

# **Test Rapport for «Testing Bank» Programvare**

Skrevet av Oskar Firat

1. Identifikasjon .....	2
2. Sammendrag.....	3
3. Avvik i forhold til test plan.....	3
4. Evaluering av testens grundighet.....	5
5. Gjenstående aktiviteter/oppgaver.....	5
6. Avvik i produktet.....	6
7. Vurdering av produktet som er testet.....	6
8. Sammendrag av aktiviteter og lærdom.....	8
1. Identifikasjon.....	9
2. Innledning.....	10
3. Testobjekter.....	11
4. Fremgangsmåte.....	12
4.1 Overordnet teststrategi for.....	13
4.2 Prosedyre for planlegging og gjennomføring av test.....	15
4.3 Kriterier for grundighet i testen.....	16
5. Godkjenningsskriterier.....	17
6. Testmiljø.....	18
7. Resultat dokumenter.....	19

# 1. Identifikasjon

Navn på slutt rapporten: Test rapport for "Testing Bank" programvare

Versjon: 1

«En test av programmet TestingBank»

**Forfattere:** Oskar Firat

**Dato:** 16.03.2019

Denne rapporten tar for seg alle tester som er gjort for programmet «Testingbank», dvs. den delen som bankkunder bruker og den delen Admin bruker, og testene som er gjennomført er enhets-, integrasjons-, og systemtest.

## 2. Sammendrag

Oppsummer evalueringen av det produkt som er testet. Spesielt beskriv kritikaliteten av produktet, slik at de som leser denne rapporten vet hvor kritisk produktet og dens funksjoner er! Identifiser testmiljøet som er brukt.

Dette er et program som krever høy sikkerhet siden dette skal brukes av bank og bankkunder og hvor man ønsker å ha et så sikkert system som mulig for å forhindre at verdier kan gå tapt og at bankens omdømme svekkes. Vi har derfor hatt som mål og forsøke å teste alle metoder og alle deler av programmet. Vi har vurdert det slik at systemet bør kunne håndtere ulike input data fra kundene som både er riktig og feil, som f.eks. riktig personnummer og feil personnummer. Systemet bør kunne gi en feilmelding og samtidig forhindre tilgang på informasjon om f.eks. konto eller konti når kunden taster inn feil, enten hensikten er god eller dårlig.

Våre tester viste at programvaren har en del feil, både for brukersiden og Admin siden, når det ble tastet inn feil data. Systemet burde i mye større grad gi tilbakemeldinger i form av feilmeldinger og ha et regex system som validerer input data ved feil input ikke lar brukeren få lov til å kunne fortsette videre.

### 3. Avvik i forhold til test plan

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

I enhetstesten så hadde vi ingen test plan, så her kan vi ikke dokumentere noe avvik. Her testet vi databasen med adminDatabaseStub og bankDatabaseStub. Enkelte av testene mot stubene gikk via adminLogikk og bankLogikk. Her gikk testene ut på å sjekke om metodene virket slik de skulle. Her ble tester i adminTest og bankTest som hadde valid testdata iført endelsen OK, mens invalid testdata endte med endelsen Feil i metode navnet.

I integrasjonstesten så fant vi følgende avvik:

Når vi tastet inn feil personnummer ved henting av betalinger så forventet vi å få en feilmelding tilbake. Systemet gav ingen tilbakemelding. Man bør forvente at systemet

gir en feilmelding når feil personnummer blir tastet inn. Når det ble testet på inntasting av feil kontonummer, så forventet vi også å få en feilmelding tilbake, som f.eks. feil inntastet kontonummer, prøv på nytt. I stedet får man en feilmelding hvor det står personnummer: null, kontonummer: 1234, saldo: null, type: null, valuta: null, transaksjoner []. Ingen oppklarende feilmelding blir gitt i dette tilfelle.

Når det blir tastet inn feil transaksjons id (TxID), i dette tilfelle -1, så forventer man at systemet skal gi en feilmelding, men ingen feilmelding blir returnert. I stedet får man en del html kode og en gjennomført transaksjon som har en annen TxID enn det som blir tastet inn.

Om man vil endre kundeinfo og lar feltene for inntasting stå tomme, så får man en OK tilbakemelding. Altså ingen tilbakemelding som gjør kunden oppmerksom på at han må taste inn informasjon han ønsker å endre. Det samme er det om kunden skulle taste inn personnummer på feil form. Heller ikke her gir systemet tilbakemelding om at inntastingen er feil. Det samme gjelder for registrering av betalinger.

Ved en situasjon hvor databasen er nede eller ikke mulig å få kontakt med, så får man ikke tilbakemelding om at banken eller databasen ikke er mulig å nå, eller f.eks. «Banken er ikke tilgjengelig for tiden, prøv igjen senere». Det ble også testet om det var mulig å hente betalinger når databasen var nede, også her ble det ikke returnert noen feilmelding. Det samme resultatet ble registrert når det ble testet på å hente konti med databasen nede.

Ved systemtesten så fant vi følgende avvik:

Når man er på admin bruker og vil endre kunde info så er det mulig å fjerne alt bortsett fra poststed og personnummer og sende inn med whitespace. Her forventes det at du får en feilmelding at man ikke kan endre kundeinfo feltene til bare whitespace.

Om man vil registrere en konto til en kunde som admin får man tilbakemelding når man skriver inn et feil kontonummer men kontoen blir registrert. I tillegg hvis admin skal endre kunde kunde konto. Så kan man både endre kontoen med tomme felt og med feil verdier(verdier som da ikke bør være tillatt eks. kontonummer med færre enn 11 siffer).

Hvis man som kunde vil registrere betalinger er det mulig å registrere med betalingen med feil kontonummer og eller feil dato i tillegg kan man også skrive inn en negativ verdi på beløp (eks. -200). Her får man feilmelding på dato format og kontonummer men det går ann å registrere betalingen uansett.

Det var også noe feil med å endre kundeinfo, man kunne endre alt av infoen til for eksempel kun "a", dette skapte problemer ved at man kan endre passord til a men når man skal logge inn med samme kunde senere så går det ikke ann å logge inn og man får feilmeldingen "passord må være minst 6 bokstaver". I tillegg må et postnummer være 4 siffer og tlf nummer 8 siffer, men dette fikk man ingen feilmelding på.

Det var også en feil med å se transaksjoner som kunde, her kan man skrive inn feil "fra dato" men fortsatt få opp alle transaksjonene.

## 4. Evaluering av testens grundighet

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

Vi hadde som målsetning å teste alle funksjoner/metoder, som henting, endring, betaling, innlogging, transaksjoner, registreringer og hva som skjedde når databasen er nede.

I enhetstesten så ble 95 % eller mer av den planlagte testen utført og vi fant ingen åpne A eller B feil. Her kunne vi kanskje ha testet mer uventede verdier som kan bli skrevet inn av en bruker. Altså, så kunne vi ha hatt større variasjon i testdata. Dette ble ikke gjort grunnet at vi mente at testdata er tilstrekkelig, grunnet invalid data som ble brukt som testdata.

I integrasjonstesten så ble alle API'er utført, og her fant vi ingen åpne A feil, men vi

fant B feil. Et eksempel på en slik åpen B feil er ved registrering av kunde med ingen inndata i innskrivingsfeltene og resultatet er at man får registrert en kunde i databasen uten verdier.

I system testen ble alle brukerhistoriene og deres tester utført og ingen åpne A feil ble registrert. Her kunne vi kanskje ha satt inn mer skadelig testdata som for eksempel `$test = new skadeligklasse();`. Altså, forsøke å forårsake en fatale error. Her kunne vi da ha sett om systemet krasjer. Grunnen til at dette ikke ble gjort er grunnet at dette går mer ut på sikkerhetstesting.

## 5. Gjenstående aktiviteter/oppgaver

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

I hovedsak så testet vi bare for feil og for å finne feil. Det ble ikke gjort noen endringer i selve koden eller programvaren fra vår side. Det som da gjenstår er å rette alle feil som ble funnet og så gjennomføre nye tester for å se om det har oppstått nye feil og endringer som må rettes på.

## 6. Avvik i produktet

Gi et sammendrag av testresultatene. Identifiser alle avvik som ble funnet. Oppsummer deres løsning og beskriv ny test / regresjonstest som er utført etter retting. Identifiser alle gjenstående uløste avvik. Gi en begrunnelse for at de ikke er rettet.

Resultatet av testene viser at enhetstesten ga en godkjenning på 95% og mer, mens integrasjonstesten gav et mye dårligere resultat som en følge av mangel på validering av inntastet data og mangel på feilmeldinger, her viste også testen at de fleste feilene var av alvorlig grad. Stort sett de samme testresultatene fant vi for systemtesten, med mange av de samme feilene og også her med feil av alvorlig grad.

Avvik som ble identifisert, var mangel på validering og å gi administrator og bankkunder som bruker systemet feilmeldinger når man enten hadde tastet inn feil passord eller kontonummer eller latt være å skrive inn noe i felter som skal fylles ut.

Forslag til løsninger på feilene vi fant er hovedsakelig å bruke validering av inntastet data og gi feilmeldinger. Når det gjelder ny test/regresjonstest som en følge av

retting, så gjennomførte vi ingen rettinger. Vår testing hadde bare som mål å finne feil. Begrunnelsen for at vi ikke rettet feil er at vi forstod oppgaven slik at den gikk ut på å finne feil og ikke rette feil.

## 7. Vurdering av produktet som er testet

Gi en totalvurdering av hvert testobjekt med dets eventuelle begrensninger. Dette skal baseres på testresultatene og på godkjenningskriteriene som tidligere er planlagt brukt (se test plan, etc.). Gi et estimat av risikoen med å sette produktet i drift. Er det noen spesielle forutsetninger som er tatt?

Slik vi forstår dette spørsmålet så ønsker man en overordnet vurdering av metoder/funksjoner til de ulike testobjektene basert på testresultatene og godkjenningskriteriene.

Om vi begynner med enhetstesten så satte vi godkjenningskriteriet til 95 % eller høyere og testen gav oss et resultat på over 95 % som vi anser som akseptabelt. Alle testene passerte uten feil. Grunnen til at testen ikke kunne gi et mulig resultat på 100 % skyldes at adminLogikk og bankLogikk ikke brukes som database i enhetstesten, i stedet bruker man en databasestub. Imidlertid så viser all testing i forhold til adminDatabasestub og bankdatabasestub at hver av dem gir et resultat på 100 %. Godkjenningskriteriet ble satt til kategori C. "Kategori C: Feil med mindre alvorlige konsekvenser, for eksempel mindre grad av ustabilitet, mindre funksjons-/egenskapsmangler etc."

Med integrasjonstesting fant vi flere feil. For bank delen så handlet feilene om at man ikke fikk tilbakemelding om man tastet inn f.eks. feil personnummer og kontonummer, at man ikke fikk feilmelding når databasen var nede og man forsøkte å logge inn eller hente konti med databasen nede.

I admin delen så var de viktigste feilene registrere Kunde, Endre kunde info og slette kunde. I testingen viser det seg at det er mulig å registrere ny kunde og endre kundeinfo uten noe inndata. Det samme gjelder på fleste av filene som finnes. Den andre mest vanlig feilen er at man får ikke noe feilmelding tilbake og testen gir OK som tilbakemelding. For eksempel i funksjonen slettKunde gis det ikke noen feilmelding tilbake hvis man skriver et personnummer som ikke finnes i databasen, mens tilbakemeldingen sier Ok.

Godkjenningskriteriet settes til B og C.

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

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

Om man legger sammen sammen godkjente tester i bank og admin delene så blir det totalt godkjente antall tester på 70 %. Dette mener vi er under det som er akseptabelt.

System testen har sine klare mangler ved at den ikke validerer og i tillegg godkjenner invalid data. Det er en åpen risiko at man kan for eksempel skrive -100 (se - 2.3.5 Kunde: Registrere Betalinger). Det er også et problem at hvis man skriver 50.50 (se i Registrere Betalinger), så forsvinner 0 fra 50.50 etter komma. Dermed mister kunden 45 øre. Dette er et relativt alvorlig problem. Det er åpenbare risiko derfor å sette systemet i drift. Dermed bør dette fikses, før den nye versjonen utgis. Basert på godkjenningsskriteriene, så så er det ingen krasj av systemet ved invalid testdata, men det bør fortsatt forekomme fiksing av feilene som er oppgitt.

I systemtestingen fikk vi mange av feilene som også ligger i integrasjonstesting og enhetstesten sett fra brukeren sin side.

Vårt estimat av risikoen med å sette dette produktet i drift er at risikoen er høy, dvs. at produktet inneholder såpass mange feil at det vil kunne skade organisasjonen som tar det i bruk sitt omdømme. Det er stor grunn til å tro at brukerne vil bli frustrerte, irriterte og finne det lite brukervennlig. Vi forutsetter da at det ikke har blitt gjort noen endringer i systemet/koden.

## 8. Sammendrag av aktiviteter og lærdom

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

Testingen har vært interessant og lærerik, men fra starten av så har læringskurven vært bratt og ikke alt har vært like tilgjengelig og lett å sette seg inn i, men ettersom testingen skred frem så har forståelsen og viktigheten av testing i stigende grad blitt større. Installeringen av NetBeans var uproblematisk, mens PHPUnit, som skal integreres i Netbeans hadde sine utfordringer, det vil si at det ikke var like lett å forstå hvordan det skulle brukes. men når de utfordringene var overkommet, så viste det seg at enhetstesten med disse verktøyene var lett å bruke og etter vår mening et godt verktøy å bruke i forbindelse med enhetstesting.

Xampp var veldig greit å bruke, ganske enkelt og bra server software der man kun trenger å skru på to tjenester for å hoste løsningen lokalt.

I integrasjonstesting så ble det brukt programmet SoapUI som det var litt vanskelig å bli kjent med og ble ikke oppfatta som så veldig brukervennlig.



Bruken av test plan i Excel var et god hjelpemiddel som bidro til å ha en god oversikt over testene som skulle kjøres og resultatene av testene, i tillegg til hvilke testdata som ble brukt og forventet og det reelle testresultatet.

En annen ting var at det var ikke mulig å endre fonten og eller skalere programmet til riktig oppløsning, hvor en i gruppen hadde en pc skjerm med 4k oppløsning.

Løsningen for dette var å redusere oppløsningen så man kunne bruke programmet.

Selenium IDE var lett å installere da dette er bare en google chrome add on. Det var litt knotete å lære seg bruken av programmet i starten. Men etter et par timer var vi godt kjent med funksjonalitetene og hvordan programmet fungerte. Duplisering av tester var en god funksjonalitet å ha, og sparte oss mye tid ved at vi skal å manuelt logg oss inn hele tiden. Det var også positivt at programmet kunne kopiere command i testene. Det som var mest negativt var at den (eller kanskje vi ikke fant denne funksjonaliteten) ikke kunne skrive ut eller gi et automatisk skjerm bilde av command kjøringene. Dermed ble det mye arbeid med å dokumentere dette. MTM inneholder utrolig mange funksjonaliteter og kan være veldig omfattende når man prøver å dirigere seg rundt på programvaren. En annen ting var at det var at det gikk ikke ann å lage duplikater av steps, for eksempel "1.resette databasen 2. gå til index 3. logge inn etc...". Disse "teststepsene" er inkludert på hver eneste test case og det å ikke kunne kopiere dem krevde ekstremt mye mere tid. I ettertid vil man tenke at en sånn funksjon eksistere, det er bare veldig vanskelig å eventuelt finne frem når det er en haug av andre funksjonaliteter.

Vi trekker den lærdommen ut av denne testen at vi gjerne skulle ha utformet en bedre testplan. Vi kunne ønsket oss mer tid til testing fordi vi antar at det er flere ting vi kunne ha testet på.

# Testplan

## 1. Identifikasjon

**Laget når:** 21.01.2019

**Av forfatter:** Oskar Firat

**Sist oppdatert når:** 16.03.2019

**Hva testes her:** TestingBank med bruk av Enhetstest, Integrasjonstest og Systemtest.

**Versjonsnummer av denne planen:** 8

## 2. Innledning

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

Her fikk vi som oppgave å teste systemet "TestingBank" som er en applikasjon som er en enkel bank tjeneste for kunder. Systemet har også en administrasjonsdel for ansatte. Dermed kan systemet deles opp i en kunde del og administrasjonsdel. Systemet tjenester blir implementert ved hjelp av funksjonaliteter som har blitt laget i TestingBank systemet. TestingBank systemet er en mappe som inneholder flere andre mapper som utgjør systemet som helhet. Her utgjør de ulike mappene API'er, DAL, View, BLL, osv.

Funksjonaliteten som presenteres i Testingbank for kunde og administrasjon er som følge:

Funksjoner for Administrasjons delen:

- Se oversikt over alle kundene
- Se oversikt over alle kontoene
- Registrere ny Kunde
- Endre Informasjon på Kundene

- Slette en Kunde fra databasen
- Registrere ny Konto for kunden
- Endringer på Kontoene til kunder
- Slette Konto fra databasen

Funksjoner for Kundens delen:

- Sjekke Saldo
- Registrere og utføre Betalinger
- Se oversikt over Betalinger
- Se oversikt over Transaksjoner
- Endre Kunde Informasjonen

### 3. Testobjekter

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

Beskriv de funksjoner og egenskaper som skal testes og evt. som ikke skal testes.

Er det noen prioritering av testene?

De funksjonene som skal testes er i mappene API (inneholder programvarens grensesnitt som påkaller ulike tjenester), BLL (logikken for hvordan nettstedet fungerer), DAL (som gir tilgang til data, og innehar database kontakt) og View (som er innehar en interface på hva brukeren ser). BLL inneholder adminLogikk og bankLogikk. Mens DAL inneholder adminDatabase og bankdatabase (som er PHP kode) med funksjoner og SQL spørringer. View mappen inneholder funksjoner og kode som gjør det mulig å presentere innhold som hentes i databasen og presenteres for bankkunder og admin, altså et visuelt grensesnitt som viser innholdet i nettleser. Her tester vi blant annet validering av data. Når vi startet med enhetstesting måtte vi erstatte adminDatabase og bankdatabase i DAL mappen med adminDatabaseStub.php og bankDatabaseStub.php. Dette er fordi vi ofte måtte foreta kjøring av test ved hver endring av test scriptet i adminTest og bankTest som kjørte mot adminDatabaseStub.php og bankDatabaseStub.php. Her ville det ta lengre tid å kjøre mot databasen, samt at databasen vil måtte endres for hver

kjøring. Her stubene et speilbilde av selve databasen og erstatter da denne.

I enhetstesting så testes alle funksjoner som er å finne i de vanlige databasene for admin og bank. Her er det viktig å påminne at vi tester individuelle programenheter i enhetstesting og dermed forventes det ved input et bestemt output. I disse tilfelle skal `adminDatabaseStub.php` og `bankDatabaseStub.php` virke lik databasen og gi samme output som databasen ved en gitt input. Men i noen tester så må testdata gå gjennom `adminLogikk` og `bankLogikk`. Dermed måtte funksjoner i `adminLogikk/bankLogikk` prioriteres og deretter funksjoner i `bankDatabaseStuben`. I for eksempel `bankLogikk` så foretas en `Regex` i funksjonen `sjekkLoggInn` og deretter videre til `bankDatabaseStub.php` med `sjekkLoggInn`. Dermed må egenskapene teste både denne for godkjent og feil for `Regex` og prioritere denne funksjonen for deretter å gå videre til `bankDatabaseStuben`. Derfor tok verdiene som det ble testet med, høyde for dette først.

Deretter startet vi med integrasjonstesting som her tester sammensatte komponenter og om samspillet oppfører seg som ønsket. Enkelt forklart man tester en del av programmet, men ikke hele. Mappen `API` som ligger alle funksjoner i blir brukt i integrasjonstesting, men unntatt `apiAdminHeader` og `apiBankheader`. For å resette databasen til opprinnelig versjon etter hver endring brukes `InitDB.php` og `InitDB.sql` som ligger i `DAL` mappen. Filen `InitDB.php` resetter innholdet i selve databasen og har et innhold som aktiverer SQL koden, `InitDB.sql` som inneholder selve SQL koden. Derfor måtte dette prioriteres først.

For å teste hver eneste funksjon i administrasjons og kunden delen må man logges inn som admin eller kunde, derfor måtte egenskap verdiene prioritere logg inn brukernavn og passord. Ved å logge som admin kan man teste alle funksjonen som finnes i admin delen av systemet og ved loggin som kunde teste alle funksjoner som tilbys for kunden.

Etter integrasjonstesting, så startet vi med systemtesting. Systemtest er integrerte komponenter som skaper en versjon av systemet. Med andre ord her testes et integrert system. I systemtesting måtte vi starte med å spesifisere brukerhistorie for å kunne hva som skal testes. Her måtte vi som integrasjonstesting prioritere logg inn funksjon i `Selenium IDE` (plugin i nettleseren `Chrome`). Her måtte vi også gi egenskap verdiene som ga mulighet for å logge inn. Her testet vi funksjoner som er viktig for kunden og administrasjonen. Her blir også data test for å se på input validering. Det er etter bruk av `Selenium` at `MTM` (`Microsoft Test Management`)

System) brukes for spesifisere input data som er brukt i Selenium.

## 4. Fremgangsmåte

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

I enhetstesten, som er den første testen vi gjennomførte så ble det brukt en databasestub som speiler den egentlige databasen. Det blir gjort for å spare tid siden databasen stadig må resettes under en enhetstest. Altså, her testes det ikke på den egentlige databasen. Testen er automatisert og PHPUnit blir brukt sammen med Netbeans. Her er data som benyttet ulike verdier for å se om de virker slik som databasen hadde fungert. Altså, om verdiene i input fungerer med forventet output.

Integrasjonstesten blir gjennomført med SoapUI og flere metoder blir testet samtidig. F.eks. Logg inn og henteKonti. Testingen skjer i forhold til selve databasen hvor det blir brukt InitDB.php og InitDB.sql for å resette databasen når endringer har blitt gjort som en følge av testingen. Et eksempel på det er når man tester om det er mulig å endre kundeinformasjon. Siden norske bokstaven (å, æ og ø) er ikke lesbart i SoapUI skal de forandres i InitDB.sql med (a, ae og o). I alle testet blir brukt test dataene som er skrevet i testplanen for hver eneste test, og alle testene er automatisert og som allerede nevnt blir SoapUi brukt med HTTP protokoll.

Testingen skal først forekomme med brukerhistorie først og deretter skal dette resultere i hva som skal testes. Dermed er det brukerens krav som testes. Selenium og MTM blir brukt i systemtestingen, hvor Selenium er automatisert, mens bruk av MTM ikke er det. Selenium først, for så deretter å foreta MTM. Her skal MTM deretter gå Selenium etter i sømmene. I denne testen ble det undersøkt hvordan systemet fungerer for kunder av banksystemet og admin. Dermed test vi om brukerens input ga forventet output. Det forventet output her betyr også feilmeldinger som gis brukeren under validering.

### 4.1 Overordnet teststrategi for

- Enhetstest
- Integrasjonstest
- Systemtest

Målet med testingen er å kvalitetssikre programvare og finne så mange feil som mulig i systemet. Dette er viktig fordi system skal brukes av mennesker som vi vet

kan gjøre mange feil og at disse feilene kan få betydelige konsekvenser hvis de ikke

er rettet opp. Her kan feil ha store økonomiske og bli mer kostelig jo lengre ut i utviklingsprosessen man kommer. Derfor må testing utføres og for å sørge for at man går systematisk til verks så bør man ha en overordnet teststrategi.

Dette systemet er et system hvor man henter ut informasjon, setter inn informasjon og gjør endringer i informasjonen lagret i en database, så derfor er det viktig at det testes på disse operasjonene. Her utføres disse operasjoner av funksjoner. Derfor bør hver enkelt funksjon testes. Derfor begynner med en enhetstest og så en integrasjonstest og avslutter testingen med en systemtest. I testingen vår så vil feil vi finner, ikke bli endret eller korrigert/rettet opp.

Vi hadde som felles overordnet strategi for alle testene som vi foretok å kunne automatisere, dokumentere og teste så mye som mulig. Med automatisering kunne vi spare tid, ressurser og gå mer metodisk til verk. Dette ga oss bedre mulighet å dokumentere fremgang på hvordan vi hadde testet. Et viktig mål var å finne og rapportere feil. Dette er for at de vi leverer rapporten til kan se hva som forårsaket feilen og vurdere om programmeringen skal endres, slik at feilen unngås.

### Enhetstesten

Her har vi som strategi å gå gjennom alle funksjoner som brukes i adminLogikk og

bankLogikk i BLL mappen. Disse funksjonene er også koblet adminDatabase og bankdatabase, som forklarer opprettelsen av stubene i DAL mappen. Derfor måtte

disse også testes. Her var det viktig å også teste funksjoner som også allerede laget

i stuben. For eksempel ble det test med phpunit at funksjonen hentAlleKunder.

### Integrasjonstesten

Her satte vi opp en test plan i Excel, ut fra API mappen delte vi opp funksjoner i

Admin og Bank delen hvor vi satte opp forventede resultater av tester på de tre operasjonene, henting av informasjon, endring av informasjon og sette inn informasjon. Her ble avvik og godkjente tester registrert. Alle testene blir utført med

SoapUI som bruker HTTP protokoll for å automatisere.

### System testing

For oss hadde vi som strategi å oppfylle brukerens krav. Disse kravene ble delt opp i

administrasjon og kunde. Derfra tenkte vi på hvilken krav hver av disse gruppene kunne ha til nettstedet. Derfor laget vi bruker historier for hver av gruppene som resulterte i syv for admin og seks for kunde delen. Her delte vi opp admin og kunde

del, og hver av dem ble en feature i MTM. Deretter ble det laget en test suite per brukerhistorie. Feil vi fant i MTM for å kunne dokumentere.

## 4.2 Prosedyre for planlegging og gjennomføring av test

### Hvordan skal testene spesifiseres og hvordan skal de kjøres?

Enhetstesting ble planlagt med bruk av mappen DAL og BLL. DAL mappen har databasestubbene som er en gjenspeiling av databasen, og BLL har adminLogikk og bankLogikk som det ble planlagt testet på. De blir testet med adminTest og bankTest som består av "navnet til metoden"\_OK og "navnet til metoden"\_Feil. Der OK tilsvarer at testen skal gi OK som resultat og Feil skal gi Feil som resultat.

I integrasjonstesting så ble det ikke brukt en databasestub men selve databasen. Admin og Bank delen ble delt i to selvstendige tester. I Excel regneark ble det skrevet inn planlagte tester som skulle gjennomføres for både Admin og Bank. For Bank delen ble det laget tester for henting, registrering og endring. Det ble satt opp hva man forventet å få som resultat og det ble notert ned hvilke inndata som skal brukes. Et eksempel er logg inn, hvor man forventet å få OK hvis vellykket innlogging. I motsatt fall, om man skriver inn feil brukernavn eller passord, så forventer man å få melding "Feil i innloggingen". Inndata som ble brukt var

personnummer for brukernavn og eget passord registrert i databasen for passord. For Admin delen ble det gjort det samme, men i tillegg ble det laget test for sletting. Alle testene ble kjørt og resultatet ble sammenlignet med hva som stod i test planen. Om det var avvik så ble det registrert og merket med rød farge i Excel arket, i motsatt fall med grønn farge og testen gav forventet tilbakemelding. I systemtesting delte vi først bruker historiene opp i admin og kunde. Deretter så foretok vi en test for hver bruker historie. Her valgte vi å spesifisere testene med hva vi forventet i hver bruker historie. Altså, så testet vi hver brukerhistorie på forskjellige måter og dokumenterte om de tilsvarte til forventninger (da fikk de endelse OK i test casene), mens hvis de ikke svarte til forventningene eller ikke fungerte som forventet så fikk de i endelsen Feil). De ble kjørt ved at vi startet alltid med å logge oss inn og resette databasen. Kjøringen for MTM er gjort manuelt og dermed må hver steg beskrives.

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

Overordnet var det nødvendig med XamPP og nettleser med nettforbindelse. Kriteriene for at enhetstesten kunne kjøres er installering av Netbeans, PHPUnit og at man laget en admin databasestub for testing av admin delen og en tilsvarende for testing av bank delen, altså en bankdatabasestub. For integrasjonstesting var det nødvendig å installere SoapUI og ha tilgang på databasen i tillegg til API. Det var også nødvendig med initDB.sql og initDB.php for å gjenopprette databasen for både integrasjonstesting og systemtesting. For systemtestingen måtte vi bruke MTM (passord og brukernavn ble gitt av foreleser), i tillegg til Selenium sammen med Chrom.

### **Hva er kriteriene for at testene er kjørt?**

I enhetstesten så er indikasjonen på at testen gir et positivt resultat eller testen er vellykket en grønn Code coverage, ellers så vil Code coverage bli rød. Det siste vil indikere at testen mangler noe. I denne testen, en test av adminLogikk og bankLogikk, så forventet vi en Code coverage på over 95 %, mens adminDatabasestub og bankDatabasestub på 100%. Grunnen til at adminLogikk og bankLogikk er forventet at adminLogikk og bankLogikk kan ikke kobles til databasen, dermed det er umulig å få 100%.

Med SoapUI fikk vi en klar indikasjon på om testen var vellykket visualisert med



grønn farge for godkjent og rød for ikke godkjent, i tillegg fikk man en melding hvor det står “Failed”(med rød bakgrunn) om testen ikke var godkjent, og “Finished” (med grønn bakgrunn) om testen var godkjent, slik ble vi informert om at testen var kjørt. I systemtesting så vil Selenium antyde med grønn eller rød for at forventet resultat har forekommet. Grønn indikerer riktig, mens rød indikerer et annet resultat enn det forventede output. I MTM må dette gjøres manuelt for hver test case. Her må det spesifiseres om det er rød eller grønn.

### 4.3 Kriterier for grundighet i testen

På enhetstest-nivå bør det dokumenteres graden av dekning av kodelinjer og betingelser (“code coverage”). Ved integrasjonstest bør alle interne grensesnitt være utført (API’er). For systemtest bør det dokumenteres hvorvidt krav i spesifikasjonen / brukerhistorier er testet.

Kriteriet for grundighet i enhetstesten er å sjekke hva code coverage viser for å kunne danne seg et bilde av grundighet. Den vil indikere og vise i hvor stor grad det som blir testet fungerer.

I integrasjonstesten så ble alle APIene testet, med unntak av apiAdminHeader og apiBankheader og det bør dem, og det ble gjort mer enn en test på hver funksjon. Alle testene som skal endre noe i databasen bør har reset fordi at alle testene skal gjøres på den opprinnelig databasen.

I systemtesting, så har vi foretatt bruker historie for admin og kunde. Her har alle de forskjellige bruker historiene blitt testet. Dermed er alle brukerhistoriene blitt test, samt med forskjellige input data. Her er alle brukerhistoriene blitt dokumentert i MTM, samt om hvordan hver test er utført. Her er det også blitt redegjort om testen er vellykket eller har feilet.

## 5. Godkjenningskriterier

Et eksempel på et godkjenningskriteriene kan være:

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

Ved rapportering av feil, skal feil og avvik kategoriseres. Dette vil bli brukt som

underlag for godkjennings- og underkjenningskriterier av testene. Følgende kategorier kan f.eks benyttes for feil og avvik:

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

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

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

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

1. Enhetstesten er godkjent når 95 % eller mer av planlagt testing er utført og det er ingen åpne A feil.
2. Integrasjonstesten regnes som godkjent hvis alle interne grensesnitt (API'er) er utført og det er ingen åpne A og B feil.
3. System testen regnes som godkjent hvis alle brukerhistorier og deres respektive tester blir utført og det er ingen åpne A , B eller C feil.

## 6. Testmiljø

Beskriv de ulike testmiljøene og verktøy benyttet for de tre test typene.

1. NetBeans IDE Dev (201804200002)
2. SoapUI 5.4.0
3. MTM
4. PHPUnit v.6.5.9
5. XAMPP v3.2.2 [Compiled: Nov 12<sup>th</sup>2015]
6. Xdebug v2.6.1
7. Google Chrome version 72.0.3626.121 (Official Build) (64-bit)

## 8. Selenium ide 3.5.8 for chrome

Servere som ble brukt (database server):

1. Server: 127.0.0.1 via TCP/IP
2. Server type: MariaDB
3. Server version: 10.1.28-MariaDB - mariadb.org binary distribution
4. Protocol versjon:10
5. Bruker: root@localhost
6. Server charset: UTF-8 Unicode (utf8)

Web server:

1. Apache/2.4.28 (Win32) OpenSSL/1.0.2l PHP/7.1.10
2. Database client Version:libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: b396954eeb2d1d9ed7902b8bae237b287f21ad9e \$
3. PHP extension: mysqli, curl, mbstring
4. PHP version: 7.1.10

## 7. Resultat dokumenter

1. Sluttrapport
2. Testplan (integrasjonstesting)
3. Testprogrammer (Stubs, API.....)
4. Test manuskript
5. Brukerhistorier for systemtesting
6. Rapport (Code coverage)
7. Beskrivelse av inputdata og outputdata (Excel ark)
8. Testlogg (All dokumentasjon av testresultater + XML fil)

## 1. Enhetstesting

### TestingBank: Test Results

Test Results x Output - PHPUnit (TestingBank)

TestingBank x

Tests passed: 100.00 %

- testhentKonti\_DBStubFeil passed (0.002 s)
- testhentSaldi\_DBStubOK passed (0.004 s)
- testhentSaldi\_DBStubFeil passed (0.002 s)
- testregistrerBetaling\_DBStubOk passed (0.002 s)
- testregistrerBetaling\_DBStubFeil passed (0.002 s)
- testhentBetaling\_DBStubOk passed (0.002 s)
- testutforBetaling\_DBStubOk passed (0.005 s)
- testutforBetaling\_DBStubFeil passed (0.004 s)
- testendreKundeInfoOk passed (0.005 s)
- testhentKundeInfoOk passed (0.007 s)
- testhentKundeInfoFeil passed (0.002 s)
- test\_slettKundeOk passed (0.002 s)
- test\_slettKunde\_Feil passed (0.002 s)
- test\_registrerKundeOk passed (0.002 s)
- test\_registrerKundeFeil passed (0.002 s)
- test\_registrerKundeFeil1 passed (0.002 s)
- test\_registrerKundeTom passed (0.002 s)
- test\_hentEnKunde passed (0.002 s)
- test\_hentAlleKunderOk passed (0.004 s)
- testDatoFeilTransaksjoner passed (0.003 s)
- testIngenTransaksjoner passed (0.004 s)
- testEnTransaksjon passed (0.004 s)
- testToTransaksjoner passed (0.004 s)
- testAlleTransaksjoner passed (0.005 s)
- [PHPUnit] hentTransaksjonerTest passed
  - testDatoFeilTransaksjoner passed (0.002 s)
  - testIngenTransaksjoner passed (0.003 s)
  - testEnTransaksjon passed (0.003 s)
  - testToTransaksjoner passed (0.004 s)
  - testAlleTransaksjoner passed (0.005 s)

## Admin

### AdminTest.php

#### test\_hentAlleKunder

```
<?php
include_once '../Model/domeneModell.php';
include_once '../DAL/adminDatabaseStub.php';
include_once '../BLL/adminLogikk.php';

class adminTest extends PHPUnit\Framework\TestCase
{

function test_hentAlleKunderOk()
{
    // arrange
    $Admin= new Admin(new DBStub());
    // act
    $kunder= $Admin->hentAlleKunder();
    // assert
    $this->assertEquals("01010122344",$kunder[0]->personnummer);
    $this->assertEquals("Per",$kunder[0]->fornavn);
    $this->assertEquals("Hansen",$kunder[0]->etternavn);

    $this->assertEquals("Osloveien 82",$kunder[0]->adresse);
    $this->assertEquals("3270",$kunder[0]->postnr);
    $this->assertEquals("12345678",$kunder[0]->telefonnr);
    $this->assertEquals("HeiHei",$kunder[0]->passord);
    $this->assertEquals("Asker",$kunder[0]->poststed);

    $this->assertEquals("01010110523",$kunder[1]->personnummer);
    $this->assertEquals("Lene",$kunder[1]->fornavn);
    $this->assertEquals("Jensen",$kunder[1]->etternavn);

    $this->assertEquals("Akerveien 22",$kunder[1]->adresse);
    $this->assertEquals("", $kunder[1]->postnr);
    $this->assertEquals("22224444",$kunder[1]->telefonnr);
    $this->assertEquals("HeiHei",$kunder[1]->passord);
    $this->assertEquals("", $kunder[1]->poststed);
}
```

## testendreKundeInfo

```
public function testendreKundeInfoOk()
{
    $kunde1 = new kunde();
    $kunde1->fornavn = "Lene";
    $kunde1->etternavn = "Jensen";
    $kunde1->personnummer = "01010110523";
    $kunde1->adresse = "Askerveien 22";
    $kunde1->postnr = "";
    $kunde1->poststed = "Asker";
    $kunde1->telefonnr = "111111111";
    $kunde1->passord = "HeiHei";

    $kunde2 = new kunde();
    $kunde2->fornavn = "Per";
    $kunde2->etternavn = "Testsen";
    $kunde2->personnummer = "12345678901";
    $kunde2->adresse = "Osloveien 82";
    $kunde2->postnr = "1234";
    $kunde2->poststed = "";
    $kunde2->telefonnr = "12345678";
    $kunde2->passord = "HeiHei";

    $Admin= new Admin(new DBStub());
    // act
    $endreKundeInfoDBStub1 = $Admin->endreKundeInfo($kunde1);
    $endreKundeInfoDBStub2 = $Admin->endreKundeInfo($kunde2);
    // assert
    $this->assertEquals("", $endreKundeInfoDBStub1[0]->postnr);
    $this->assertEquals("Asker", $endreKundeInfoDBStub1[0]->poststed);
    $this->assertEquals("Lene", $endreKundeInfoDBStub1[0]->fornavn);
    $this->assertEquals("Jensen", $endreKundeInfoDBStub1[0]->etternavn);
    $this->assertEquals("Askerveien 22", $endreKundeInfoDBStub1[0]->adresse);
    $this->assertEquals("111111111", $endreKundeInfoDBStub1[0]->telefonnr);// Kunden
    byttet telefonnummer.
    $this->assertEquals("HeiHei", $endreKundeInfoDBStub1[0]->passord);
```

```

$this->assertEquals("01010110523", $endreKundeInfoIDBStub1[0]->personnummer);
$this->assertEquals("OK", $endreKundeInfoIDBStub1[1]);
$this->assertEquals("1234", $endreKundeInfoIDBStub2[0]->postnr);
$this->assertEquals("", $endreKundeInfoIDBStub2[0]->poststed);
$this->assertEquals("Per", $endreKundeInfoIDBStub2[0]->fornavn);
$this->assertEquals("Testsen", $endreKundeInfoIDBStub2[0]->etternavn); //Kunden
byttet etternavn.
$this->assertEquals("Osloveien 82", $endreKundeInfoIDBStub2[0]->adresse);
$this->assertEquals("12345678", $endreKundeInfoIDBStub2[0]->telefonnr);
$this->assertEquals("HeiHei", $endreKundeInfoIDBStub2[0]->passord);
$this->assertEquals("12345678901", $endreKundeInfoIDBStub2[0]->personnummer);
$this->assertEquals("OK", $endreKundeInfoIDBStub2[1]);
}

```

### test\_registrerKunde

```

function test_registrerKundeOk()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lene";
    $kunde->etternavn = "Jensen";
    $kunde->personnummer = "10123456723";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "3270";
    $kunde->poststed = "Asker";
    $kunde->telefonnr = "22224444";
    $kunde->passord = "HeiHei";

    $Admin = new Admin(new DBStub());
    // act
    $registrerKundeOk = $Admin->registrerKunde($kunde);
    // assert
    $this->assertEquals("OK", $registrerKundeOk);
}

```

//spør læreren

```

function test_registrerKunde_Feil()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lasae";
    $kunde->etternavn = "Jenassen";
    $kunde->personnummer = "12345678901";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "1";
    $kunde->poststed = "1";
    $kunde->telefonnr = "22224444";
}

```

```

    $kunde->passord = "HeiHei";

    $Admin = new Admin(new DBStub());
    // act
    $registrerKundeFeil = $Admin->registrerKunde($kunde);
    // assert
    $this->assertEquals("Feil",$registrerKundeFeil);
}

function test_registrerKunde_Feil1()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lasae";
    $kunde->etternavn = "Jenassen";
    $kunde->personnummer = "12345678901";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "3242";
    $kunde->poststed = "dgfd";
    $kunde->telefonnr = "22224444";
    $kunde->passord = "HeiHei";

    $Admin = new Admin(new DBStub());
    // act
    $registrerKundeFeil = $Admin->registrerKunde($kunde);
    // assert
    $this->assertEquals("Feil",$registrerKundeFeil);
}

function test_registrerKunde_tom()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lasae";
    $kunde->etternavn = "Jenassen";
    $kunde->personnummer = "12345690113";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "";
    $kunde->poststed = "";
    $kunde->telefonnr = "22224444";
    $kunde->passord = "HeiHei";

    $Admin = new Admin(new DBStub());
    // act
    $registrerKundeTom = $Admin->registrerKunde($kunde);
    // assert
    $this->assertEquals("Tom",$registrerKundeTom);
}

```



### **test\_slettKunde**

```
function test_slettKundeOK()
{
    // arrange
    $Admin = new Admin(new DBStub());
    $personnummer= 01010122344;
    // act
    $OK = $Admin->slettKunde($personnummer);
    // assert
    $this->assertEquals("OK",$OK);
}

function test_slettKundeFeil()
{
    // arrange
    $Admin = new Admin(new DBStub());
    $personnummer= 06556765757;
    // act
    $OK = $Admin->slettKunde($personnummer);
    // assert
    $this->assertEquals("Feil",$OK);
}
```

### **test\_registrerKonto**

```
function test_registrerKontoOk()
{
    // arrange
    $Admin = new Admin(new DBStub());
    $konto = new konto();
    $konto->kontonummer = "109010123456";
    $konto->personnummer = "12345678901";
    $konto->saldo = "656";
    $konto->type = "Lønnskonto";
    $konto->valuta = "NOK";
    // act
    $OK= $Admin->registrerKonto($konto);
    // assert
    $this->assertEquals("OK",$OK);
}

function test_registrerKontoFeil()
```

```

{
  $Admin = new Admin(new DBStub());
  $konto = new konto();
  $konto->kontonummer = "12345678902";
  $konto->personnummer = "12345678906";
  $konto->saldo = "576";
  $konto->type = "Lønnskonto";
  $konto->valuta = "NOK";
  // act
  $feil= $Admin->registrerKonto($konto);

  // assert
  $this->assertEquals("Feil personnummer",$feil);
}

```

```

function test_registrerKontoTom (){
  // arrange
  $konto =new konto();

  $Admin = new Admin(new DBStub());
  // act

  $OK= $Admin->registrerKonto($konto);
  // assert
  $this->assertEquals("Feil",$OK);
}

```

### **test\_endreKonto**

```

function test_endreKontoOk()
{
  // arrange
  $Admin = new Admin(new DBStub());
  $konto = new konto();
  $konto->kontonummer = "435432643279";
  $konto->personnummer = "12345678901";
  $konto->saldo = "76";
  $konto->type = "Lønnskonto";
  $konto->valuta = "NOK";
  // act
  $OK = $Admin->endreKonto($konto);
  // assert
  $this->assertEquals($OK, "OK");
}

```

```

function test_endreKontoFeilPersonnummer()
{

```

```

// arrange
$Admin = new Admin(new DBStub());
$konto = new konto();
$konto->kontonummer = "232343243256";
$konto->personnummer = "12345678901";
$konto->saldo = "76";
$konto->type = "Lønnskonto";
$konto->valuta = "NOK";
// act
$OK = $Admin->endreKonto($konto);
// assert
$this->assertEquals($OK, "Feil");
}

```

```

function test_endreKontoFeilKontonummer()
{
    // arrange
    $Admin = new Admin(new DBStub());
    $konto = new konto();
    $konto->kontonummer = "435432643279";
    $konto->personnummer = "12343575345";
    $konto->saldo = "76";
    $konto->type = "Lønnskonto";
    $konto->valuta = "NOK";
    // act
    $OK = $Admin->endreKonto($konto);
    // assert
    $this->assertEquals($OK, "Feil");
}

```

### **test\_slettKonto**

```

function test_slettKontoOk()
{
    // arrange
    $konto= new konto();
    $konto->kontonummer= "12343432344";
    $Admin = new DBStub(new DBStub());
    // act
    $OK = $Admin->slettKonto($konto->kontonummer);
    // assert
    $this->assertEquals("OK",$OK);
}

```

```

function test_slettKontoFeil()
{
    // arrange
    $Admin = new Admin(new DBStub());

```

```

    $konto= new konto();
    $konto->kontonummer= "342";
    // act
    $OK = $Admin->slettKonto($konto->kontonummer);
    // assert
    $this->assertEquals("Feil",$OK);
}

```

### **test\_hentAlleKonti**

```

function test_hentAlleKontiOk()
{
    // arrange
    $Admin= new Admin(new DBStub());
    // act
    $konti= $Admin->hentAlleKonti();
    // assert
    $this->assertEquals("105010123456",$konti[0]->kontonummer);
    $this->assertEquals("01010110523",$konti[0]->personnummer);
    $this->assertEquals("720",$konti[0]->saldo);
    $this->assertEquals("Lønnskonto",$konti[0]->type);
    $this->assertEquals("NOK",$konti[0]->valuta);

    $this->assertEquals("10502012345",$konti[1]->kontonummer);
    $this->assertEquals("01010110523",$konti[1]->personnummer);
    $this->assertEquals("100500",$konti[1]->saldo);
    $this->assertEquals("Sparekonto",$konti[1]->type);
    $this->assertEquals("NOK",$konti[1]->valuta);

    $this->assertEquals("22334412345",$konti[2]->kontonummer);
    $this->assertEquals("01010110523",$konti[2]->personnummer);
    $this->assertEquals("10234.5",$konti[2]->saldo);
    $this->assertEquals("Brukskonto",$konti[2]->type);
    $this->assertEquals("NOK",$konti[2]->valuta);
}
}

```

## **Bank**

### **bankTest.php**

### **TestsjekkLoggInn\_BankLogikk**

```

<?php
include_once '../Model/domeneModell.php';
include_once '../DAL/databaseStub.php';

```

```
include_once '../BLL/bankLogikk.php';
```

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

```
    public function testsjekkLoggInn_BankLogikk_FeilPersonnummer()
    {
        // arrange
        $feilpersonnummer = 1;
        $riktigpassord = "heihei";
        $bank = new Bank(new BankDBStub());
        // act
        $hentEnKundeMedFeilPersonnummer = $bank->sjekkLoggInn($feilpersonnummer,
        $riktigpassord);
        // assert
        $this->assertEquals("Feil i personnummer", $hentEnKundeMedFeilPersonnummer);
    }

    public function testsjekkLoggInn_BankLogikk_FeilPassord()
    {
        // arrange
        $riktigpersonnummer = "01010110523";
        $feilpassord = "????";
        $bank = new Bank(new BankDBStub());
        // act
        $hentEnKundeMedFeilPassord = $bank->sjekkLoggInn($riktigpersonnummer,
        $feilpassord);
        // assert
        $this->assertEquals("Feil i passord", $hentEnKundeMedFeilPassord);
    }
}
```

### **testsjekkLoggInn\_DBStub**

```
public function testsjekkLoggInn_DBStubOk()
{
    // arrange
    $riktigpersonnummer = "01010110523";
    $riktigpassord = "heihei";
    $bank = new Bank(new BankDBStub());
    // act
    $hentEnKundeMedriktiginfolDBStub = $bank->sjekkLoggInn($riktigpersonnummer,
    $riktigpassord);
    // assert
    $this->assertEquals("OK", $hentEnKundeMedriktiginfolDBStub);
}
```

```

public function testsjekkLoggInn_DBStubFeil()
{
    // arrange
    $riktigpersonnummer = "111111111111";
    $riktigpassord = "heihei";
    $bank = new Bank(new BankDBStub());
    // act
    $hentEnKundeMedfeilinfoIDBStub = $bank->sjekkLoggInn($riktigpersonnummer,
$riktigpassord);
    // assert
    $this->assertEquals("Feil", $hentEnKundeMedfeilinfoIDBStub);
}

```

### **testhentKonti\_DBStub**

```

function testhentKonti_DBStubOk()
{
    $riktigPersonnummer = "01010110523";
    $bank = new Bank(new BankDBStub());
    $hentKontiMedRiktigDBStub = $bank->hentKonti($riktigPersonnummer);
    $this->assertEquals("105010123456", $hentKontiMedRiktigDBStub[0]->kontonummer);
    $this->assertEquals("105020123456", $hentKontiMedRiktigDBStub[1]->kontonummer);
    $this->assertEquals("22334412345", $hentKontiMedRiktigDBStub[2]->kontonummer);
}

```

```

function testhentKonti_DBStubFeil()
{
    $feilPersonnummer = "111111111111";
    $bank = new Bank(new BankDBStub());
    $hentKontiMedFeilDBStub = $bank->hentKonti($feilPersonnummer);
    $this->assertEquals("", $hentKontiMedFeilDBStub);
}

```

### **testhentSaldi\_DBStub**

```

function testhentSaldi_DBStubOK()
{
    $riktigPersonnummer = "01010110523";
    $bank = new Bank(new BankDBStub());
    $hentSaldiMedRiktigIDBStub = $bank->hentSaldi($riktigPersonnummer);
    $this->assertEquals("01010110523", $hentSaldiMedRiktigIDBStub[0]->personnummer);
    $this->assertEquals("105010123456", $hentSaldiMedRiktigIDBStub[0]->kontonummer);
    $this->assertEquals("720", $hentSaldiMedRiktigIDBStub[0]->saldo);
    $this->assertEquals("Lønnskonto", $hentSaldiMedRiktigIDBStub[0]->type);
}

```

```

$this->assertEquals("NOK",$hentSaldiMedRiktigIDBStub[0]->valuta);
$this->assertEquals("01010110523",$hentSaldiMedRiktigIDBStub[1]->personnummer);
$this->assertEquals("105020123456",$hentSaldiMedRiktigIDBStub[1]->kontonummer);
$this->assertEquals("100500",$hentSaldiMedRiktigIDBStub[1]->saldo);
$this->assertEquals("Sparekonto",$hentSaldiMedRiktigIDBStub[1]->type);
$this->assertEquals("NOK",$hentSaldiMedRiktigIDBStub[1]->valuta);
$this->assertEquals("01010110523",$hentSaldiMedRiktigIDBStub[2]->personnummer);
$this->assertEquals("22334412345",$hentSaldiMedRiktigIDBStub[2]->kontonummer);
$this->assertEquals("10234.5",$hentSaldiMedRiktigIDBStub[2]->saldo);
$this->assertEquals("Brukskonto",$hentSaldiMedRiktigIDBStub[2]->type);
$this->assertEquals("NOK",$hentSaldiMedRiktigIDBStub[2]->valuta);
}

function testhentSaldi_DBStubFeil()
{
    $feilPersonnummer = "111111111111";
    $bank = new Bank(new BankDBStub());
    $hentSaldiMedFeilIDBStub = $bank->hentSaldi($feilPersonnummer);
    $this->assertEquals("", $hentSaldiMedFeilIDBStub);
}

```

### **testregistrerBetaling\_DBStub**

```

public function testregistrerBetaling_DBStubOk()
{
    // arrange
    $konto = "105010123456";
    $transaksjon = "1";
    $bank = new Bank(new BankDBStub());
    // act
    $registrerBetalingMedriktiginfolDBStub = $bank->registrerBetaling($konto,
    $transaksjon);
    // assert
    $this->assertEquals("OK", $registrerBetalingMedriktiginfolDBStub);
}

public function testregistrerBetaling_DBStubFeil()
{
    // arrange

```

```

    $konto = "1";
    $transaksjon = "11";
    $bank = new Bank(new BankDBStub());
    // act
    $registrerBetalingMedfeilinfoDBStub = $bank->registrerBetaling($konto, $transaksjon);
    // assert
    $this->assertEquals("Feil", $registrerBetalingMedfeilinfoDBStub);
}

```

### **testhentBetaling\_DBStub**

```

public function testhentBetaling_DBStubOk()
{
    // arrange
    $riktigPersonnummer1 = "01010110523";
    $riktigPersonnummer2 = "12345678901";
    $bank = new Bank(new BankDBStub());
    // act
    $hentBetalingMedOkinfoDBStub1 = $bank->hentBetaling($riktigPersonnummer1);
    $hentBetalingMedOkinfoDBStub2 = $bank->hentBetaling($riktigPersonnummer2);
    // assert
    $this->assertEquals("", $hentBetalingMedOkinfoDBStub1);
    $this->assertEquals("", $hentBetalingMedOkinfoDBStub2);
}

```

### **testutforBetaling\_DBStub**

```

public function testutforBetaling_DBStubOk()
{
    // arrange
    $TxID1 = "1";
    $TxID2 = "2";
    $TxID4 = "4";
    $TxID7 = "7";
    $TxID8 = "8";
    $TxID9 = "9";

    $bank = new Bank(new BankDBStub());
    // act
    $utforBetalingMedOkinfoDBStub1 = $bank->utforBetaling($TxID1);
    $utforBetalingMedOkinfoDBStub2 = $bank->utforBetaling($TxID2);
    $utforBetalingMedOkinfoDBStub3 = $bank->utforBetaling($TxID4);
    $utforBetalingMedOkinfoDBStub4 = $bank->utforBetaling($TxID7);
    $utforBetalingMedOkinfoDBStub5 = $bank->utforBetaling($TxID8);
    $utforBetalingMedOkinfoDBStub6 = $bank->utforBetaling($TxID9);
}

```



```

//assert
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub1[0]->avventer);
$this->assertEquals("-5074.2", $utforBetalingMedOKinfoIDBStub1[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub1[2]);
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub2[0]->avventer);
$this->assertEquals("-5575.1", $utforBetalingMedOKinfoIDBStub2[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub2[2]);
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub3[0]->avventer);
$this->assertEquals("-174.2", $utforBetalingMedOKinfoIDBStub3[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub3[2]);
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub4[0]->avventer);
$this->assertEquals("-8174.7", $utforBetalingMedOKinfoIDBStub4[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub4[2]);
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub5[0]->avventer);
$this->assertEquals("-5189.7", $utforBetalingMedOKinfoIDBStub5[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub5[2]);
$this->assertEquals("0", $utforBetalingMedOKinfoIDBStub6[0]->avventer);
$this->assertEquals("-5299.7", $utforBetalingMedOKinfoIDBStub6[1]->saldo);
$this->assertEquals("OK", $utforBetalingMedOKinfoIDBStub6[2]);
}

```

```

public function testutforBetaling_DBStubFeil()
{

```

```

    // arrange

```

```

    $TxID1 = Null;

```

```

    $TxID2 = "0";

```

```

    $TxID3 = "10";

```

```

    $TxID4 = "3";

```

```

    $TxID5 = "5";

```

```

    $TxID6 = "6";

```

```

    $bank = new Bank(new BankDBStub());

```

```

    // act

```

```

    $utforBetalingMedFeilinfoIDBStub1 = $bank->utforBetaling($TxID1);

```

```

    $utforBetalingMedFeilinfoIDBStub2 = $bank->utforBetaling($TxID2);

```

```

    $utforBetalingMedFeilinfoIDBStub3 = $bank->utforBetaling($TxID3);

```

```

    $utforBetalingMedFeilinfoIDBStub4 = $bank->utforBetaling($TxID4);

```

```

    $utforBetalingMedFeilinfoIDBStub5 = $bank->utforBetaling($TxID5);

```

```

    $utforBetalingMedFeilinfoIDBStub6 = $bank->utforBetaling($TxID6);

```

```

    // assert

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub1);

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub2);

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub3);

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub4[0]);

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub5[0]);

```

```

    $this->assertEquals("Feil", $utforBetalingMedFeilinfoIDBStub6[0]);

```

```

}

```

### testendreKundeInfo

```
public function testendreKundeInfoOk()
```

```
{
    $kunde1 = new kunde();
    $kunde1->fornavn = "Lene";
    $kunde1->etternavn = "Jensen";
    $kunde1->personnummer = "01010110523";
    $kunde1->adresse = "Askerveien 22";
    $kunde1->postnr = "";
    $kunde1->poststed = "Asker";
    $kunde1->telefonnr = "11111111"; // Kunden ville bytte telefon nummer.
    $kunde1->passord = "HeiHei";

    $kunde2 = new kunde();
    $kunde2->fornavn = "Per";
    $kunde2->etternavn = "Testmannen"; // Kunden ville bytte etternavn.
    $kunde2->personnummer = "12345678901";
    $kunde2->adresse = "Osloveien 82";
    $kunde2->postnr = "1234";
    $kunde2->poststed = "";
    $kunde2->telefonnr = "12345678";
    $kunde2->passord = "HeiHei";

    $bank = new Bank(new BankDBStub());
    // act
    $endreKundeInfoDBStub1 = $bank->endreKundeInfo($kunde1);
    $endreKundeInfoDBStub2 = $bank->endreKundeInfo($kunde2);
    // assert
    $this->assertEquals("", $endreKundeInfoDBStub1[0]->postnr);
    $this->assertEquals("Asker", $endreKundeInfoDBStub1[0]->poststed);
    $this->assertEquals("Lene", $endreKundeInfoDBStub1[0]->fornavn);
    $this->assertEquals("Jensen", $endreKundeInfoDBStub1[0]->etternavn);
    $this->assertEquals("Askerveien 22", $endreKundeInfoDBStub1[0]->adresse);
    $this->assertEquals("11111111", $endreKundeInfoDBStub1[0]->telefonnr);
    $this->assertEquals("HeiHei", $endreKundeInfoDBStub1[0]->passord);
    $this->assertEquals("01010110523", $endreKundeInfoDBStub1[0]->personnummer);
    $this->assertEquals("OK", $endreKundeInfoDBStub1[1]);
    $this->assertEquals("1234", $endreKundeInfoDBStub2[0]->postnr);
    $this->assertEquals("", $endreKundeInfoDBStub2[0]->poststed);
    $this->assertEquals("Per", $endreKundeInfoDBStub2[0]->fornavn);
    $this->assertEquals("Testmannen", $endreKundeInfoDBStub2[0]->etternavn);
    $this->assertEquals("Osloveien 82", $endreKundeInfoDBStub2[0]->adresse);
    $this->assertEquals("12345678", $endreKundeInfoDBStub2[0]->telefonnr);
    $this->assertEquals("HeiHei", $endreKundeInfoDBStub2[0]->passord);
    $this->assertEquals("12345678901", $endreKundeInfoDBStub2[0]->personnummer);
    $this->assertEquals("OK", $endreKundeInfoDBStub2[1]);
}
```

### testhentKundeInfo

```
public function testhentKundeInfoOk()
{
    $kunde1 = new kunde();
    $kunde1->fornavn = "Lene";
    $kunde1->etternavn = "Jensen";
    $kunde1->personnummer = "01010110523";
    $kunde1->adresse = "Askerveien 22";
    $kunde1->postnr = "3270";
    $kunde1->poststed = "Asker";
    $kunde1->telefonnr = "22224444";
    $kunde1->passord = "HeiHei";

    $bank = new Bank(new BankDBStub());
    // act
    $sendreKundeInfoDBStub1 = $bank->hentKundeInfo($kunde1->personnummer);

    // assert
    $this->assertEquals("3270", $sendreKundeInfoDBStub1[0]->postnr);
    $this->assertEquals("Asker", $sendreKundeInfoDBStub1[0]->poststed);
    $this->assertEquals("Lene", $sendreKundeInfoDBStub1[0]->fornavn);
    $this->assertEquals("Jensen", $sendreKundeInfoDBStub1[0]->etternavn);
    $this->assertEquals("Askerveien 22", $sendreKundeInfoDBStub1[0]->adresse);
    $this->assertEquals("22224444", $sendreKundeInfoDBStub1[0]->telefonnr);
    $this->assertEquals("HeiHei", $sendreKundeInfoDBStub1[0]->passord);
    $this->assertEquals("01010110523", $sendreKundeInfoDBStub1[0]->personnummer);
}

public function testhentKundeInfoFeil()
{
    $kunde1 = new kunde();
    $kunde1->fornavn = "Feil";
    $kunde1->etternavn = "Feilemann";
    $kunde1->personnummer = "";
    $kunde1->adresse = "Feileveien 72";
    $kunde1->postnr = "";
    $kunde1->poststed = "";
    $kunde1->telefonnr = "12345678";
    $kunde1->passord = "FeilFeil";
    $kunde2 = new kunde();
    $kunde2->fornavn = "Per";
    $kunde2->etternavn = "Hansen";
    $kunde2->personnummer = "12345678901";
    $kunde2->adresse = "Osloveien 82";
    $kunde2->postnr = "1234";
```

```

$kunde2->poststed = "";
$kunde2->telefonnr = "12345678";
$kunde2->passord = "HeiHei";

$bank = new Bank(new BankDBStub());
// act
$sendreKundeInfoDBStub1 = $bank->hentKundeInfo($kunde1->personnummer);
$sendreKundeInfoDBStub2 = $bank->hentKundeInfo($kunde2->personnummer);
// assert
$this->assertEquals("Feil", $sendreKundeInfoDBStub1);
$this->assertEquals("Feil", $sendreKundeInfoDBStub2);
}

```

### **test\_slettKunde**

function test\_slettKundeOk()

```

{
    // arrange
    $bank = new BankDBStub();
    $personnummer= 01010122344;
    // act
    $slettKundeOk = $bank->slettKunde($personnummer);
    // assert
    $this->assertEquals("OK",$slettKundeOk);
}

```

function test\_slettKunde\_Feil()

```

{
    // arrange
    $bank = new BankDBStub();
    $personnummer= 06556765757;
    // act
    $slettKundeFeil = $bank->slettKunde($personnummer);
    // assert
    $this->assertEquals("Feil",$slettKundeFeil);
}

```

### **test\_registrerKunde**

function test\_registrerKundeOk()

```

{

```

```

$kunde = new kunde();
$kunde->fornavn = "Lene";
$kunde->etternavn = "Jensen";
$kunde->personnummer = "10123456723";
$kunde->adresse = "Askerveien 22";
$kunde->postnr = "3270";
$kunde->poststed = "Asker";
$kunde->telefonnr = "22224444";
$kunde->passord = "HeiHei";

$bank = new BankDBStub();
// act
$registrerKundeOk = $bank->registrerKunde($kunde);
// assert
$this->assertEquals("OK",$registrerKundeOk);
}

```

```

function test_registrerKundeFeil()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lasae";
    $kunde->etternavn = "Jenassen";
    $kunde->personnummer = "12345678901";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "1";
    $kunde->poststed = "1";
    $kunde->telefonnr = "22224444";
    $kunde->passord = "HeiHei";

    $bank = new BankDBStub();
    // act
    $registrerKundeFeil = $bank->registrerKunde($kunde);
    // assert
    $this->assertEquals("Feil",$registrerKundeFeil);
}

```

```

function test_registrerKundeFeil1()
{
    $kunde = new kunde();
    $kunde->fornavn = "Lasae";
    $kunde->etternavn = "Jenassen";
    $kunde->personnummer = "12345678901";
    $kunde->adresse = "Askerveien 22";
    $kunde->postnr = "3242";
    $kunde->poststed = "dgfd";
    $kunde->telefonnr = "22224444";
    $kunde->passord = "HeiHei";
}

```

```

$bank = new BankDBStub();
// act
$registrerKundeFeil = $bank->registrerKunde($kunde);
// assert
$this->assertEquals("Feil",$registrerKundeFeil);
}

```

```

function test_registrerKundeTom()
{
    $kunde = new kunde();
    $kunde->fornavn = "";
    $kunde->etternavn = "";
    $kunde->personnummer = "";
    $kunde->adresse = "";
    $kunde->postnr = "";
    $kunde->poststed = "";
    $kunde->telefonnr = "";
    $kunde->passord = "";

    $bank = new BankDBStub();
    // act
    $registrerKundeTom = $bank->registrerKunde($kunde);
    // assert
    $this->assertEquals("Tom",$registrerKundeTom);
}

```

### **test\_hentEnKunde**

```

function test_hentEnKunde()
{
    $personnummer = "12345678901";
    // arrange
    $bank = new BankDBStub();
    // act
    $hentEnKunde = $bank->hentEnKunde($personnummer);
    // assert
    $this->assertEquals("12345678901",$hentEnKunde->personnummer);
    $this->assertEquals("Per Olsen",$hentEnKunde->navn);
    $this->assertEquals("Osloveien 82, 0270 Oslo",$hentEnKunde->adresse);
    $this->assertEquals("12345678",$hentEnKunde->telefonnr);
}

```

### **test\_hentAlleKunder**

```

function test_hentAlleKunderOk()

```

```

{
  // arrange
  $bank = new BankDBStub();
  // act
  $hentAlleKunder = $bank->hentAlleKunder();
  // assert
  $this->assertEquals("01010122344",$hentAlleKunder[0]->personnummer);
  $this->assertEquals("Per Olsen",$hentAlleKunder[0]->navn);
  $this->assertEquals("Osloveien 82 0270 Oslo",$hentAlleKunder[0]->adresse);
  $this->assertEquals("12345678",$hentAlleKunder[0]->telefonnr);
  $this->assertEquals("01010122344",$hentAlleKunder[1]->personnummer);
  $this->assertEquals("Line Jensen",$hentAlleKunder[1]->navn);
  $this->assertEquals("Askerveien 100, 1379 Asker",$hentAlleKunder[1]->adresse);
  $this->assertEquals("92876789",$hentAlleKunder[1]->telefonnr);
  $this->assertEquals("02020233455",$hentAlleKunder[2]->personnummer);
  $this->assertEquals("Ole Olsen",$hentAlleKunder[2]->navn);
  $this->assertEquals("Bærumsveien 23, 1234 Bærum",$hentAlleKunder[2]->adresse);
  $this->assertEquals("99889988",$hentAlleKunder[2]->telefonnr);
}

```

### **testDatoFeilTransaksjoner**

```

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);
}

```

### **testIngenTransaksjoner**

```

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);
}

```

### testEnTransaksjon

```

public function testEnTransaksjon()
{
    // arrange
    $kontoNr = "10502023523";
    $fraDato = '2015-03-26';
    $tilDato = '2015-03-26';
    $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);
}

```

### testToTransaksjoner

```

public function testToTransaksjoner()
{
    // arrange
    $kontoNr = "10502023523";
    $fraDato = '2015-03-27';
    $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);
$this->assertEquals('2015-03-27',$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);
}

```

### testAlleTransaksjoner

```

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);
}
}

```

## AdminLogikk.php Code Coverage

1	<?php	
2	include_once '../DAL/adminDatabase.php';	
3	//include_once '../DAL/adminDatabaseStub.php';	
4	class Admin	
5	{	
6	private \$db;	
7	function __construct(\$innDb=null)	
8	{	
9	if(\$innDb==null)	
10	{	
11	\$this->db=new AdminDB();	
12	}	
13	else	
14	{	
15	\$this->db=\$innDb;	
16	}	
17	}	
18		
19	function hentAlleKunder()	
20	{	
21	\$kunder= \$this->db->hentAlleKunder();	
22	return \$kunder;	
23	}	
24		
25	function endreKundeInfo(\$kunde)	
26	{	
27	\$OK= \$this->db->endreKundeInfo(\$kunde);	
28	return \$OK;	
29	}	
30		
31	function registrerKunde(\$kunde)	
32	{	
33	\$OK = \$this->db->registrerKunde(\$kunde);	
34	return \$OK;	
35	}	
36		
37	function slettKunde(\$personnummer)	
38	{	
39	\$OK = \$this->db->slettKunde(\$personnummer);	
40	return \$OK;	
41	}	
42		
43	function registrerKonto(\$konto)	
44	{	
45	\$OK = \$this->db->registerKonto(\$konto);	
46	return \$OK;	
47	}	
48		
49	function endreKonto(\$konto)	
50	{	
51	\$OK = \$this->db->endreKonto(\$konto);	
52	return \$OK;	
53	}	
54	function hentAlleKonti()	
55	{	
56	\$konti = \$this->db->hentAlleKonti();	
57	return \$konti;	
58	}	
59	function slettKonto(\$kontonummer)	
60	{	
61	\$OK = \$this->db->slettKonto(\$kontonummer);	
62	return \$OK;	
63	}	
64	}	

## BankLogikk.php Code Coverage

1	<?php	
2	include_once '../DAL/bankDatabase.php';	
3	include_once '../DAL/bankDatabaseStub.php';	
4	class Bank	
5	{	
6	private \$db;	
7	function __construct(\$innDb=null)	
8	{	
9	if (\$innDb==null)	
10	{	
11	\$this->db=new BankDB();	
12	}	
13	else	
14	{	
15	\$this->db=\$innDb;	
16	}	
17	}	
18		
19	function hentTransaksjoner(\$kontoNr, \$fraDato, \$tilDato)	
20	{	
21	\$konto= \$this->db->hentTransaksjoner(\$kontoNr, \$fraDato, \$tilDato);	
22	return \$konto;	
23	}	
24		
25	function sjekkLoggInn(\$personnummer,\$passord)	
26	{	
27	if (!preg_match("/[0-9]{11}/", \$personnummer))	
28	{	
29	return "Feil i personnummer";	
30	}	
31		
32	if (!preg_match("/.{6,30}/", \$passord))	
33	{	
34	return "Feil i passord";	
35	}	
36		
37	\$OK = \$this->db->sjekkLoggInn(\$personnummer,\$passord);	
38	return \$OK;	
39	}	
40		
41	function hentKonti(\$personnummer)	
42	{	
43	\$konti = \$this->db->hentKonti(\$personnummer);	
44	return \$konti;	
45	}	
46		
47	function hentSaldi(\$personnummer)	
48	{	
49	\$saldi = \$this->db->hentSaldi(\$personnummer);	
50	return \$saldi;	
51	}	
52		
53	function registrerBetaling(\$kontoNr, \$transaksjon)	
54	{	
55	\$ok = \$this->db->registrerBetaling(\$kontoNr, \$transaksjon);	
56	return \$ok;	
57	}	
58		
59	function hentBetalinger(\$personnummer)	
60	{	
61	\$betalinger = \$this->db->hentBetalinger(\$personnummer);	
62	return \$betalinger;	
63	}	
64		

```

65 function utforBetalning($TxID)
66 {
67     $ok = $this->db->utforBetalning($TxID);
68     return $ok;
69 }
70
71 function endreKundeInfo($kunde)
72 {
73     $OK= $this->db->endreKundeInfo($kunde);
74     return $OK;
75 }
76 function hentKundeInfo($personnummer)
77 {
78     $kunde= $this->db->hentKundeInfo($personnummer);
79     return $kunde;
80 }
81 }

```

### AdminDatabaseStub.php

```

<?php
include_once '../Model/domeneModell.php';
class DBStub
{
function hentAlleKunder(){
    $alleKunder = array();
    $kunde1 = new kunde();
    $kunde1->personnummer = "01010122344";
    $kunde1->fornavn = "Per";
    $kunde1->etternavn = "Hansen";
    $kunde1->adresse = "Osloveien 82";
    $kunde1->postnr="3270";
    $kunde1->telefonnr="12345678";
    $kunde1->passord="HeiHei";
    $kunde1->poststed="Asker";
    $alleKunder[]=$kunde1;
    $kunde2 = new kunde();
    $kunde2->personnummer = "01010110523";
    $kunde2->fornavn = "Lene";
    $kunde2->etternavn = "Jensen";
    $kunde2->adresse = "Akerveien 22";
    $kunde2->postnr=null;
    $kunde2->telefonnr="22224444";
    $kunde2->passord="HeiHei";
    $kunde2->poststed=null;
    $alleKunder[]=$kunde2;
    return $alleKunder;
}

function endreKundeInfo($kunde)
{
    if($kunde->postnr == "1234" ){
        $alleEndreInfo = array();
        $EndreInfo1 = new kunde();

```

```

$EndreInfo1->postnr = $kunde->postnr;
$EndreInfo1->poststed = $kunde->poststed;
$EndreInfo1->fornavn = $kunde->fornavn;
$EndreInfo1->etternavn = $kunde->etternavn;
$EndreInfo1->adresse = $kunde->adresse;
$EndreInfo1->telefonnr = $kunde->telefonnr;
$EndreInfo1->passord = $kunde->passord;
$EndreInfo1->personnummer = $kunde->personnummer;
$alleEndreInfo[] = $EndreInfo1;
$EndreInfo2 = "OK";
$alleEndreInfo[] = $EndreInfo2;
return $alleEndreInfo;
} else {
    $alleEndreInfo = array();
    $EndreInfo1 = new kunde();
    $EndreInfo1->postnr = $kunde->postnr;
    $EndreInfo1->poststed = $kunde->poststed;
    $EndreInfo1->fornavn = $kunde->fornavn;
    $EndreInfo1->etternavn = $kunde->etternavn;
    $EndreInfo1->adresse = $kunde->adresse;
    $EndreInfo1->telefonnr = $kunde->telefonnr;
    $EndreInfo1->passord = $kunde->passord;
    $EndreInfo1->personnummer = $kunde->personnummer;
    $alleEndreInfo[] = $EndreInfo1;
    $EndreInfo2 = "OK";
    $alleEndreInfo[] = $EndreInfo2;
    return $alleEndreInfo;
}
}

```

```

function registrerKunde($kunde)
{

    if($kunde->postnr == "1" && $kunde->poststed == "1" )
    {
        return "Feil";
    }

    if($kunde->personnummer == "12345678901")
    {
        return "Feil";
    }

    if($kunde->postnr == "" || $kunde->poststed == "" )
    {
        return "Tom";
    }
}

```

```
}  
return "OK";  
}
```

```
function slettKunde($personnummer)  
{  
    $kunde = new kunde();  
    $kunde->personnummer = 01010122344;  
    if($personnummer==$kunde->personnummer)  
    {  
        return "OK";  
    }  
    return "Feil";  
}
```

```
function registerKonto($konto)  
{  
    if($konto->personnummer == "12345678901")  
    {  
        return "OK";  
    } else if (count($konto->personnummer) && $konto->personnummer != "12345678901")  
{  
        return "Feil personnummer";  
    } else {  
        return "Feil";  
    }  
}
```

```
function endreKonto($konto)  
{  
  
    $konto1 = new konto();  
    $konto1->kontonummer ="435432643279";  
    $konto1->personnummer = "12345678901";  
  
    if($konto1->personnummer !=$konto->personnummer )  
    {  
  
        return "Feil";  
    }  
    if($konto1->kontonummer !=$konto->kontonummer)  
    {  
  
        return "Feil";  
    }  
}
```

```

    }

    else{
        return "OK";
    }
}

```

```

}

```

```

function slettKonto($kontonummer)
{
    $konto = new konto();
    $konto->kontonummer = "12343432344";
    if($konto->kontonummer==$kontonummer)
    {
        return "OK";
    } else {

        return "Feil";
    }
}

```

```

function hentAlleKonti(){
    $konti = array();
    $konto1 = new konto();
    $konto1->kontonummer = "105010123456";
    $konto1->personnummer ="01010110523";
    $konto1->saldo = "720";
    $konto1->type="Lønnskonto";
    $konto1->valuta="NOK";
    $konti[]=$konto1;

    $konto2 = new konto();
    $konto2->kontonummer = "10502012345";
    $konto2->personnummer ="01010110523";
    $konto2->saldo = "100500";
    $konto2->type="Sparekonto";
    $konto2->valuta="NOK";
    $konti[]=$konto2;

    $konto3 = new konto();
    $konto3->kontonummer = "22334412345";
    $konto3->personnummer ="01010110523";
    $konto3->saldo = "10234.5";
    $konto3->type="Brukskonto";
    $konto3->valuta="NOK";
    $konti[]=$konto3;
    return $konti;
}

```

```
}  
}
```

### **BankDatabaseStub**

```
<?php  
include_once '../Model/domeneModell.php';  
class BankDBStub  
{  
  
    function sjekkLoggInn($personnummer,$passord)  
    {  
        if($personnummer=="01010110523" && $passord == "heihei")  
        {  
            return "OK";  
        }  
        else  
        {  
            return "Feil";  
        }  
    }  
  
    function hentKonti($Personnummer)  
    {  
        if ($Personnummer == "01010110523"){  
            $alleHentKonti = array();  
            $hentkonti1 = new konto();  
            $hentkonti1->kontonummer = "105010123456";  
            $alleHentKonti[] = $hentkonti1;  
            $hentkonti2 = new konto();  
            $hentkonti2->kontonummer = "105020123456";  
            $alleHentKonti[] = $hentkonti2;  
            $hentkonti3 = new konto();  
            $hentkonti3->kontonummer = "22334412345";  
            $alleHentKonti[] = $hentkonti3;  
            return $alleHentKonti;  
        } else {  
            return "";  
        }  
    }  
  
    function hentSaldi($Personnummer)  
    {  
        if($Personnummer == "01010110523"){  
            $alleHentSaldi = array();  
            $hentSaldi1 = new konto();  
            $hentSaldi1->personnummer = "$Personnummer";
```



```

$hentSaldi1->kontonummer = "105010123456";
$hentSaldi1->saldo = "720";
$hentSaldi1->type = "Lønnskonto";
$hentSaldi1->valuta = "NOK";
$alleHentSaldi[] = $hentSaldi1;
$hentSaldi2 = new konto();
$hentSaldi2->personnummer = "$Personnummer";
$hentSaldi2->kontonummer = "105020123456";
$hentSaldi2->saldo = "100500";
$hentSaldi2->type = "Sparekonto";
$hentSaldi2->valuta = "NOK";
$alleHentSaldi[] = $hentSaldi2;
$hentSaldi3 = new konto();
$hentSaldi3->personnummer = "$Personnummer";
$hentSaldi3->kontonummer = "22334412345";
$hentSaldi3->saldo = "10234.5";
$hentSaldi3->type = "Brukskonto";
$hentSaldi3->valuta = "NOK";
$alleHentSaldi[] = $hentSaldi3;
return $alleHentSaldi;
} else {
    return "";
}
}

function registrerBetaling($kontoNr, $transaksjon)
{
    if($kontoNr == "105010123456" && $transaksjon == "1")
    {
        return "OK";
    }
    else
    {
        return "Feil";
    }
}

function hentBetaling($personnummer)
{
    $betaling = "";
    return $betaling;
}

function utforBetaling($TxID)
{
    if($TxID == 1){
        $allebetaling = array();
        $hentBetaling1 = new transaksjon();
    }
}

```

```

    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-5074.2";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 2){
    $allebetaling = array();
    $hentBetaling1 = new transaksjon();
    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-5575.1";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 3){
    $allebetaling = array();
    $hentBetaling3 = "Feil";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 4){
    $allebetaling = array();
    $hentBetaling1 = new transaksjon();
    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-174.2";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 5){
    $allebetaling = array();
    $hentBetaling3 = "Feil";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 6){
    $allebetaling = array();
    $hentBetaling3 = "Feil";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 7){
    $allebetaling = array();
    $hentBetaling1 = new transaksjon();

```

```

    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-8174.7";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 8){
    $allebetaling = array();
    $hentBetaling1 = new transaksjon();
    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-5189.7";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else if ($TxID == 9){
    $allebetaling = array();
    $hentBetaling1 = new transaksjon();
    $hentBetaling1->avventer = "0";
    $allebetaling[] = $hentBetaling1;
    $hentBetaling2 = new konto();
    $hentBetaling2->saldo = "-5299.7";
    $allebetaling[] = $hentBetaling2;
    $hentBetaling3 = "OK";
    $allebetaling[] = $hentBetaling3;
    return $allebetaling;
} else {
    return "Feil";
}
}

function endreKundeInfo($kunde)
{
    if($kunde->postnr == "1234" ){
        $alleEndreInfo = array();
        $EndreInfo1 = new kunde();
        $EndreInfo1->postnr = $kunde->postnr;
        $EndreInfo1->poststed = $kunde->poststed;
        $EndreInfo1->fornavn = $kunde->fornavn;
        $EndreInfo1->etternavn = $kunde->etternavn;
        $EndreInfo1->adresse = $kunde->adresse;
        $EndreInfo1->telefonnr = $kunde->telefonnr;
        $EndreInfo1->passord = $kunde->passord;
        $EndreInfo1->personnummer = $kunde->personnummer;
    }
}

```

```

        $alleEndreInfo[] = $EndreInfo1;
        $EndreInfo2 = "OK";
        $alleEndreInfo[] = $EndreInfo2;
        return $alleEndreInfo;
    } else {
        $alleEndreInfo = array();
        $EndreInfo1 = new kunde();
        $EndreInfo1->postnr = $kunde->postnr;
        $EndreInfo1->poststed = $kunde->poststed;
        $EndreInfo1->fornavn = $kunde->fornavn;
        $EndreInfo1->etternavn = $kunde->etternavn;
        $EndreInfo1->adresse = $kunde->adresse;
        $EndreInfo1->telefonnr = $kunde->telefonnr;
        $EndreInfo1->passord = $kunde->passord;
        $EndreInfo1->personnummer = $kunde->personnummer;
        $alleEndreInfo[] = $EndreInfo1;
        $EndreInfo2 = "OK";
        $alleEndreInfo[] = $EndreInfo2;
        return $alleEndreInfo;
    }
}

```

```

function hentKundeInfo($personnummer){
    if($personnummer == "01010110523"){
        $allehentKundeInfo = array();
        $hentKundeInfo = new kunde();
        $hentKundeInfo->postnr = "3270";
        $hentKundeInfo->poststed = "Asker";
        $hentKundeInfo->fornavn = "Lene";
        $hentKundeInfo->etternavn = "Jensen";
        $hentKundeInfo->adresse = "Askerveien 22";
        $hentKundeInfo->telefonnr = "22224444";
        $hentKundeInfo->passord = "HeiHei";
        $hentKundeInfo->personnummer = $personnummer;
        $allehentKundeInfo[] = $hentKundeInfo;
        return $allehentKundeInfo;
    } else if($personnummer == "12345678901") {
        return "Feil";
    } else {
        return "Feil";
    }
}

```

```

function registrerKunde($kunde)
{

if($kunde->postnr == "1" && $kunde->poststed == "1" )
{

```

```

        return "Feil";

    }
    if($kunde->personnummer == "12345678901")
    {

        return "Feil";
    }

    if($kunde->postnr == "" || $kunde->poststed == "" )
    {
        return "Tom";
    }
    return "OK";
}

function slettKunde($personnummer)
{
    $kunde = new kunde();
    $kunde->personnummer = 01010122344;
    if($personnummer == $kunde->personnummer)
    {
        return "OK";
    }
    return "Feil";
}

function hentEnKunde($personnummer)
{
    $enKunde = new kunde();
    $enKunde->personnummer =$personnummer;
    $enKunde->navn = "Per Olsen";
    $enKunde->adresse = "Osloveien 82, 0270 Oslo";
    $enKunde->telefonnr="12345678";
    return $enKunde;
}

function hentAlleKunder()
{
    $alleKunder = array();
    $kunde1 = new kunde();
    $kunde1->personnummer ="01010122344";
    $kunde1->navn = "Per Olsen";
    $kunde1->adresse = "Osloveien 82 0270 Oslo";
    $kunde1->telefonnr="12345678";
    $alleKunder[]=$kunde1;
    $kunde2 = new kunde();
    $kunde2->personnummer ="01010122344";

```

```

$kunde2->navn = "Line Jensen";
$kunde2->adresse = "Askerveien 100, 1379 Asker";
$kunde2->telefonnr="92876789";
$alleKunder[]=$kunde2;
$kunde3 = new kunde();
$kunde3->personnummer ="02020233455";
$kunde3->navn = "Ole Olsen";
$kunde3->adresse = "Bærumsveien 23, 1234 Bærum";
$kunde3->telefonnr="99889988";
$alleKunder[]=$kunde3;
return $alleKunder;
}

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;
}
}

```

## Code Coverage

Total Coverage: <span style="color: green;">21.38 %</span>				
Filename	Coverage	Total	Not Executed	
 adminDatabaseStub.php	<div><div style="width: 100%;">100.00 %</div></div>	104	0	0
 bankDatabaseStub.php	<div><div style="width: 100%;">100.00 %</div></div>	252	0	0
 bankLogikk.php	<div><div style="width: 96.15%;">96.15 %</div></div>	26	1	1
 adminLogikk.php	<div><div style="width: 95.00%;">95.00 %</div></div>	20	1	1
<b>Total</b>	<div><div style="width: 21.38%; background-color: red;">21.38 %</div></div>	1871	1471	

## 2. Integrasjonstesting

### TestPlan for Integrasjonstesting

#### Admin EndreKonto

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	Test Scenario	Som Admin vil jeg kunne endre konto info	Total Passed	5			In the "Status" column, use Pass, Fail, or Not Started.		
3	Test Environment	SoapUI 5.4.0	Total Failed	1			Totals and color highlighting will update automatically based on the Status column		
4	Test Data Source	<a href="http://localhost/TestingBank/API/adminEndreKonto.php">http://localhost/TestingBank/API/adminEndreKonto.php</a>	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	EK1	Gjenopprette Databasen	Databasen ble gjenopprettet	bruker:Admin password:Admin	vellykket	Pass	Esmaeel	25-02-19 2:10pm	
10	EK2	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin password:Admin	Admin	Pass	Esmaeel	25-02-19 2:10pm	
11	EK3	Tester om endring i konto er mulig	OK	kontonummer:105010123456 personnummer: 01010110523 saldo:1000 type:Lønnskonto valuta:NOK	OK	Pass	Esmaeel	25-02-19 2:10pm	
		Lar feltet være tom eller skriver feil personNr som ikke er registrert i Databasen	Feil i Personnummer	kontonummer:105010123456 personnummer: 10101010 saldo:8989 type:Lønnskonto valuta:NOK	Feil i Personnummer	Pass	Esmaeel	25-02-19 2:10pm	
12	EK4	Lar feltet være tom eller skriver feil KontonNr som ikke er registrert i Databasen	Feil i kontonummer	kontonummer:10101010 personnummer: 01010110523 saldo:8989 type:Lønnskonto valuta:NOK	Feil i kontonummer	Pass	Esmaeel	25-02-19 2:10pm	
		Lar Saldo, Type og Valuta feltetene være tomme	Feil	kontonummer:105010123456 personnummer: 01010110523 type: valuta: saldo: OK	OK	Fail	Esmaeel	25-02-19 2:10pm	Mangler feilhåndtering. Jeg fikk OK og HTML kode.
14	EK6								

## Admin EndreKunde

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	Test Scenario	Som Admin vil jeg kunne endre kundeinfo	Total Passed	3			In the "Status" column, use Pass, Fail, or Not Started.		
3	Test Environment	SoapUI 5.4.0	Total Failed	3			Totals and color highlighting will update automatically based on the Status column		
4	Test Data Source	<a href="http://localhost/TestingBank/API/adminEndreKunde.php">http://localhost/TestingBank/API/adminEndreKunde.php</a>	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	EKD1	Gjenopprette Databasen	Databasen ble gjenopprettet	bruker:Admin password:Admin	vellykket	Pass	Esmaeel	25-02-19 2:10pm	
10	EKD2	Logge inn med riktig bruker og passord	Admin er logget inn	personnummer:10101010101 Fornavn:Oskar Etternavn:Hansen adresse:Oslogata 5 Postnr:3270 Poststed:Oslo Telefonnr:67346743 Passord:HeiOf	Admin	Pass	Esmaeel	25-02-19 2:10pm	
11	EKD3	Alt riktig	OK	personnummer:22222222222 Fornavn:Oskar Etternavn:Hansen adresse:Oslogata 5 Postnr:3270 Poststed:Asker Telefonnr:67346743 Passord:HeiOf	OK	Pass	Esmaeel	25-02-19 2:10pm	
		Skriver feil personNr som ikke er registrert i Databasen	Feil i personnummer	personnummer:22222222222 Fornavn:Oskar Etternavn:Hansen adresse:Oslogata 5 Postnr:3270 Poststed:Asker Telefonnr:67346743 Passord:HeiOf	OK	Fail	Esmaeel	25-02-19 2:10pm	Får ingen feilmelding
12	EKD4	Lar personNr fletet være tom	Feil i personnummer	personnummer: Fornavn:Oskar Etternavn:Hansen adresse:Oslogata 5 Postnr:3270 Poststed:Asker Telefonnr:67346743 Passord:HeiOf	OK	Fail	Esmaeel	25-02-19 2:10pm	Får ingen feilmelding
		Lar alle felt være tomme	Feil i fornavn, etternavn, adresse, Postnr, Poststed, Telefonnr, Passord	personnummer:01010110523 Fornavn: Etternavn: adresse: Postnr: Poststed: Telefonnr: Passord:	OK	Fail	Esmaeel	25-02-19 2:10pm	Mangler feilhåndtering. Jeg fikk OK og HTML kode.
14									

## Admin hentkonti

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	Test Scenario	Som Admin vil jeg kunne se alle kontainfo	Total Passed	2			In the "Status" column, use Pass, Fail, or Not Started.		
3	Test Environment	SoapUI 5.4.0	Total Failed	0			Totals and color highlighting will update automatically based on the Status column		
4	Test Data Source	<a href="http://localhost/TestingBank/API/adminHentKonti.php">http://localhost/TestingBank/API/adminHentKonti.php</a>	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	HK1	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin password:Admin	Admin	Pass	Esmaeel	25-02-19 2:10pm	
10	HK2	Henter kontoinfo	Ingen test parameter	[{"Kontonummer":"105010123456","Personnummer":"01010110523","Saldo":"720","Type":"Lønnskonto","Valuta":"NOK"}, {"Kontonummer":"105020123456","Personnummer":"01010110523","Saldo":"100500","Type":"Sparekonto","Valuta":"NOK"}, {"Kontonummer":"22334412345","Personnummer":"01010110523","Saldo":"102345","Type":"Brukskonto","Valuta":"NOK"}]	[{"Kontonummer":"105010123456","Personnummer":"01010110523","Saldo":"720","Type":"Lønnskonto","Valuta":"NOK"}, {"Kontonummer":"105020123456","Personnummer":"01010110523","Saldo":"100500","Type":"Sparekonto","Valuta":"NOK"}, {"Kontonummer":"22334412345","Personnummer":"01010110523","Saldo":"102345","Type":"Brukskonto","Valuta":"NOK"}]	Pass	Esmaeel	25-02-19 2:10pm	

## Admin RegisterKonto



1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>			
2	Test Scenario	Som Admin vil jeg kunne registrere ny konto	Total Passed	6			In the "Status" column, use Pass, Fail, or Not Started.			
3	Test Environment	SoapUI 5.4.0	Total Failed	1			Totals and color highlighting will update automatically based on the Status column			
4	Test Data Source	http://localhost/TestingBank/API/adminRegisterKonto.php	Total Remaining	0						
5										
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	RK1	Gjenopprette Databasen	Databasen ble gjenopprettet		Databasen ble gjenopprettet	Pass	Esmaeel	25-02-19 2:10pm		
10	RK2	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin    passord:Admin	Admin er logget inn	Pass	Esmaeel	25-02-19 2:10pm		
11	RK3	Registrer ny konto	OK	personnummer:10101010101 type:Brukskonto valuta:US	OK	Pass	Esmaeel	25-02-19 2:10pm		
12	RK5	Prøver å registrere en konto med tomme felt Kontonummer,Saldo,Type,Valuta	Feil	kontonummer:0101010523    saldo:    type:    OK valuta:		Fail	Esmaeel	25-02-19 2:10pm	Mangler feilhåndtering. Burde ikke registreres ny konto i databasen med tommefelt	
13	RK5	Prøver å registrere kontonummer som eksisterer i databasen		200 personnummer:0101010523 type:Brukskonto valuta:US	Feil	Pass	Esmaeel	25-02-19 2:10pm	Her har jeg sjekket statuskode 200. som betyr at den har utført requesten	
14	RK5	Tom Personnummer	Feil i Personnummer	kontonummer:10101010101 personnummer:    saldo:8989    type:Lønnskonto valuta:NOK	Feil i kontonummer	Pass	Esmaeel	25-02-19 2:10pm		
15	RK5	Feil Personnummer	Feil i Personnummer	kontonummer:10101010101 personnummer:01010523    saldo:8989    type:Lønnskonto valuta:NOK	Feil i kontonummer	Pass	Esmaeel	25-02-19 2:10pm		

## Admin RegisterKunde

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>			
2	Test Scenario	Som Admin vil jeg kunne registrere ny kunde	Total Passed	5			In the "Status" column, use Pass, Fail, or Not Started.			
3	Test Environment	SoapUI 5.4.0	Total Failed	2			Totals and color highlighting will update automatically based on the Status column			
4	Test Data Source	http://localhost/TestingBank/API/adminRegisterKunde.php	Total Remaining	0						
5										
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	RK1	Gjenopprette Databasen	Databasen ble gjenopprettet		Databasen ble gjenopprettet	Pass	Esmaeel	25-02-19 2:10pm		
10	RK2	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin    passord:Admin	Admin er logget inn	Pass	Esmaeel	25-02-19 2:10pm		
11	RK3	Registrer ny kunde	OK	personnummer:10101010101 Fornavn:Oskar    Etternavn:Hansen adresse:Oslogata 5    Postnr:3270 Poststed:Asker    Telefonnr:67346743 Passord:HeiOf	OK	Pass	Esmaeel	25-02-19 2:10pm		
12	RK4	Tom personnummer	Feil	personnummer:    Fornavn:Oskar Etternavn:Hansen adresse:Oslogata 5 Postnr:3270    Poststed:Asker Telefonnr:67346743    Passord:HeiOf	Feil	Pass	Esmaeel	25-02-19 2:10pm		
13	RK4	Feil personnummer	Feil i personnummer	personnummer:22222222 Fornavn:Oskar    Etternavn:Hansen adresse:Oslogata 5    Postnr:3270 Poststed:Asker    Telefonnr:67346743 Passord:HeiOf	OK	Fail	Esmaeel	25-02-19 2:10pm	Burde ikke registreres kunde med ugyldig personnummer	
14	RK4	All Felt tom	Feil		OK	Fail	Esmaeel	25-02-19 2:10pm	Mangler feilhåndtering. Burde ikke registreres i databasen	
15	RK5	Prøver å registrere personnummer som eksisterer i databasen	Feil	personnummer:10101010101 Fornavn:Oskar    Etternavn:Hansen adresse:Oslogata 5    Postnr:3270 Poststed:Asker    Telefonnr:67346743 Passord:HeiOf	Feil	Pass	Esmaeel	25-02-19 2:10pm	Personnummere finnes allerede i databasen, så får feil melding	

## Admin SlettKonto

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>			
2	Test Scenario	Som Admin vil jeg kunne slette Konto	Total Passed	5			In the "Status" column, use Pass, Fail, or Not Started.			
3	Test Environment	SoapUI 5.4.0	Total Failed	0			Totals and color highlighting will update automatically based on the Status column			
4	Test Data Source	http://localhost/TestingBank/API/adminSlettKonto.php	Total Remaining	0						
5										
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	SK1	Gjenopprette Databasen	Databasen ble gjenopprettet		Databasen ble gjenopprettet	Pass	Esmaeel	25-02-19 2:10pm		
10	SK2	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin    passord:Admin	Admin er logget inn	Pass	Esmaeel	25-02-19 2:10pm		
11	SK3	Skriver kontonummer som finnes i databasen	OK	kontonummer:105010123456	OK	Pass	Esmaeel	25-02-19 2:10pm		
12	SK4	Lar Kontonummer felt være tom	Feil i kontonummer	kontonummer:	Feil i kontonummer	Pass	Esmaeel	25-02-19 2:10pm		
13	SK4	Skriver kontonummer som ikke finnes i databasen	Feil i kontonummer	kontonummer:111111111111	Feil i kontonummer	Pass	Esmaeel	25-02-19 2:10pm		

## Admin Logginn

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>			
2	Test Scenario	Som Admin vil jeg logge inn	Total Passed	3			In the "Status" column, use Pass, Fail, or Not Started.			
3	Test Environment	SoapUI 5.4.0	Total Failed	0			Totals and color highlighting will update automatically based on the Status column			
4	Test Data Source	http://localhost/TestingBank/API/logginnAdmin.php	Total Remaining	0						
5										
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	LI 1	Prøver å logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin    passord:Admin	Admin er logget inn	Pass	Esmaeel	25-02-19 2:10pm		
10	LI 2	Prøver å logge feil bruker og riktig passord	Feil i innlogging	bruker:Oskar    passord:Admin	Feil i innlogging	Pass	Esmaeel	25-02-19 2:15pm		
11	LI 3	Prøver å logge bruker og riktig passord	Feil i innlogging	bruker:Admin    passord:Oscar	Feil i innlogging	Pass	Esmaeel	01-01-14 1:40pm		

## Admin SlettKunde

1	Test Info			Test Summary			Instructions		
2	Test Scenario	Som Admin vil jeg kunne slette Kunde			Total Passed	3	In the "Status" column, use Pass, Fail, or Not Started.		
3	Test Environment	SoapUI 5.4.0			Total Failed	2	Totals and color highlighting will update automatically based on the Status column		
4	Test Data Source	<a href="http://localhost/TestingBank/API/adminSlettKunde.php">http://localhost/TestingBank/API/adminSlettKunde.php</a>			Total Remaining	0			
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	RK1	Gjenopprett Database	Databasen ble gjenopprettet		vellykket	Pass	Esmael	25-02-19 2:10pm	
10	RK2	Logge inn med riktig bruker og passord	Admin er logget inn	bruker:Admin password:Admin	Admin er logget inn	Pass	Esmael	25-02-19 2:10pm	
11	RK3	Slett kunde	OK	personnummer:01010110523	OK	Pass	Esmael	25-02-19 2:10pm	
12	RK5	Skriver personnummer som ikke finnes i databasen	Feil i personnummer	personnummer: 11111111111	OK	Fail	Esmael	25-02-19 2:10pm	Får ikke feilmelding
13	RK5	Lar personnummer felt være tom	Feil i personnummer	personnummer:	OK	Fail	Esmael	25-02-19 2:10pm	Får ikke feilmelding

## Admin HentAlleKunder

2	Test Info			Test Summary			Instructions			
3	Test Scenario	Som Admin vil jeg kunne se alle kundeinfo			Total Passed	2	In the "Status" column, use Pass, Fail, or Not Started.			
4	Test Environment	SoapUI 5.4.0			Total Failed	0	Totals and color highlighting will update automatically based on the Status column			
5	Test Data Source	<a href="http://localhost/TestingBank/API/hentAlleKunder.php">http://localhost/TestingBank/API/hentAlleKunder.php</a>			Total Remaining	0				
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	HK1	Logge inn med riktig bruker og passord Sjekk alle kundeinfo	Admin er logget inn	bruker:Admin password:Admin	Admin er logget inn	Pass	Esmael	25-02-19 2:10pm		
			<pre>ixz[{"Personnummer":"01010110523","Fornavn":"Lene","Ettermavn":"Jensen","Adresse":"A skerveien 22","Postnr":"3270","Telefonnr":"22234444","Passord":"HeiHei","Poststed":"Asker"}, {"Personnummer":"12345678901","Fornavn":"Per","Ettermavn":"Hansen","Adresse":"Osloveien 82","Postnr":null,"Telefonnr":"12345678","Passord":"HeiHei","Poststed":null}]</pre>	<pre>ixz[{"Personnummer":"01010110523","Fornavn":"Lene","Ettermavn":"Jensen","Adresse":"Askerveien 22","Postnr":"3270","Telefonnr":"22234444","Passord":"HeiHei","Poststed":"Asker"}, {"Personnummer":"12345678901","Fornavn":"Per","Ettermavn":"Hansen","Adresse":"Osloveien 82","Postnr":null,"Telefonnr":"12345678","Passord":"HeiHei","Poststed":null}]</pre>		Pass	Esmael	25-02-19 2:10pm		
10	HK2									

## Bank HentBetaling

2	Test Info			Test Summary		Instructions				
3	Test Scenario	Som kunde vil jeg se om betalingen er gjennomført og			Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started.			
4	Test Environment	SoapUI 5.5.0			Total Failed	1	Totals and color highlighting will update automatically based on the Status column			
5	Test Data Source	<a href="http://localhost/testingBank/API/hentBetaling.php">http://localhost/testingBank/API/hentBetaling.php</a>			Total Remaining	0				
6										
7										
8	Test Case ID	Test Case beskrivelse	Forventet resultat	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
9	HB-0001	Se om en betaling blir registrert	<pre>[[{"TxID": "10", "FraTilKontonummer": "20102012345", "Belop": "10000", "Dato": "2015-03-15", "Melding": "funkl", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lonnkonto", "Valuta": "NOK" }]</pre>	Personnummer: 01010110523	<pre>[[{"TxID": "10", "FraTilKontonummer": "20102012345", "Belop": "10000", "Dato": "2015-03-15", "Melding": "funkl", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lonnkonto", "Valuta": "NOK" }]</pre>	Pass	Ronny K.	25/02/19 15:00	Betalingen ble registrert.	
10	HB-0002	Se om man får en feilmelding om man taster inn feil personnummer	Feilmelding, feil personnr.	Personnummer 0101	Ingen feilmelding	Fail	Ronny K.	25/02/19 15:10	Fikk ingen feilmelding. Tiltross for feil personnummer ble dette returnert: <pre>[[{"TxID": "10", "FraTilKontonummer": "20102012345", "Belop": "10000", "Dato": "2015-03-15", "Melding": "funkl", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lonnkonto", "Valuta": "NOK" }]]</pre>	

## Bank HnetKonti

1	Test Info		Test Summary		Instructions				
2	Test Scenario	Se om kunde kan hente konti	Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started.				
3	Test Environment	SoapUI 5.5.0	Total Failed	0	Totals and color highlighting will update automatically based on the Status column				
4	Test Data Source	http://localhost/testingBank/API/hentKonti.php	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
	HK-0001	Kunde skal hente sine konti	[{"Kontonummer": "105010123456"}, {"Kontonummer": "105020123456"}, {"Kontonummer": "105030123456"}, {"Kontonummer": "22334412345"}]	Ingen testdata, ingen variabler	[{"Kontonummer": "105010123456"}, {"Kontonummer": "105020123456"}, {"Kontonummer": "105030123456"}, {"Kontonummer": "22334412345"}]	Pass	Ronny K.	22-2-19 2:30pm	Fikk forventet resultat.
9									

## Bank HnetKundeInfo

1	Test Info		Test Summary				Instructions			
2	Test Scenario	Kunde vil hente sin kundeinformasjon	Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started.					
3	Test Environment	SoapUI 5.5.0	Total Failed	0	Totals and color highlighting will update automatically based on the Status column					
4	Test Data Source	http://localhost/testingBank/API/hentKundeInfo.php	Total Remaining	0						
5										
6										
7										
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments	
	HKI-0001	Kunden skal hente sin kunde informasjon	{ "personnummer": "01010110523", "fornavn": "Lene", "etternavn": "Jensen", "adresse": "Askerveien 22", "postnr": "3270", "poststed": "Asker", "telefonnr": "22224444", "passord": "HeiHei" }	Ingen testdata, men på forhånd logget inn.	{ "personnummer": "01010110523", "fornavn": "Lene", "etternavn": "Jensen", "adresse": "Askerveien 22", "postnr": "3270", "poststed": "Asker", "telefonnr": "22224444", "passord": "HeiHei" }	Pass	Ronny K.	22-02-19 2:30pm		
9										

## Bank HentSaldi

1

2

3

4

5

6

7

8

9

Test Info		Test Summary		Instructions			
Test Scenario	Se om kunde kan hente sine saldi	Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started.			
Test Environment	SoapUI 5.5.0	Total Failed	0	Totals and color highlighting will update automatically based on the Status column			
Test Data Source	http://localhost/testingBank/API/hentSaldi.php	Total Remaining	0				

## Bank HentTrasaksjoner

1	Test Info		Test Summary		Instructions				
2	Test Scenario	Kunde skal kunne hente gjennomførte transaksjoner	Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started. Totals and color highlighting will update automatically based on the Status column				
3	Test Environment	SoapUI 5.5.0	Total Failed	1					
4	Test Data Source	http://localhost/testingBank/API/hentTransaksjoner	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	HT-0001	Kunde skal kunne hente og se sine transaksjoner	{ "personnummer": "01010110523", "kontonummer": "105010123456", "saldo": "720", "type": "Lønnskonto", "valuta": "NOK", "transaksjoner": [ { "fraTilKontonummer": "20102012345", "transaksjonBelop": "400.4", "belop": null, "dato": "2015-03-20", "melding": "Innebtaling", "avventer": null }, { "fraTilKontonummer": "20102012347", "transaksjonBelop": "-5000.5", "belop": null, "dato": "2016-03-20", "melding": "Utbetaling", "avventer": null } ] }	kontoNr: 105010123456, fraDato: 2015-03-20, tilDato: 2016-03-20	{ "personnummer": "01010110523", "kontonummer": "105010123456", "saldo": "720", "type": "Lønnskonto", "valuta": "NOK", "transaksjoner": [ { "fraTilKontonummer": "20102012345", "transaksjonBelop": "400.4", "belop": null, "dato": "2015-03-20", "melding": "Innebtaling", "avventer": null }, { "fraTilKontonummer": "20102012347", "transaksjonBelop": "-5000.5", "belop": null, "dato": "2016-03-20", "melding": "Utbetaling", "avventer": null } ] }	Pass	Ronny K.	21-02-19 1:30pm	
10	HT-0002	Kunde skriver inn feil kontonummer	Feilmelding: feil kontonummer	kontoNr: 1234, fraDato: 2015-03-20, tilDato: 2016-03-20	{ "personnummer": null, "kontonummer": "1234", "saldo": null, "type": null, "valuta": null, "transaksjoner": [] }	Fail	Ronny K.	21-02-19 1:35pm	Her ble det returnert en del HTML-kode i tillegg til dette: {"personnummer":null,"kontonummer":"1234","saldo":null,"type":null,"valuta":null,"transaksjoner":[]}. Det ble forventet at det ble returnert en feilmelding som: "Feil kontonummer", det ble det ikke.

## Bank loggut

1	Test Info		Test Summary		Instructions In the "Status" column, use Pass, Fail, or Not Started. Totals and color highlighting will update automatically based on the Status column				
2	Test Scenario	Teste om kunde kan logge ut og om man kan hente ko	Total Passed	2					
3	Test Environment	SoapUI	Total Failed	0					
4	Test Data Source	http://localhost/testingBank/Index.php	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	LU-0001	Teste om kunde er logget ut	Å komme til index siden for innlogging	Ingen	Kom til index siden og innlogging	Pass	Ronny K.	02-03-19 1:30pm	
10	LU-0002	Teste om man kan hente konti etter utlogging	Feil innlogging	Ingen	Feil innlogging	Pass	Ronny K.	01-03-14 1:35pm	

## Bank UtførBetaling

1	Test Info		Test Summary		Instructions				
2	Test Scenario	Kunde skal kunne se utførte betalinger	Total Passed	1	In the "Status" column, use Pass, Fail, or Not Started. Totals and color highlighting will update automatically based on the Status column				
3	Test Environment	SoapUI 5.5.0	Total Failed	1					
4	Test Data Source	http://localhost/testingBank/API/utførBetaling.php	Total Remaining	0					
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	UB-0001	Skal vise gjennomført betaling	{ "TxID": "10", "FraTilKontonummer": "22334412345", "Belop": "500", "Dato": "2015-03-17", "Melding": "funkt!", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lønnskonto", "Valuta": "NOK" }	TxID : 1	{ "TxID": "10", "FraTilKontonummer": "22334412345", "Belop": "500", "Dato": "2015-03-17", "Melding": "funkt!", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lønnskonto", "Valuta": "NOK" }	Pass	Ronny K.	27-02-19 1:30pm	Det ble returnert en gjennomført betaling.

UB-0002	Skal vise feilmelding når når transaksjons ID er feil.	Feil melding	TxID : -1	 <font size="1"><table class="xdebug-error xe-notice" dir="ltr" border="1" cellpadding="0" cellspacing="1"><tr><th align="left" bgcolor="#f57900" colspan="5"><span style="background-color: #cc0000; color: #fce94f; font-size: x-large;>( ! )</span>Notice: Trying to get property of non-object in C:\xampp\htdocs\TestingBank\DAL\bankDatabase.php on line <td>133</td></tr><tr><th align="left" bgcolor="#e6b96e" colspan="5">Call Stack</th></tr><tr><th align="center" bgcolor="#eeeeec">#</th><th align="left" bgcolor="#eeeeec">Tim	Fail	Ronny K.	29-02-19 1:35pm	Viste gjennomført betaling(se under) og en del HTML kode. [[ "TxID": "10", "FraTilkontonummer": "22334412345", "Belop": "500", "Dato": "2015-03-17", "Melding": "funkt", "Kontonummer": "105010123456", "Avventer": "1", "Personnummer": "01010110523", "Saldo": "720", "Type": "Lønnskonto", "Valuta": "NOK" ]]
---------	--	--------------	-----------	---	------	----------	-----------------	--

## Bank EndreKundeInfo

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	<b>Test Scenario</b>	Kunde skal kunne endre kundeinformasjon	<b>Total Passed</b>	1			In the "Status" column, use Pass, Fail, or Not Started.		
3	<b>Test Environment</b>	SoapUI	<b>Total Failed</b>	2			Totals and color highlighting will update automatically based on the Status column		
4	<b>Test Data Source</b>	<a href="http://localhost/testingBank/API/endreKundeInfo.php">http://localhost/testingBank/API/endreKundeInfo.php</a>	<b>Total Remaining</b>	0					
5									
6									
7									
8	<b>Test Case ID</b>	<b>Test Case Description</b>	<b>Expected Result</b>	<b>Test Data</b>	<b>Actual Result</b>	<b>Status</b>	<b>Tester</b>	<b>Date/Time</b>	<b>Comments</b>
9	EKI-0001	Skal teste om kunde info kan endres. Her endres fornavn og byttes ut med "Blablaba".	OK	personnummer: 01010110523, fornavn: Blablaba, etternavn: Jensen, adresse: Askerveien 22, postnr: 3270, poststed: Asker, telefonnr:22224444, passord: HeiHei	OK	Pass	Ronny K.	23-02-19 1:30pm	
10	EKI-0002	Se om man får feilmelding når ingen data blir tastet inn	Feil melding	Tomme felt	OK	Fail	Ronny K.	23-02-19 1:35pm	Her fikk man en del html kode i tillegg til "OK". Forventet å få en feilmelding, f.eks. tast inn navn, adresse etc. som du vil endre.
11	EKI-0003	Se om man får feilmelding med feil inn data	Feil melding	Feil personnummer: 010101	OK	Fail	Ronny K.	23-02-19 1:40pm	

## Bank RegisterBetaling

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	<b>Test Scenario</b>	Som kunde ønsker jeg å få registrert en betaling	<b>Total Passed</b>	1			In the "Status" column, use Pass, Fail, or Not Started.		
3	<b>Test Environment</b>	SoapUI	<b>Total Failed</b>	2			Totals and color highlighting will update automatically based on the Status column		
4	<b>Test Data Source</b>	<a href="http://localhost/testingBank/API/registerBetaling.php">http://localhost/testingBank/API/registerBetaling.php</a>	<b>Total Remaining</b>	0					
5									
6									
7									
8	<b>Test Case ID</b>	<b>Test Case Description</b>	<b>Expected Result</b>	<b>Test Data</b>	<b>Actual Result</b>	<b>Status</b>	<b>Tester</b>	<b>Date/Time</b>	<b>Comments</b>
9	RB-0001	Registrere en betaling	OK	tilkonto: 20102012345, dato: 2015-03-15, beløp: 10000, melding: funkt, kontoNr: 105010123456.	OK	Pass	Ronny K.	25-02-14 1:30pm	Betalig ble registrert.
10	RB-0002	Se om man får feilmelding om feltene er tomme	Feilmelding	Tomme felt	OK	Fail	Ronny K.	01-03-14 1:40pm	Fikk html kode og "OK", men ingen feil melding.
11	RB-0003	Se om man får feilmelding om inndata er feil	Feilmelding	Feil kontoNr: 123fg45, feil dato: 2020-03-05	OK	Fail	Ronny K.	03-03-19 2:30pm	Fikk ingen feilmelding for feil innskrevne data, men det ble returnert OK og en del html kode.

## Bank Logginn

1	<b>Test Info</b>		<b>Test Summary</b>				<b>Instructions</b>		
2	<b>Test Scenario</b>	Se om kunden kan logge seg inn i banken	<b>Total Passed</b>	3			In the "Status" column, use Pass, Fail, or Not Started.		
3	<b>Test Environment</b>	SoapUI	<b>Total Failed</b>	0			Totals and color highlighting will update automatically based on the Status column		
4	<b>Test Data Source</b>	<a href="http://localhost/testingBank/API/logginn.php">http://localhost/testingBank/API/logginn.php</a>	<b>Total Remaining</b>	0					
5									
6									
7									
8	<b>Test Case ID</b>	<b>Test Case Description</b>	<b>Expected Result</b>	<b>Test Data</b>	<b>Actual Result</b>	<b>Status</b>	<b>Tester</b>	<b>Date/Time</b>	<b>Comments</b>
9	LI-0001	Logge inn som bankkunde	OK	personnummer: 01010110523, passord: Heihei	OK	Pass	Ronny K.	23-02-19 1:30pm	Logg inn gikk bra
10	LI-0002	Se om kunde kan logg inn uten inndata	Feil i innlogging	Ingen testdata	Feil i innlogging	Pass	Ronny K.	23-02-19 1:35pm	Feilmelding når man forsøkte å logge inn uten å skrive inn personnummer og passord. Ble ikke logget inn. Her fikk man i tillegg noe html kode i tillegg til feilmelding.
11	LI-0003	Logge inn med feil personnummer og passord	Feil i innlogging	personnummer:12345, passord: bruker	Feil i innlogging	Pass	Ronny K.	23-02-19 1:40pm	Ble ikke logget inn. Fikk feilmelding når man forsøkte å logge inn med feil personnummer og passord.

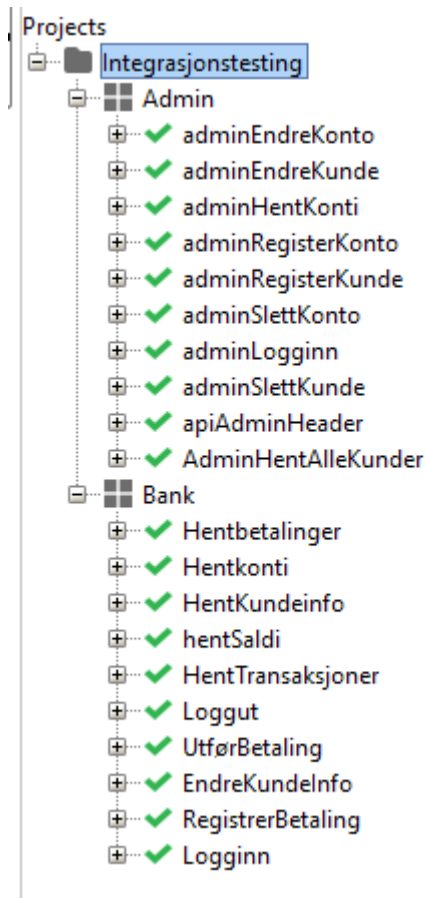
## Bank HenteBatalinger ( ikke tilgjengelig databasen)

1	Test Info		Test Summary				Instructions		
2	Test Scenario		Total Passed		0		In the "Status" column, use Pass, Fail, or Not Started.		
3	Test Environment		Total Failed		2		Totals and color highlighting will update automatically based on the Status column		
4	Test Data Source		Total Remaining		0				
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	HDBBIT-0001	Forsøker å logge inn for å hente betalinger.	Får ikke kontakt med banken, prøv igjen senere.	personnummer: 01010110523; passord:Heihei	Ingen returnerte data. Ingen feilmelding.	Fail	Ronny K.	23-02-19 1:30pm	Når man logger inn for å hente betalinger, så får man ingen tilbakemeldinger.
10	HDBBIT-0001	Hente betalinger når database ikke er tilgjengelig. Dette krever innlogging.	Får ikke kontakt med banken, prøv igjen senere.	personnummer: 01010110523	Ingen returnerte data. Ingen feilmelding.	Fail	Ronny K.	23-02-19 1:30pm	Når man forsøker å hente betalinger når database ikke er tilgjengelig, så får man ingen tilbake melding.

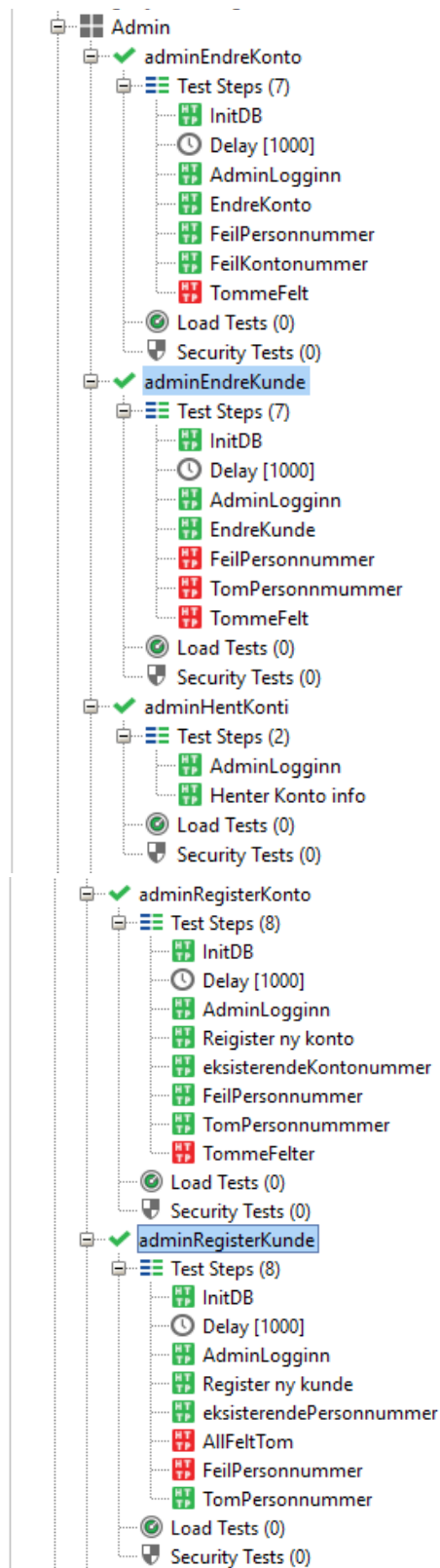
## Bank HenteKonti ( ikke tilgjengelig databasen)

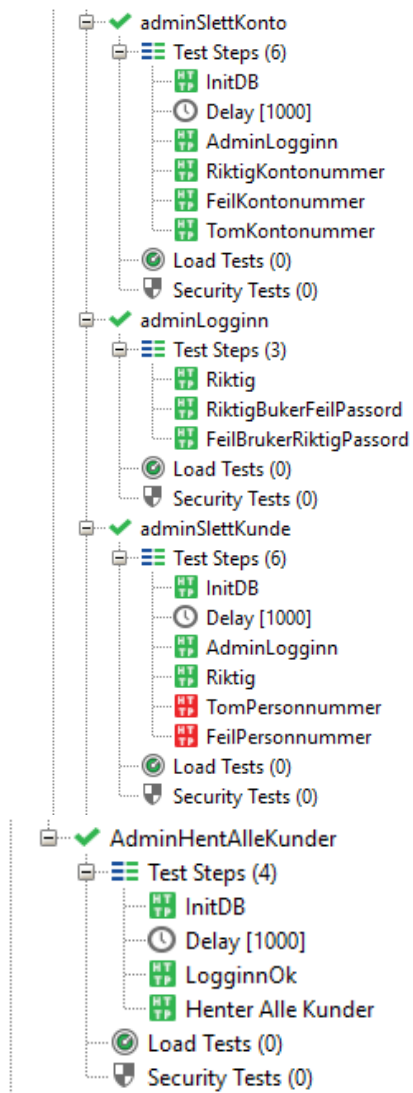
1	Test Info		Test Summary			Instructions			
2	Test Scenario		Total Passed			In the "Status" column, use Pass, Fail, or Not Started.			
3	Test Environment		Total Failed			Totals and color highlighting will update automatically based on the Status column.			
4	Test Data Source		Total Remaining						
5									
6									
7									
8	Test Case ID	Test Case Description	Expected Result	Test Data	Actual Result	Status	Tester	Date/Time	Comments
9	HKDBIT-0001	Forsøker å logge inn for å hente konti.	Får ikke kontakt med banken, prøv igjen senere.	personnummer: 01010110523, passord:Heihei	Ingen returnerte data. Ingen feilmelding.	Fail	Ronny K.	23-02-19 1:30pm	Når man forsøker å logge inn for å hente betalinger, og databasen ikke er tilgjengelig, så får man ingen tilbakemeldinger.
10	HKDBIT-0001	Hente konti når database ikke er tilgjengelig. Dette krever innlogging.	Får ikke kontakt med banken, prøv igjen senere.	Ingen.	Ingen returnerte data. Ingen feilmelding.	Fail	Ronny K.	23-02-19 1:30pm	Når man forsøker å hente konti når database ikke er tilgjengelig, så får man ingen tilbake melding.

## integrasjonstesting



## Admin



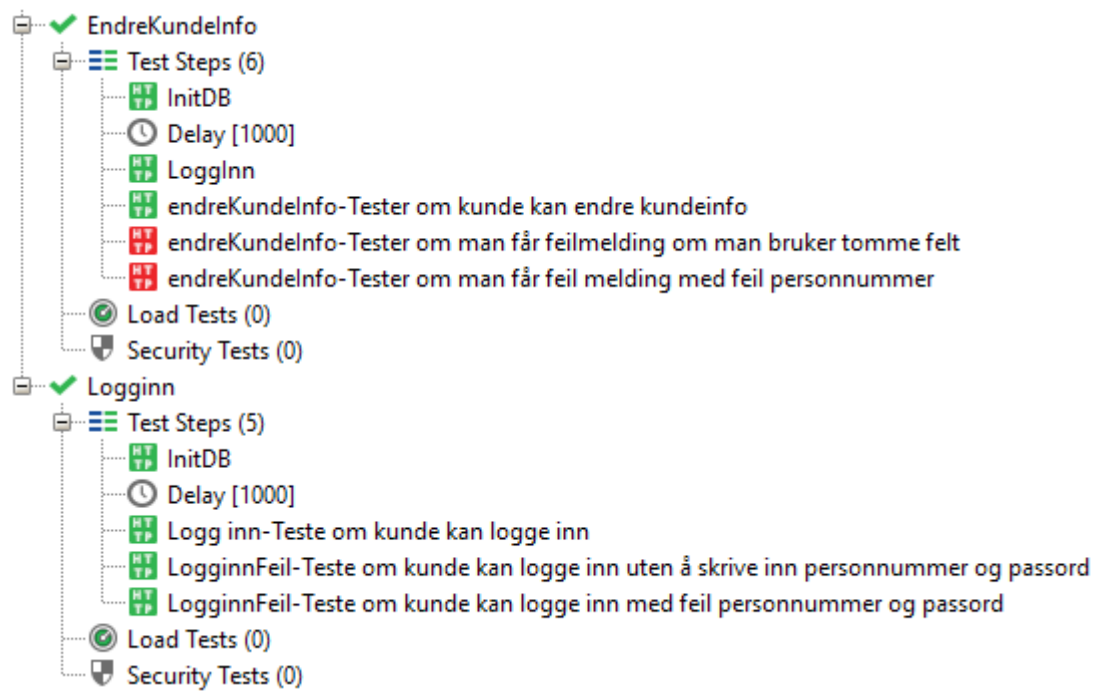


**Bank**









### 3. Systemtesting

Her er alle brukerhistoriene og testene som er resulterte av dem. Deretter er det Selenium IDE skjermdump og MTM bilder.

## Brukerhistorie for TestingBank

### 1. For Admin

1.1 Som Administrator ønsker jeg å kunne logge inn for å utføre operasjoner på nettstedet.

- 1.1.1 Admin: Logg Inn OK (Tom felt) (Forsøke å feile med vilje)
- 1.1.2 Admin: Logg Inn OK (Forsøke å feile med vilje)
- 1.1.3 Admin: Logg Inn OK (SQL injection 1)
- 1.1.4 Admin: Logg Inn OK (SQL injection 2)
- 1.1.5 Admin: Logg Inn OK

1.2 Som Administrator ønsker jeg å kunne registrere kunde for å opprette en ny kunde.

- 1.2.1 Admin: Registrere Kunde Feil (har ikke validering)
- 1.2.2 Admin: Registrere Kunde OK (Æ, Ø, Å)
- 1.2.3 Admin: Registrere Kunde OK

1.3 Som Administrator ønsker jeg å kunne endre kunde informasjon for å oppdatere kunde opplysninger.

- 1.3.1 Admin: Endre Kunde Informasjon Feil (Personnummer kan ikke endres, det er ingen validering, asker dukker opp som default på Poststed)
- 1.3.2 Admin: Endre Kunde Informasjon OK (Æ, Ø, Å)
- 1.3.3 Admin: Endre Kunde Informasjon OK

1.4 Som Administrator ønsker jeg å kunne slette en kunde for å fjerne en kunde.

- 1.4.1 Admin: Slette en Kunde OK (Forsøke å feile med vilje)
- 1.4.2 Admin: Slette en Kunde OK

1.5 Som Administrator ønsker jeg å kunne registrere konto for en kunde sånn at kunden kan foreta service og tjenestene nettstedet tilbyr.

- 1.5.1 Admin: Registrere Konto Feil (Kontonummer Feil, Hvis personnummer er riktig, så går det)
- 1.5.2 Admin: Registrere Konto OK (Personnummer Feil, lar ikke registrere)
- 1.5.3 Admin: Registrere Konto OK (lar ikke registrere, Ingen validering på Type eller Valuta)
- 1.5.4 Admin: Registrere Konto OK (Æ, Ø, Å)
- 1.5.5 Admin: Registrere Konto OK

1.6 Som Administrator ønsker jeg å kunne endre kunde konto opplysninger for å endre konto tilstanden til kunde.

- 1.6.1 Admin: Endre Kunde Konto Feil (Tom felt, 0 saldo og mangler validering)
- 1.6.2 Admin: Endre Kunde Konto Feil (Feil verdi, Aksepterer Fiktiv Valuta)
- 1.6.3 Admin: Endre Kunde Konto OK (Æ, Ø, Å)
- 1.6.4 Admin: Endre Kunde Konto OK

1.7 Som Administrator ønsker jeg å kunne slette en konto for å fjerne en konto for en kunde.

- 1.7.1 Admin: Slette en Konto OK (Forsøke å feile med vilje)
- 1.7.2 Admin: Slette en Konto OK

### 2. For Kunde

2.1 Som Kunde ønsker jeg å kunne logge inn for å utføre tjenester som er tilbudt av nettstedet.

- 2.1.1 Kunde: Logg Inn OK (Tom felt) (Forsoke å feile med vilje)
- 2.1.2 Kunde: Logg Inn OK (Forsoke å feile med vilje)
- 2.1.3 Kunde: Logg Inn OK (SQL injection 1)
- 2.1.4 Kunde: Logg Inn OK (SQL injection 2)
- 2.1.5 Kunde: Logg Inn OK
- 2.2 Som Kunde ønsker jeg å kunne se saldo for å se aktiviteter som er gjort på mine personnummer.
- 2.2.1 Kunde: Se Saldo OK
- 2.3 Som Kunde ønsker jeg å kunne registrere betalinger for å betale kunder av meg for tjenester eller service de har utført.
- 2.3.1 Kunde: Registrere Betalinger Feil (Feil Kontonummer, skal ikke virke)
- 2.3.2 Kunde: Registrere Betalinger Feil (Feil Betalingsdato, skal ikke virke)
- 2.3.3 Kunde: Registrere Betalinger Feil (Feil Betalingsdato og Feil Kontonummer, skal ikke virke)
- 2.3.4 Kunde: Registrere Betalinger Feil (Tom felt, skal ikke virke)
- 2.3.5 Kunde: Registrere Betalinger Feil (Minus saldo)
- 2.3.6 Kunde: Registrere Betalinger OK (Problem med å skrive 50.50 og nullen blir borte)
- 2.4 Som Kunde ønsker jeg å kunne se betalinger som er utført for å holde oversikt over betalinger gjort av meg som kunde.
- 2.4.1 Kunde: Se Betalinger OK
- 2.5 Som Kunde ønsker jeg å kunne se transaksjoner som er gjort for kontonummer for å vite hvilke transaksjoner som er gjort av meg som kunde.
- 2.5.1 Kunde: Se Transaksjoner Feil (Feil Fra dato, denne virker)
- 2.5.2 Kunde: Se Transaksjoner OK (Feil Til dato, virker ikke)
- 2.5.3 Kunde: Se Transaksjoner OK (Feil Fra dato og Feil Til dato) (Virker ikke)
- 2.5.4 Kunde: Se Transaksjoner OK
- 2.6 Som Kunde ønsker jeg å kunne endre min kunde informasjon for å oppdatere opplysningene mine.
- 2.6.1 Kunde: Endre Kunde Informasjon Feil (mangler validering)
- 2.6.2 Kunde: Endre Kunde Informasjon OK (Æ, Ø, Å)
- 2.6.3 Kunde: Endre Kunde Informasjon OK
- 2.6.4 Kunde: Endre Kunde Informasjon OK (Tilbake knappen virker)

## Selenium IDE

1.1.1 Admin: Logg Inn OK (Tom felt) (Forsoke å f... ⋮

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	
8	<i>assert text</i>	id=feilBruker	Bruker må være 2 til 20 tegn
9	<i>click</i>	css=html	
10	<i>click</i>	id=password	
11	<i>type</i>	id=password	
12	<i>assert text</i>	id=feilPassord	Minimum 4 tegn

### 1.1.2 Admin: Logg Inn OK (Forsoke å feile med vi...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	Bruker må være 2 til 20 tegn
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	css=html	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	Feilpassord
11	<i>click</i>	id=loggInn	
12	<i>assert text</i>	id=feilMelding	Feil personnummer / passord!

### 1.1.3 Admin: Logg Inn OK (SQL injection 1)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	" or ""="
7	<i>click</i>	css=html	
8	<i>click</i>	id=loggInn	
9	<i>click</i>	id=loggInn	
10	<i>click</i>	id=password	
11	<i>type</i>	id=password	" or ""="
12	<i>click</i>	id=loggInn	
13	<i>assert text</i>	id=feilMelding	Feil personnummer / passord!

### 1.1.4 Admin: Logg Inn OK (SQL injection 2)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	`Ole` - - _`
7	click	css=html	
8	click	id=loggInn	
9	click	css=body	
10	click	id=passord	
11	type	id=passord	`Ole` - - _`
12	click	id=loggInn	
13	assert text	id=feilMelding	Feil personnummer / passord!

### 1.1.5 Admin: Logg Inn OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	css=html	
8	click	id=passord	
9	type	id=passord	Admin
10	click	css=html	
11	click	id=loggInn	
12	click	css=html	

### 1.2.1 Admin: Registrere Kunde Feil (har ikke vali...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=passord	
8	type	id=passord	Admin
9	click	id=loggInn	
10	click	linkText=Registrer kunde	
11	assert text	linkText=Registrer kunde	Registrer kunde
12	click	id=personnummer	

13	<i>type</i>	id=personnummer	1
14	<i>assert text</i>	id=feilPersonnummer	Personnummeret må være 11 siffer
15	<i>click</i>	id=fornavn	
16	<i>type</i>	id=fornavn	1
17	<i>assert text</i>	id=feilFornavn	Fornavnet må være mellom 2 og 30 tegn
18	<i>click</i>	id=etternavn	
19	<i>type</i>	id=etternavn	1
20	<i>assert text</i>	id=feilEtternavn	Etternavn må være mellom 2 og 30 tegn
21	<i>type</i>	id=adresse	1
22	<i>assert text</i>	id=feilAdresse	Adresse må være mellom 2 og 30 tegn
23	<i>type</i>	id=postnr	1
24	<i>assert text</i>	id=feilPostnr	Postnr må være 4 siffer
25	<i>type</i>	id=poststed	1
26	<i>assert text</i>	id=feilPoststed	Poststed må være mellom 2 og 30 siffer
27	<i>type</i>	id=telefonnr	1
28	<i>assert text</i>	id=feilTelefonnr	Telefonnr må være 8 siffer
29	<i>type</i>	id=passord	1
30	<i>click</i>	id=regKunde	

## 1.2.2 Admin: Registrere Kunde OK (Æ, Ø, Å)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=passord	
8	<i>type</i>	id=passord	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Registrer kunde	
11	<i>assert text</i>	linkText=Registrer kunde	Registrer kunde
12	<i>click</i>	id=personnummer	
13	<i>type</i>	id=personnummer	1111111111
14	<i>click</i>	id=fornavn	
15	<i>type</i>	id=fornavn	Øyvind
16	<i>type</i>	id=etternavn	Åserud
17	<i>type</i>	id=adresse	Æreveien 34
18	<i>type</i>	id=postnr	0578
19	<i>type</i>	id=poststed	Oslo
20	<i>type</i>	id=telefonnr	22334455
21	<i>type</i>	id=passord	passord1
22	<i>click</i>	id=regKunde	
23	<i>click</i>	css=html	

## 1.2.3 Admin: Registrere Kunde OK



1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=password	
8	type	id=password	Admin
9	click	id=loggInn	
10	click	linkText=Registrer kunde	
11	assert text	linkText=Registrer kunde	Registrer kunde
12	click	id=personnummer	
13	type	id=personnummer	1111111111
14	click	id=fornavn	
15	type	id=fornavn	Ole
16	type	id=etternavn	Nordmann
17	type	id=adresse	Osloveien 34
18	type	id=postnr	0578
19	type	id=poststed	Oslo
20	type	id=telefonnr	22334455
21	type	id=password	password1
22	click	id=regKunde	
23	click	css=html	

### 1.3.1 Admin: Endre Kunde Informasjon Feil

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=password	
8	type	id=password	Admin
9	click	id=loggInn	
10	click	linkText=Endre kunde	
11	assert text	linkText=Endre kunde	Endre kunde
12	click	id=personnummer2	

13	<i>click</i>	id=fornavn2	
14	<i>type</i>	id=fornavn2	
15	<i>click</i>	id=etternavn2	
16	<i>type</i>	id=etternavn2	
17	<i>click</i>	id=adresse2	
18	<i>type</i>	id=adresse2	
19	<i>click</i>	id=postnr2	
20	<i>type</i>	id=postnr2	
21	<i>click</i>	id=poststed2	
22	<i>type</i>	id=poststed2	
23	<i>click</i>	id=telefonnr2	
24	<i>type</i>	id=telefonnr2	
25	<i>click</i>	id=password2	
26	<i>type</i>	id=password2	
27	<i>click</i>	css=tr:nth-child(2) .btn-success	
28	<i>assert text</i>	id=fornavn2	
29	<i>assert text</i>	id=etternavn2	
30	<i>assert text</i>	id=adresse2	
31	<i>assert text</i>	id=postnr2	
32	<i>assert text</i>	id=telefonnr2	
33	<i>assert text</i>	id=password2	

### 1.3.2 Admin: Endre Kunde Informasjon OK (Æ, Ø...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=password	
8	<i>type</i>	id=password	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Endre kunde	
11	<i>assert text</i>	linkText=Endre kunde	Endre kunde
12	<i>click</i>	id=fornavn1	

13	type	id=fornavn1	Øyvind
14	click	id=etternavn1	
15	type	id=etternavn1	Åserud
16	click	id=adresse1	
17	type	id=adresse1	Æreveien 56
18	click	id=postnr1	
19	type	id=postnr1	0578
20	click	id=poststed1	
21	type	id=poststed1	Oslo
22	click	id=telefonnr1	
23	type	id=telefonnr1	33224455
24	click	id=password1	
25	type	id=password1	password2
26	click	linkText=Endre	
27	assert text	linkText=Endre	Endre

### 1.3.3 Admin: Endre Kunde Informasjon OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=password	
8	type	id=password	Admin
9	click	id=loggInn	
10	click	linkText=Endre kunde	
11	assert text	linkText=Endre kunde	Endre kunde
12	click	id=fornavn1	
13	type	id=fornavn1	Tom
14	click	id=etternavn1	
15	type	id=etternavn1	Skogheim
16	click	id=adresse1	
17	type	id=adresse1	Bakkerud veien 56
18	click	id=postnr1	
19	type	id=postnr1	0578
20	click	id=poststed1	
21	type	id=poststed1	Oslo
22	click	id=telefonnr1	
23	type	id=telefonnr1	33224455
24	click	id=password1	

25	<i>type</i>	id=passord1	passord2
26	<i>click</i>	linkText=Endre	
27	<i>assert text</i>	linkText=Endre	Endre

### 1.4.1 Admin: Slette Kunde OK (Forsoke å feile m...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=passord	
9	<i>type</i>	id=passord	Admin
10	<i>click</i>	id=loggInn	
11	<i>choose cancel on next confirmation</i>		
12	<i>click</i>	css=tr:nth-child(2) .btn-danger	
13	<i>assert confirmation</i>	Slett kunde med personnummer 12345678901	
14	<i>webdriver choose cancel on visible confirmation</i>		
15	<i>assert text</i>	id=fornavn2	

### 1.4.2 Admin: Slette Kunde OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=passord	
9	<i>type</i>	id=passord	Admin
10	<i>click</i>	id=loggInn	
11	<i>choose ok on next confirmation</i>		
12	<i>click</i>	linkText=Slett	
13	<i>assert text</i>	linkText=Slett	Slett
14	<i>webdriver choose ok on visible confirmation</i>		

### 1.5.1 Admin: Registrere Konto Feil (Kontonumme...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=password	
9	<i>type</i>	id=password	Admin
10	<i>click</i>	id=login	
11	<i>click</i>	linkText=Register konto	
12	<i>assert text</i>	linkText=Register konto	Register konto
13	<i>click</i>	id=kontoNr	
14	<i>type</i>	id=kontoNr	a
15	<i>assert text</i>	id=feilKontonummer	Kontonummeret må være 11 siffer
16	<i>click</i>	id=type	
17	<i>type</i>	id=type	a
18	<i>click</i>	id=personnummer	
19	<i>type</i>	id=personnummer	01010110523
20	<i>click</i>	id=valuta	
21	<i>type</i>	id=valuta	a
22	<i>click</i>	css=.container:nth-child(3)	
23	<i>click</i>	id=regKonto	

## 1.5.2 Admin: Registrere Konto OK (Personnumm...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=password	
9	<i>type</i>	id=password	Admin
10	<i>click</i>	id=login	
11	<i>click</i>	linkText=Register konto	
12	<i>assert text</i>	linkText=Register konto	Register konto

13	click	id=kontoNr	
14	type	id=kontoNr	105010123456
15	click	id=type	
16	type	id=type	a
17	click	id=personnummer	
18	type	id=personnummer	a
19	assert text	id=feilPersonnummer	Personnummeret må være 11 siffer
20	click	id=valuta	
21	type	id=valuta	a
22	click	css=.container:nth-child(3)	
23	click	id=regKonto	
24	assert text	id=feilPersonnr	Personnummer finnes ikke!

### 1.5.3 Admin: Registrere Konto OK (Kontonumme...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	//		
5	click	linkText=Admin	
6	click	id=bruker	
7	type	id=bruker	Admin
8	click	id=passord	
9	type	id=passord	Admin
10	click	id=loggInn	
11	click	linkText=Register konto	
12	assert text	linkText=Register konto	Register konto
13	click	id=kontoNr	
14	type	id=kontoNr	a
15	assert text	id=feilKontonummer	Kontonummeret må være 11 siffer
16	click	id=type	
17	type	id=type	a
18	click	id=personnummer	
19	type	id=personnummer	a
20	assert text	id=feilPersonnummer	Personnummeret må være 11 siffer
21	click	id=valuta	
22	type	id=valuta	a
23	click	css=.container:nth-child(3)	
24	click	id=regKonto	
25	assert text	id=feilPersonnr	Personnummer finnes ikke!

### 1.5.4 Admin: Registrere Konto OK (Æ, Ø, Å)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=passord	
9	<i>type</i>	id=passord	Admin
10	<i>click</i>	id=loggInn	
11	<i>click</i>	linkText=Register konto	
12	<i>assert text</i>	linkText=Register konto	Register konto
13	<i>click</i>	id=kontoNr	
14	<i>type</i>	id=kontoNr	111111111111
15	<i>click</i>	id=type	
16	<i>type</i>	id=type	Øvrige kostnader
17	<i>click</i>	id=valuta	
18	<i>type</i>	id=valuta	Øvelsesvaluta
19	<i>click</i>	id=personnummer	
20	<i>type</i>	id=personnummer	01010110523
21	<i>click</i>	id=regKonto	

### 1.5.5 Admin: Registrere Konto OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	//		
5	<i>click</i>	linkText=Admin	
6	<i>click</i>	id=bruker	
7	<i>type</i>	id=bruker	Admin
8	<i>click</i>	id=passord	
9	<i>type</i>	id=passord	Admin
10	<i>click</i>	id=loggInn	
11	<i>click</i>	linkText=Register konto	
12	<i>assert text</i>	linkText=Register konto	Register konto

13	<i>click</i>	id=kontoNr	
14	<i>type</i>	id=kontoNr	11111111111
15	<i>click</i>	id=type	
16	<i>type</i>	id=type	Reisekonto
17	<i>click</i>	id=valuta	
18	<i>type</i>	id=valuta	NOK
19	<i>click</i>	id=personnummer	
20	<i>type</i>	id=personnummer	01010110523
21	<i>click</i>	id=regKonto	

### 1.6.1 Admin: Endre Kunde Konto Feil (Tom felt)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=passord	
8	<i>type</i>	id=passord	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Endre konto	
11	<i>assert text</i>	linkText=Endre konto	Endre konto
12	<i>click</i>	id=kontonummer1	
13	<i>type</i>	id=saldo1	
14	<i>type</i>	id=type1	
15	<i>type</i>	id=valuta1	
16	<i>click</i>	linkText=Endre	

### 1.6.2 Admin: Endre Kunde Konto Feil (Feil verdi, ...)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=passord	
8	<i>type</i>	id=passord	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Endre konto	
11	<i>assert text</i>	linkText=Endre konto	Endre konto
12	<i>click</i>	id=saldo1	



13	click	id=saldo1	
14	double click	id=saldo1	
15	click	id=saldo1	
16	type	id=saldo1	-50.50
17	click	id=type1	
18	type	id=type1	Reisekonto
19	click	id=valuta1	
20	click	id=valuta1	
21	double click	id=valuta1	
22	click	id=valuta1	
23	type	id=valuta1	Minegenvaluta
24	click	id=personnummer1	
25	click	linkText=Endre	

### 1.6.3 Admin: Endre Kunde Konto OK (Æ, Ø, Å)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=passord	
8	type	id=passord	Admin
9	click	id=loggInn	
10	click	linkText=Endre konto	
11	assert text	linkText=Endre konto	Endre konto
12	click	id=saldo1	
13	click	id=saldo1	
14	double click	id=saldo1	
15	click	id=saldo1	
16	type	id=saldo1	1500.00
17	click	id=type1	
18	type	id=type1	Øvrekostnader
19	click	id=valuta1	
20	click	id=valuta1	
21	double click	id=valuta1	
22	type	id=valuta1	Ørevaluta
23	click	linkText=Endre	

### 1.6.4 Admin: Endre Kunde Konto OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=passord	
8	<i>type</i>	id=passord	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Endre konto	
11	<i>assert text</i>	linkText=Endre konto	Endre konto
12	<i>click</i>	id=saldo1	
13	<i>click</i>	id=saldo1	
14	<i>double click</i>	id=saldo1	
15	<i>click</i>	id=saldo1	
16	<i>type</i>	id=saldo1	1500.00
17	<i>click</i>	id=type1	
18	<i>type</i>	id=type1	Reisekonto
19	<i>click</i>	id=valuta1	
20	<i>click</i>	id=valuta1	
21	<i>double click</i>	id=valuta1	
22	<i>type</i>	id=valuta1	EURO
23	<i>click</i>	linkText=Endre	

### 1.7.1 Admin: Slette en Konto OK (Forsoke å feile ...)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>click</i>	linkText=Admin	
5	<i>click</i>	id=bruker	
6	<i>type</i>	id=bruker	Admin
7	<i>click</i>	id=passord	
8	<i>type</i>	id=passord	Admin
9	<i>click</i>	id=loggInn	
10	<i>click</i>	linkText=Endre konto	
11	<i>assert text</i>	linkText=Endre konto	Endre konto
12	<i>choose cancel on next confirmation</i>		
13	<i>click</i>	linkText=Slett	
14	<i>assert text</i>	linkText=Slett	Slett
15	<i>assert confirmation</i>	Slett kunde med kontonummer 105010123456	
16	<i>webdriver choose cancel on visible confirmation</i>		

### 1.7.2 Admin: Slette en Konto OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	click	linkText=Admin	
5	click	id=bruker	
6	type	id=bruker	Admin
7	click	id=password	
8	type	id=password	Admin
9	click	id=loggInn	
10	click	linkText=Endre konto	
11	assert text	linkText=Endre konto	Endre konto
12	choose ok on next confirmation		
13	click	linkText=Slett	
14	webdriver choose ok on visible confirmation		
15	click	css=html	
16	click	css=html	
17	double click	css=html	

### 2.1.1 Kunde: Logg Inn OK (Tom felt) (Forsoke å f...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	
8	assert text	id=feilPersonNr	Personnummer nå være 11 siffer
9	click	id=password	
10	type	id=password	
11	assert text	id=feilPassord	Minimum 6 tegn
12	click	id=loggInn	
13	assert text	id=feilMelding	Feil personnummer / passord!

### 2.1.2 Kunde: Logg Inn OK (Forsoke å feile med vi...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	type	id=password	Feilpassord
10	click	id=loggInn	
11	assert text	id=feilMelding	Feil personnummer / passord!

### 2.1.3 Kunde: Logg Inn OK (SQL injection 1)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	" or ""="
8	assert text	id=feilPersonNr	Personnummer nå være 11 siffer
9	click	id=passord	
10	click	id=loggInn	
11	click	id=passord	
12	type	id=passord	" or ""="
13	click	id=loggInn	
14	assert text	id=feilMelding	Feil personnummer / passord!

### 2.1.4 Kunde: Logg Inn OK (SQL injection 2)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	`Ole` - - _ `
8	assert text	id=feilPersonNr	Personnummer nå være 11 siffer
9	click	id=passord	
10	click	id=loggInn	
11	click	id=passord	
12	type	id=passord	`Ole` - - _ `
13	click	id=loggInn	
14	assert text	id=feilMelding	Feil personnummer / passord!

### 2.1.5 Kunde: Logg Inn OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	click	id=password	
10	type	id=password	HeiHei
11	click	id=loggInn	
12	click	css=th:nth-child(1)	
13	assert text	css=th:nth-child(1)	Kontonr

### 2.2.1 Kunde: Se Saldo OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	click	id=password	
10	type	id=password	HeiHei
11	click	id=loggInn	
12	click	linkText=Saldo	
13	assert text	linkText=Saldo	Saldo
14	click	css=html	

### 2.3.1 Kunde: Registrere Betalinger Feil (Feil Kont...)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	click	id=password	
10	type	id=password	HeiHei
11	click	id=loggInn	
12	click	linkText=Register betaling	

### 2.3.2 Kunde: Registrere Betalinger Feil (Feil Beta...)

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	click	id=password	
10	type	id=password	HeiHei
11	click	id=loggInn	
12	click	linkText=Register betaling	
13	assert text	linkText=Register betaling	Register betaling
14	click	id=dato	
15	type	id=dato	A
16	assert text	id=feilDato	Dato må være på formatet yyyy-mm-dd
17	click	id=tilKonto	
18	type	id=tilKonto	105010123456
19	mouse down	id=belop	
20	mouse up	css=.col-md-5:nth-child(4) > label	
21	type	id=belop	A
22	click	id=melding	
23	type	id=melding	A
24	click	css=.container:nth-child(3)	
25	click	id=regBetaling	
26	click	css=.btn	

### 2.3.3 Kunde: Registrere Betalinger Feil (Feil Beta...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=password	
9	click	id=password	
10	type	id=password	HeiHei
11	click	id=loggInn	
12	click	linkText=Register betaling	

13	<i>assert text</i>	linkText=Register betaling	Register betaling
14	<i>click</i>	id=dato	
15	<i>type</i>	id=dato	A
16	<i>assert text</i>	id=feilDato	Dato må være på formatet yyyy-mm-dd
17	<i>click</i>	id=tilKonto	
18	<i>type</i>	id=tilKonto	A
19	<i>assert text</i>	id=feilTilKontonummer	Kontonummeret må være 11 siffer
20	<i>mouse down</i>	id=belop	
21	<i>mouse up</i>	css=.col-md-5:nth-child(4) > label	
22	<i>type</i>	id=belop	A
23	<i>click</i>	id=melding	
24	<i>type</i>	id=melding	A
25	<i>click</i>	css=.container:nth-child(3)	
26	<i>click</i>	id=regBetaling	
27	<i>click</i>	css=.btn	

### 2.3.4 Kunde: Registrere Betalinger Feil (Tom felt)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=passord	
9	<i>click</i>	id=passord	
10	<i>type</i>	id=passord	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Register betaling	
13	<i>assert text</i>	linkText=Register betaling	Register betaling
14	<i>click</i>	id=dato	
15	<i>type</i>	id=dato	
16	<i>assert text</i>	id=feilDato	Dato må være på formatet yyyy-mm-dd
17	<i>click</i>	id=tilKonto	
18	<i>type</i>	id=tilKonto	
19	<i>assert text</i>	id=feilTilKontonummer	Kontonummeret må være 11 siffer
20	<i>mouse down</i>	id=belop	
21	<i>mouse up</i>	css=.col-md-5:nth-child(4) > label	
22	<i>type</i>	id=belop	
23	<i>click</i>	id=melding	
24	<i>type</i>	id=melding	
25	<i>click</i>	css=.container:nth-child(3)	
26	<i>click</i>	id=regBetaling	
27	<i>click</i>	css=.btn	

### 2.3.5 Kunde: Registrere Betalinger Feil (Minus sa...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Register betaling	
13	<i>assert text</i>	linkText=Register betaling	Register betaling
14	<i>click</i>	id=dato	
15	<i>type</i>	id=dato	2010-10-10
16	<i>click</i>	id=tilKonto	
17	<i>type</i>	id=tilKonto	105010123456
18	<i>mouse down</i>	id=belop	
19	<i>mouse up</i>	css=.col-md-5:nth-child(4) > label	
20	<i>type</i>	id=belop	-100
21	<i>click</i>	id=melding	
22	<i>type</i>	id=melding	Transaksjon1
23	<i>click</i>	css=.container:nth-child(3)	
24	<i>click</i>	id=regBetaling	
25	<i>click</i>	css=.btn	

## 2.3.6 Kunde: Registrere Betalinger OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Register betaling	



13	<i>assert text</i>	linkText=Register betaling	Register betaling
14	<i>click</i>	id=dato	
15	<i>type</i>	id=dato	2010-10-10
16	<i>click</i>	id=tilKonto	
17	<i>type</i>	id=tilKonto	105010123456
18	<i>mouse down</i>	id=belop	
19	<i>mouse up</i>	css=.col-md-5:nth-child(4) > label	
20	<i>type</i>	id=belop	1500.25
21	<i>click</i>	id=melding	
22	<i>type</i>	id=melding	Transaksjon1
23	<i>click</i>	css=.container:nth-child(3)	
24	<i>click</i>	id=regBetaling	
25	<i>click</i>	css=.btn	

## 2.4.1 Kunde: Se Betalinger OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Register betaling	
13	<i>assert text</i>	linkText=Register betaling	Register betaling
14	<i>click</i>	id=dato	
15	<i>type</i>	id=dato	2010-11-11
16	<i>click</i>	id=tilKonto	
17	<i>type</i>	id=tilKonto	105010123456
18	<i>mouse down</i>	id=belop	
19	<i>mouse up</i>	css=.col-md-5:nth-child(4) > label	
20	<i>type</i>	id=belop	1500
21	<i>click</i>	id=melding	
22	<i>type</i>	id=melding	Transaksjon2
23	<i>click</i>	id=regBetaling	
24	<i>click</i>	linkText=Utfør betalinger	
25	<i>assert text</i>	linkText=Utfør betalinger	Utfør betalinger
26	<i>click</i>	css=.btn	

## 2.5.1 Kunde: Se Transaksjoner Feil (Feil Fra dato)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Transaksjoner	Transaksjoner
13	<i>assert text</i>	linkText=Transaksjoner	Transaksjoner
14	<i>click</i>	id=fraDato	
15	<i>type</i>	id=fraDato	a
16	<i>click</i>	id=tilDato	
17	<i>type</i>	id=tilDato	2016-01-01
18	<i>click</i>	id=sok	

## 2.5.2 Kunde: Se Transaksjoner OK (Feil Til dato)

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Transaksjoner	Transaksjoner
13	<i>assert text</i>	linkText=Transaksjoner	Transaksjoner
14	<i>click</i>	id=fraDato	
15	<i>type</i>	id=fraDato	2011-01-01
16	<i>click</i>	id=tilDato	
17	<i>type</i>	id=tilDato	a
18	<i>click</i>	id=sok	

### 2.5.3 Kunde: Se Transaksjoner OK (Feil Fra dato ...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=passord	
9	click	id=passord	
10	type	id=passord	HeiHei
11	click	id=loggInn	
12	click	linkText=Transaksjoner	Transaksjoner
13	assert text	linkText=Transaksjoner	Transaksjoner
14	click	id=fraDato	
15	type	id=fraDato	a
16	click	id=tilDato	
17	type	id=tilDato	a
18	click	id=sok	

### 2.5.4 Kunde: Se Transaksjoner OK

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=passord	
9	click	id=passord	
10	type	id=passord	HeiHei
11	click	id=loggInn	
12	click	linkText=Transaksjoner	Transaksjoner
13	assert text	linkText=Transaksjoner	Transaksjoner
14	click	id=fraDato	
15	type	id=fraDato	2011-01-01
16	click	id=tilDato	
17	type	id=tilDato	2016-01-01
18	click	id=sok	

### 2.6.1 Kunde: Endre Kunde Informasjon Feil

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=passord	
9	click	id=passord	
10	type	id=passord	HeiHei
11	click	id=loggInn	
12	click	linkText=Kunde-info	
13	assert text	linkText=Kunde-info	Kunde-info
14	click	id=fornavn	
15	type	id=fornavn	a
16	type	id=etternavn	a
17	type	id=adresse	a
18	type	id=postnr	a
19	type	id=poststed	a
20	type	id=telefonnr	a
21	type	id=passord	a
22	click	css=.row:nth-child(11)	
23	click	id=endre	

## 2.6.2 Kunde: Endre Kunde Informasjon OK (Æ,Ø,...

1	open	/TestingBank/DAL/InitDB.php	
2	pause	1000	
3	open	http://localhost/TestingBank/Index.php	
4	set window size	1184x680	
5	click	linkText=Logg inn	
6	click	id=personNr	
7	type	id=personNr	01010110523
8	click	id=passord	
9	click	id=passord	
10	type	id=passord	HeiHei
11	click	id=loggInn	
12	click	linkText=Kunde-info	

13	<i>assert text</i>	linkText=Kunde-info	Kunde-info
14	<i>click</i>	id=fornavn	
15	<i>type</i>	id=fornavn	Øyvind
16	<i>type</i>	id=etternavn	Åserud
17	<i>type</i>	id=adresse	Æreveien 24
18	<i>type</i>	id=postnr	0578
19	<i>type</i>	id=poststed	Oslo
20	<i>type</i>	id=telefonnr	22334455
21	<i>type</i>	id=passord	HeiHei
22	<i>click</i>	css=.row:nth-child(11)	
23	<i>click</i>	id=endre	

### 2.6.3 Kunde: Endre Kunde Informasjon OK

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=passord	
9	<i>click</i>	id=passord	
10	<i>type</i>	id=passord	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Kunde-info	
13	<i>assert text</i>	linkText=Kunde-info	Kunde-info
14	<i>click</i>	id=fornavn	
15	<i>type</i>	id=fornavn	Lena
16	<i>type</i>	id=etternavn	Skogheim
17	<i>type</i>	id=adresse	Bakkerud veien 24
18	<i>type</i>	id=postnr	0578
19	<i>type</i>	id=poststed	Oslo
20	<i>type</i>	id=telefonnr	22334455
21	<i>type</i>	id=passord	HeiHei
22	<i>click</i>	css=.row:nth-child(11)	
23	<i>click</i>	id=endre	

### 2.6.4 Kunde: Endre Kunde Informasjon OK (Tilba...

1	<i>open</i>	/TestingBank/DAL/InitDB.php	
2	<i>pause</i>	1000	
3	<i>open</i>	http://localhost/TestingBank/Index.php	
4	<i>set window size</i>	1184x680	
5	<i>click</i>	linkText=Logg inn	
6	<i>click</i>	id=personNr	
7	<i>type</i>	id=personNr	01010110523
8	<i>click</i>	id=password	
9	<i>click</i>	id=password	
10	<i>type</i>	id=password	HeiHei
11	<i>click</i>	id=loggInn	
12	<i>click</i>	linkText=Kunde-info	
13	<i>assert text</i>	linkText=Kunde-info	Kunde-info
14	<i>click</i>	id=fornavn	
15	<i>type</i>	id=fornavn	Lena
16	<i>type</i>	id=etternavn	Skogheim
17	<i>type</i>	id=adresse	Bakkerud veien 24
18	<i>type</i>	id=postnr	0578
19	<i>type</i>	id=poststed	Oslo
20	<i>type</i>	id=telefonnr	22334455
21	<i>type</i>	id=password	HeiHei
22	<i>click</i>	css=.row:nth-child(11)	
23	<i>click</i>	id=endre	
24	<i>click</i>	linkText=Tilbake	

## MTM DOKUMENT

# Test plan [18](#): testplan

### Properties

Area Path:	Testing123
Iteration:	Testing123\Iteration 1
Owner:	s326164
State:	Active
Start date:	lørdag 9. mars 2019
End date:	lørdag 16. mars 2019

### Suite hierarchy

Static suite: testplan (ID: [19](#))

Requirement-based suite: 2 : Som Administrator ønsker jeg kunne logge inn for å utføre operasjoner på nettstedet (ID: [20](#))

Requirement-based suite: 3 : Som Administrator ønsker jeg å kunne registrere kunde for å opprette en ny kunde (ID: [21](#))

Requirement-based suite: 4 : Som Administrator ønsker jeg å kunne endre

kundeinformasjon for å oppdatere kundeopplysninger (ID: [22](#))

Requirement-based suite: 5 : Som Administrator ønsker jeg å kunne slette en kunde for å fjerne en kunde (ID: [23](#))

Requirement-based suite: 6 : Som Administrator ønsker jeg å kunne registrere konto for en kunde sånn at kunden kan foreta service og tjeneste nettstedet tilbyr (ID: [24](#))

Requirement-based suite: 7 : Som Administrator ønsker jeg å kunne endre kunde konto konto opplysninger for endre konto tilstanden til kunde (ID: [25](#))

Requirement-based suite: 8 : Som Administrator ønsker jeg å kunne slette en konto for å fjerne en konto for en kunde (ID: [26](#))

Requirement-based suite: 10 : Som Kunde ønsker jeg å kunne logge inn for å utføre tjenester som er tilbudt av nettstedet (ID: [27](#))

Requirement-based suite: 11 : Som kunde ønsker jeg å kunne se saldo for å se aktiviteter som er gjort på min personnummer (ID: [28](#))

Requirement-based suite: 12 : Som kunder ønsker jeg å kunne registrere betalinger for å betale kunder av meg for tjenester eller service (ID: [29](#))

Requirement-based suite: 13 : Som kunder ønsker jeg å kunne se betalinger som er utført for å holde oversikt over betalinger gjort av meg som kunde (ID: [30](#))

Requirement-based suite: 14 : Som kunder ønsker jeg å kunne se transaksjoner som er gjort for kontonummer for å vite hvilke transaksjoner som er gjort av meg som kunde (ID: [31](#))

Requirement-based suite: 15 : Som kunde ønsker jeg å kunne endre min kundeinformasjon for å oppdatere opplysningene mine (ID: [32](#))

## Configurations

### CONFIGURATIONS IN TEST PLAN

Id	Name
----	------

1	Windows 10
---	------------

Configuration variables

Operating System: Windows 10



## Test suite [19](#): testplan

## Test suite [20](#): 2 : Som Administrator ønsker jeg kunne logge inn for å utføre operasjoner på nettstedet

Test cases (5)

### Test case [33](#): Feil logg inn

#### STEPS

#	Action	Expected value	Attachments
1	skriv inn URL: <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>	får frem logg inn side	<a href="#">startside.PNG</a> (15K)
2	klikk logg inn	får fram en logg inn form	
3	fill inn brukernavn og passord		<a href="#">fyltinn.PNG</a> (24K)
4	logger inn	forventer en feilmelding	<a href="#">feilmelding.PNG</a> (30K)

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	tirsdag 12. mars 2019	0	

### Test case [34](#): Logg inn riktig

#### STEPS

#	Action	Expected value	Attachments
1	gå inn på URL: <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>	forventer start side	<a href="#">startside.PNG</a> (15K)
2	Klikk logg inn	forventer å få fram logg inn form	
3	fill inn riktig info		<a href="#">fyllinnR.PNG</a> (28K)

4      klikk logg inn      forventer å komme til admin side      [admin side.PNG](#) (42K)

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passe d</a>	s326164	Windows10	s326164	tirsdag 12. mars 2019	0	

## Test case [65](#): Admin logg inn (Tom felt)

STEPS

#	Action	Expected value	Attachments
1	resette databasen med InitDB.php	Ok	
2	gå til innlogging side Testingbank/Index.php		
3	klikk på logg inn		<a href="#">startside.PNG</a> (15K)
4	la input feltene være tomme		<a href="#">TomtFelt admin.PNG</a> (29K)
5	logg inn	Forventer at man ikke kommer inn	

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passe d</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [66](#): Logg inn (SQL injection 1)

STEPS

#	Action	Expected value	Attachments
1	Resetter databasen med InitDB.php	Ok	
2	går til Testingbank/index.php		
3	klikker på Admin		
4	fyller inn sql injection kommando " or ""=""		<a href="#">sqlinject1.PNG</a> (21K)
5	klikker logg inn	Feil personnummer/passord!	

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passe d</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [67](#): Logg inn (SQL injection 2)

### STEPS

#	Action	Expected value	Attachments
1	Ressetter database /DAL/initDB.php	Ok	
2	går til /TestingBank/index.php		
3	klikker på Admin		
4	fyller inn sql injection `Ole`--_`		<a href="#">sqlinje2.PNG</a> (18K)
5	klikker logg inn	Feil personnummer/pass ord!	

### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test suite [21](#): 3 : Som Administrator ønsker jeg å kunne registrere kunde for å opprette en ny kunde

### Test cases (3)

## Test case [35](#): Registrere kunde feil

### STEPS

#	Action	Expected value	Attachments
1	gå til URL: <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
2	Logg inn som admin	kommer inn på admin side	<a href="#">admin side.PNG</a> (42K)
3	klikk på registrer kunde	kommer inn på kunde registrerings siden	<a href="#">kundeReg.PNG</a> (59K)
4	Fyll inn feil personnummer	personnummer må være 11 siffer	<a href="#">feilpersonnummer.PNG</a> (16K)
5	fill inn feil fornavn	fornavn må være mellom 2 og 30 tegn	<a href="#">feilFornavn.PNG</a> (8K)
6	fill inn feil etternavn	etternavn må være mellom 2 og 30 tegn	<a href="#">FeilEtternavn.PNG</a> (11K)

7	fill inn feil adresse	adresse må være mellom 2 og 30 tegn	<a href="#">feilAdresse.PNG</a> (10K)
8	fill inn feil postnr	postnr må være 4 siffer	<a href="#">feilPostnummer.PNG</a> (7K)
9	fill inn feil poststed	poststed må være mellom 2 og 30 siffer	<a href="#">feilPostSted.PNG</a> (10K)
10	fill inn feil telefonnr	telefonnr må være 8 siffer	<a href="#">feilTlf.PNG</a> (7K)
11	fill inn passord		
12	klikk registrer	registrering ikke går gjennom	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s3261	Windows	s3261	søndag 17. mars 2019	0	

## Test case [36](#): registrer kunde

#### STEPS

#	Action	Expected value	Attachments
1	reset database med <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	ok	
2	gå inn på <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	logg inn som admin		<a href="#">fyllinnR.PNG</a> (28K)
4	kommer til admin side	kommer til admin side	<a href="#">admin side.PNG</a> (42K)
5	klikker på registrer ny kunde	kommer til registrer ny kunde side	<a href="#">kundeReg.PNG</a> (59K)
6	Fyll inn alt info		<a href="#">regNyKundeRiktig.PNG</a> (50K)
7	Kunde registrert	Kunde er registrert	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s3261	Windows	s3261	tirsdag 12. mars 2019	0	

## Test case [68](#): Registrere Kunde(Æ,Ø,Å)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen	OK	

	/TestingBank/Dal/initDB.php		
2	går til /TestingBank/index.php	kommer til index side	
3	klikker på Admin	blir tatt til admin logg inn side	
4	logger inn som admin		<a href="#">fyllinnR.PNG</a> (28K)
5	klikker logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker på registrer kunde	Kunde registrerings side	<a href="#">kundeReg.PNG</a> (59K)
7	Fyller inn info		<a href="#">ÆØÅ.PNG</a> (49K)
8	Klikker på endre		

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test suite [22](#): 4 : Som Administrator ønsker jeg å kunne endre kundeinformasjon for å oppdatere kundeopplysninger

Test cases (3)

### Test case [37](#): Endre kundeinformasjon feil

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen på <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	ok	
2	går til <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikk på admin		
4	fill inn innloggingsinfo		<a href="#">fyllinnR.PNG</a> (28K)
5	logger inn som admin	kommer til admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker på endre		
7	fjerner alt av kunde info		<a href="#">endreInfo.PNG</a> (42K)
8	klikker på endre	Forventer at endringer ikke skal	

bli gjort

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [38](#): Endre kunde info riktig

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	ok	
2	går til logginn siden <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker på admin	kommer til admin login	
4	yller inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	logger inn	forventer å komme til admin side	<a href="#">admin side.PNG</a> (42K)
6	yller inn ny info til kunde		<a href="#">endreInfoRiktig.PNG</a> (41K)
7	klikker på endre	forventer at ny kunde info blir lagret	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Pass</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [69](#): Endre kunde informasjon(Æ,Ø,Å)

#### STEPS

#	Action	Expected value	Attachments
1	Resette database <a href="/testingBank/Dal/InitDB.php">/testingBank/Dal/InitDB.php</a>	Ok	
2	gå til /index.php		
3	klikk på admin	Admin logg inn side	
4	yll inn riktig info		<a href="#">fyllinnR.PNG</a> (28K)
5	logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk på endre kunde	kunde info side	<a href="#">endreInfo.PNG</a> (42K)

7 fyll inn ny info med Æ, Ø og Å

[kundeINFOØÆÅ.PNG](#)  
[G](#) (34K)

8 klikker på endre

Kunde info endret

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test suite [23](#): 5 : Som Administrator ønsker jeg å kunne slette en kunde for å fjerne en kunde

Test cases (2)

### Test case [39](#): Slette kunde (forsøke å feile)

#### STEPS

#	Action	Expected value	Attachments
1	resette databasen på <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	ok	
2	gå til logginn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikke på admin		
4	fill inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	logg inn	kommer til admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker slett	forventer at pop-up bekreftelse	
7	klikker avbryt	forventer at kunde ikke er slettet	<a href="#">vilDuSletteKunde.PNG</a> (34K)

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

### Test case [40](#): Slette kunde

#### STEPS

#	Action	Expected value	Attachments
1	resette databasen på <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	går til <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker på admin	forventer å komme til admin logging	
4	fyller inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikker logg inn	forventer å komme til admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk slett		<a href="#">slett.PNG</a> (2K)
7	klikk ok	Kunde slettet	<a href="#">vilDuSletteKunde.PNG</a> (34K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Pass</a>	s326164	Windows 10	s326164	søndag 10. mars 2019	0	

**Test suite [24](#): 6 : Som Administrator ønsker jeg å kunne registrere konto for en kunde sånn at kunden kan foreta service og tjeneste nettstedet tilbyr**

Test cases (5)

**Test case [41](#): Registrere konto feil (kontonummer feil)**

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	Fyll inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
4	klikk logg inn	kommer til hoved side	<a href="#">admin side.PNG</a> (42K)
5	klikk registrer konto	registeringside	



6	fill inn feil kontonummer	kontonummeret må være 11 siffer	<a href="#">kontonummerFeil.PNG (16K)</a>
7	filler inn resterende info		<a href="#">kontonummer fill inn info.PNG (27K)</a>
8	klikker registrer konto	Konto ikke registrert	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [42](#): Registrere Konto admin (personnummer feil)

#### STEPS

#	Action	Expected value	Attachments
1	resetter database på <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	ok	
2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker på admin		
4	filler inn admin logg inn info		<a href="#">fillinnR.PNG (28K)</a>
5	logger inn	Kommer til Admin side	<a href="#">admin side.PNG (42K)</a>
6	klikk registrer konto		
7	filler inn info		<a href="#">kontoInfo.PNG (34K)</a>
8	filler inn feil personnummer	personnummer må være 11 siffer	<a href="#">feilpersonnummer.PNG (16K)</a>
9	klikk registrer konto	personnummer finnes ikke	<a href="#">personnummer finnes ikke.PNG (23K)</a>

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [43](#): Registrere konto feil (kontonummer og personnummer)

#### STEPS

#	Action	Expected value	Attachments
1	resetter datanbasen på <a href="http://localhost/TestingBank/DAL/InitDB">http://localhost/TestingBank/DAL/InitDB</a>	ok	

	B.php		
2	går til logg inn side på <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikk på admin		
4	fill inn admin logg inn info		<a href="#">fylldinnR.PNG</a> (28K)
5	klikk logg inn	kommer til admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk registrer konto		
7	fill inn feil kontonummer	kontonummer må være 11 siffer	<a href="#">kontonummerFeil.PNG</a> (16K)
8	fill inn feil personnummer	personnummer må være 11 siffer	<a href="#">personnummer feil.PNG</a> (9K)
9	klikker registrer	personnummer finnes ikke	<a href="#">personnummer finnes ikke.PNG</a> (23K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [44](#): Registrere Konto Ok

#### STEPS

#	Action	Expected value	Attachments
1	resette databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikk admin		
4	fill inn admin info		<a href="#">fylldinnR.PNG</a> (28K)
5	logg inn	kommer til admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk registrere konto		
7	Fyll inn all riktig info		<a href="#">riktig info konto.PNG</a> (34K)
8	klikk registrer konto	Konto registrert	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [70](#): Registrere konto (Æ, Ø, Å)

### STEPS

#	Action	Expected value	Attachments
1	Resette databasen /initDB.php	Ok	
2	gå til siden TestingBank/index.php		
3	klikk på Admin	Admin logg inn side	
4	Fyll inn admin logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikk logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk registrer konto	registrerings side	
7	fill inn ny konto info		<a href="#">kontoØÆÅ.PNG</a> (48K)
8	klikk registrer	Ny konto registrert	

### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows 10	s326164	søndag 17. mars 2019	0	

## Test suite [25](#): 7 : Som Administrator ønsker jeg å kunne endre kunde konto opplysninger for endre konto tilstanden til kunde

### Test cases (4)

## Test case [45](#): Endre Kunde Konto Feil (tom felt)

### STEPS

#	Action	Expected value	Attachments
1	resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker admin		
4	fyller inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)

5	klikker logg inn	admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker endre konto	endre konto side	<a href="#">endreKontoSide.PNG</a> (73K)
7	Fjerner all info(tømmer feltene)		<a href="#">tommeKontoFelt.PNG</a> (41K)
8	klikker på endre	At det ikke lagres	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<b>Failed</b>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [46](#): Endre kunde Konto Feil (feil verdi)

#### STEPS

#	Action	Expected value	Attachments
1	resett databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	gå til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikk på admin		
4	fill inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikk logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk endre konto		
7	fill inn feil konto info		<a href="#">endrekontoFeilINFO.PNG</a> (32K)
8	klikk endre	Ikke blir endret	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<b>Failed</b>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [47](#): Endre KundeInfo Ok

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	

2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker på admin		
4	yller inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikker på logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker på endre konto	Endre konto side	
7	yll inn info		<a href="#">EndreKontoRiktigOK.PNG</a> (33K)
8	klikk på endre	Konto endret	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [71](#): Endre kunde konto (Æ, Ø, Å)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen InitDB.php	Ok	
2	Går til siden /TestingBank/index.php		
3	Klikker på admin	Admin logg inn side	
4	yller inn admnin logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikker logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikker på endre konto	endre konto side	
7	yller inn ny informasjon med Æ, Ø og Å		<a href="#">endreKontoøæå.PNG</a> (54K)
8	klikker på endre	informasjon endret	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test suite [26](#): 8 : Som Administrator ønsker jeg å

# kunne slette en konto for å fjerne en konto for en kunde

## Test cases (2)

### Test case [48](#): Slette en Konto (forsøke å feile)

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	gå på logg inn siden <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikk på admin		
4	fill inn admin logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikk logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk på endre konto		
7	klikk på slett konto	bekreftelse pop-up	<a href="#">slettKonto.PNG</a> (64K)
8	klikk på avbryt		

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Pass</a>	s3261	Windows	s3261	søndag 17. mars 2019	0	

### Test case [49](#): Slett Konto Ok

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen <a href="http://localhost/TestingBank/DAL/InitDB.php">http://localhost/TestingBank/DAL/InitDB.php</a>	Ok	
2	går til logg inn side <a href="http://localhost/TestingBank/Index.php">http://localhost/TestingBank/Index.php</a>		
3	klikker Admin		
4	filler inn logg inn info		<a href="#">fyllinnR.PNG</a> (28K)
5	klikk logg inn	Admin side	<a href="#">admin side.PNG</a> (42K)
6	klikk endre konto	Endre konto side	

**7**    **klikk slett**

**bekreftelse pop-up**

[slettKonto.PNG](#) (64K)

**8**      **klikk Ok**

## Konto slettet

## LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows 10	s326164	søndag 10. mars 2019	0	

**Test suite [27](#): 10 : Som Kunde ønsker jeg å kunne logge inn for å utføre tjenester som er tilbudt av nettstedet**

### Test cases (4)

## Test case 50: Logg inn feil

## STEPS

#	Action	Expected value	Attachments
1	Resetter database http://localhost/TestingBank/DAL/InitDB.php	Ok	
2	logg inn side http://localhost/TestingBank/Index.php		
3	klikker logg inn		
4	fyll inn feil logg inn info		<a href="#">feilInfo.PNG</a> (22K)
5	klikker logg på	feil personnummer og passord	<a href="#">feilLogginn.PNG</a> (24K)

### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

### Test case 51: Logg inn Ok

## STEPS

#	Action	Expected value	Attachments
1	Resetter databasen		
2	går til logg inn siden		
3	klikker logg inn		
4	yller inn info		<a href="#">riktigInfo.PNG</a> (21K)
5	klikker logg inn	kommer til kunde side	<a href="#">KundeSide.PNG</a> (86K)

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [72](#): logg inn (SQL injection 1)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen /InitDB.php	Ok	
2	gå til siden /TestingBank/Index.php		
3	klikk på logg inn	logg inn side	
4	Fyller inn " or ""=" på begge felt	personnummer må være 11 siffer	<a href="#">sglii.PNG</a> (27K)
5	klikker logg inn	Feil personnummer/passord!	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [73](#): Logg inn (SQL injection 2)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen med /InitDB.PHP	Ok	
2	går til siden /TestingBank/index.php		
3	klikk på logg inn	logg inn side	



- |   |   |                                   |   |
|---|---|-----------------------------------|---|
| 4 | fyll inn personnummer og passord med<br>`Ole`--_` | personnummer må<br>være 11 siffer | <a href="#">KundeSQLin.PNG</a><br>(38K)<br><a href="#">riktigInfo.PNG</a> (21K) |
| 5 | klikklogg inn                                     | Feil<br>personnummer/pass<br>ord! |   |

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

Test suite [28](#): 11 : Som kunde ønsker jeg å kunne se saldo for å se aktiviteter som er gjort på min personnummer

Test cases (1)

#### Test case [52](#): Se saldo

##### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen	Ok	
2	går tillogg inn side		
3	yller inn riktig info		<a href="#">riktigInfo.PNG</a> (21K)
4	logger inn	Kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	Klikker på Saldo	Får saldo	

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test suite [29](#): 12 : Som kunder ønsker jeg å kunne registrere betalinger for å betale kunder av meg for tjenester eller service

### Test cases (5)

#### Test case [53](#): Registrere betalinger Feil (Feil kontonummer)

##### STEPS

#	Action	Expected value	Attachments
1	resetter database	Ok	
2	går til logg inn side		
3	fyller inn riktig info		<a href="#">riktigInfo.PNG</a> (21K)
4	logger inn	Kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	klikker registerer betalinger	registrer betalinger side	<a href="#">registrerBetalingerSIDE.PNG</a> (86K)
6	fill inn kontonummer	kontonummer må være 11 siffer	
7	fill inn resten av info		<a href="#">FeilbetalingInfo.PNG</a> (67K)
8	klikk registrer betaling	Betalingen ikke gjennomført	

##### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows 10	s326164	søndag 10. mars 2019	0	

#### Test case [54](#): Registrere betalinger Feil (feil DATO)

##### STEPS

#	Action	Expected value	Attachments
1	Ressete databasen	Ok	
2	gå til logg inn side		
3	fill inn info		<a href="#">riktigInfo.PNG</a> (21K)
4	klikker på logg inn	Kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	klikk registrer betalinger	registrere betalinger side	<a href="#">registrerBetalingerSIDE.PNG</a> (86K)

6	fyller inn dato	Dato må være format yyyy-mm-dd	<a href="#">feilDato.PNG</a> (16K)
7	Fyller inn resten av info		
8	klikker registrer betaling	registrering gjennomført	<a href="#">DATOFEILREG.PNG</a> (29K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [55](#): Registrere betalinger feil (betalingsdato og kontonummer)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter database	Ok	
2	Går til logg inn side		
3	Fyller inn logg inn info		<a href="#">riktigInfo.PNG</a> (21K)
4	klikker logg inn	Kunde Side	<a href="#">KundeSide.PNG</a> (86K)
5	klikker registrer betalinger	registrer betalinger side	<a href="#">registrerBetalingerSIDE.PNG</a> (86K)
6	Fyller inn feil kontonummer	kontonummer må være 11 siffer	
7	fyller inn feil dato	dato må være på formate yyyy-mm-dd	
8	fyller inn resten av info		<a href="#">regbetalFeilDatoogKonto.PNG</a> (52K)
9	klikker registrer betaling	Betaling ikke registrert	<a href="#">betalinggikkgjennom.PNG</a> (64K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [56](#): Registrering betaling Ok

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen	Ok	
2	går til logg inn side		

3	klikker logg inn	
4	yller inn logg inn info	
5	logger inn	kunde side
6	klikker registrer betaling	registrer betaling side
7	Fyller inn all info	<a href="#">regRiktig.PNG</a> (58K)
8	klikker registrer betaling	betaling registrert

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [75](#): Registrer betaling (minus saldo)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen med /InitDB.php	Ok	
2	går til siden /TestingBank/Index.php		
3	klikker på logg inn	Kunde logg inn side	
4	yller inn riktig personnummer og passord		<a href="#">riktigInfo.PNG</a> (21K)
5	klikker logg inn	Kunde side	<a href="#">KundeSide.PNG</a> (86K)
6	klikker på registrer betaling	registrer betaling side	<a href="#">regbet.PNG</a> (47K)
7	Fyller inn info med minus verdi i saldo-feltet		<a href="#">Capture.PNG</a> (46K)
8	klikker på registrer betaling	betalingen ikke blir registrert	<a href="#">betREGA.PNG</a> (27K)

#### LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

Test suite [30](#): 13 : Som kunde ønsker jeg å kunne se betalinger som er utført for å holde oversikt over

# betalinger gjort av meg som kunde

Test cases (1)

## Test case [57](#): Se betalinger Ok

### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen		
2	går til logg inn side		
3	yller inn logg inn info		
4	logger inn	kunde side	
5	klikker registrer betalinger	registrer betalinger side	
6	registrerer en betaling		
7	klikker på utfør betalinger	Får oppå betalinger	<a href="#">regbet.PNG</a> (47K)

### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows 10	s326164	søndag 10. mars 2019	0	

Test suite [31](#): 14 : Som kunde ønsker jeg å kunne se transaksjoner som er gjort for kontonummer for å vite hvilke transaksjoner som er gjort av meg som kunde

Test cases (4)

## Test case [58](#): Se transaksjoner Feil (feil fra dato)

### STEPS

#	Action	Expected value	Attachments
1	resetter databasen		
2	går til logg inn side		
3	yller inn logg inn info		
4	logger inn	kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	klikker på transaksjoner	transaksjoner side	<a href="#">transaksjonside.PNG</a> (39K)
6	yller inn feil info		<a href="#">feilFraDAto.PNG</a> (29K)
7	klikker søk		<a href="#">transUtført.PNG</a> (60K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [59](#): Se transaksjoner feil (feil til dato)

#### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen		
2	gå til logg inn side		
3	yll inn logg inn informasjon		
4	klikk logg inn	kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	klikker på transaksjoner	transaksjoner side	<a href="#">transaksjonside.PNG</a> (39K)
6	yller inn feil til dato		
7	klikker søk		<a href="#">feiltildatoSøk.PNG</a> (41K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [60](#): Se transaksjoner feil (fra dato og til dato)

#### STEPS

#	Action	Expected value	Attachments
1	resetter database	Ok	
2	Går til logg inn side		
3	yller inn logg inn informasjon		
4	klikker logg inn	kunde side	
5	klikker på transaksjoner	transaksjoner side	
6	yller inn feil til og fra dato		
7	klikker søk		<a href="#">tilogfradato.PNG</a> (61K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	

## Test case [61](#): Se transaksjoner OK

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen	Ok	
2	Går til logg inn side		
3	yller inn logg inn informasjon		
4	klikker logg inn	kunde side	
5	klikker på transaksjoner	transaksjoner side	
6	yller inn fra og til dato		
7	klikker søk		<a href="#">riktigTrans.PNG</a> (115K)

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test suite [32](#): 15 : Som kunde ønsker jeg å kunne

# endre min kundeinformasjon for å oppdatere opplysningene mine

## Test cases (4)

### Test case [62](#): Endre kundeinformasjon feil

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen	Ok	
2	gå til logg inn side		
3	fyller inn logg inn info		
4	klikker logg inn	kunde side	<a href="#">KundeSide.PNG</a> (86K)
5	klikker på kunde info	kunde info side	<a href="#">kundeInfo.PNG</a> (50K)
6	fyller inn feil informasjon		<a href="#">endreInfoFeil.PNG</a> (67K)
7	klikker på endre info	forventer feilmelding og at info ikke blir endret	

#### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Failed</a>	s326164	Windows 10	s326164	søndag 17. mars 2019	0	

### Test case [63](#): Endre kunde informasjon Ok

#### STEPS

#	Action	Expected value	Attachments
1	resetter databasen	Ok	
2	Går til logg inn side		
3	fyller inn logg inn info		<a href="#">riktigInfo.PNG</a> (21K)
4	klikker logg inn	Kunde side	
5	klikker Kunde-info	Kunde info side	<a href="#">kundeInfo.PNG</a> (50K)
6	fyller inn riktig info		<a href="#">k2.PNG</a> (61K)



## 7 Klikker på endre info

## Endret kunde info

### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [64](#): Endre kundeinformasjon tilbake

### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen	Ok	
2	Går til logg inn side		
3	Fyller inn riktig logg inn info		<a href="#">riktigInfo.PNG</a> (21K)
4	Klikker logg inn	Kunde Side	<a href="#">KundeSide.PNG</a> (86K)
5	Klikker på kunde-info	Kunde Info side	<a href="#">kundeInfo.PNG</a> (50K)
6	Fyller inn all info		
7	Klikker på tilbake	Kunde side	<a href="#">KundeSide.PNG</a> (86K)

### LATEST TEST OUTCOME

Outcome	Test runner	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passerd</a>	s326164	Windows10	s326164	søndag 10. mars 2019	0	

## Test case [74](#): Endre kunde informasjon(Æ,Ø,Å)

### STEPS

#	Action	Expected value	Attachments
1	Resetter databasen /InitDB.php	Ok	
2	går til siden TestingBank/Index.php		
3	klikk på logg inn	kommer til logg inn side	
4	fyller inn riktig personnummer og passord		<a href="#">riktigInfo.PNG</a> (21K)
5	klikker på logg inn	kommer til kunde side	
6	klikker på kunde info	kunde info side	

7 Fyller inn ny info med Æ, Ø og Å

[endreInfoØÆÅ.PNG](#)  
(42K)

8 klikker på endre info

Info endret

LATEST TEST OUTCOME

Outcome	Tester	Configuration	Run by	Date completed	Duration in seconds	Build number
<a href="#">Passed</a>	s326164	Windows10	s326164	søndag 17. mars 2019	0	