
Gruppe 1

Systemutviklingsprosjekt TDAT 3022
Hovedrapport

Versjon 1.0

Kristian Kampenhøy, Simon Lilleeng

Forord

Systemutviklingsprosjekt er et valgfag på høstsemesteret på linjen til dataingeniør 3. Der skal elevene velge en systemutviklingsoppgave som er relevant for deres utdanning og videre utvikling i faget. Vi valgte denne oppgaven med en enighet om å utfordre oss selv faglig og samtidig få et innblikk i hvordan prosessen ved å utvikle et system i bedrift er. Oppgaven vi har valgt fikk vi ved å ta kontakt med bedriften Fixrate AS. Vi har gjennom utviklingstiden hatt utrolig stort læringsutbytte og ser derfor fram til å fortsette i dette fagfeltet.

Vi vil takke vår veileder og fagansvarlig Alexander Holt for tilbakemeldinger gjennom prosjekttiden. Nils Tesdal for å ha tipset oss om denne mulighet. Også en stor takk til de på Fixrate som har gitt oss denne oppgaven, vært behjelpelig, svart på alle våre spørsmål og fått oss til å føle oss velkommen i deres lokaler. Vi ser derfor fram til et videre samarbeid med dere.

Oppgavetekst

Fixrates visjon er å utvikle en god kundeopplevelse som gir kundene sine en enklere tilgang til bedre betingelser. Fixrate har idag en markeds plass for store bankinnskudd og gjør det enklere for organisasjoner med god likviditet å finne det beste tilgjengelige tilbudet på innskuddsprodukter hos norske banker.

Fixrate ønsker derfor å se på mulighetene til å få noen av sine funksjoner inn i en mobilapplikasjon slik at kunder enkelt kan ha tilgang til markedet, gjøre mål og motta notifikasjoner når nye annonser havner på markedet til Fixrate. Slik kan kundene alltid være oppdatert og samtidig ikke gå glipp av gode tilbud.

Oppgaven vi fikk tildelt av bedriften var å utvikle en applikasjon ut fra den eksisterende web-plattformen deres. Bedriften ønsket en applikasjon som skulle hjelpe kundene deres å effektivisere prosessen av kjøp på nettstedet deres, samt motta pushnotifikasjoner på mobilene sine ved nye endringer og gjøre mål.

Applikasjonen skal utvikles slik at den er kompatibel for kunder som bruker både Android- og iOS-mobiler.

Sammendrag

Oppgaven består i å utvikle en applikasjon som er kompatibel både med iOS og Android. Fixrate ønsker en måte å utvide sitt nettsted til en applikasjon og har utstedt denne prototypingen til oss. Gruppen går på 3. år dataingeniør og har dette prosjektet som et valgfag for å få enda bedre erfaring med systemutvikling samt erfaring innad bedrift. Det har ikke vært fokus på applikasjonsutvikling tidligere i studiet, og dette er et nytt område for gruppen.

Gruppen har gjennom prosjekttiden brukt mange nye teknologier og biblioteker for å oppnå den ønskede funksjonaliteten i applikasjonen og har slik blitt utfordret faglig. Videre i rapporten skal vi i detalj gå gjennom hvilke teknologi vi har valgt å bruke gjennom utviklingsprosessen og hvorfor disse ble valgt. Beskrive hvilke resultater vi har oppnådd gjennom utviklingen og frem til sluttprodukt. Til slutt forteller vi om hvilke problemer og utfordringer vi har støtt på underveis og hvordan vi valgte å løse disse.

For videre lesing utenom rapporten henviser vi til vedleggene som er listet opp helt til slutt samt. referansene for nærmere detaljer.

Innholdsfortegnelse

Forord	2
Oppgavetekst	3
Sammendrag	4
Innholdsfortegnelse	5
Kapittel 1: Introduksjon og relevans	6
Kapittel 2: Valg av teknologi og metode	7
React Native	7
React Redux	7
NodeJS	8
RabbitMQ	8
Expo Client	8
Jest og Enzyme	9
Annet	9
Kapittel 3: Resultater	10
Ingeniørfaglige resultater:	10
Administrative resultater:	11
Utviklingsprosessen	11
Kapittel 4 - Diskusjon	12
Hva har vi lært	12
Problemer og utfordringer underveis	13
Ressursbruk	14
Resultater	14
Kapittel 5: Konklusjon og videre arbeid	15
Referanser	16
Vedlegg	16

Kapittel 1: Introduksjon og relevans

Fixrates ønske om utviklingen av denne applikasjonen er at prosessen kundene må gjennom fra en bestilling er gjort til den er ferdigstilt kan være litt lang. De vil derfor at kundene skal varsles når det er deres tur å signere dokumenter. Disse signeres selvfølgelig digitalt gjennom et sikkert system, men Fixrate vil at den aktuelle parten skal få beskjed i form av varsel på telefon og at aktiviteten blir lagt inn i en liste av gjøremål i applikasjonen. I dagens løsning må kunden logge inn i nettleser for deretter å sjekke selv hvor langt i prosessen bestillingen er. Dette tror Fixrate kan effektiviseres ytterligere ved å gjøre kunder oppmerksomme på nye hendelser og annonser på markedsplassen gjennom varslinger på telefon. De ønsker også at alle gjøremål for brukerne skal havne i en liste slik at brukerne kan få et bilde av dagens agenda og hva som må gjøres.

Kapittel 2: Valg av teknologi og metode

Prosjektet vårt bruker en rekke teknologier og rammeverk for å oppnå utviklingen av selve applikasjon, kommunikasjon, datalagring, hvordan håndtere data i applikasjonen og hvordan vi skal sende pushvarsler ut til brukerne.

Prosjektet har tatt i bruk teknologiene:

- React
- React Native
- React Redux
- Expo-Client
- ExpoAPI
- Apache Maven
- RabbitMQ
- MySQL
- NodeJS
- JEST og Enzyme

React Native

For utviklingen av applikasjonen har vi valgt å støtte oss på rammeverket React-Native som er en teknologi basert på React. React Native er laget spesifikt for å brukes til utvikling av mobilapplikasjoner og mange store applikasjoner på markedet idag er utviklet ved bruk av React Native. Noen eksempler er Facebook, Instagram og Skype. Teknologien bruker vanlig javascript og react bare med sine unike elementer for oppbygging av en mobilapplikasjon. React Native har den klare fordelen ved at all kode vi skriver fungerer på både Android og iOS. Det finnes selvfølgelig noen bibliotek som spesifikt har enten Android eller iOS som målgruppe.

Ved vanlig apputvikling er det vanlig å ha to forskjellige kildekoder, en for android og en til iOS. Ved å bruke React Native slipper vi alt dette ekstra arbeidet og sparer derfor endel tid. Siden dette prosjektet er på siden av andre valgfag er det derfor begrenset hvor mye tid vi får brukt. Derfor virket dette som et fornuftig valg siden vi har begrenset til ingen kunnskap innenfor apputvikling. For utfyllende informasjon om React Native se referanse [React Native].

React Redux

Siden applikasjonen er avhengig av mye data så må dette håndteres innad i applikasjonen. Flere deler av applikasjonen bruker samme data. Derfor må vi ha en måte å håndtere denne dataen og dens endringer gjennom hele applikasjonen. For å slippe å videreføre all data ned gjennom komponentene i applikasjonen har vi valgt å bruke rammeverket React Redux. All data vil endre tilstand etterhvert som brukeren endrer på data. React Redux hjelper derfor med tilstands håndtering. Dette gjør at hvis brukeren endrer på data inne i en komponent som også brukes i en annen, så vil React Redux varsle komponenten om at data har

blitt endret. Slik vil samtlige komponenter som bruker samme type data alltid holdes oppdatert. For utfyllende informasjon om React Redux se referanse [React Redux].

NodeJS

Siden applikasjon kun er en prototype har vi ikke koblet oss til på backenden til Fixrate. Dette er av flere grunner, en av dem er at all data som sendes fra backenden deres ikke trengs i applikasjonen. Dette vil derfor virke overflødig siden vi ikke trenger absolutt all data. I fremtiden vil Fixrate ordne egne endepunkter for uthenting av data ment for applikasjonen. For å simulere datautveksling mellom applikasjon og server har vi derfor valgt å sette opp en enkel nodeJS-server for dette formålet samt en egen mySQL-database for å lagre all data vi sender til serveren. Vi har valgt å bruke nodeJS fordi det er noe vi har jobbet med opptil flere ganger gjennom utdanningen vår og er veldig enkelt å sette opp. Slik har vi oppnådd kommunikasjon mellom applikasjon, server og database.

RabbitMQ

Siden alle brukere av applikasjonen skal motta varsler på telefonene sine når noe skjer, en event, har vi satt opp en apache-klient som kun står å lytter på nye eventer. Apache-klient vår bruker et rammeverk som heter RabbitMQ og i tillegg har kontakt med mySQL-databasen vår. RabbitMQ er en “message-broker” som er veldig mye brukt i dag og finnes allerede i den eksisterende løsningen til Fixrate. Man kan se på RabbitMQ som et postkontor hvor post sendes, og deretter velger postkontoret hvilket tiltak som må gjøres alt etter hvor posten skal.

Slik har vi oppnådd pushvarsler i applikasjon vår. RabbitMQ får nemlig beskjed når en ny annonse legges ut på markedet til Fixrate for deretter å sende ut en post-request til et API som håndterer utsending av pushvarslene. RabbitMQ har kontakt med mySQL-databasen vår for å hente ut ID'en til brukerne som skal motta pushvarsler. Se vedlegget med arkitekturoversikt for en bedre oversikt over hvordan denne kommunikasjonen fungerer. For utfyllende informasjon om RabbitMQ se referanse [RabbitMQ].

Expo Client

Expo Client er bare en applikasjon man laster ned på telefonen sin. Gjennom utviklingen av applikasjon kan vi enkelt starte opp vår applikasjon inni Expo Client. Slik har vi alltid full kontroll over hvordan applikasjon vår blir seende ut. Siden det er strenge krav for å slippe ut applikasjoner på AppStore(iOS) og Playstore(Android) er ikke dette reelt i prosjekttiden. Derfor vil vi under demonstrasjon av applikasjonen kjøre den gjennom Expo Client. Om produkteier ønsker å starte opp applikasjonen selv, er han derfor nødt til å installere Expo Client på sin egen telefon også. For utfyllende informasjon om Expo se referanse [Expo].

Jest og Enzyme

Jest og Enzyme er test biblioteker en bruker for enhetstesting og for å teste applikasjonen i sin helhet med flere komponenter som skal kjøres sammen. Jest er brukt for å teste komponenter på klient-siden og på server-siden. I dette systemet er Enzyme brukt sammen med Jest, der Jest er hoveddrammeverket for testingen.

Enzyme fra Airbnb gir god tilgang til å teste komponentene på en god måte. Biblioteket tilbyr funksjonene som shallow, og render. Shallow er en funksjon som for eksempel lar oss teste komponenten på et nivå, der den ikke er avhengig av at komponenten tar inn de propsene som den tilbyr. Med shallow kan vi ta skjermbilder av komponenten som blir lagret i en fil. Ved senere kjøring vil det blitt tatt flere skjermbilder og sammenligner dette med den lagrede filen. På denne måten kan en kontrollere komponentens sin konsistent i rendering, og oppdatere skjermbildene ved eventuelle oppdateringer på komponenten.

Annet

Innloggingen som brukes i applikasjonen er en ferdig komponent som allerede brukes på Fixrate's egne nettsider. Dette er rett og slett bare et utsnitt, en iFrame, som tilbyr sikker innlogging gjennom bankID. Denne har vi implementert i vår applikasjon for å oppnå den samme sikre innloggingen til brukerne gjennom React Native WebView. WebView er React Native sin egen iFrame.

Det kan også nevnes at mySQL-databasen er satt opp på en egen privat server slik at alle instanser av applikasjon har tilgang til samme data. Videre er det tenkt at node-serveren også skal opp på denne serveren siden den idag kun kjøres lokalt. Ved å få den opp på serveren muliggjør vi all bruk av applikasjonen for deltakere som vil prøve den selv på demonstrasjonsdagen.

Kapittel 3: Resultater

Ingeniørfaglige resultater

I Visjonsdokumentet ble det definert Resultatmål for prosjektet. Fleste av disse har vi klart å tilfredsstille, da også noen av disse ble satt til side på grunn av fortløpende endringer underveis i denne prosessen. Vi har tilpasset oss utviklingen på de tilbakemeldingene vi har fått fra veileder og produkteier hos Fixrate, etter diskusjoner og demo.

Det første og andre målet vi satt oss var å forenkle prosessen ved bestilling av bankinnskudd fra innskyter til bank, og bruker skal kunne logge inn ved hjelp av bankID, samt push-varsler ved nye steg i prosessen. Målene ble gjennomført, men også noe endret. Siden dette prosjektet skal jobbes videre med i en bacheloroppgave, er resultatmålene noe endret med bakgrunn i at det har blitt prioritert å legge vekt på de teknologiene som skal brukes, er implementert og testet slik at de fungerer godt i lag. Med de teknologiene på plass ligger det til rette for å implementere de funksjonalitetene som er tenkt til å utføre de målene som er satt.

Dagens system tar i bruk et webview som bruker en “dummy” bankID logg inn. Dette webviewet er lignende som det som brukes i nåværende web-applikasjon. Dette webviewet returnerer et event som vi håndterer på klientsiden for å autentisere brukerne for å logge inn. Gruppen har ikke klart å få til slik at den bruker med IOS telefon kan bruke denne bankID innloggingen. Dette var et problem som gruppen møtte på underveis, og det ble brukt mye tid på å feilsøke årsaken til problemet. Problemet ble ikke løst, og det ble avgjort at gruppen skulle legge dette til side, og heller fokusere på videre arbeid. Det ble da lagt til en vei rundt autentiseringen slik at IOS brukere kan logge seg inn med data satt statisk.

For push-varsler bruker vi Expo sitt API. Vi valgte å bruke dette API’et da det er sikkert at dette vil fungere sammen med resten av applikasjonen, også siden vi allerede bruker Expo-client for å kjøre applikasjonen i utgangspunktet. Det ble gjort mye undersøkelse på hvilke biblioteker som var tilgjengelige for å utføre push-varsling, og vi landet på å bruke Expo sitt API da det virket mest stabilt og sikkert.

Vi hadde et mål med å ha et fungerende system innen første kvartal 2019, og det målet har vi klart. Dagens system er en prototype, det er mulighet til å se tilgjengelige annonser på markedsplassen, se mer utfyllende informasjon om annonsen og banken som har lagt ut annonsen, og det er implementert en “todo” liste med en oversikt over hvilke gjøremål brukeren har. Prototypen er per dags dato ikke koblet opp mot Fixrate sine servere, det er da laget statiske data som applikasjonen bruker for å visualisere hvordan gangen vil være.

Som nevnt over er det lagt opp slik at det er muligheter for videreutvikle applikasjonen. De teknologiene som skal brukes videre på bacheloroppgaven ligger til rette, og vi har utviklet en prototype som er klar for brukertesting.

Administrative resultater

De første ukene gikk som det var planlagt i henhold til fremdriftsplanen, i den første uken som var satt av til oppstart ble det brukt tid på egenlæring i React-Native. Påfølgende uke fikk vi satt opp innloggingen gjennom webviewet som er en “dummy” innlogging med bankid autentisering.

Senere utover ble det noen avvik fra fremdriftsplanen. Det ble utført demo-visning for produkteier og veileder. Under denne demoen, ble det gitt tilbakemeldinger på hvilke forandringer som var ønskelig. Da det også ble lagt vekt på andre prioriteringer for oss. Java-backend implementeringen som vi hadde satt av fra uke 45-47 (se fremdriftsplan i prosjekthåndbok) ble endret til at vi skulle lage en RabbitMQ server. Under denne demoen ble det også flere av skjermene som vi hadde lagt til applikasjonen lagt til side, da det ikke var ønskelig å ha det. Etter demoen ble det lagt vekt på at vi skulle ha fokus på implementeringen av push-varsler, noe som kom litt tidligere i planene enn det vi hadde sett for oss.

Uke 48-49 var det i fremdriftsplanen planlagt at vi skulle jobbe med å implementere websocket server for live oppdateringer av det innholdet som skal vises frem på applikasjonen. Denne implementasjonen ble også en del som ikke skal implementeres denne omgangen. Grunnen var at det ikke ble en prioriteringssak for prototypen. Det ble heller avgjort at vi skal jobbe med statiske data, og heller avvente websocket server implementasjonen når dette systemet skal kobles mot Fixrate sine servere.

Utviklingsprosessen

Den utviklingsprosessen vi har hatt under dette prosjektet har basert seg på smidig utviklingsprosesser. Vi har fulgt den fremdriftsplanen som ble satt opp i startfasen av prosjektperioden, samt med de User-Stories som ble laget. Underveis har vi implementert de funksjonalitetene som vi definerte i user-stories.

Underveis ble det gjennomført en demo for produkteier og veileder hos Fixrate. Under denne demoen ble det avklart at flere av de user-stories'ene som vi hadde definert nødvendigvis ikke var behov for. Det ble noen forandringer i planene som vi hadde satt, samtidig som at flere punkter i fremdriftsplanen ble endret. For eksempel ble alle punktene fra user-stories'ene angående bruker som skal se prosessen under etter at den bestilling har blitt gjort, ble satt på vent og ikke prioritert under denne prosessen. Dette systemet skal vi jobbe videre med på bacheloroppgaven. Og derfor har enkelte punkter og funksjonaliteter som var planlagt at skulle implementeres blitt utsatt til over nyåret. Fokuset ble forandret til å lage en prototype, der de teknologiene som vi skal bruke videre er implementert.

Kapittel 4 - Diskusjon

I startfasen av prosjektet og under planleggingen ble det fort fastsatt at det var veldig mye ny teknologi vi skulle ta i bruk. Det ble fort kartlagt hva vi trengte å se mer på før vi startet skikkelig opp med prosjektet og hvor vi kunne finne nødvendig dokumentasjon.

Hva har vi lært

I startfasen ble det satt av mye tid til å gjennomføre et betalt kurs om React Native på Udemy.com. Dette viste seg å gi utrolig stor avkastning under utviklingen siden ingen av oss hadde noe forkunnskap eller erfaringer i React Native. Selv om det ble brukt rundt 40 timer på dette kurset, er samtlige i prosjektgruppen enige om at det var verdt det siden alternativet ville vært å starte med null erfaring. Dette ville nok medført til veldig mange timer på å finne løsninger på problemer som egentlig er veldig enkle, men som vi lærte oss gjennom dette kurset. Oppbyggingen til kurset var slik at man startet med det aller enkleste og bygde det gradvis opp til det punktet hvor man lagde fullstendige applikasjoner. For fullstendig informasjon angående kurset se referanse [Udemy-Kurs].

Videre har gruppen også blitt vant til å lese dokumentasjon som er produsert til de valgte teknologiene og bibliotekene. Denne måten å lære seg teknologi på har vist seg som veldig positiv og gitt en veldig god forståelse på hvordan teknologien fungerer og “best-practice” for implementering. Dette er ikke noe gruppen har vært vant til tidligere eller brukt mye tid på og vi har derfor lært oss å bruke dokumentasjon veldig godt gjennom prosjekttiden.

Gruppen hadde tidligere erfaringer med React og webutvikling gjennom tidligere fag i utdanningen, noe som har gitt en fordel i dette prosjektet. Med det grunnlaget vi hadde fra før, så har gruppen hatt mulighet til å lære seg enda bedre hvordan ting fungerer og henger sammen. Dette har derfor gitt en veldig positiv avkastning da gruppen var komfortabel med javascript å react. Det har derfor vært rom for gruppemedlemmene å utfordre seg enda mer faglig å lære seg enda mer i disse teknologiene.

Problemer og utfordringer underveis

Applikasjonen har en innlogging som baserer seg på bankID. Denne innloggingen har bydd på mange problemer under prosjekttiden. Den har i to tilfeller sluttet å fungere som tenkt og dette har medført til mye feilsøking og ekstra timer hvor vi var nødt å stoppe alt arbeid å gå tilbake til innloggingen igjen for å løse problemet. Første gang den sluttet å fungere klarte vi å finne ut av feilen. Feilen her var at applikasjonen ikke klarte å utveksle cookies med serveren den prøvde å autentisere seg for. Den andre gangen innloggingen sluttet å fungere brukte gruppen ca. 40 timer for å feilsøke å prøve nye ting for å få den til å fungere igjen. Her er det viktig å merke seg at det kun er på iOS innloggingen ikke fungerer, men fungerer på Android som normalt. Det ble ikke funnet en løsning til denne feilen, og det er per i dag fortsatt ikke noe løsning. Gruppen så seg derfor nødt til å legge dette til side siden det fortsatt var mye gjenstående arbeid i applikasjonen og innloggingen hadde blitt brukt alt for mye tid på. Dagens løsning ble å lagre gyldig innloggingsdata statisk inntil videre.

Det har også vist seg vanskelig å finne hjelp til problemer vi har møtt siden React Native er en teknologi under stadig utvikling og det skjer mye endringer. Med vanskelig menes det at løsninger som ble funnet på forum som stackoverflow [Stackoverflow] kunne være utdatert siden løsningene til problemet ofte var flere år gamle og react er en teknologi som stadig oppdateres.

3-parts bibliotek som vi kom over kunne også virke som den endelige løsningen til et problem, men her var også problemet at biblioteket ikke ble holdt oppdatert av eieren og var derfor utdatert ifht nyeste versjon av React Native. Dette ble rett og slett løst ved at vi lagde våre egne løsninger i React Native. Dette var både positivt og negativt. Vi var derfor i noen tilfeller nødt til å lage egne løsninger, noe som gir enda bedre læring, men samtidig øker timeforbruket på utviklingen av applikasjonen.

Fixrate ønsket at vi skulle se på muligheten for å implementere innlogging vha. biometrisk data som fingerskanning og face-id. Gruppen begynte å undersøke mulighetene for dette, men det viste seg at vi hadde behov for tredjeparts biblioteker som ikke fungerte i dagens løsning av React Native. Vi fikk så beskjed om å legge dette litt til side å heller fokusere på målene som var satt for å heller ta opp emnet i senere tid.

Gruppen har ikke hatt demonstrasjon av applikasjonen ofte nok for Fixrate. Dette er av flere grunner. Først måtte man komme opp med en funksjonell prototype som man kunne vise frem, noe som tok tid. Siden prosjektet kjøres parallelt med andre fag så gikk det ikke like fort å få en funksjonell prototype. Ved første demo hadde produsert veldig mye. Mye av dette, som nevnt tidligere, ble kuttet bort etter første demonstrasjon, fordi Fixrate ikke ønsket å ha det i applikasjon sin likevel. Vi har derfor konkludert med at i fremtiden skal en ha demonstrasjoner hyppigere slik at mye unødvendig arbeid ikke forekommer.

Ressursbruk

Som nevnt tidligere har gruppen brukt mye dokumentasjon som er gitt ut av på de offisielle sidene til de valgte teknologiene samt innhentet kunnskap gjennom gjennomføring av kurset på Udemy. Dette har vært en positiv opplevelse siden dokumentasjon tidligere har vært krevende å lese. Nå som vi har erfaring med mye av teknologien blir derfor dokumentasjon enklere å lese samt mer forståelig. Dette har resultert i at gruppen har fått en økt forståelse for dokumentasjon som er skrevet på høyt faglig nivå og vi har derfor lært veldig mye av det. Øvrige ressurser er knyttet til forum og da spesielt Stackoverflow og Issue-boards til bibliotekene vi har brukt på Github. På Stackoverflow har vi alltid funnet en trådstarter som har opplevd samme problem som oss og har fått flere forslag til løsninger. Derfor har forumet på Stackoverflow vært en veldig stor ressurs. Ved problemer til bibliotek har vi stadig vekk sjekket issue-boardet til biblioteket for å se om det er en bug eller om andre brukere av biblioteket har opplevd samme problem.

Resultater

Resultatene som ble satt i visjonsdokumentet blir sett på som oppnådd forutenom innloggingen på iOS vha. bankID. Innloggingen ble som nevnt tidligere lagt til side til fordel av å få resten av funksjonaliteten i applikasjonen på plass da gruppen ikke kom noen vei med løsningen av problemet.

Vi har implementert pushvarsler i applikasjonen, realisert en markeds plass på mobil med samme funksjonalitet som på nettstedet deres og lagt inn en ønsket side hvor bruker kan se nødvendige gjøremål som for eksempel at det er mottatt en ny bestilling eller ankommet en ny annonse på markedet. Dette står i samsvar med de målene som ble satt i starten av prosjektet og betegnes derfor som fullført.

Applikasjonens GUI er forsøkt best mulig likt fra Fixrate's egne nettsider for å oppnå mest mulig likhet mellom applikasjon og nettsted.

Resultat i form av kildekode kan diskuteres og kanskje, enkelte steder, oppleves som litt rotete med tungvinte løsninger. Dette er et resultat av mye ny teknologi som blir tatt i bruk og lite til null erfaring med implementasjon av løsningene. Mye av implementasjon og forståelse viste seg å være utfordrende, men ga ønsket resultat til slutt. Mye ble gjort med innstillingen om å først få det til å fungere, deretter re-faktorisere koden for å oppnå en bedre leselig kode.

Kapittel 5: Konklusjon og videre arbeid

Som nevnt tidligere ble ikke innlogging vha bankID på iOS innfridd. Dette var en følge av en feil som gruppen ikke klarte å finne en løsning på og videre lagt til side for å få resten av målene implementert. I senere tid ble også muligheten til å signere dokumenter elektronisk gjennom appen fjernet som funksjonalitet av Fixrate siden dette er en administrativ oppgave som de mente ikke har tilhørighet i en mobilapplikasjon. Listen over disse funksjonelle egenskapene finner man i kapittel 6 i det vedlagte visjonsdokumentet.

Applikasjon har derfor de funksjonelle egenskapene som ble satt i kapittel 6 i visjonsdokumentet. Når det kommer til effektivisering av Fixrates tjeneste samt gjøre brukerne mer oppmerksom på endringer i markedet er vanskelig å si. Det har gjennom utviklingstiden ikke blitt foretatt noen form for brukertester, og det er derfor vanskelig å konkludere om disse målene er nådd. I fremtiden vil det bli gjort brukertester på kundebasen til Fixrate slik at man kan få tilbakemeldinger på hva som virker bra og dårlig. Mangelen på brukertester er rett og slett et resultat på for lite tid da man må ha en funksjonell prototype veldig tidlig, noe som ikke var mulig gjennom prosjekttiden.

Videre arbeid skjer i vårsemesteret 2020 som en bacheloroppgave. Første prioritet videre vil være å få innloggingen med bankID til å fungere som tenkt på iOS. Videre er det ment at det skal lages enda mer funksjonalitet samt. koble seg til på backenden til Fixrate slik at vi kan få en ny versjon av selve applikasjonen. Applikasjonen i dag er å anse som versjon 1.

Høstens arbeid var ment som prototype for videre arbeid. Etterhvert som videre arbeid pågår skal det også bli gjort brukertester slik at man kan få tilbakemeldinger på nåværende prototype å få innskudd til hva som bør gjøres ytterligere med applikasjonen.

Under vårsemesteret er det planlagt møte med flere ansatte i Fixrate, som vil være med på å definere flere funksjonaliteter og forbedringer til applikasjonen.

Referanser

[Udemy-Kurs]Schwarzmüller, M. (2019). [online] Udemy. Available at: <https://www.udemy.com/course/react-native-the-practical-guide/> [Accessed 28 Sep. 2019].

[React Native] Facebook.github.io. (2019). *React Native · A framework for building native apps using React*. [online] Available at: <https://facebook.github.io/react-native/> [Accessed 28 Sep. 2019].

[React Redux] Redux.js.org. (2019). *Usage with React · Redux*. [online] Available at: <https://redux.js.org/basics/usage-with-react> [Accessed 28 Sep. 2019].

[RabbitMQ] Rabbitmq.com. (2019). *Messaging that just works — RabbitMQ*. [online] Available at: <https://www.rabbitmq.com/> [Accessed 28 Sep. 2019].

[Expo] Expo.io. (2019). [online] Available at: <https://expo.io/> [Accessed 28 Sep. 2019].

Vedlegg

Prosjekthåndbok

Visjondokument

Kravdokumentasjon

Timeliste

Systemarkitektur

Systemdokumentasjon