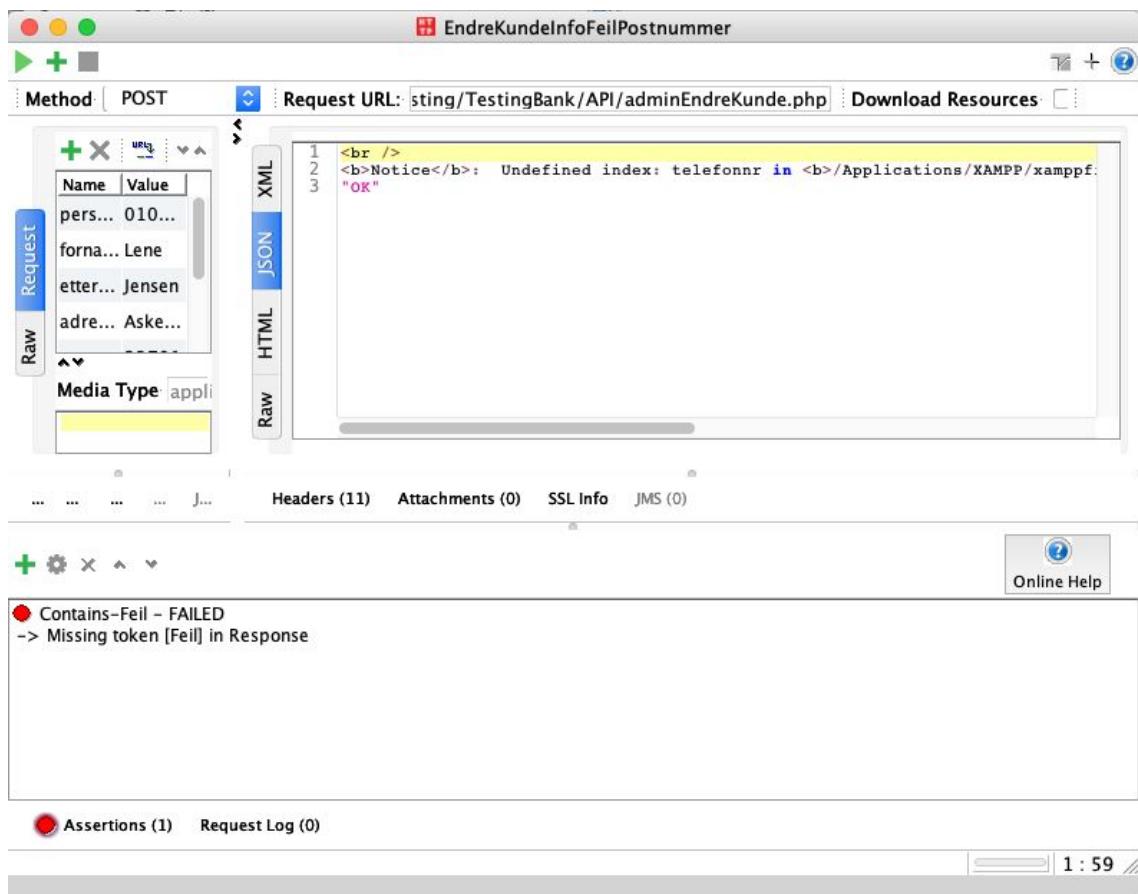
Method: POST Request URL: sting/TestingBank/API/adminEndreKunde.php Download Resources

Raw Request XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil – FAILED  
-> Missing token [Feil] in Response

Assertions (1) Request Log (0) 1 : 59



EndreKundeInfoFeilTelefonnummer

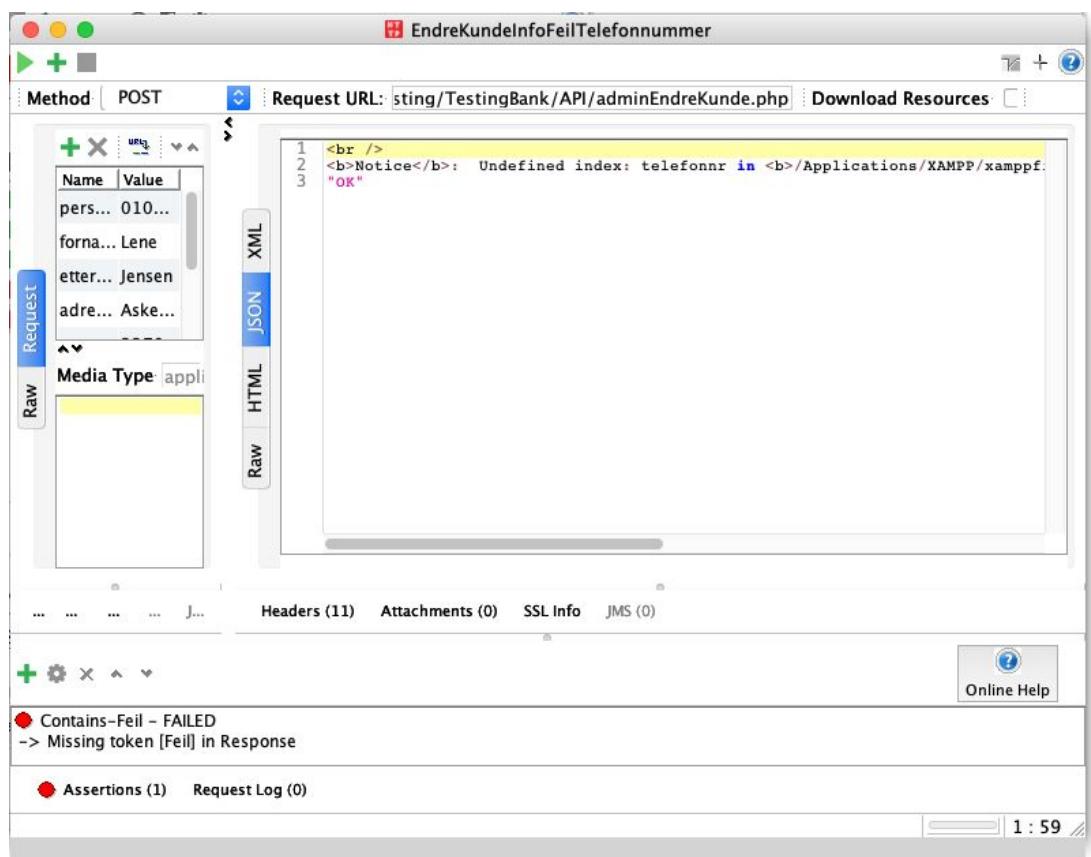
Method: POST Request URL: sting/TestingBank/API/adminEndreKunde.php Download Resources

Raw Request XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil – FAILED  
-> Missing token [Feil] in Response

Assertions (1) Request Log (0) 1 : 59



## AdminHentKonti:

The screenshot shows the SoapUI interface for a successful API request. The top bar displays the title "LoggInnAdmin-OK". The "Method" dropdown is set to "GET" and the "Request URL" is "/testing/TestingBank/API/loggInnAdmin.php". The "Raw" tab under the "Request" pane shows the following XML payload:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>
<Auth>
<Name>bruker</Name>
<Value>Admin</Value>
</Auth>
<Auth>
<Name>passord</Name>
<Value>Admin</Value>
</Auth>
</soap:Header>
<soap:Body>
<Admin></Admin>
</soap:Body>
</soap:Envelope>
```

The "Response" pane shows the JSON response:

```
[{"name": "Admin"}]
```

The status bar at the bottom right indicates the time is 1:59.

The screenshot shows the SoapUI interface for a successful API request to get all accounts. The top bar displays the title "AdminHentAlleKonti". The "Method" dropdown is set to "GET" and the "Request URL" is "/testing/TestingBank/API/adminHentKonti.php". The "Raw" tab under the "Request" pane shows the following XML payload:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>
<Auth>
<Name>unq</Name>
<Value>Admin</Value>
</Auth>
<Auth>
<Name>passord</Name>
<Value>Admin</Value>
</Auth>
</soap:Header>
<soap:Body>
<AdminHentAlleKonti>
</AdminHentAlleKonti>
</soap:Body>
</soap:Envelope>
```

The "Response" pane shows the JSON response:

```
[{"Kontonummer": "105010123456", "Personnummer": "01010110523", "Saldo": "820", "Type": "L\u00e4nnskonto", "Valuta": "NOK"}, {"Kontonummer": "105020123456", "Personnummer": "01010110523", "Saldo": "100500", "Type": "Sparekonto", "Valuta": "NOK"}, {"Kontonummer": "222223344444", "Personnummer": "01010110523", "Saldo": "100000", "Type": "Sparekonto", "Valuta": "NOK"}]
```

The status bar at the bottom right indicates the time is 1:59.

AdminSlettKonto1/7

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request

Raw

Name Value  
konto... 1050...

Media Type application/json

XML

JSON

HTML

Raw

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows the SoapUI interface for a DELETE request to the endpoint AdminSlettKonto.php. The request payload contains a single key-value pair: 'konto...' with value '1050...'. The response is a simple JSON object with a single key '1' and value '"OK"'. The Assertions panel indicates that the response contains the string 'OK' and is valid. The Request Log panel shows one entry.

AdminSlettKonto2/7

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request

Raw

Name Value  
konto... 1050...

Media Type application/json

XML

JSON

HTML

Raw

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows the SoapUI interface for a second DELETE request to the same endpoint. The setup and response are identical to the first one, with a single key-value pair in the payload and a response containing '1 "OK"'. The Assertions panel again confirms the presence of 'OK' in the response. The Request Log panel shows one entry.

AdminSlettKonto3/7

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request Raw

Name Value  
konto... 2233...

Media Type application/json

XML JSON HTML Raw

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows the SoapUI interface for a DELETE request to the endpoint 'adminSlettKonto.php'. The request payload contains a single parameter 'konto...' with the value '2233...'. The response is a simple JSON object with a single key '1' and the value '"OK"'. The status bar at the bottom indicates the time as '1 : 59'.

AdminSlettKonto4/7

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request Raw

Name Value  
konto... 2222...

Media Type application/json

XML JSON HTML Raw

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows the SoapUI interface for a DELETE request to the endpoint 'adminSlettKonto.php'. The request payload contains a single parameter 'konto...' with the value '2222...'. The response is a simple JSON object with a single key '1' and the value '"OK"'. The status bar at the bottom indicates the time as '1 : 59'.

Method: GET Request URL: testing/TestingBank/API/adminHentKonti.php Download Resources

```

1 [
2   {
3     "Kontonummer": "33333662222",
4     "Personnummer": "11029700112",
5     "Saldo": "45",
6     "Type": "Sparekonto",
7     "Valuta": "USD"
8   },
9   {
10    "Kontonummer": "44444992222",
11    "Personnummer": "11029700112",
12    "Saldo": "200",
13    "Type": "Lånskonto",
14    "Valuta": "USD"
15  },
16  {
17    "Kontonummer": "55555224444",
18    "Personnummer": "12345123456",
19    "Saldo": "3",
20    "Type": "Brukskonto",
21    "Valuta": "SEK"
22  }
23 ]

```

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

TreKontoer - VALID  
TreKontoer-Count - VALID

Assertions (2) Request Log (0)

Method: DELETE Request URL: testing/TestingBank/API/adminSlettKonto.php Download Resources

```

1 "OK"

```

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

AdminSlettKonto6/7

Method: DELETE Request URL: testing/TestingBank/API/adminSlettKonto.php Download Resources

Request

Raw

Media Type application/json

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK – VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows the SoapUI interface with a successful DELETE request to 'AdminSlettKonto.php'. The response body contains the string 'OK'.

AdminHentEnKonti

Method: GET Request URL: testing/TestingBank/API/adminHentKonti.php Download Resources

Request

Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

EnKonto – VALID

EnKonto-Count – VALID

Assertions for this request

Assertions (2) Request Log (0)

1 : 59

The screenshot shows the SoapUI interface with a successful GET request to 'AdminHentKonti.php'. The response body is a JSON array containing one object:

```
[{"Kontonummer": "55555224444", "Personnummer": "12345123456", "Saldo": "3", "Type": "Brukskonto", "Valuta": "SEK"}]
```

AdminSlettKonto7/7

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Raw Request

Name Value  
konto... 5555...

Media Type application/json

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a SoapUI test step named 'AdminSlettKonto7/7'. The request method is set to 'DELETE' and the URL is 'esting/TestingBank/API/adminSlettKonto.php'. The request body contains a single parameter 'konto...' with the value '5555...'. The media type is specified as 'application/json'. The response body is displayed in the JSON tab, showing the string 'OK' in red. Below the request and response panes, there are tabs for Headers (11), Attachments (0), SSL Info, and JMS (0). The status bar at the bottom right indicates '1 : 59'. The Assertions section shows one assertion labeled 'Contains-OK - VALID'.

AdminHentIngenKonti

Method: GET Request URL: esting/TestingBank/API/adminHentKonti.php Download Resources

Raw Request

Name Value

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

IngenKontoer - VALID  
IngenKontoer-Count - VALID

Assertions (2) Request Log (0)

1 : 59

The screenshot shows a SoapUI test step named 'AdminHentIngenKonti'. The request method is set to 'GET' and the URL is 'esting/TestingBank/API/adminHentKonti.php'. The request body is empty. The response body is displayed in the JSON tab, showing '<Empty JSON content>' in yellow. Below the request and response panes, there are tabs for Headers (11), Attachments (0), SSL Info, and JMS (0). The status bar at the bottom right indicates '1 : 59'. The Assertions section shows two assertions labeled 'IngenKontoer - VALID' and 'IngenKontoer-Count - VALID'.

InitDB

Method: GET Request URL: http://localhost/testing/TestingBank/InitDB.php Download Resources

Raw Request

Name	Value
------	-------

XML JSON HTML Raw Headers (8) Attachments (0) SSL Info JMS (0)

Contains - VALID

Assertions (1) Request Log (0) 1 : 59

The screenshot shows the SoapUI interface with a successful response for the 'InitDB' API. The response body contains the string "OK". The status bar at the bottom indicates the assertion was successful ("Contains - VALID").

### AdminRegisterKunde:

LoggInnAdmin-OK

Method: GET Request URL: http://localhost/testing/TestingBank/API/loggInnAdmin.php Download Resources

Raw Request

Name	Value
bruker	Admin
passord	Admin

XML JSON HTML Raw Headers (12) Attachments (0) SSL Info JMS (0)

Online Help

AdminLoggInn-OK - VALID

Assertions (1) Request Log (0) 1 : 59

The screenshot shows the SoapUI interface with a successful response for the 'AdminRegisterKunde' API. The response body contains the string "Admin". The status bar at the bottom indicates the assertion was successful ("AdminLoggInn-OK - VALID").

AdminRegistrerKundeRiktigpostNr

Method POST Request URL: g/TestingBank/API/adminRegistrerKunde.php Download Resources

Request Raw

Name	Value
pers...	010...
forna...	Hans
etter...	Hans...
adre...	Ikaro...
postnr	2040

Media Type application/json

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK – VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test case titled 'AdminRegistrerKundeRiktigpostNr'. The request pane displays a POST method with a JSON payload containing customer details like name, surname, address, and postal code. The response pane shows a single JSON object with the value 'OK'. The status bar at the bottom indicates the test was valid ('Contains-OK – VALID') and took 1 second and 59 milliseconds.

AdminRegistrerKundeEksisterendePostNr

Method POST Request URL: g/TestingBank/API/adminRegistrerKunde.php Download Resources

Request Raw

Name	Value
pers...	010...
forna...	Bjorn
etter...	Stjerne
adre...	Karl ...
postnr	2040

Media Type application/json

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK – VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test case titled 'AdminRegistrerKundeEksisterendePostNr'. The setup is identical to the first one, but the surname ('forna...') is changed to 'Bjorn'. The response is still 'OK', indicating the test was valid ('Contains-OK – VALID').

AdminRegistrerKundeNyFeilPostNr

Method: POST Request URL: g/TestingBank/API/adminRegistrerKunde.php Download Resources

Request

Raw

Name Value  
pers... 121...  
forna... Sindre  
etter... Suut  
adre... Bobl...  
postnr 123...  
Media Type application/json

XML

1 "Feil"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test step for the endpoint 'g/TestingBank/API/adminRegistrerKunde.php'. The request method is POST. The request body is a JSON object with fields: pers... (121...), forna... (Sindre), etter... (Suut), adre... (Bobl...), and postnr (123...). The XML tab shows the response body: 1 "Feil". The status bar at the bottom right indicates the time is 1:59.

AdminRegistrerKundEksPostNrFeilTelefonNr

Method: POST Request URL: g/TestingBank/API/adminRegistrerKunde.php Download Resources

Request

Raw

Name Value  
pers... 060...  
forna... Kjartan  
etter... Kjuus  
adre... Ned...  
postnr 1234  
Media Type application/json

XML

1 "Feil"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test step for the endpoint 'g/TestingBank/API/adminRegistrerKunde.php'. The request method is POST. The request body is a JSON object with fields: pers... (060...), forna... (Kjartan), etter... (Kjuus), adre... (Ned...), and postnr (1234). The XML tab shows the response body: 1 "Feil". The status bar at the bottom right indicates the time is 1:59.

## AdminRegistrerKonto:

The screenshot shows the SoapUI interface with a successful login request. The request URL is `t/testing/TestingBank/API/loggInnAdmin.php`. The method is GET, and the request body contains the following parameters:

Name	Value
bruker	Admin
passord	Admin

The response body shows the result of the login:

```
1 "Admin"
```

Assertions (1) Request Log (0)

The screenshot shows the SoapUI interface with a request to list all accounts. The request URL is `t/testing/TestingBank/API/adminHentKonti.php`. The method is GET, and the request body is empty.

The response body displays a JSON array of account details:

```
1 [
2   {
3     "Kontonummer": "105010123456",
4     "Personnummer": "01010110523",
5     "Saldo": "720",
6     "Type": "Lønnskonto",
7     "Valuta": "NOK"
8   },
9   {
10    "Kontonummer": "105020123456",
11    "Personnummer": "01010110523",
12    "Saldo": "100500",
13    "Type": "Sparekonto",
14    "Valuta": "NOK"
15  },
16  {
17    "Kontonummer": "22222334444",
18    "Personnummer": "11029700112",
19    "Saldo": "9000000",
20    "Type": "Brukskonto",
21    "Valuta": "USD"
22  },
23]
```

Assertions (2) Request Log (0)

**AdminRegistrerKonto**

Method: POST Request URL: /TestingBank/API/adminRegistrerKonto.php Download Resources

Raw Request

Name	Value
perso...	1234...
konto...	6666...
type	Bruks...
valuta	NOK

Media Type application/json

XML JSON HTML Raw

```
1 "OK"
```

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0) 1 : 59

**AdminHentAlleKonti=8**

Method: GET Request URL: /TestingBank/API/adminHentKonti.php Download Resources

Raw Request

Name	Value
------	-------

XML JSON HTML Raw

```

1 [
2   {
3     "Kontonummer": "105010123456",
4     "Personnummer": "01010110523",
5     "Saldo": "720",
6     "Type": "Lønnskonto",
7     "Valuta": "NOK"
8   },
9   {
10    "Kontonummer": "105020123456",
11    "Personnummer": "01010110523",
12    "Saldo": "100500",
13    "Type": "Sparekonto",
14    "Valuta": "NOK"
15  },
16  {
17    "Kontonummer": "222223344444",
18  }
]

```

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

AlleKontoer - VALID  
AlleKontoer-Count - VALID

Assertions (2) Request Log (0) 1 : 59

AdminRegistrerKontoFeilValuta

Method: POST Request URL: ig/TestingBank/API/adminRegistrerKonto.php Download Resources

Request Raw

Name Value  
perso... 1234...  
konto... 9876...  
type Spare...  
valuta Kroner

Media Type application/json

XML JSON HTML Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows a SoapUI test step named "AdminRegistrerKontoFeilValuta". The request pane displays a POST method with a JSON payload containing four parameters: perso, konto, type, and valuta, all set to their respective values. The response pane shows a single JSON object with the value "Feil". Below the main interface, status indicators show one assertion passed and no requests in the log.

AdminRegistrerKontoFeilKontonummer

Method: POST Request URL: ig/TestingBank/API/adminRegistrerKonto.php Download Resources

Request Raw

Name Value  
perso... 1234...  
konto... 1234...  
type Spare...  
valuta NOK

Media Type application/json

XML JSON HTML Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot is identical to the one above, showing the same test step and results. It demonstrates that the API returns "Feil" regardless of the konto number provided in the request.

AdminRegistrerKonto ikke Registrert i Bank

Method: POST Request URL: /TestingBank/API/adminRegistrerKonto.php Download Resources

Raw Request

Name	Value
perso...	6666...
konto...	1234...
type	Spare...
valuta	NOK

Media Type: application/json

XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - VALID

Assertions (1) Request Log (0) 1 : 59

The screenshot shows a SoapUI test case titled 'AdminRegistrerKonto ikke Registrert i Bank'. The request method is POST, and the URL is /TestingBank/API/adminRegistrerKonto.php. The request body contains four parameters: perso... (6666...), konto... (1234...), type (Spare...), and valuta (NOK). The response body is displayed in JSON format, showing the error message 'Feil i personnummer' (Error in person number). The status bar at the bottom indicates 'Contains-Feil - VALID'.

### AdminSlettKonto:

LoggInnAdmin-OK

Method: GET Request URL: /testing/TestingBank/API/loggInnAdmin.php Download Resources

Raw Request

Name	Value
bruker	Admin
passord	Admin

XML JSON HTML Raw Headers (12) Attachments (0) SSL Info JMS (0)

Online Help

AdminLoggInn-OK ~ VALID

Assertions (1) Request Log (0) 1 : 59

The screenshot shows a SoapUI test case titled 'LoggInnAdmin-OK'. The request method is GET, and the URL is /testing/TestingBank/API/loggInnAdmin.php. The request body contains two parameters: bruker (Admin) and passord (Admin). The response body is displayed in JSON format, showing the message 'Admin'. The status bar at the bottom indicates 'AdminLoggInn-OK ~ VALID'.

**AdminSlettKontoRiktig**

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request

Name Value  
konto... 6666...

Media Type application/json

Raw XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0) 1 : 59

This screenshot shows the SoapUI interface for a successful DELETE request. The 'Request' tab displays a JSON payload with a single entry 'konto...' set to '6666...'. The 'Response' tab shows a single JSON object with the value 'OK'. The status bar at the bottom indicates 'Contains-OK - VALID'.

**AdminSlettKontoFeil**

Method: DELETE Request URL: esting/TestingBank/API/adminSlettKonto.php Download Resources

Request

Name Value  
konto... 9999...

Media Type application/json

Raw XML JSON HTML Raw Headers (11) Attachments (0) SSL Info JMS (0)

Contains-Feil - VALID

Assertions (1) Request Log (0) 1 : 59

This screenshot shows the SoapUI interface for an unsuccessful DELETE request. The 'Request' tab displays a JSON payload with a single entry 'konto...' set to '9999...'. The 'Response' tab shows a single JSON object with the value 'Feil kontonummer'. The status bar at the bottom indicates 'Contains-Feil - VALID'.

## AdminSlettKunde:

Screenshot of a web testing tool interface showing a successful login request.

**Request URL:** t/testing/TestingBank/API/loggInnAdmin.php

**Method:** GET

**Raw Request Data:**

Name	Value
bruker	Admin
passord	Admin

**Response Body:**

1	"Admin"

**Assertions:** 1 Assertion passed (VALID).

Screenshot of a web testing tool interface showing a successful delete request to delete a customer.

**Request URL:** t/testing/TestingBank/API/adminSlettKunde.php

**Method:** DELETE

**Raw Request Data:**

Name	Value
perso...	0103...

**Media Type:** application/json

**Response Body:**

1	"OK"

**Assertions:** 1 Assertion passed (VALID).

**AdminSlettKundeFeil**

Method: DELETE Request URL: /testing/TestingBank/API/adminSlettKunde.php Download Resources

Request

Name	Value
perso...	9999...

Media Type: application/json

Response

```
1 "OK"
```

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-Feil - FAILED  
-> Missing token [Feil] in Response

Assertions (1) Request Log (0)

1 : 59

### HentAlleKunder:

**LoggInnAdmin-OK**

Method: GET Request URL: /testing/TestingBank/API/loggInnAdmin.php Download Resources

Request

Name	Value
bruker	Admin
passord	Admin

Response

```
1 "Admin"
```

Headers (12) Attachments (0) SSL Info JMS (0)

Online Help

Contains - VALID

Assertions (1) Request Log (0)

1 : 59

HentAlleKunder=5

Method: GET Request URL: /testing/TestingBank/API/hentAlleKunder.php Download Resources

Request

Raw XML JSON

Raw Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

AlleKunder – VALID  
AlleKunder-Count – VALID

Assertions (2) Request Log (0)

```
1 [ {  
2     "Personnummer": "01010110523",  
3     "Fornavn": "Lene",  
4     "Etternavn": "Jensen",  
5     "Adresse": "Askerveien 22",  
6     "Postnr": "3270",  
7     "Telefonnr": "22224444",  
8     "Passord": "Passord1!",  
9     "Poststed": "Asker"  
10 },  
11 {  
12     "Personnummer": "01020312345",  
13     "Fornavn": "Hans",  
14     "Etternavn": "Hansen",  
15     "Adresse": "Tkarosveien 73",  
16     "Postnr": "2040",  
17 }
```

AdminSlettKunde1

Method: DELETE Request URL: /testing/TestingBank/API/adminSlettKunde.php Download Resources

Request

Raw XML JSON

Raw Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK – VALID

Assertions (1) Request Log (0)

```
1 "OK"
```

AdminSlettKunde2

Method: DELETE Request URL: testing/TestingBank/API/adminSlettKunde.php Download Resources

Request

Raw

Media Type: application/json

XML

JSON

HTML

Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

The screenshot shows the SoapUI interface with a successful DELETE request to 'adminSlettKunde.php'. The response body contains the string 'OK'.

HentAlleKunder=3

Method: GET Request URL: testing/TestingBank/API/hentAlleKunder.php Download Resources

Request

Raw

Raw

XML

JSON

HTML

Raw

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

AlleKunder - VALID  
AlleKunder-Count - VALID

Assertions (2) Request Log (0)

1 : 59

The screenshot shows the SoapUI interface with a successful GET request to 'hentAlleKunder.php'. The response body is a JSON array containing two objects, each representing a customer record:

```
[{"Personnummer": "11029700112", "Fornavn": "Zak", "Etternavn": "Ibrahim", "Adresse": "Nesveien", "Postnr": "1440", "Telefonnr": "12020509", "Passord": "Passord1!", "Poststed": "Nesoya"}, {"Personnummer": "12345123456", "Fornavn": "Johan", "Etternavn": "Olsson", "Adresse": "Svenskevegen", "Postnr": null}]
```

AdminSlettKunde3

Method: DELETE Request URL: testing/TestingBank/API/adminSlettKunde.php Download Resources

Raw Request

Name Value  
person... 12345...

Media Type application

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test step named 'AdminSlettKunde3'. The request pane displays a DELETE method with a single parameter 'person...' set to '12345...'. The response pane shows the JSON result '1 "OK"'. Below the main interface, a status bar indicates '1 : 59'.

AdminSlettKunde4

Method: DELETE Request URL: testing/TestingBank/API/adminSlettKunde.php Download Resources

Raw Request

Name Value  
person... 12345...

Media Type application

1 "OK"

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

Contains-OK - VALID

Assertions for this request

Assertions (1) Request Log (0)

1 : 59

This screenshot shows a SoapUI test step named 'AdminSlettKunde4'. It has the same configuration as the previous step, including the DELETE method and the parameter 'person...' set to '12345...'. The response pane shows the JSON result '1 "OK"'. A tooltip 'Assertions for this request' is visible above the assertions section. Below the main interface, a status bar indicates '1 : 59'.

HentAlleKunder=1

Method | GET Request URL: /testing/TestingBank/API/hentAlleKunder.php Download Resources

Raw Request XML JSON HTML

```
1 [ {  
2     "Personnummer": "11029700112",  
3     "Fornavn": "Zak",  
4     "Etternavn": "Ibrahim",  
5     "Adresse": "Nesveien",  
6     "Postnr": "1440",  
7     "Telefonnr": "12020569",  
8     "Passord": "Passord1!",  
9     "Poststed": "Nesoya"  
10 } ]
```

Headers (11) Attachments (0) SSL Info JMS (0)

AlleKunder - VALID  
AlleKunder-Count - VALID

Assertions (2) Request Log (0)

AdminSlettKunde5

Method | DELETE Request URL: /testing/TestingBank/API/adminSlettKunde.php Download Resources

Raw Request XML JSON HTML

```
1 "OK"
```

Headers (11) Attachments (0) SSL Info JMS (0)

Contains-OK - VALID

Assertions (1) Request Log (0)

HentAlleKunder=

Method: GET Request URL: /testing/TestingBank/API/hentAlleKunder.php Download Resources

Raw Request

1 <Empty JSON content>

Headers (11) Attachments (0) SSL Info JMS (0)

Online Help

AlleKunder - VALID  
AlleKunder-Count - VALID

Assertions for this request

Assertions (2) Request Log (0) 1 : 59

### 10.3.3. Systemtest

#### 10.3.3.1. Katalon Recorder

##### Oversikt:

The screenshot shows the Katalon Recorder interface with the title "Test Suites" at the top. Below it is a list of test cases under the suite "Systemtest\_TestingBank". The test cases are listed as green bullet points:

- InitDB
- LoggInn
- LoggUt
- LoggInn
- Transaksjoner - se kontonr
- Transaksjoner - alle betalinger
- Transaksjoner - spesifik periode
- SeKundeInfo
- OppdatereKundeInfo
- RegBetaling - KontoOversikt
- RegBetaling
- UtførBetaling
- SeSaldo
- LoggInnAdmin
- AdminEndreKunde
- AdminEndreKonto
- AdminOpprettNyKunde
- AdminOpprettKonto
- AdminSlettKonto
- AdminSlettKunde
- AdminSeAlleKunder

At the bottom of the interface, there are two status indicators: "Passed: 21" and "Failed: 0".

## Bank funksjoner

### InitDB:

open	http://localhost/testing/TestingBank/InitDB.php
pause	1000

**Som kunde ønsker jeg å logge inn på nettbanken min, slik at jeg får tilgang til nettbank funksjoner:**

open	http://localhost/testing/TestingBank/Index.php
click	link=Logg inn
click	id=personNr
type	id=personNr
assertValue	id=personNr
type	id=passord
assertValue	id=passord
click	id=loggInn

open	http://localhost/testing/TestingBank/View/saldo.php
click	link=Logg ut

open	http://localhost/testing/TestingBank/Index.php
click	link=Logg inn
click	id=personNr
type	id=personNr
assertValue	id=personNr
type	id=passord
assertValue	id=passord
click	id=loggInn
assertText	id=personNr
	Saldo for personnummer 01010110523

**Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal se mine transaksjoner, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenat:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Transaksjoner	
click	id=kontoNr	
select	id=kontoNr	label=22334412345
assertValue	id=kontoNr	22334412345

**Som kunde ønsker jeg å kunne se alle betalingene mine, slik at jeg får en oversikt:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Transaksjoner	
click	id=sok	

**Som kunde ønsker jeg å kunne se betalingene mine for en spesifikk periode:**

open	http://localhost/testing/TestingBank/View/transaksjon.php	
click	id=fraDato	
type	id=fraDato	2015-03-15
type	id=tilDato	2015-03-30
click	id=sok	

**Som kunde ønsker jeg å kunne se informasjonen som er lagret om meg, slik at jeg kan verifisere at den er korrekt:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Kunde-info	

**Som kunde vil jeg kunne oppdatere informasjonen min, slik at mine opplysninger blir riktige:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Kunde-info	
click	id=fornavn	
type	id=fornavn	Elin
assertValue	id=fornavn	Elin
type	id=etternavn	Holde
assertValue	id=etternavn	Holde
type	id=adresse	Prestmoen, 2b
assertValue	id=adresse	Prestmoen, 2b
type	id=postnr	2040
assertValue	id=postnr	2040
type	id=poststed	Kløfta
assertValue	id=poststed	Kløfta
type	id=telefonnr	+4797407320
assertValue	id=telefonnr	+4797407320
click	id=password	
type	id=password	Passord1!22
assertValue	id=password	Passord1!22
click	id=endre	

**Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal registrere en betaling, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenat:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Register betaling	
click	id=kontoNr	
select	id=kontoNr	22334412345
assertValue	id=kontoNr	22334412345

**Som kunde ønsker jeg å kunne registrere en betaling med en melding, slik at mottaker vet hva de får betalt for:**

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Register betaling	
click	id=kontoNr	
select	id=kontoNr	label=22334412345
assertValue	id=kontoNr	22334412345
click	id=tilKonto	
type	id=tilKonto	111111111111
assertValue	id=tilKonto	111111111111
type	id=dato	2019-09-09
assertValue	id=dato	2019-09-09
type	id=belop	666
assertValue	id=belop	666
type	id=melding	Dette er en systemtest
assertValue	id=melding	Dette er en systemtest
click	id=regBetaling	
assertText	//h2	Utfør betalinger

### Som kunde ønsker jeg å utføre en betaling som er blitt registrert:

open	http://localhost/testing/TestingBank/View/saldo.php	
click	link=Utfør betalinger	
assertText	//div[@id='listBetalinger']/table/tbody/tr/td[1]	
click	//div[@id='listBetalinger']/table/tbody/tr/td[6]/button	
assertText	//div[@id='listBetalinger']/table/thead/tr/th[5]	Melding

### Som kunde ønsker jeg å se informasjon om kontiene mine, slik at jeg har en oversikt over saldo, kontonr, type og valuta, på hver enkelt konto:

open	http://localhost/testing/TestingBank/View/utførBetalinger.php	
click	link=Saldo	
assertText	//div[@id='listSaldi']/table/tbody/tr/td	
assertText	id=personNr	Saldo for personnummer 01010110523

### Admin funksjoner

### Som administrator ønsker jeg å kunne logge inn, slik at jeg får tilgang til funksjonene i applikasjonen som jeg trenger til jobben min:

open	http://localhost/testing/TestingBank/index.php	
click	link=Admin	
click	id=bruker	
type	id=bruker	Admin
assertValue	id=bruker	Admin
click	id=passord	
type	id=passord	Admin
assertValue	id=passord	Admin
click	id=loggInn	
assertValue	id=personnummer1	01010110523

**Som administrator vil jeg kunne endre informasjon om en kunde for å ha riktig informasjon registrert:**

open	http://localhost/testing/TestingBank/View/adminKunde.php	
click	//div[@id='listKunde']/table/tbody/tr/td[2]	
type	id=fornavn1	Test
assertValue	id=fornavn1	Test
type	id=etternavn1	Test
assertValue	id=etternavn1	Test
type	id=adresse1	Test 22
assertValue	id=adresse1	Test 22
type	id=postnr1	1010
assertValue	id=postnr1	1010
click	link=Endre	
type	id=poststed1	Test
assertValue	id=poststed1	Test
type	id=telefonnr1	69696969
assertValue	id=telefonnr1	69696969
click	id=personnummer1	
click	id=passord1	
type	id=passord1	Passord2?
assertValue	id=passord1	Passord2?
click	link=Endre	

**Som administrator vil jeg kunne endre konto på en kunde for å ha riktig informasjon registrert:**

open	http://localhost/testing/TestingBank/View/adminKunde.php	
click	link=Endre konto	
click	id=saldo1	
type	id=saldo1	150
assertValue	id=saldo1	150
click	id=type1	
type	id=type1	Brukskonto
assertValue	id=type1	Brukskonto
click	id=valuta1	
type	id=valuta1	SEK
assertValue	id=valuta1	SEK
click	link=Endre	

**Som administrator vil jeg kunne opprette en ny kunde for at kunden skal kunne  
bruke nettbanken:**

open	http://localhost/testing/TestingBank/View/adminKunde.php	
click	link=Registrer kunde	
click	id=personnummer	
type	id=personnummer	1
assertValue	id=personnummer	1
click	id=fornavn	
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
click	id=etternavn	
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
click	id=adresse	
type	id=adresse	sætreveien 15
assertValue	id=adresse	sætreveien 15
click	id=postnr	
type	id=postnr	1415
assertValue	id=postnr	1415
click	id=poststed	
type	id=poststed	oppegård
assertValue	id=poststed	oppegård
click	id=telefonnr	
type	id=telefonnr	98859288
assertValue	id=telefonnr	98859288
click	id=password	
type	id=password	Admin

assertValue	id=password	Admin
click	id=regKunde	
assertValue	id=password2	Admin
click	link=Registrer kunde	
click	id=personnummer	
type	id=personnummer	01010110532
assertValue	id=personnummer	01010110532
click	id=fornavn	
type	id=fornavn	007
assertValue	id=fornavn	007
type	id=etternavn	test
assertValue	id=etternavn	test
click	id=adresse	
type	id=adresse	testveien 15
assertValue	id=adresse	testveien 15
click	id=postnr	
type	id=postnr	1515
assertValue	id=postnr	1515
click	id=poststed	
type	id=poststed	test
assertValue	id=poststed	test
click	id=telefonnr	
type	id=telefonnr	98859269
assertValue	id=telefonnr	98859269
click	id=password	

type	id=password	Admin
assertValue	id=password	Admin
click	id=regKunde	
assertValue	id=fornavn2	007
click	link=Registrer kunde	
click	id=personnummer	
type	id=personnummer	01010110555
assertValue	id=personnummer	01010110555
click	id=fornavn	
type	id=fornavn	test
assertValue	id=fornavn	test
click	id=etternavn	
type	id=etternavn	1
assertValue	id=etternavn	1
click	id=adresse	
type	id=adresse	testveien22
assertValue	id=adresse	testveien22
click	id=postnr	
type	id=postnr	1212
assertValue	id=postnr	1212
click	id=poststed	
type	id=poststed	køfta
assertValue	id=poststed	køfta
click	id=telefonnr	
type	id=telefonnr	98859288

assertValue	id=postaar	1212
click	id=poststed	
type	id=poststed	klefta
assertValue	id=poststed	klefta
click	id=telefonnr	
type	id=telefonnr	98859288
assertValue	id=telefonnr	98859288
click	id=password	
type	id=password	Admin
assertValue	id=password	Admin
click	id=regKunde	
assertValue	id=etternavn3	1
click	link=Registrer kunde	
click	id=personnummer	
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
click	id=fornavn	
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
click	id=etternavn	
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
⋮ click	id=adresse	
type	id=adresse	1
assertValue	id=adresse	1
⋮	id=submit	Submit

type	id=adresse	1
assertValue	id=adresse	1
type	id=postnr	1717
assertValue	id=postnr	1717
click	id=poststed	
type	id=poststed	skolen
assertValue	id=poststed	skolen
type	id=telefonnr	99999999
assertValue	id=telefonnr	99999999
click	id=password	
type	id=password	Admin
assertValue	id=password	Admin
click	id=regKunde	
assertValue	id=adresse8	1
click	link=Registrer kunde	
click	id=personnummer	
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
type	id=adresse	skoleveien 1
assertValue	id=adresse	skoleveien 1
click	id=postnr	

click		
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
type	id=adresse	skoleveien 1
assertValue	id=adresse	skoleveien 1
click		
type	id=postnr	1
assertValue	id=postnr	1
click		
type	id=poststed	skolen
assertValue	id=poststed	skolen
click		
type	id=password	Admin
assertValue	id=password	Admin
click		
type	id=telefonnr	98859269
assertValue	id=telefonnr	98859269
click		
click	link=Register kunde	
click		
type	id=personnummer	12345678910

type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
click	id=fornavn	
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
click	id=etternavn	
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
click	id=adresse	
type	id=adresse	skolen 1
assertValue	id=adresse	skolen 1
click	id=postnr	
type	id=postnr	1212
assertValue	id=postnr	1212
click	id=poststed	
type	id=poststed	1
assertValue	id=poststed	1
type	id=telefonnr	98859269
assertValue	id=telefonnr	98859269
click	id=password	
type	id=password	Admin
assertValue	id=password	Admin
click	id=regKunde	
click	link=Registrer kunde	

click	id=personnummer	
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
click	id=fornavn	
type	id=fornavn	jostein
assertValue	id=fornavn	jostein
click	id=etternavn	
type	id=etternavn	haugen
assertValue	id=etternavn	haugen
click	id=adresse	
type	id=adresse	skolen 2
assertValue	id=adresse	skolen 2
click	id=postnr	
type	id=postnr	1212
assertValue	id=postnr	1212
click	id=poststed	
type	id=poststed	skolen
assertValue	id=poststed	skolen
click	id=telefonnr	
type	id=telefonnr	dsd
assertValue	id=telefonnr	dsd
click	id=password	
type	id=password	Admin
assertValue	id=password	Admin
click	id=reqKunde	

assertValue		id=personnummer	12345678910
click		id=fornavn	
type		id=fornavn	jostein
assertValue		id=fornavn	jostein
click		id=etternavn	
type		id=etternavn	hauge
assertValue		id=etternavn	hauge
click		id=adresse	
type		id=adresse	skolen 3
assertValue		id=adresse	skolen 3
click		id=postnr	
type		id=postnr	1212
assertValue		id=postnr	1212
click		id=poststed	
type		id=poststed	skolen
assertValue		id=poststed	skolen
click		id=telefonnr	
type		id=telefonnr	98859269
assertValue		id=telefonnr	98859269
click		id=passord	
type		id=passord	Admin
assertValue		id=passord	Admin
click		id=regKunde	
assertValue		id=personnummer6	12345123456

**Som administrator vil jeg kunne opprette en konto på en spesifikk kunde for at kunden skal ha et sted å putte pengene sine:**

open	http://localhost/testing/TestingBank/View/adminKunde.php	
click	link=Register konto	
click	id=kontoNr	
type	id=kontoNr	1
assertValue	id=kontoNr	1
click	id=type	
type	id=type	brukskonto
assertValue	id=type	brukskonto
click	id=personnummer	
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
click	id=valuta	
type	id=valuta	150
assertValue	id=valuta	150
click	id=regKonto	
click	link=Register konto	
click	id=kontoNr	
type	id=kontoNr	123456789101
assertValue	id=kontoNr	123456789101
click	id=type	
type	id=type	sparekonto
assertValue	id=type	sparekonto
click	id=valuta	
type	id=valuta	100
assertValue	id=valuta	100

click	id=personnummer	
type	id=personnummer	1
assertValue	id=personnummer	1
click	id=regKonto	
click	link=Register konto	
click	id=kontoNr	
type	id=kontoNr	123456789101
assertValue	id=kontoNr	123456789101
click	id=type	
type	id=type	sparekonto
assertValue	id=type	sparekonto
type	id=valuta	700
assertValue	id=valuta	700
type	id=personnummer	12345678910
assertValue	id=personnummer	12345678910
click	id=regKonto	
click	link=Endre konto	
assertText	//h2	Endre konto

**Som administrator vil jeg kunne slette en konto fra en kunde for å rydde bort en ubrukt konto:**

open	http://localhost/testing/TestingBank/View/adminKunde.php
click	link=Endre konto
chooseOkOnNextConfirmation	
click	link=Slett

**Som administrator vil jeg kunne slette en kunde hvis kunden bytter bank eller dør:**

open	http://localhost/testing/TestingBank/View/adminKunde.php
chooseOkOnNextConfirmation	
click	link=Slett

**Som administrator vil jeg kunne få se alle kundene i banken for å få en oversikt over kundene i banken:**

open	http://localhost/testing/TestingBank/View/adminEndreKonto.php
click	link=Endre kunde
assertValue	id=etternavn1

**Som bruker av nettbanken ønsker jeg å kunne logge ut:**

open	http://localhost/testing/TestingBank/View/saldo.php
click	link=Logg ut

### 10.3.3.2. MTM-Rapport

#### Bank funksjoner

Test plan: TestPlan for Kunde TestingBank (ID: 15)

13/04/2020, 12:04

#### Test plan [15](#): TestPlan for Kunde TestingBank

##### Properties

Area Path:	Testing-bank-systemtest
Iteration:	Testing-bank-systemtest\Iteration 1
Owner:	s333943
State:	Active
Start date:	Wednesday, March 11, 2020
End date:	Wednesday, March 18, 2020

##### Suite hierarchy

Static suite: TestPlan for Kunde TestingBank (ID: [16](#))

Requirement-based suite: 7 : Som kunde ønsker jeg å logge inn på nettbanken min, slik at jeg får tilgang til nettbank funksjoner (ID: [18](#))

Requirement-based suite: 8 : Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal se mine transaksjoner, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenat (ID: [19](#))

Requirement-based suite: 9 : Som kunde vil jeg kunne sjekke betalingene mine, slik at jeg får en oversikt (ID: [20](#))

Requirement-based suite: 54 : Som kunde ønsker jeg å kunne se betalingene mine for en spesifikk periode (ID: [56](#))

Requirement-based suite: 10 : Som kunde ønsker jeg å kunne se informasjonen som er lagret om meg, slik at jeg kan verifisere at den er korrekt (ID: [21](#))

Requirement-based suite: 11 : Som kunde vil jeg kunne oppdatere informasjonen min, slik at mine opplysninger blir riktige (ID: [22](#))

Requirement-based suite: 12 : Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal registrere en betaling, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenatt (ID: [23](#))

Requirement-based suite: 13 : Som kunde ønsker jeg å kunne registrere en betaling med en melding, slik at mottaker vet hva de får betalt for (ID: [24](#))

Requirement-based suite: 63 : Som kunde ønsker jeg å utføre en betaling som er blitt registrert (ID: [64](#))

Requirement-based suite: 14 : Som kunde ønsker jeg å se informasjon om kontoene mine, slik at jeg har en oversikt over saldo, kontonr, type og valuta, på hver enkelt konto (ID: [25](#))

#### Test suite [18](#): 7 : Som kunde ønsker jeg å logge inn på nettbanken min, slik at jeg får tilgang til nettbank funksjoner

##### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">7</a> : Som kunde ønsker jeg å logge inn på nettbanken min, slik at jeg får tilgang til nettbank funksjoner
Configurations:	Windows 10
Test cases (1)	

#### Test case [44](#): LoggInn Kunde

##### STEPS

#	Action	Expected value	Attachments
1	Skriv inn URL: <a href="http://localhost/testing/TestingBank/index">http://localhost/testing/TestingBank/index</a>	Indexside kommer opp	

	x.php		<a href="#">Screenshot 2020-03-11 at 11.30_18.png (75K)</a>
2	trykke på logg inn	Åpner loggInn-side	<a href="#">Screenshot 2020-03-11 at 11.08_26.png (120K)</a>
3	skrive inn brukernavn	Skriver 01010110523	<a href="#">Screenshot 2020-03-11 at 11.11_49.png (119K)</a>
4	skrive inn passord	Skriver inn Passord!!	<a href="#">Screenshot 2020-03-11 at 11.12_59.png (105K)</a>
5	trykke på logg-inn-knapp	Går til /testing/TestingBank/View/saldo.php	<a href="#">Screenshot 2020-03-11 at 11.13_20.png (144K)</a>

**LATEST TEST OUTCOME**

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333943 Windows 10 s333939 Friday, April 3, 2020 0

Test suite [19](#): 8 : Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal se mine transaksjoner, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenat

## Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">8: Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal se mine transaksjoner, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenat</a>
Configurations:	Windows 10
Test cases (1)	

Test case [49](#): Kunde oversikt kontonummer

STEPS		Expected value	Attachments
#	Action		
1	Se startside etter login	ser url: http://localhost/testing/	<a href="#">Screenshot 2020-03-11 at 11.13_20.png (144K)</a>
2	Trykke transaksjoner	TestingBank/View/saldo.php	<a href="#">Screenshot 2020-03-11 at 11.44_34.png (133K)</a>
3	Trykke på dropdown	se Transaksjoner for konto side	<a href="#">Screenshot 2020-03-11 at 11.46_36.png (252K)</a>
4	Trykke på kontonr	er	<a href="#">Screenshot 2020-03-11 at 11.46_58.png (138K)</a>
		Se at kontonummer endrer seg	

**LATEST TEST OUTCOME**

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

Test suite [20](#): 9 : Som kunde vil jeg kunne sjekke betalingene mine, slik at jeg får en oversikt

## Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">2: Som kunde ønsker jeg å kunne se alle betalingene mine, slik at jeg får en oversikt</a>
Configurations:	Windows 10
<b>Test cases (1)</b>	

### Test case [55](#): se alle Transaksjoner

#### STEPS

#	Action	Expected value	Attachments
1	Se startside etter logginn	url: <a href="http://localhost/testing/TestBank/View/Saldo.php">http://localhost/testing/TestBank/View/Saldo.php</a>	<a href="#">Screenshot 2020-03-11 at 11.13.20.png</a> (144K)
2	Trykk på Transaksjoner	url: <a href="http://localhost/testing/TestBank/View/Transaksjon.php">http://localhost/testing/TestBank/View/Transaksjon.php</a>	<a href="#">Screenshot 2020-03-11 at 11.44.34.png</a> (133K)
3	trykke på søker-knapp	se alle transaksjoner	<a href="#">Screenshot 2020-03-11 at 11.55.05.png</a> (236K)

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

### Test suite [56](#): 54 : Som kunde ønsker jeg å kunne se betalingene mine for en spesifikk periode

## Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">54: Som kunde ønsker jeg å kunne se betalingene mine for en spesifikk periode</a>
Configurations:	Windows 10
<b>Test cases (1)</b>	

### Test case [57](#): Sjekke transaksjoner for en måned

#### STEPS

#	Action	Expected value	Attachments
1	Se at url er <a href="http://localhost/testing/TestingBank/View/transaksjon vindu saksjon.php">http://localhost/testing/TestingBank/View/transaksjon vindu saksjon.php</a>	fradato: 2015-03-15 tilDato: 2015-03-30	<a href="#">Screenshot 2020-03-11 at 11.44.34.png</a> (133K)
2	Skriv til og fra dato	se liste over transaksjoner	<a href="#">Screenshot 2020-03-11 at 12.10.25.png</a> (144K)
3	trykk på søker-knapp		<a href="#">Screenshot 2020-03-11 at 12.10.25.png</a> (201K)

**LATEST TEST OUTCOME**

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333943 Windows 10 s333943  
Monday, April 6, 2020 0

## Test suite [21](#): 10 : Som kunde ønsker jeg å kunne se informasjonen som er lagret om meg, slik at jeg kan verifisere at den er korrekt

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">10</a> : Som kunde ønsker jeg å kunne se informasjonen som er lagret om meg, slik at jeg kan verifisere at den er korrekt
Configurations:	Windows 10

### Test cases (1)

#### Test case [58](#): Se KundelInfo

##### STEPS

#	Action	Expected value	Attachments
1	Se Startside etter logginn	<a href="http://localhost/testing/Testing/Bank/View/Saldo.php">http://localhost/testing/Testing/Bank/View/Saldo.php</a>	<a href="#">Screenshot 2020-03-11 at 11.13.20.png</a> (144K)
2	klikk på Kunde-Info	Kommer til siden <a href="http://localhost/testing/Testing/View/kundelInfo.php">http://localhost/testing/Testing/View/kundelInfo.php</a>	<a href="#">Screenshot 2020-03-11 at 12.24.45.png</a> (183K)

**LATEST TEST OUTCOME**

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

## Test suite [22](#): 11 : Som kunde vil jeg kunne oppdatere informasjonen min, slik at mine opplysninger blir riktige

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">11</a> : Som kunde vil jeg kunne oppdatere informasjonen min, slik at mine opplysninger blir riktige
Configurations:	Windows 10
Test cases (1)	

### Test case [59](#): Oppdatere Kunde-Info

**STEPS**

#	Action	Expected value	Attachments
1	Startside etter logginn	url: http://localhost/testing/Vie w/saldo.php	<a href="#">Screenshot 2020-03-11 at 11.13.20.png</a> (144K)
2	Trykk på kunde-info	url: http://localhost/testing/Vie w/kundeInfo.php	<a href="#">Screenshot 2020-03-11 at 12.24.45.png</a> (183K)
3	endre alle felt	Alle felt er endret	<a href="#">Screenshot 2020-03-11 at 12.33.51.png</a> (186K)
4	Trykk endre-info-knapp	endring blir lagret	<a href="#">Screenshot 2020-03-11 at 12.35.01.png</a> (183K)

**LATEST TEST OUTCOME**

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333943 Windows 10 s333943 Monday, April 6, 2020 0

Test suite [23](#): 12 : Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal registrere en betaling, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenatt

**Properties**

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">12</a> : Som kunde ønsker jeg å få en oversikt over alle kontoene mine når jeg skal registrere en betaling, og kunne velge en av disse, slik at jeg ikke trenger å huske kontonr mine utenatt
Configurations:	Windows 10
Test cases (1)	

**Test case [60](#): Oversikt over kontoer på regBetalinger****STEPS**

#	Action	Expected value	Attachments
1	se Startside etter login	http://localhost/testing/Testing Bank/View/saldo.php	<a href="#">Screenshot 2020-03-11 at 11.43.50.png</a> (142K)
2	trykk på registrer-betaling	åpner side http://localhost/testin g/TestingBank/View/registerBe taling.php	<a href="#">Screenshot 2020-03-11 at 12.45.29.png</a> (145K)
3	Trykk på dropdown	viser alle tilgjengelige kontonu mmer	<a href="#">Screenshot 2020-03-11 at 12.47.12.png</a> (263K)
4	velge kontonummer i dropdown	viser valgt kontonummer	<a href="#">Screenshot 2020-03-11 at 12.48.00.png</a> (149K)

**LATEST TEST OUTCOME**

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

## Test suite [24](#): 13 : Som kunde ønsker jeg å kunne registrere en betaling med en melding, slik at mottaker vet hva de får betalt for

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">13</a> : Som kunde ønsker jeg å kunne registrere en betaling med en melding som viser hva de får betalt for
Configurations: Test cases (1)	Windows 10

### Test case [62](#): registrere en betaling

#### STEPS

#	Action	Expected value
1	se startside etter logginn url: <a href="http://localhost/testing/TestingBank/View/Saldo.php">http://localhost/testing/TestingBank/View/Saldo.php</a>	
2	Trykke på registrer betaling	går til registrer-betaling side
3	velge kontonr	velger kontonummer i dropdown
4	Fylle ut tilKontonr, betalingsdato, beløp, m felter er fylt ut	elding
5	trykke på registrer betaling	betalingen lagres, flytter side til <a href="#">Screenshot 2020-03-11 at 13.06</a> 1 utfør betaling: <a href="http://localhost/testing/Betalinger.php">http://localhost/testing/Betalinger.php</a>

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
Passed	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

## Test suite [64](#): 63 : Som kunde ønsker jeg å utføre en betaling som er blitt registrert

## Properties

---

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">63</a> : Som kunde ønsker jeg å utføre en betaling som er blitt registrert
Configurations:	Windows 10
Test cases (1)	

---

### Test case [65](#): utfør betaling

#### STEPS

#	Action	Expected value	Attachments
1	se startside etter logginn	url: http://localhost/testing/Test	<a href="#">Screenshot 2020-03-11 at 11.43</a>
2	trykk på utfør-betaling-knapp	fullfører betaling	<a href="#">50.png (142K)</a> <a href="#">Screenshot 2020-03-11 at 13.15.15.png (97K)</a>

#### LATEST TEST OUTCOME

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333943 Windows 10 s333943 Monday, April 6, 2020 0

## Test suite [25](#): 14 : Som kunde ønsker jeg å se informasjon om kontoene mine, slik at jeg har en oversikt over saldo, kontonr, type og valuta, på hver enkelt konto

## Properties

---

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">14</a> : Som kunde ønsker jeg å se informasjon om kontoene mine, slik at jeg har en oversikt over saldo, kontonr, type og valuta, på hver enkelt konto
Configurations:	Windows 10
Test cases (1)	

---

### Test case [61](#): Se Saldo

#### STEPS

#	Action	Expected value	Attachments
1	Stå på register-betaling side	url: http://localhost/testing/Test	<a href="#">Screenshot 2020-03-11 at 12.45</a>
2	Trykk på saldo-knapp	går til saldo side: http://localhost/testing/Test	<a href="#">registerBetaling.</a> <a href="#">29.png (145K)</a> php

**LATEST TEST OUTCOME**

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
Passed	s333943	Windows 10	s333943	Monday, April 6, 2020	0	

---

## Admin funksjoner

Test plan: TestPlan for Admin TestingBank (Id: 34)

13/04/2020, 11:56

### Test plan [34](#): TestPlan for Admin TestingBank

#### Properties

Area Path:	Testing-bank-systemtest
Iteration:	Testing-bank-systemtest\Iteration 1
Owner:	s333982
State:	Active
Start date:	Wednesday, March 11, 2020
End date:	Wednesday, March 18, 2020

#### Suite hierarchy

Static suite: TestPlan for Admin TestingBank (ID: [35](#))

Requirement-based suite: 66 : Som bruker av nettbanken ønsker jeg å kunne logge ut (ID: [70](#))

Requirement-based suite: 26 : Som administrator ønsker jeg å kunne logge inn, slik at jeg får tilgang til funksjonene i applikasjonen som jeg trenger til jobben min (ID: [36](#))

Requirement-based suite: 27 : Som administrator vil jeg kunne endre konto på en kunde for å ha riktig informasjon registrert (ID: [37](#))

Requirement-based suite: 28 : Som administrator vil jeg kunne endre informasjon om en kunde for å ha riktig informasjon registrert (ID: [38](#))

Requirement-based suite: 29 : Som administrator vil jeg kunne opprette en ny kunde for at kunden skal kunne bruke nettbanken (ID: [39](#))

Requirement-based suite: 30 : Som administrator vil jeg kunne opprette en konto på en spesifikk kunde for at kunden skal ha et sted å putte pengene sine (ID: [40](#))

Requirement-based suite: 31 : Som administrator vil jeg kunne slette en konto fra en kunde for å rydde bort en ubrukt konto (ID: [41](#))

Requirement-based suite: 32 : Som administrator vil jeg kunne slette en kunde hvis kunden bytter bank eller dør (ID: [42](#))

Requirement-based suite: 33 : Som administrator vil jeg kunne få se alle kundene i banken for å få en oversikt over kundene i banken (ID: [43](#))

### Test suite [70](#): 66 : Som bruker av nettbanken ønsker jeg å kunne logge ut

#### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">66</a> : Som bruker av nettbanken ønsker jeg å kunne logge ut
Configurations:	Windows 10

#### Test cases (1)

### Test case [71](#): LoggUt

#### STEPS

#	Action	Expected value	Attachments
1	Sjekk at man er på admin-siden	<a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a>	<a href="#">1.Admin - LoggetInn.png</a> (191 K)

Test plan: TestPlan for Admin TestingBank (id: 34)	13/04/2020, 11:56
2 Trykk på Logg Ut	<a href="http://localhost/testing/Testing_2_Admin - LoggetUt.png">http://localhost/testing/Testing_2_Admin - LoggetUt.png</a> (109K <u>Bank/Index.php</u> )
<b>LINKS</b>	
ID WorkItemType	
<u>66</u> User Story	Link type Tests
<b>LATEST TEST OUTCOME</b>	
Outcome Tester Configuration Run by Date completed Duration in seconds Build number <a href="#">Passed</a> s333939 Windows 10 s333943 Monday, April 13, 2020 0	

Test suite [36](#): 26 : Som administrator ønsker jeg å kunne logge inn, slik at jeg får tilgang til funksjonene i applikasjonen som jeg trenger til jobben min

#### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">26</a> : Som administrator ønsker jeg å kunne logge inn, slik at jeg får tilgang til funksjonene i applikasjonen som jeg trenger til jobben min
Configurations:	Windows 10
Test cases (1)	

#### Test case [45](#): LoggInnAdmin

##### STEPS

#	Action	Expected value	Attachments
1	Skriv inn i URL <a href="http://localhost/testing/TestingBank/Index">http://localhost/testing/TestingBank/Index</a> -siden er åpen	<a href="#">1.Forside-LoggInn.png</a> (45K) x.php	
2	Trykke på Admin i nav-bar	Åpne URL <a href="http://localhost/testi_2.Admin- ikkeLoggInn.png">http://localhost/testi_2.Admin- ikkeLoggInn.png</a> (ng/TestingBank/View/loggInn 55K) Admin.php	
3	Skrive inn brukernavn og passord	Åpne URL <a href="http://localhost/testi_3.Admin-LoggInn.png">http://localhost/testi_3.Admin-LoggInn.png</a> (57K) ng/TestingBank/View/loggInnB rukernavn & Passord: Admin.php Admin - Admin	
4	Trykke logg inn-knappen	Åpner endre kunde-side URL: <a href="#">4.Admin - LoggetInn.png</a> (90K) <a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a>	

##### LINKS

ID WorkItemType	Link type	Title
<u>26</u> User Story	Tests	Som administrator ønsker jeg å kunne logge inn, slik at jeg får tilgang til funksjonene i applikasjonen som jeg trenger til jobben min

##### LATEST TEST OUTCOME

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333982 Windows 10 s333943 Monday, April 6, 2020 0

## Test suite [37](#): 27 : Som administrator vil jeg kunne endre konto på en kunde for å ha riktig informasjon registrert

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">27</a> : Som administrator vil jeg kunne endre konto på en kunde for å ha riktig informasjon registrert
Configurations:	Windows 10
Test cases (1)	

### Test case [46](#): AdminEndreKonto

#### STEPS

#	Action	Expected value	Attachments
1	Sjekk at du er inne i admin-sidene	URL: <a href="http://localhost/testing/TestPlanForAdminTestingBank/View/adminKunde.php">http://localhost/testing/TestPlanForAdminTestingBank/View/adminKunde.php</a>	<a href="#">1.Admin - LoggetInn.png</a> (90K)
2	Trykke på endre konto i nav-bar	URL: <a href="http://localhost/testing/TestPlanForAdminTestingBank/View/adminEndreKonto.php">http://localhost/testing/TestPlanForAdminTestingBank/View/adminEndreKonto.php</a>	<a href="#">2.Admin - EndreKonto.png</a> (11K)
3	Skrive inn nytt saldo, type, valuta på en konto	saldo: 150 - type: Brukskonto - <a href="#">3.Admin - EndretKonto-kopi.png</a> Valuta: SEK	<a href="#">3.Admin - EndretKonto-kopi.png</a> (107K)
4	Trykke på endre konto-knappen	Kontoen blir endret til de nye verdiene	<a href="#">3.Admin - EndretKonto.png</a> (107K)
5			

#### LINKS

ID	WorkItemType	Link type	Title
<a href="#">27</a>	User Story	Tests	Som administrator vil jeg kunne endre konto på en kunde for å ha riktig informasjon registrert

#### LATEST TEST OUTCOME Outcome

Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333982	Windows 10	s333943	Monday, April 6, 2020	0

## Test suite [38](#): 28 : Som administrator vil jeg kunne endre informasjon om en kunde for å ha riktig informasjon registrert

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">28</a> : Som administrator vil jeg kunne endre informasjon om en kunde for å ha riktig informasjon registrert
Configurations:	Windows 10

## Test cases (1)

### Test case 47: AdminEndreKunde

#### STEPS

#	Action	Expected value	Attachments
1	Sjekk at man er på endre kunde-siden Bank\View\adminKunde.php K)	http://localhost/testing/Testing 1.Admin - EndreKunde.png (89K)	<a href="#">1.Admin - EndreKunde.png</a> (89K)
2	Endre persnr, fornavn, etternavn, adresse, postnr, poststed , tlf, passord for en kunde	<a href="#">2.Admin - EndreKunde.png</a> (88K)	<a href="#">2.Admin - EndreKunde.png</a> (88K)
3	Trykke på endre-knappen Kunden blir endret og får de ny e verdiene comment: validering funker i kke	<a href="#">3.Admin - EndreKunde-kopi.png</a> (88K) http://localhost/testing/Testing Bank\View\adminKunde.php	<a href="#">3.Admin - EndreKunde-kopi.png</a> (88K)

#### LINKS

ID	WorkItemType	Link type	Title
<a href="#">28</a>			User Story Tests Som administrator vil jeg kunne endre informasjon om en kunde for å ha riktig informasjon registrert

#### LATEST TEST OUTCOME

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333982 Windows 10 s333943 Monday, April 6, 2020 0

### Test suite 39: 29 : Som administrator vil jeg kunne opprette en ny kunde for at kunden skal kunne bruke nettbanken

#### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">29: Som administrator vil jeg kunne opprette en ny kunde for at kunden skal kunne bruke nettbanke</a> n
Configurations:	Windows 10
Test cases (1)	

### Test case 48: AdminOpprettNyKunde

#### STEPS

#	Action	Expected value	Attachments
1	Sjekk at man er på admin-sidene Bank\View\adminKunde.php 89K)	http://localhost/testing/Testing 1.Admin - OpprettKunde.png (89K)	<a href="#">1.Admin - OpprettKunde.png</a> (89K)

2	Trykk på registrer kunde	Åpner registrer kunde-siden <a href="http://localhost/testing/TestingBank/View/adminRegisterKunde.php">http://localhost/testing/TestingBank/View/adminRegisterKunde.php</a>	<a href="#">2.Admin - RegistrerKunde.png</a> (73K)
3	Feil PersonNummer - Alt annet riktig	Rød valideringsmelding	<a href="#">3.Admin - RegKunde_FeilPersonNr.png</a> (88K)
4	Trykke på registrer kunde-knappen	Kunde blir (ikke) registrert.	
5	Feil Fornavn - Alt annet riktig	Rød valideringsmelding	<a href="#">5.Admin - RegKunde_FeilFornavn.png</a> (87K)
6	Trykke på registrer kunde-knappen	Kunde blir ikke registrert.	
7	Feil Etternavn - Alt annet riktig		<a href="#">6.Admin - RegKunde_FeilEtternavn.png</a> (88K)
8	Trykke på registrer kunde-knappen	Kunde blir ikke registrert. Rød valideringsmelding	
9	Feil Adresse - Alt annet riktig		<a href="#">7.Admin - RegKunde_FeilAdresse.png</a> (86K)
10	Trykke på registrer kunde-knappen	Kunde blir ikke registrert. Rød valideringsmelding	
11	Feil PostNr - Alt annet riktig		<a href="#">8.Admin - RegKunde_FeilPostNr.png</a> (87K)
12	Trykke på registrer kunde-knappen	Kunde blir ikke registrert. Rød valideringsmelding	
13	Feil PostSted - Alt annet riktig		<a href="#">9.Admin - RegKunde_FeilPoststed.png</a> (87K)
14	Trykke på registrer kunde-knappen	Kunde blir ikke registrert. Rød valideringsmelding	
15	Feil Telefonnummer - Alt annet riktig		<a href="#">10.Admin - RegKunde_FeilTlfNr.png</a> (84K)
16	Trykke på registrer kunde-knappen	Kunde blir ikke registrert. Rød valideringsmelding	
17	Alle felt fylt ut riktig		<a href="#">11.Admin - RegKunde_NyKunde.png</a> (80K)
18	Trykke på registrer kunde-knappen	Ny kunde registrert med riktig <a href="#">12.Admin - RegKunde_Opprettst.png</a> (99K)	

**LINKS**

ID	WorkItemType	Link type	Title
<a href="#">29</a>	User Story	Tests	Som administrator vil jeg kunne opprette en ny kunde for at kunden skal kunne bruke nettbanken

**LATEST TEST OUTCOME**

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Failed\\_s333982](#) Windows 10 s333943 Monday, April 6, 2020 0

**Test suite [40](#): 30 : Som administrator vil jeg kunne opprette en konto på en spesifikk kunde for at kunden skal ha et sted å putte pengene sine**

Properties

State: Completed  
 Type: Requirement-based Suite  
 Requirement: [30: Som administrator vil jeg kunne opprette en konto på en spesifikk kunde for at kunden skal ha et sted å putte pengene sine](#)  
 Configurations: Windows 10  
 Test cases (1)

## Test case [50](#): AdminOpprettKonto

### STEPS

#	Action	Expected value	Attachments
1	Sjekk at man er inne i admin-sidene	<a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a> (90K)	<a href="#">0.Admin - LoggetInn-kopi.png</a> (90K)
2	Trykke på registrer konto i nav-bar	Åpner registrer konto-siden <a href="http://localhost/testing/TestingBank/View/adminRegisterKonto.php">http://localhost/testing/TestingBank/View/adminRegisterKonto.php</a>	<a href="#">1.Admin - RegKonto_Forside.png</a> (61K)
3	Feil Kontonummer - Alt annet riktig	Rød valideringsmelding	<a href="#">2.Admin - RegKonto_FeilKoNr.png</a> (71K)
4	Trykk på registrer konto-knappen	Konto blir (ikke) registrert.	
5	Feil Personnummer - alt annet riktig	Rød valideringsmelding	<a href="#">3.Admin - RegKonto_FeilPersNr.png</a> (68K)
6	Trykk på registrer konto-knappen	Konto blir ikke registrert fordi personnummer ikke finnes i databasen, men det er mulig å lage en kunde med feil personnummer, så regexen fungerer ikke her heller	
7	Alle felt fylles inn riktig		<a href="#">4.Admin- RegKonto_AltRiktig.png</a> (82K)
8	Trykk på registrer konto-knappen	Konto registreres	
9	Åpner kundeoversikten	<a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a> (102K)	<a href="#">5.Admin - SideEtterKontoReg.png</a> (102K)
10	Trykke på endreKonto	<a href="http://localhost/testing/TestingBank/View/adminEndreKonto.php">http://localhost/testing/TestingBank/View/adminEndreKonto.php</a> (122K)	<a href="#">6.Admin-RegKonto_LagtInn.png</a> (122K)
11	Sjekke at konto er opprettet		

### LINKS

ID	WorkItemType	Link type	Title
<a href="#">30</a>	User Story	Tests	Som administrator vil jeg kunne opprette en konto på en spesifikk kunde for at kunden skal ha et sted å putte pengene sine

### LATEST TEST OUTCOME Outcome

Tester	Configurations	Run by	Date completed	Duration in seconds	Build number
--------	----------------	--------	----------------	---------------------	--------------

Failed s333982 Windows 10 s333943 Monday, April 6, 2020 0

## Test suite [41](#): 31 : Som administrator vil jeg kunne slette en konto fra en kunde for å rydde bort en ubrukt konto

### Properties

State:	Completed
Type:	Requirement-based Suite
Requirement:	<a href="#">31</a> : Som administrator vil jeg kunne slette en konto fra en kunde for å rydde bort en ubrukt konto
Configurations:	Windows 10
Test cases (1)	

### Test case [51](#): AdminSletteKonto

#### STEPS

#	Action	Expected value	Attachments
1	Sjekk at man er inne i admin-sidene	http://localhost/testing/TestingBank/View/adminKunde.php	<a href="#">1.Admin - SlettKunde Forside.png</a> (104K)
2	Trykke på endre konto i nav-bar	Åpner endre konto-siden http://localhost/testing/TestingBank/View/adminEndreKonto.php	<a href="#">2.Admin - SlettKunde EndreKonto_Forside.png</a> (121K)
3	Trykk på slett-knappen som tilhører den ene kontoen	En alert med bekrefteelse dukker opp	<a href="#">3.Admin - SlettKunde Før Sletting.png</a> (137K)
4	Trykker OK på alerten	Kontoen blir slettet http://localhost/testing/TestingBank/View/adminEndreKonto.php	<a href="#">4.Admin - SlettKunde EtterSletting.png</a> (112K)
5	Sjekk at kontoen er slettet		

#### LINKS

ID	WorkItemType	Link type	Title
<a href="#">31</a>	User Story	Tests	Som administrator vil jeg kunne slette en konto fra en kunde for å rydde bort en ubrukt konto

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333982	Windows 10	s333943	Thursday, March 12, 2020	0	

## Test suite [42](#): 32 : Som administrator vil jeg kunne slette en kunde hvis kunden bytter bank eller dør

### Properties

State:	Completed
--------	-----------

Type: Requirement-based Suite  
 Requirement: [32](#): Som administrator vil jeg kunne slette en kunde hvis kunden bytter bank eller dør  
 Configurations: Windows 10  
 Test cases (1)

### Test case [52](#): AdminSletteKunde

#### STEPS

#	Action	Expected value	Attachments
1	Sjekke at man er i adminEndreKunde-siden	Åpner endre kunde-siden <a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a>	<a href="#">1.Admin - SlettKunde_Forside.png</a> (87K)
2	Trykke på slett-knappen som tilhører en kunde	Alert med bekrefelse dukker opp Kunden blir slettet og URL er	<a href="#">2.Admin - SlettKunde_ErSlettning.png</a> (101K) <a href="#">3.Admin - SlettKunde_EtterSletting.png</a> (50K)
3	Trykk OK i alerten		
4	Sjekke at kunden er slettet fra databasen		<a href="http://localhost/testing/TestingBank/View/adminKunde.php">http://localhost/testing/TestingBank/View/adminKunde.php</a>

#### LINKS

ID	WorkItemType	Link type	Title
<a href="#">32</a>	User Story	Tests	Som administrator vil jeg kunne slette en kunde hvis kunden bytter bank eller dør

#### LATEST TEST OUTCOME

Outcome Tester Configuration Run by Date completed Duration in seconds Build number [Passed](#) s333982 Windows 10 s333943 Monday, April 6, 2020 0

### Test suite [43](#): 33 : Som administrator vil jeg kunne få se alle kundene i banken for å få en oversikt over kundene i banken

#### Properties

State: Completed  
 Type: Requirement-based Suite  
 Requirement: [33](#): Som administrator vil jeg kunne få se alle kundene i banken for å få en oversikt over kundene i banken  
 Configurations: Windows 10  
 Test cases (1)

### Test case [53](#): AdminSeAlleKunder

#### STEPS

#	Action	Expected value	Attachments
1	Sjekke at man er inne på endre konto	<a href="http://localhost/testing/TestingBank/View/adminEndreKonto.php">http://localhost/testing/TestingBank/View/adminEndreKonto.php</a>	<a href="#">1.Admin - SlettKunde_Forside.png</a> (87K)

2 Trykke på endre kunde-knappen i nav-bar http://localhost/testing/Testing 2.Admin - SlettKunde\_FørSlett  
Bank/View/adminKunde.php [ing.png](#) (101K)

3 Se på kunder [3.Admin - SlettKunde\\_EtterSle  
ttning.png](#) (50K)

**LINKS**

ID	WorkItemType	Link type	Title
<a href="#">33</a>	User Story	Tests	Som administrator vil jeg kunne få se alle kundene i banken for å få en oversikt over kundene i banken

**LATEST TEST OUTCOME**

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s333982	Windows 10	s333943	Monday, April 6, 2020	0	