

Shahed Saeed Abbas

Bacheloroppgave

Nøkkelfri Adgangskontroll

Bacheloroppgave i dataingeniør

Veilede: Nils Tesdal

Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Bacheloroppgave

2020



Shahed Saeed Abbas

Nøkkelfri AdgangsKontroll
Bacheloroppgave i dataingeniør
Veilede: Nils Tesdal

Bacheloroppgave
Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden

Forord

Denne bacheloroppgaven ble skrevet som avsluttende innlevering i forbindelse med studieretning dataingeniør ved Norges teknisk-naturvitenskapelig universitet (NTNU) i Trondheim. Jeg vil takke min veileder for sin tilgjengelighet og hjelp og rask svar ved krisetider. Jeg vil også takke både Sodvin Systemer sin daglig leder Tore Kjørsvik som ga meg sjanse for å fullføre dette prosjektet, og John Myren som veiledet meg under utviklingsprosessen.

Dette prosjektet var på en måte veldig lærerikt for meg på mange områder og erfaringer som jeg fikk i løpet av utviklingsprosessen har åpnet veien til meg for å bygge videre på det jeg har lært og gjort.

Oppgaveteksten

Oppgaven var å lage en prototype applikasjon på telefon som kan fungere som nøkkel for å låse opp dør. Applikasjonen skal i vårt konkrete tilfelle kunne erstatte nøkkelbrikke/plastkort for medlemmer som ønsker å trene på et treningssenter.

Prinsippet er:

Sodvin Systemer ønsker å få laget en hybridapplikasjon som benytter Native funksjonalitet for å kommunisere med Bluetooth Beacons. En Bluetooth Beacon vil identifisere stedet/døren som skal låses opp. Brukeren er autentisert i appen, og vi vet dermed hvilket medlem som ønsker å låse opp døren. Appen vil kommunisere med tjenester på Sodvin Systemer sin servere via webservice (REST api), og det er disse serverside tjenestene som faktisk validerer kombinasjonen medlem/dør og som evt. sørger for å låse opp døren.

Sammendrag

I denne oppgaven ble det utviklet en prototype applikasjon for både iOS og Android med kun en kodebase som fungerer på begge plattformer. Denne applikasjon skal fungere som nøkkel for å åpner dører i treningssenteret. Denne prototypen ble utviklet for at firma kan senere utvikle det videre eller implementere funksjonaliteten som åpner døra i deres eksisterende applikasjon. Dette var en ide som vil digitaliserer treningssenteret sin plastkort eller brikke som åpner dørene til treningssenteret.

Løsning basert på å implementere ibeacon ved å bruke react native rammeverket. Det ble brukt google sin firebase authentication for å autentiserer brukere. Det ble også brukt en enkel mySql database for å registrere beacons.

Innhold

<i>Forord</i>	1
<i>Oppgaveteksten</i>	2
<i>Sammendrag</i>	3
<i>Kodesnutt</i>	8
<i>Akronymer</i>	9
<i>Kapittel 1 Introduksjon og relevans</i>	10
1.1 Bakgrunn	10
1.2 Problemstilling	10
1.3 Effektmål	11
1.3.1	11
1.3.2	11
1.4 Oppgavens struktur	11
<i>Kapittel 2 Teori</i>	12
Innledning	12
2.1 Design teori	12
2.1.1 Interaksjonsdesign.....	12
2.1.2 Brukskvalitet.....	13
2.2 Ibeacon(Bluetooth Low Energy BLE)	13
2.2.1 Hva er Ibeacon(BLE)	13
2.2.2 Hvor denne teknologien kan ble benyttet.....	14
2.3 Metoder.....	14
2.3.1 Scrum.....	15
2.3.2 Det agile manifesto	15
2.3.3 Scrum tre pilarer	16
2.3.4 Scrum artefakter(Scrum artifacts)	16
2.3.5 Scrum roller og seremonier	16
2.4 Database.....	17
2.4.1 Databasehåndteringssystem (Database management system DBMS).....	17
2.4.2 SQL språket	17
2.5 applikasjons teknologier.....	18
2.5.1 React	18
2.5.2 React Native	18
2.5.3 JSX (JavaScript XML).....	19
2.5.4 Watchman	19
2.5.5 Biblioteker (Libraries)	19
2.5.6 Firebase Autorisering(Firebase Authentication)	20
2.5.7 Wireframe	20
<i>Kapittel 3 Valg av teknologi og metodikken</i>	21
3.1 Innledning	21
3.2 Utviklingsmetodikk	21
3.3 Native vs Hybrid mobilapplikasjon	21

3.4 Innlogging	22
3.5 Design	23
3.6 iBeacon(Bluetooth Low Energy BLE)	23
3.7 Backend	24
3.8 Verktøy	24
3.8.1 VS code	24
3.8.2 flaticon	24
3.8.3 Locate Beacon	24
3.8.3 draw.io	24
Kapittel 4 Resultater	25
4.1 Innledning	25
4.2 Vitenskapelig resultat.....	25
4.2.1 Design.....	25
4.2.4 Produktet	29
4.3 Ingeniørfaglig Resultat.....	30
4.4 Administrative Resultater	34
Kapittel 5 Drøfting av Resultater	35
5.1 Vitenskapelig resultater – diskusjon.....	35
5.1.1 Design.....	35
5.1.2 Produktet	35
5.2 Ingeniørfaglig resultater – diskusjon.....	36
5.3 Evaluering av Systemet som helhet	36
Kapittel 6 Konklusjon og videre arbeid	37
6.1 Konklusjon og videre arbeid	37
Vedlegg A Prosjekthåndbok	38
A.1 Framdriftsplan	38
A.2 Møteinnkallinger	38
A.3 Referater	41
A.4 Timeliste.....	44
B.1 Innledning	52
B.2 Sammendrag – problem og produkt.....	52
2.1 Problem Sammendrag	52
2.2 Produktsammendrag	52
B.3 Overordnet beskrivelse av interessenter og brukere	52
B.4 User stories	53
B.5 Produktets funksjonelle egenskaper	53
B.6 Ikke funksjonell krav	54
B.7 Prototype	54

<i>Vedlegg C Systemdokumentasjon</i>	58
1. Introduksjon	59
C.1 Hensikt	60
C.2 Arkitektur	60
2.1 Arkitekturskisse	60
2.2 Tekstlig beskrivelse	60
C.3 Prosjektstruktur	61
C.5 server-tjeneste	64
C.6 Sikkerhet	65
C.7 Dokumentasjon av kildekode, Installasjon og kjøring	65
<i>Referanser</i>	68

Figurer

Figure 1: illustrasjon på interaksjonsdesign.....	12
Figure 2: beacons typer.....	13
Figure 3: illustrasjon på hvordan beacon fungerer	14
Figure 4: scrum-faser	15
Figure 5: Scrum prosesser.....	16
Figure 6: iterative prosessen	21
Figure 7: viser virkemåten til autentisering	23
Figure 8: skjermbilder av en fungerende applikasjon på simulator	26
Figure 9: skjermbilder av det siste designe som kjører på emulator.....	28
Figure 10: viser eksempel på tilbakemelding som en interaksjonsdesign prinsippet	29
Figure 11: viser tjenester firebase tilbyr og Sign-in providers	31
Figure 12: viser liste over registrerte brukere	32
Figure 13: viser resultat fra konsoll på oppdagede beacons	33
Figure 14: viser en del av beacons informasjon på skjermen til applikasjon	34
Figure 15: viser resultat når det sende en http-request til serveren for å åpne døra	34
Figur A 1: viser framdriftsplan	38
Figur B. 1: første utkast prototype	55
Figur B. 2: andre utkast prototype	57
Figur C. 1: Arkitekturskisse	60
Figur C. 2: Prosjektstruktur.....	61
Figur C. 3: Frontend og backend struktur	62
Figur C. 4: Struktur på Frontend mappe	63
Figur C. 5: beacons tabell i database	64
Figur C. 6: viser android setup instruksjon for kontakt.io beacons	66
Figur C. 7: viser iOS setup instruksjon for kontakt.io beacons	67

Kodesnutt

kodesnutt 1: En måte på å oppdage beacons og sette dem i en liste	33
kodesnutt 2: illustrerer endepunktet i serveren	64
kodesnutt 3: illustrerer http-forespørsel til firebase server for innlogging	65

Akronymer

IxD : interaksjonsdesign

BLE : Bluetooth Low Energy

DBMS: Database management system

SQL: Structured Query Language

API: Application Programming Interface

npm: Node package manager

JSX: JavaScript XML

XML: Extensible Markup Language

HTTP: Hypertext Transfer Protocol

REST :Representational state transfer

SDK: Software development kit

UI: User interface

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets

VS code : visual studio code

Kapittel 1 Introduksjon og relevans

1.1 Bakgrunn

Idag skjer digitalisering av samfunnet nesten på alle samfunnsområder. Vi får stadig nye teknologier som realiseres i virkeligheten i form av webapplikasjoner, mobilapplikasjoner eller desktop applikasjoner eller fysiske tilstander. Alt dette er med for å skape en enkel hverdag for både den enkelte og for samfunnet. Generelt sett ga oppveksten av teknologi både individuelle og samfunnet flere fordeler, men samtidig er ingen teknologi feilfri. Over flere år har teknologien endret samfunnet på flere områder og vi mennesker ble også forandret. Vi ble mer og mer avhengig av enkelte fysiske gjenstander som for eksempel mobiler. Mobiltelefoner fungerer idag som en gjenstand som brukes alt fra søking om informasjon på nett, som betalingsløsning eller som adgangskontroll og mye mer.

Overgang fra mekaniske låsesystemer til elektroniske adgangskontroller har flere argumenter. En av de største argumentene er sikkerhet, for eksempel sikring av lokalene og ha oversikt over hvem som har tilgang hvor. utfordringer med de fysiske nøklene oppstår når man mister en nøkkel og hvis sylindrene ikke byttes ut og nøklene faller hos uautoriserte folk, som kan låse seg inn i lokalene som nøkkelen gir tilgang til [1a].

Idag er det flere former for nøkkelfrie adgangssystemer. Selv om mesteparten av bedrifter bruker ikke lenger fysiske mekaniske nøkler, men nøkkelfrie adgangssystemer har også en form av fysiske løsninger som for eksempel plastbrikke som brukes av noen treningssentrene. Andre former av adgangssystemer som ble utviklet over tid er adgangskort og dette er et plastkort med innebygget elektronikk for trådløs kryptert kommunikasjon med kortleseren. Sånne kort må følges med pinkode som man taster inn for å låse opp. Andre former for nøkkelfrie adgangssystemer er biometri som innebære skanning av øye eller en finger for å autorisere om denne personen har tilgang eller ikke[2]. Videre en annen form for å åpne døra uten være noe behov for fysisk plastkort eller plastbrikke er en applikasjon på mobil som kommuniserer trådløst enten ved å bruke NFC teknologi eller bluetooth. Sistnevnte er kanskje den meste aktuelt for oss fordi vi kan lett glemme kortet eller den plastbrikken når vi skal dra til jobb, treningssenter eller andre plasser som krever adgangsnøkkel, men det er kanskje veldig sjeldent at vi glemmer mobilen fordi den er hoved gjenstand i det daglig. Idag er det derfor mest vanlig å utvikle mobilapplikasjoner som kan være nyttig for de enkelte.

1.2 Problemstilling

Problemstilling som jeg har valgt:

“Hvordan kan man ivareta interaksjonsdesign og brukskvalitet i en hybrid app på en lett og mindre tidkrevende måte”

“Hvordan kan man utvikle en prototype hybrid app som benytter native funksjonalitet til kommunisere med Bluetooth Beacons som skal identifisere døra som skal låses opp?”

1.3 Effektmål

1.3.1

Sodvin Systemer AS ønsker å lage en applikasjon på telefon som kan fungere som nøkkel for og låse opp dør. De ønsker å lage en hybrid app som benytter native funksjonalitet for å kommunisere med Bluetooth Beacons. En Bluetooth Beacon vil identifisere stedet/døren som skal låses opp. Brukeren er autentisert i appen, og vi vet dermed hvilket medlem som ønsker å låse opp døren.

1.3.2

Målet med prosjektet er å lage en prototype app som skal fungere som nøkkel og erstatte nøkkelbrikke/plastkortet som brukes idag. Så Sodvin Systemers langsiktige mål er å utvikle en app som skal erstatte nøkkelbrikke som brukes idag i treningssenteret, og dermed gjør det lettere for medlemmer å få tilgang til senteret.

1.4 Oppgavens struktur

Etter dette kapitlet vil det komme et kapittel som beskriver teorien om de teknologier som ble brukt for å utvikle dette systemet. Kapittel 3 vil beskrive hvilke teknologier som ble valgt og utviklingsmetode som ble benyttet i utviklingsprosessen. Neste kapittel vil beskrive resultater som ble oppnådd gjennom utviklingsprosessen. Kapittel fem vil være et kapittel hvor det er diskusjon og drøfting av resultatet som ble beskrevet i forrige kapittel. I tillegg sammenligne de resultatene som ble oppnådd med de målene som ble satt i begynnelsen. Siste kapittel vil omhandle konklusjon og svar på problemstillingen. Resten av denne rapporten vil være referanser som ble benyttet og vedlegg.

Kapittel 2 Teori

Innledning

Dette kapitlet skal omhandle kunnskap og teori som ligger bak de forskjellige teknologier som ble benyttet i dette prosjektet. Selv om dette prosjektet er begrenset kun til frontend, men i et punkt var det nødvendig med litt backend. Dermed skal dette kapitlet ta med alt teori om både teknologier som ble brukt i front- og backend.

2.1 Design teori

2.1.1 Interaksjonsdesign

IxD er forkortelse for interaksjonsdesign og denne er et stort fagfelt som handler om å designe samhandling mellom mennesker og digitale systemer, objekter, tjenester og miljøer. Dette inkluderer de fleste systemer som vi bruker i hverdagen. Interaksjonsdesign handler å designe systemer som er enkle, effektive og behagelig å bruke. Interaksjonsdesign er en designvitenskap som handler om både å forstå brukerinteraksjon og om hvordan vi kan designe slike interaksjoner best mulig. Prosessen i interaksjonsdesign består av følgende trinn:

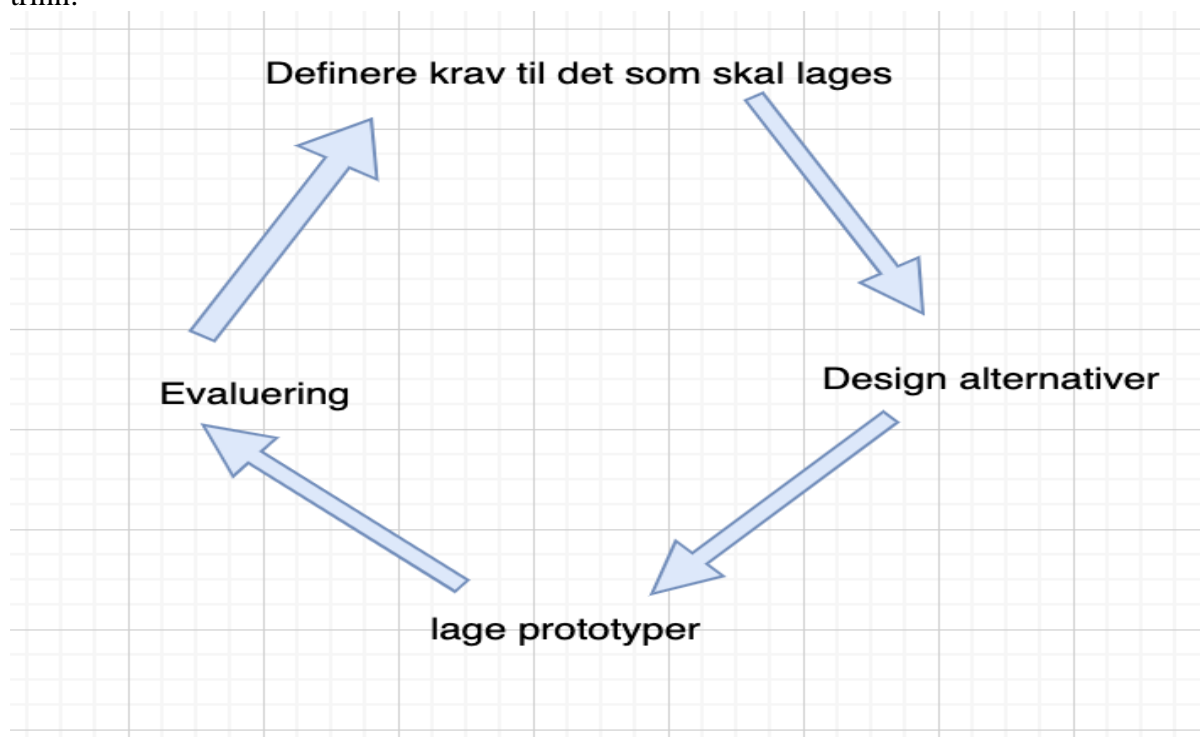


Figure 1: illustrasjon på interaksjonsdesign

Interaksjonsdesign er en iterative prosess som innebærer at disse trinnene ovenfor gjentas flere ganger til man blir fornøyd med det man laget. Det er fem prinsipper for interaksjonsdesign [1b]. Disse prinsippene er basert på boken Interaction Design: beyond human-computer interaction av Rogers, Sharp og Preece:

1. **Synlighet:** Det er viktig at funksjoner ved et system er synlige, både ved å vise at de er tilstede og å vise hvordan de skal brukes.

2. **Feedback:** Feedback handler om at systemet signaliserer til brukeren hva som foregår. For eksempel hvis brukeren trykker på send skjema, så må systemet signaliserer til brukeren om skjemaet ble sendt eller ikke ved å gi tilbakemelding.
3. **Begrensninger:** Noen ganger trenger vi å legge begrensninger på hvilke former for interaksjon som er mulig i et gitt øyeblikk, slik at det ikke oppstår feil.
4. **Konsistens:** Dette handler om å lage ting slik at brukere gjør lignende ting på lignende måter. Det sparer tid og tankeaktivitet.
5. **Tilbydelser(affordances):** Det handler om hvilke handlinger en bruker opplever at hun kan gjøre med et system. Denne opplevelse av hvilke handlinger systemet tilbyr oss vil henge sammen både med hvordan systemet er laget og hvilke erfaringer vi har gjort oss tidligere med andre systemer. For eksempel vil de fleste automatisk anta at en knapp kan trykkes på [1b].

2.1.2 Brukskvalitet

I boka «Usability Engineering» [3a], beskriver Jakob Nielsen brukskvalitet for datasystemer som sammensatt av fem faktorer, nemlig at systemet er:

1. Lett å lære seg,
2. Effektivt å bruke,
3. Lett å huske fra gang til gang,
4. Fører til få brukerfeil og å gjøre det lett å hente seg inn fra eventuelle feil.
5. Behagelig å bruke

2.2 Ibeacon(Bluetooth Low Energy BLE)

2.2.1 Hva er Ibeacon(BLE)

Ibeacon er en trådløs teknologi som ble utviklet av Apple. Den bruker bluetooth for å kommunisere. Den bruker batteri og finnes i forskjellige størrelser og utseender. Beacon kommuniserer med smarttelefoner ved å bruke bluetooth, disse beacons enhetene sender signaler til telefoner som har installert applikasjoner som interakt med beacon ellers mottar ikke beacon noe signal og den samler ikke noe data. Beacon enheten gjør mulighet for interakt med ios- og android enheter, dette gjelder imidlertid ikke bare smarttelefoner men kan være smartklokka eller andre enheter som støtter ibeacon. Nedenfor er et bilde som viser forskjellige ibeacon.



Figure 2: beacons typer[pic1]

BLE er en forkortelse for Bluetooth Low Energy og dette er teknologi som ble implementert i 4.0 standard i 2010. Fordelen med denne teknologien er at den bruker mindre strøm. Denne teknologien ble brukt for å lage treningsarmbånd. Den finske produsenten er den meste kjente for å ha laget en aktivitetsmonitor som ikke bare brukes når man trener, men man kan bruke

hele dagen og den viser hvor aktiv man er i løpet av dagen og hvor godt man sover i løpet av natten[3b][4].

Måten disse beacon-ene fungerer på er ved å plassere de på steder hvor man ønsker å kommunisere noe til en mottaker. I gjennomsnitt sende iBeacon-beacons opptil 70 meter, så når en mottaker kommer innen denne rekkevidden til en Beacon aktiviseres en eller flere apper hos mottakeren som trigger et varsel, eller viser relevant informasjon for mottakere. Informasjon som mottakeren mottar kan være spesialtilbud fra butikken som man står ved siden av eller en restaurant og andre relevante ting. Denne teknologien åpner også døren for mobilbetaling [5]. Figuren nedenfor er et eksempel som illustrerer hvordan beacon fungerer i markedsføring[6].



Figure 3: illustrasjon på hvordan beacon fungerer[6]

2.2.2 Hvor denne teknologien kan ble benyttet

Denne teknologier ble brukt i flere forskjellige ting, ikke bare markedsføring. Noen benytter denne teknologien for å spore noe for eksempel i en veldig stor bedrift med mange ansatte, så kan man spore hvor de ulike ansatte befinner seg om man trenger å ha kontakt med dem. Man kan også spore tingene sine. La oss si for eksempel at en person hele tiden glemmer hvor han/hun lagt sin headset og da kan man legge en iBeacon på den og lett finne den etterpå. Disse er bare eksempler som man peker på for å si at disse beacons brukes for lokasjonstjeneste.

Noen brukstilfeller hvor disse beacons kan benyttes er innenfor utdanningsindustri. De fleste universiteter har en del arrangementer i løpet av studieåret, vanligvis legger administrasjonen innslag om disse arrangementer på oppslagstavler til de forskjellige avdelinger og mange av disse innslagene blir ignorert og dermed blir det tap av innsats og materiale. I stedet for kan beacons sende varsler om kommende aktiviteter eller arrangementer til studentens enhet når han/hun går forbi den ansvarlig avdelingen[7]. Andre bruksområder kan være på museum og da kan man bruke beacons på de forskjellige malerier, skulpturer eller historiske momenter slik at de besøkende som kommer forbi disse får notifikasjon på deres enheter som gir informasjon og kunnskap om disse. Ibeacons har åpnet for store muligheter innenfor flere bruksområder som helse, reise, marketing osv.

2.3 Metoder

2.3.1 Scrum

I alle prosjekter er det en fordel å basere arbeidsprosessen på en velprøvd prosjektmodell. Det er flere prosjektmodeller man kan benyttes, for eksempel fossefallsmodellen som går ut på å dele prosjektet i separate faser. De fleste bedrifter bruker idag iterative metoder i deres IT-prosjekter, og denne metoden er nemlig scrum. Scrum er en strukturert rammeverk for å støtte kompleks produktutvikling. Den er iterativ og basert på trinnvis levering. Scrum er teambasert og avhengig av nærkontakt med kunden. Scrum bruker også verdiene i Agile Manifesto som skal uttales i neste delkapittel.

Det er tre faser i scrum. Den første fase er planleggingsfase hvor man etablerer de generelle målene for prosjektet og designe programvarearkitekturen. Neste fase er kjøring av sprint syklusen hvor hver syklus utvikles trinnvis. Siste fase er avslutningsfase hvor prosjektet pakkes opp og fullfører nødvendig dokumentasjon. figur nedenfor illustrerer kort disse tre fasene[8]

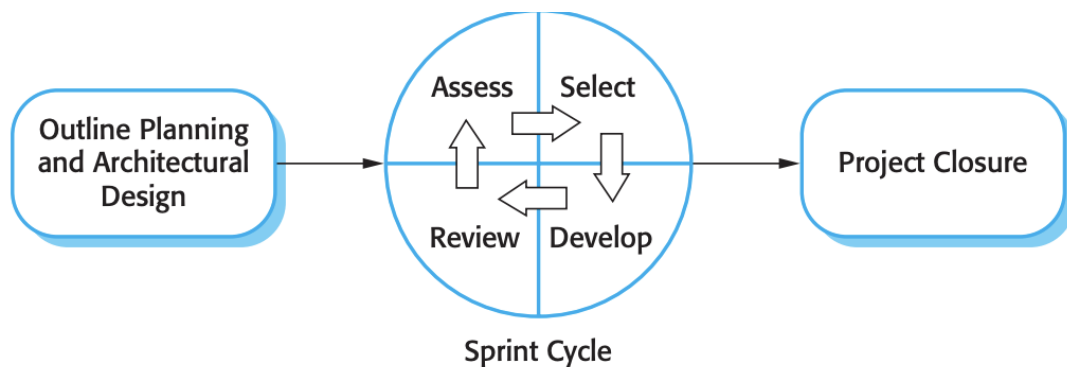


Figure 4: scrum-faser

2.3.2 Det agile manifesto

Prinsippet bak Det agile manifesto er at den gir den største prioritering til å tilfredsstille kunden gjennom tidlig og kontinuerlige leveranser av programvare. Det er mulig med endringer selv om den er seint, fordi smidige prosesser bruker endringer for å skape konkurransefortrinn for kunden[9].

De verdiene som agile manifesto står for er:

- **Personer og samspill** fremfor prosesser og verktøy.
- **Programvare som virker** fremfor omfattende dokumentasjon.
- **Samarbeid med kunden** fremfor kontraktsforhandlinger.
- **Å reagere på endringer** fremfor å følge en plan.

Dette vil med andre ord si at selv om punktene som står til høyre har en verdi, så verdsetter vi punktene til venstre enda høyere[10].

Så med denne metodikken kan man unngå å bruke veldig lang tid på en prosess som kan mislykkes på grunn av små feil fordi den er teambasert og teamet har daglige møter. Dette samarbeidet og nær kommunikasjon gjør at prosessen blir på rett spor. Dessuten har teamet nær og tett kontakt med kunden noe som gjør at teamet forstår målene på en bedre måte og kommer raskt med løsninger. I tillegg til den tidlige leveransen av prosjektdeler gjør at det blir lettere å tilpasse endringer og nye behov.

2.3.3 Scrum tre pilarer

- **Transparency** - åpen prosess: dette innebærer å representere fakta som den er. Alle involverte mennesker fra kunden til og med bidragsytere må være åpen i sin daglige omgang med de andre.
- **Inspection** - det er å kontrollere mot sprintmålet og oppdage uønsket avvik. For eksempel kan scrum teamet vise produktet på slutten av hvert sprint til kunden, og dette er en veldig effektiv måte å samle tilbakemeldinger for eventuell endring.
- **Adaptation** - det innebære tilpasning for å minske avvik, og det handler om kontinuerlig forbedring, evnen til å tilpasse seg basert på resultater fra inspeksjonen. Alle bedrifter ser på at forbedring av produktet må ha noe positivt til firma med tanke på profitt, for eksempel er det økt avkastning på investering eller er det reduserte totale eierkostnader gjennom forbedret programvarekvalitet og forbedret til kundenes og ansattes tilfredshet.

2.3.4 Scrum artefakter(Scrum artifacts)

Scrum består av disse elementene:

- Produkt backlog
- Sprint backlog
- Burndown chart

Produkt backlog er en liste over alle funksjonaliteter som ønskes i produktet. Denne listen settes vanligvis av scrum teamet og produkteier. Denne backloggen er dynamisk som innebærer at den starter med små liste som etterhvert vokser med flere krav.

Sprint backlog er en liste over oppgaver som scrum teamet forplikter seg til å fullføre i løpet av sprinten.

Burndown chart er et diagram som viser gjenværende arbeid i sprint backlog. Diagrammet bør oppdateres hver dag i løpet av det daglige scrum møtet. Diagrammet gir en enkel visualisering av sprint framgang.

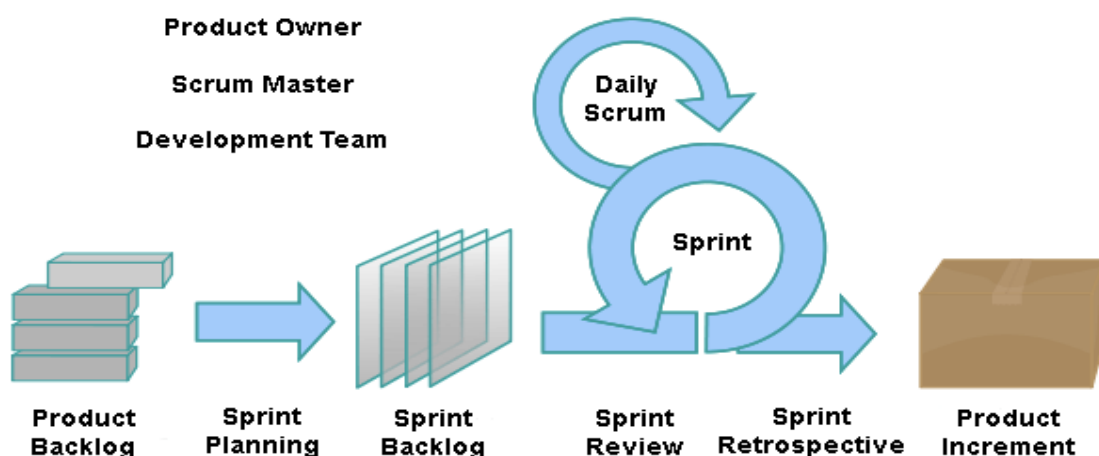


Figure 5: Scrum prosesser [pic2]

2.3.5 Scrum roller og seremonier

Det er tre hovedroller i scrum:

- Produkteier

- Scrum master
- scrum teamet

Produkteier: er kundens stemme, har ansvar for å håndtere produkt backlog, ansvarlig for lønnsomhet og det er produkteier som definerer eller justerer funksjonalitet til produktet. Dessuten er det produkteier som bestemmer utgivelsesdato og om arbeidet er akseptert eller avvist.

Scrum master: hans rolle er å sørge for at scrum teamet er i full funksjonalitet og produktiv, sørger for at prosessen følges og aktiverer samarbeid på tvers.

Scrum team: det er utviklingsgruppe som har ansvaret for å bestemme sprint målet, organiserer arbeidet sitt og vise et demo resultat som fungerer til produkteier.

Scrum har også fire seremonier:

- Sprint planleggingsmøte
- Daglig scrum møte
- Sprint gjennomgang
- Sprint retrospektiv

2.4 Database

Ramez Elmasri & Shamkant B. Navathe definerer database som en samling av relatert data, og med data mener de fakta som for eksempel navn, telefonnummer osv[11].

2.4.1 Databasehåndteringssystem (Database management system DBMS)

Databasehåndteringssystem er et datastyrt system som gjør det mulig for brukere å opprette og vedlikeholde en database. DBMS er et generelt programvaresystem som letter prosessene for å definere, konstruere, manipulere og dele databaser mellom forskjellige brukere og applikasjoner. Det å definere en database innebærer å spesifisere datatyper, datastrukturer og begrensninger for dataene som skal være lagret i databasen.

Andre viktige funksjoner som DBMS tilbyr er beskyttelse av database og vedlikeholde den over lengre perioder. Med beskyttelse menes det å beskytte database mot funksjonsfeil i maskinvare eller programvare og beskyttelse mot uautorisert eller ondsinne tilgang [11].

2.4.2 SQL språket

Sql står for Structured Query Language. Sql brukes for å sende sql spørringer mot databasehåndteringssystem for for å hente data fra database, man kan også bruke sql kommandoer for å opprette, oppdatere, eller slette en tabell i database. For eksempel for å opprette en tabell skriver man[11]:

```
CREATE TABLE student;
```

Man kan også spesifiserer hvilke kolonner som skal være med:

```
CREATE TABLE student (  
  firstname VARCHAR(15) NOT NULL,  
  lastname VARCHAR(15) NOT NULL  
);
```

Man kan også bruke sql spørringer for å hente data fra database ved å bruke de reserverte ordene som SELECT, FROM , WHERE osv. Eksempel:

```
SELECT lastname  
FROM student  
WHERE firstname = "Shahed";
```

Denne spørringen ovenfor returnerer etternavnet til 'Shahed' fra database om den er registrert.

2.5 applikasjons teknologier

2.5.1 React

React er et JavaScript bibliotek som ble utviklet av facebook. React brukes for å lage brukergrensesnitt. Det ble laget i utgangspunktet for å utvikle en enkelt-side(single-page) eller mobilapplikasjoner. Komplekse react applikasjoner krever bruk av ekstra biblioteker for tilstandsstyring og interaksjon med API. Redux, axios, React router er eksempler på slike biblioteker[12].

2.5.2 React Native

React Native er også et JavaScript rammeverk for å lage virkelige mobile applikasjoner for Android og iOS. Dette rammeverket er basert på React som er facebook javascript biblioteket. React retter seg mot nettleseren, men React Native retter seg mot mobile plattformer. React Native "bridge" påkaller de native rendering-API-ene i Objective-C (for iOS) eller Java (for Android). Dermed vil applikasjonen din gjengis ved å bruke ekte mobile UI-komponenter, ikke webvisning, og vil se ut og føles som alle andre mobilapplikasjoner[13].

2.5.2.1 React Native prosjekt alternativer

- **Expo CLI** : det er å bygge en react native prosjekt ved å bruke expo verktøy for og generere og administrere react native app-en. Expo er en tredje part tjeneste som er helt gratis å bruke. Det som expo gir er en slags administrert arbeidsutvikling for apputvikling. Det tar mye av kompleksiteten bort fordi det kan være veldig komplekst å bygge React Native-apper. Ulempen med expo at den fungerer som innpakning rundt appen og har begrensninger.
- **React Native CLI:** her utvikler man egen app som trenger at man installere en del verktøy som omtalt i(delkapittel 2.5.2.2 React Native verktøy). Denne brukes av utviklere med veldig stor erfaring og vil bygge komplekse apper.

2.5.2.2 React Native verktøy

Det er behov for noen få verktøy for at man klarer å bygge en React Native applikasjoner:

- **Xcode:** for å kunne bygge en iOS applikasjon, så er det nødvendige med Xcode IDE. React Native fungerer på iOS 7 eller høyere og dette innebærer at man krever Xcode versjon 7 eller høyere. Xcode hjelper med å teste appen på simulator[14].

- **Google Chrome:** Chrome har et utviklingsverktøy som hjelper med å debugge appen som kjører på iOS simulator eller på en virkelig iOS enhet[14].
- **Homebrew:** er “package manager” for macOS. Dette brukes for å installere andre verktøy som man trenger for å ha utviklingsmiljø oppe og kjører. Den kan installeres på <http://brew.sh/>. [14].
- **Node.js and npm:** Node.js er runtime JavaScript miljø på serversiden. I tillegg trenger vi en Node’s package manager “npm” for å installere React Native command line verktøy med andre biblioteker[14].

2.5.3 JSX (JavaScript XML)

Da facebook opprettet react bibliotek, har de opprettet en syntaks som ligner på HTML som kan brukes for å beskrive komponenter. Dette ble kalt JavaScript XML.

JSX er deklarativt markeringsspråk(Declarative Markup language) som er skrevet sammen med JavaScript for å definerer komponent oppsettet[14].

2.5.4 Watchman

Watchman er en open source verktøy som facebook har opprettet. React Native bruker Watchman for å se om det er noen endringer som skjer på kildekode på tvers av filer. Hvis den oppdager noen endringer så den ombygger JavaScript bundle automatisk[14].

2.5.5 Biblioteker (Libraries)

2.5.5.1 React Native Element

React Native Element er et styling bibliotek som har forhånds bygde UI komponenter for å erstatte de grunnleggende og begrensede react native elementer. Det ligner Bootstrap og gir brukbare stiler som er bedre til å tilpasse din applikasjon. Hver av komponentene i React Native Elements er pakket med den grunnleggende React Native <View /> -komponenten. for å installere react-native elements for react native CLI prosjekt eller react native expo basert prosjekt, kan man bruke følgende hvis man bruker npm[15]:

```
npm i react-native-elements --save
```

Men hvis man bruker yarn:
yarn add react-native-elements

2.5.5.2 Kontakt.io

kontakt.io er en cross-plattform React Native modul for å oppdage beacons med Android og iOS enheter. Fordeler med den er at den fungerer for hvilken som helst beacon[19].

2.5.5.3 Redux

et open-source JavaScript-bibliotek for å administrere applikasjons tilstand. Det brukes ofte med biblioteker som React eller Angular for å bygge brukergrensesnitt[16].

2.5.5.4 Axios

Axios er en Promise-based HTTP-klient for JavaScript som kan brukes i front-end applikasjonen og i Node.js-backend. Ved å bruke Axios er det enkelt å sende HTTP-forespørsel til REST-sluttpunkter og utføre operasjoner som for eksempel å hente data fra database[17].

2.5.5.5 React Navigation v5

React navigation versjon 5 er den siste oppdaterte bibliotek for å utvikle navigering struktur for react native applikasjoner.

2.5.6 Firebase Autorisering(Firebase Authentication)

De fleste apper trenger å identifisere brukeren sin identitet og når appen kjenner brukerens identitet, så vil appen lagre brukeren sine informasjon på en sikker måte i Cloud og gir den samme personlige opplevelser på tvers av alle brukeren enheter. Firebase Authentication tilbyr backend-tjenester, brukervennlige SDK-er og ferdige UI-biblioteker for å autentisere brukere til appen din. Den støtter autentisering ved hjelp av passord, telefonnumre, populære identitetsleverandører som Google, Facebook og Twitter og mer[18].

2.5.7 Wireframe

Wireframe er en prototype teknikk som rett og slett brukes for å skisse grensesnittet til applikasjon eller nettside som man utvikler.

Kapittel 3 Valg av teknologi og metodikken

3.1 Innledning

Dette kapittelet skal omhandle valg av metodikken og valg av teknologi, programmeringsspråk og verktøy som ble benyttet i dette prosjektet.

3.2 Utviklingsmetodikk

Under utvikling av dette prosjektet ble det brukt en tilnærmet prinsipp til agile utviklingsmetode nemlig Scrum. Dette innebærer at jeg ikke har brukt Scrum metodikken på fullt som løsning til mitt prosjekt. Det var ikke nødvendig å bruke alle prinsippene i Scrum og grunnen til dette at jeg var det eneste medlem i dette prosjektet. Det viktigste prinsippet fra Scrum som ble brukt i dette prosjektet er den iterative prosessen. Figuren nedenfor viser den iterative prosessen som ble benyttet i dette prosjektet.

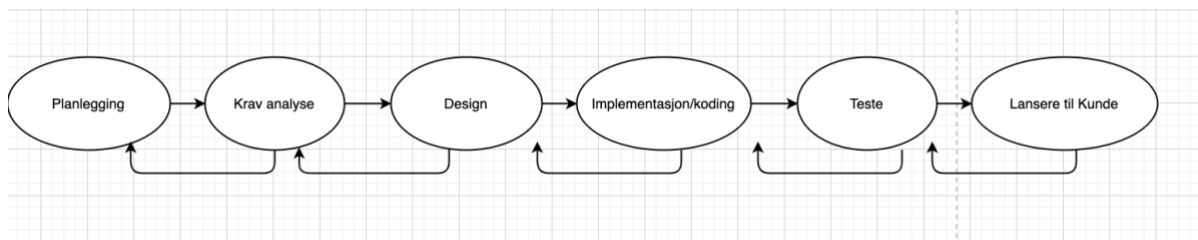


Figure 6: iterative prosessen

Første sprint startet med oppstartsmøte hvor veilederen på en måte var Scrum master, oppgavestiller som på en måte var produkteier og studenten var tilstede. I løpet av dette møtet fikk oppgavestiller formidle sin ide om prosjektet samtidig fikk oppgavestiller formidlet at i deres firma bruker ikke og har ikke noe erfaring med å bruke den teknologien(React Native kap 2.5.2) som de ba studenten å bruke i dette prosjektet. Dermed må studenten være helt selvstendig og løse hele prosjektet på egen hånd. Firmet har intensjon å innføre bruken av React Native i framtiden hvis prosjektet lykkes. Tilslutt ble alle medlemmene enige om å skrive først visjonsdokumentet og sende den til både oppgavestiller og veilederen.

I første iterasjon ble det et møte med produkteier hvor vi ble enig om å lage kravdokument som skal beskrive alle kravene i prosjektet for å være sikker på at studenten har god forståelse av oppgaven. På hver iterasjon var det et møte med oppdragsgiver(produkteier) for å vise frem det som ble gjort og gjør en vurdering om det er noe som må endres eller forbedres. Selv om studenten valgte to ukers sprint, men oppdragsgiver var ikke streng på at det måtte være to uker fordi noen sprinter tok lengre tid enn planlagt. Dessuten var det et standup møte hverdag kl 9:30 via skype for alle utviklere i firmet hvor de forteller om hva de har gjort forrige dag og hva de skal gjør idag og om det er noen problem. Jeg har vært med i standup møter, og selv om det er påminnelse på skype kalender at det er et møte kl 9:30 hverdag men de i firmaet var ikke strenge på å holde det møtet hver dag.

3.3 Native vs Hybrid mobilapplikasjon

Når det kommer til bygging av mobilapplikasjoner er det to type tilnærminger, den ene er native apper og den andre er hybrid apper. Native apper er raskere og mer responsive enn hybrid apper. Når man skal utvikle en native app, så må man bygge to forskjellige koder for både iOS og Android. For

eksempel for å utvikle en native applikasjon for iOS plattform så må man kode det enten ved å bruke objektiv-C språket eller swift språket. For å utvikle Android applikasjoner, må man bruke Java. Dessuten er utviklingsmiljøer for iOS og Android forskjellig. Xcode brukes for utvikling av iOS apper mens eclipse og Android studio brukes for å utvikle Android apper. Hybrid apper utvikles ved å bruke web teknologier(HTML, CSS og JavaScript). Hybrid apper er lettere fordi man trenger bare å lage en kodebase som kan brukes til forskjellige plattformer[14]. Hybrid apper kombinerer elementer fra både native og webapplikasjoner. Hybrid apper er lettere og raskere å utvikle en native apper. Fordelen med hybrid apper at man kan legge til nytt funksjonalitet til flere versjoner, mens med native apper, må man gjenskape hver nye funksjon man vil introdusere for hver plattform.[20]

I første møte med oppdragsgiver var det snakk om å lage en hybrid app med bruk av native funksjonalitet, men de var usikker om de viller lage en web applikasjon eller mobilapplikasjon. Til slutt ble det bestemt å bruke React Native teknologi, og dette innebærer at i dette prosjektet skal jeg lage en hybrid app med native funksjonalitet.

3.4 Innlogging

Dette er en funksjon som var ikke påkrevd av oppdragsgiver og dermed var veldig lavt prioritert fordi de har allerede en ferdig utviklet applikasjon fra før og det eneste som denne applikasjonen manglet er en funksjon som skal oppdage beacons som skal monteres i treningscenteret. Det eneste oppdragsgiver har krevd var en prototype app som fanger opp alle beacons signaler som finnes i treningscenteret, samtidig ønsket de å lage denne prototypen med en ny teknologi som de ikke har benyttet fra før. Derfor valgte jeg å lage autentisering ved å bruke firebase authentication tjeneste som Google tilbyr og som ble omtalt i kapittel(2.5.6). I denne prototype appen ble det også laget registreringsside som registrere nye brukere.

Men hvordan autentisering i mobilapplikasjoner fungerer? jo vi har en server og i vårt tilfelle firebase som vår react native app skal kommunisere med, når en bruker logger inn med sin registrerte email og passord så sendes denne informasjonen til serveren og sistnevnte vil svare enten med suksess response eller feilmelding hvis brukerens informasjon er feil. I vårt tilfelle kommuniserer vår mobil app med en statsløs server via RESTful APIs og her er det ikke sesjon som i webapplikasjoner med det som kalles "Token". Ideen er slik når en bruker logger seg inn og serveren sjekker om email og passord er riktige, så serveren oppretter en token og den bruker en privat nøkkel som bare serveren kjenner og denne token deretter returneres til appen hvor denne token kan lagres i en lagerplass som for eksempel "Redux Storage". Denne token slettes når brukeren logger seg ut. figuren nedenfor viser hvordan authentication fungerer.

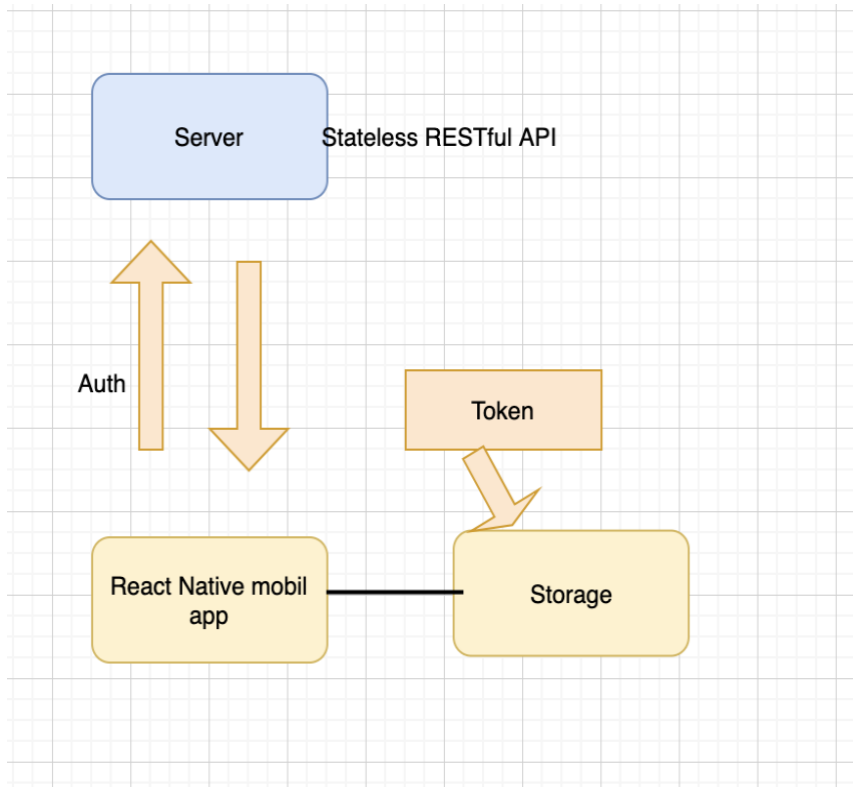


Figure 7: viser virkemåten til autentisering

3.5 Design

I designprosessen ble det brukt først Wireframe (delkapittel 2.5.7) som er et verktøy som jeg brukte for å lage en tilnærmet skisse til hvordan designet på prototype app skal se ut. I første fase av design ble det opprettet en react native expo basert prosjekt(expo CLI delkapittel 2.5.2.1). Første steget var å installere node.js som er en Javaskript runtime som man for eksempel lar bygge apper på serversiden med Javascript. Dette var en pakke som man måtte installere for å kunne kjøre javascript på min maskin og expo klient kjøre på Javascript og dermed har ikke denne pakken noe med react native å gjøre. Under designprosessen ble det brukt en del biblioteker som for eksempel Redux(delkapittel 2.5.5.3) for å håndtere applikasjons tilstand. Andre biblioteker som ble brukt er React Navigation version 5 for å kunne navigere mellom de forskjellige skjermer i appen. React Native element (delkapittel 2.5.5.1 React Native Element) ble også brukt under designprosess.

3.6 iBeacon(Bluetooth Low Energy BLE)

iBeacon som er omtalt i(delkapittel 2.2) er den vesentlig ting i dette prosjektet. Oppdragsgiver hadde gitt studenten en ibeacon for å teste den hovedfunksjonaliteten som de ønsket at studenten skal få til. Første steget var å kode en metode som oppdager beacons som tilhører en bestemt plass ved å spesifisere lokasjon og i vårt tilfelle var treningssenteret slik at appen viser bare de beacons som er i treningssenteret. Dermed var neste steget etter

applikasjonsdesign å kode en metode som oppdager beacons først. React Native har en modul som oppdager beacons med Android og iOS apper og dette var kontakt.io (delkapittel 2.5.5.2). Denne modul ble brukt i dette prosjektet. Nedenfor er et bildet av den beacon som ble brukt under testing.

3.7 Backend

Selv om dette var en frontend begrenset prosjekt men det var nødvendig å bruke i en liten grad database og kode en server. Grunnen til dette var at denne modul (kontakt.io) sin oppdaterte versjon oppdager hvilke som helst beacon og hvor som helst, og dette innebære at denne metoden kommer til å liste alle beacon som den oppdager uansett om den er i treningssenteret eller andre plasser. Derfor ble studenten nødt til å registrere de beacon som tilhører treningssenteret på en database slik at bare de registrerte beacons listes opp i skjermen. Under oppsetting av server, var jeg nødt til å installere en del dependencies som express, mysql og body-parser. Det ble også brukt axios (delkapittel 2.5.5.4) bibliotek som hjelper med å hente data fra database.

3.8 Verktøy

3.8.1 VS code

Dette er code editor som ble brukte for å utvikle dette prosjektet.

3.8.2 flaticon

Dette er nettside som ble brukt for å hente image ikoner fra.

3.8.3 Locate Beacon

Dette er en applikasjon som man installerer på mobilen for å finne informasjon om ibeacon som jeg har, og disse informasjon var for eksempel Major, Minor av ibeacon osv. Disse informasjon ble brukt for å prøve en metode for å oppdage beacons.

3.8.3 draw.io

Dette verktøyet ble brukt for å tegne illustrasjonsfigurer

Kapittel 4 Resultater

4.1 Innledning

I dette kapittelet skal jeg vise resultater i lys av problemstillingen. Resultatet av dette prosjektet er en prototype hybrid app som kan utvikles videre ved å bruke samme rammeverket eller ved å bruke et annet rammeverk eller annet programmeringsspråk.

4.2 Vitenskapelig resultat

4.2.1 Design

Design Utviklingsprosess har vært gjennom flere iterasjoner og i vedlegget visjonsdokument kan man se de første utkastene av designet.

Applikasjonen ble designet med tanke på brukere. Så å designe en applikasjon på en enkel måte og lett å lære og bruke var en viktig del av dette prosjektet. Design valget som ble gjort var basert på de fem prinsippene som ble nevnt i kapittel 2 (delkapittel 2.1.1 interaksjonsdesign). Første ting som ble gjort var å velge React-Native-Element (delkapittel 2.5.5.1) som komponentbiblioteket. Valget av denne bibliotek var ikke tilfeldig, men det var en grunn til det. Det gir ikke bare passende og attraktive komponenter, men den ga applikasjonen synlighet, konsistens, tilbakemelding(Feedback) og Affordance som er fire av de fem prinsipper fra interaksjonsdesign(delkapittel 2.1.1).

I utgangspunktet startet utviklingsprosessen ved å utvikle designet til frontend for eksempel å lage innlogging form og registreringsform. I tillegg laget jeg drawer navigation, det vil si sidebar-menyen og utviklet autentisering av brukere. Første utkast av designet ble gjort ved å bruke react native expo CLI basert prosjekt. Første utkast av design resultater vises i bildene nedenfor. Dette ble testet ved å kjøre Kommandoen **npm start** og deretter når appen kobles til metro bundler, trykker man **i** for å kjøre iOS simulator eller **e** for å kjøre det på emulator.

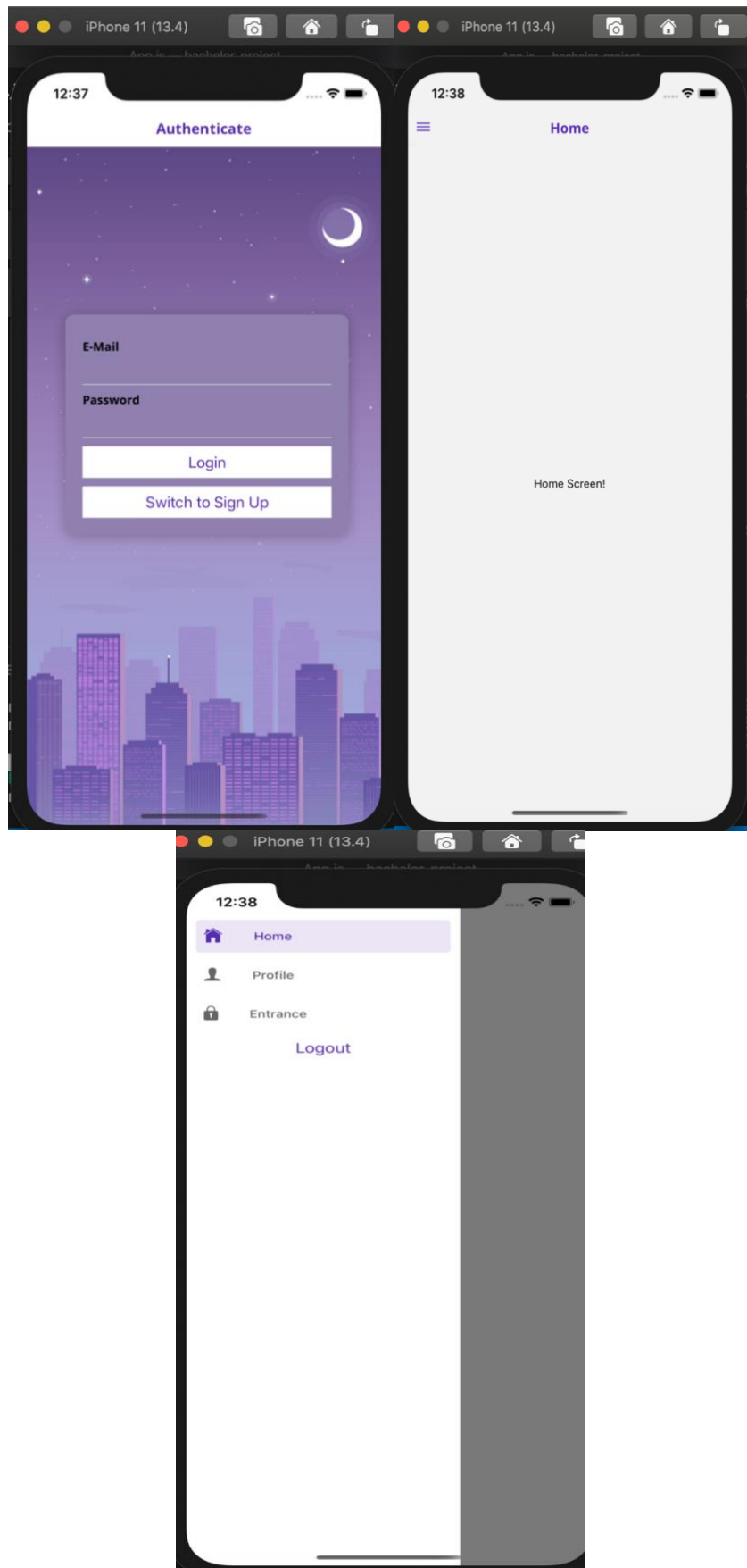


Figure 8: skjermbilder av en fungerende applikasjon på simulator

Første bilde til venstre er startup screen, denne er den første skjermen som skal vises når appen kjøres. Dette er “Authentication” skjerm og man kan switche mellom å logge seg inn hvis brukeren allerede er registrert eller registrere seg hvis brukeren ikke har registrert seg. Bildet som er på midten er den skjermen som man får når man logger seg inn. Siste bildet viser sidebar menyen som man får når man trykker på menu ikon og da får man en sidebar meny som har de forskjellige skjerner som man kan navigere mellom dem. Dessuten er det en “Logout” knapp som når man trykker kan man gå tilbake til “Authentication” skjermen.

I følge interaksjonsdesign sine prinsipper, det første prinsippet synlighet og som man ser i skjerm bildene ovenfor ser man tydelig at det er en innlogging form hvor man kan skrive både email og passord, i tillegg er komponenter veldig synlige at det der er en knapp for innlogging eller for å switche til registreringsform. Når det gjelder konsistens, så ser det første designet ovenfor mer konsistent og dette ser man veldig tydelig for eksempel knappene har samme farge og det når man trykker på de forskjellige elementene som ligger på sidebar menyen så få ikonet og skjermnavnet samme farge og dette gjør designet konsistent

Ved overgang til react native CLI prosjekt ble ikke designet endre veldig mye. Det som ble forskjell er ikonene som ble brukt i forrige design var importert fra expo, men i den siste versjonen ble det brukt bilde ikoner som er installert fra flaticon[21] nettside. Designet nedenfor ble det samme som det designet som er over, men litt mindre konsistens i forhold til elementer på sidebar menyen. Dette skjedde på grunnen av at ikonene får ikke samme farge som skjermnavnet når man trykker på den, og dette ser man tydelig i det tredje bilde nedenfor. Nedenfor er bilder for det siste designet.

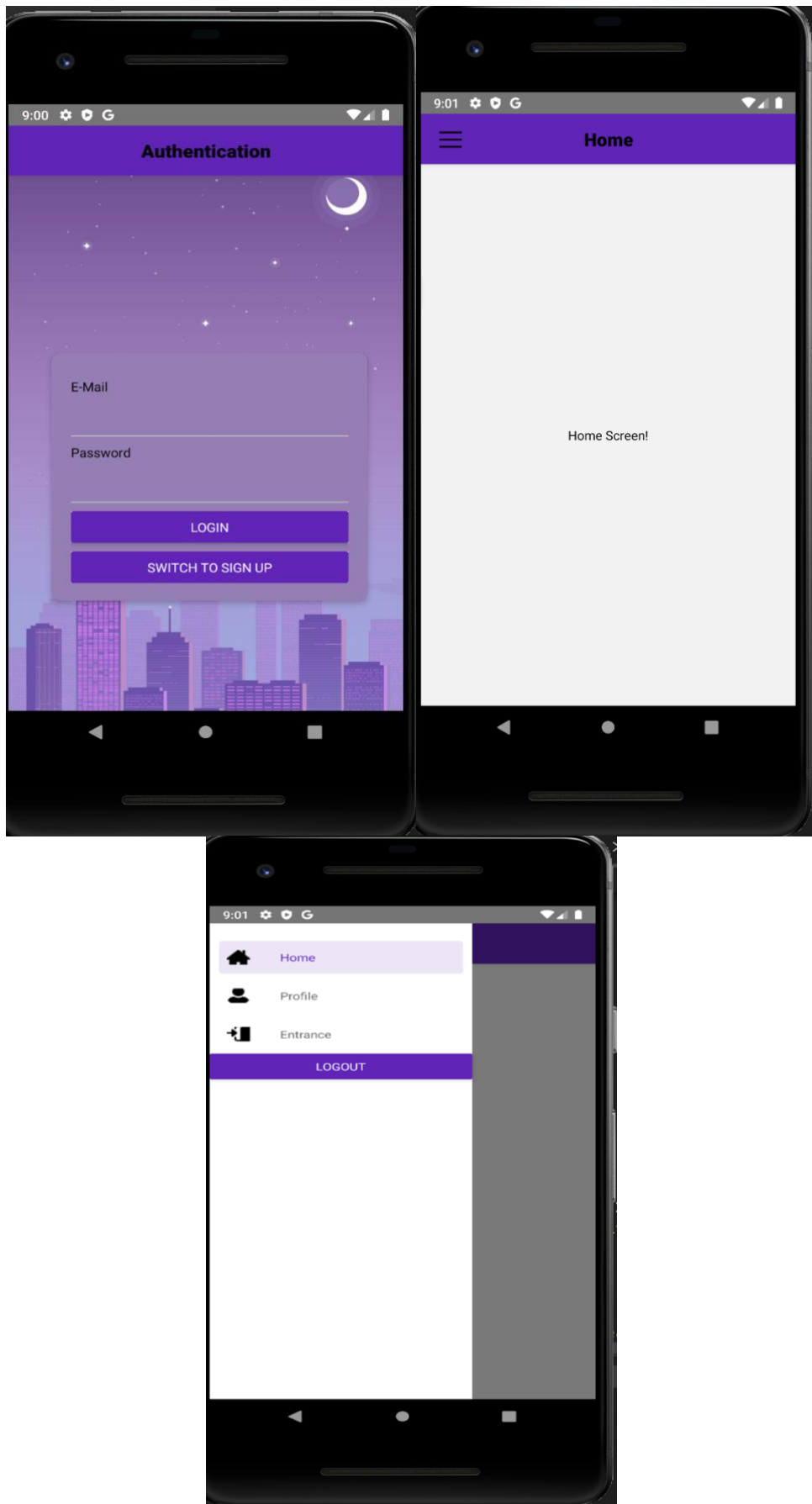


Figure 9: skjermbilder av det siste designe som kjører på emulator

Når det gjelder tilbakemelding(feedback), så applikasjonen tilbyr tilbakemelding ved innlogging eller registrering hvis bruken skriver ikke gyldige informasjon. I skjermbilde nedenfor vises det veldig tydelig at brukeren prøver å hoppe over email feltet og dermed får brukeren en tilbakemelding på rødt farge. I forhold til Affordance som er det siste prinsippet i interaksjonsdesign sine prinsipper, som ser tydelig på det høyre bildet, når man trykker på input for eksempel, får man opp tastaturet.

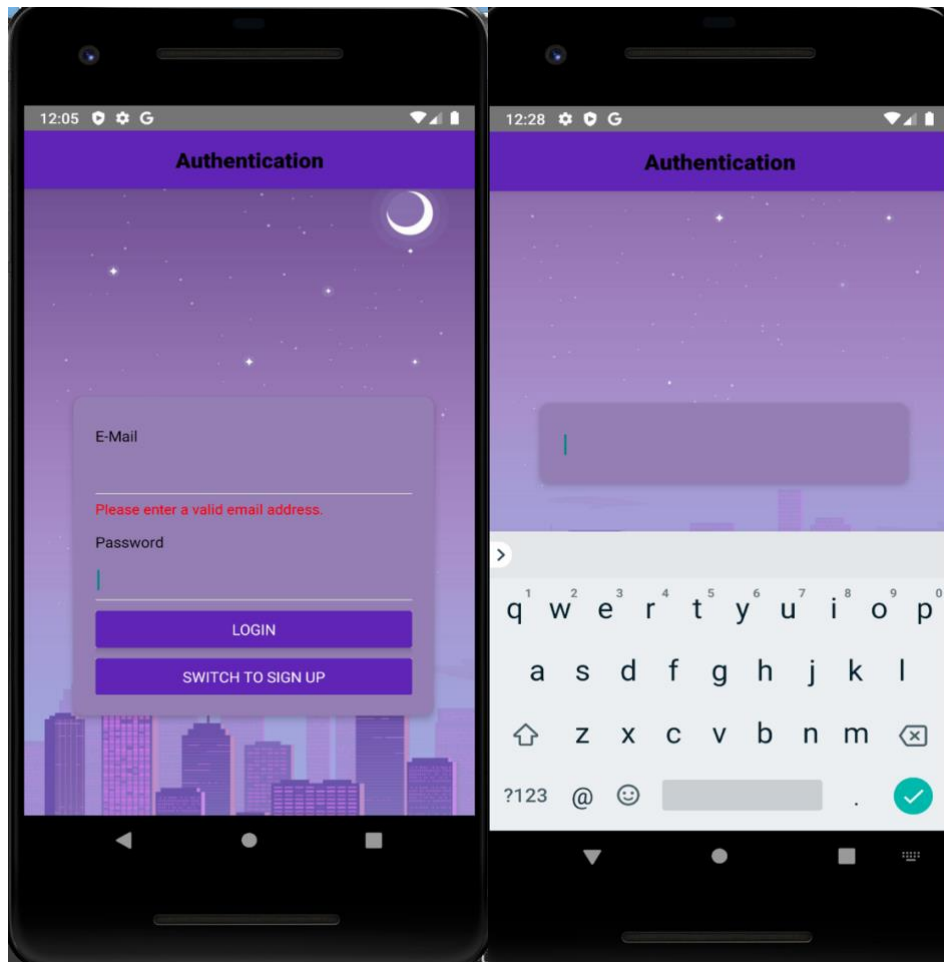


Figure 10: viser eksempel på tilbakemelding som en interaksjonsdesign prinsippet

4.2.4 Produktet

Produktet som ble levert til Sodvin Systemer er en fungerende prototype hybrid app som består av en kodebase som fungerer både for iOS og Android mobiler. Systemet består av innlogging funksjonalitet, navigering mellom de forskjellige skjermer og den hovedfunksjonalitet som var påkrevd, nemlig dørproplåsing.

Hele dette prosjektet er på et eller annet vis et utforskningsprosjekt og dette innebærer at jeg måtte utforske ulike måter for å finne den beste funksjonalitet for å låse opp døra i treningssenteret.

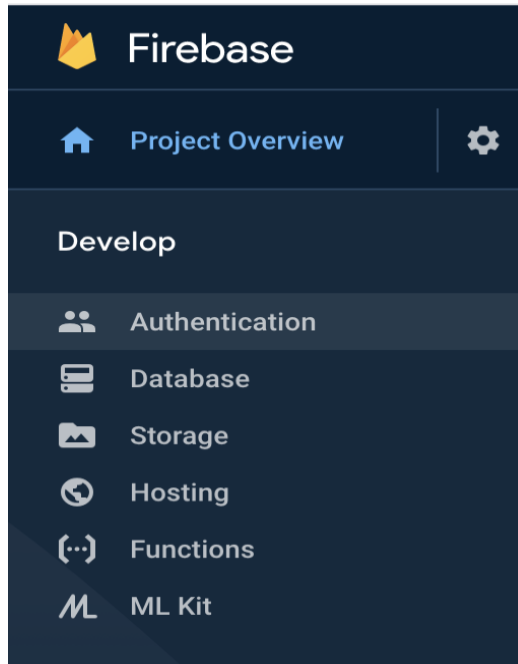
Problemstillingen ble løst først ved å søke etter en metode som kan oppdage hvilke som helst beacons. React Native rammeverk tilbyr en modul som kan oppdage beacons signaler. Selv om denne modulen oppdager hvilke som helst beacon så måtte man finne ut hvordan la appen liste opp bare de beacons som tilhører treningssenteret og da slipper brukere få en lang liste med beacons. Selv om dette prosjektet var frontend begrenset, så måtte jeg lage en enkel database og server hvor jeg registrert beacons id og dørnnavnet slik at når en beacons oppdages, så sjekkes beacons id om den er registrert i database og hvis det er det så hentes dørnnavnet i app skjermen med en knapp foran. Siste steget var å lage en metode for knappen slik at når man trykker på knappen sendes det en http-forespørsel til en server som åpner døra dersom brukeren er identifisert. Denne serveren er RESTful API server som oppdragsgiver har stilt med.

Produktet som helhet er lett å bruke, lett å lære og huske fra gang til gang og dette tilfredsstillende brukskvalitet som er beskrevet i (kapittel 2.1.2). Prosjektet som helhet dermed tilfredsstillende både interaksjonsdesign og brukskvalitet.

4.3 Ingeniørfaglig Resultat

Visjonsdokumentet beskriver ideer og alle funksjonaliteter som systemet skal inneholde. Med dette kan jeg trekke de ulike funksjonaliteter og beskrive om de ble fullført.

- Innlogging og registrering
Disse to sidene er “Authentication” og ble fullført ved å bruke min egen google konto ved å bruke firebase authentication. Dette innebærer at brukerinformasjon ble lagret i authentication side og ikke en storage eller database.



Sign-in providers

Provider	Status
<input checked="" type="checkbox"/> Email/Password	Enabled
<input type="checkbox"/> Phone	Disabled
<input type="checkbox"/> Google	Disabled
<input type="checkbox"/> Play Games	Disabled
<input type="checkbox"/> Game Center Beta	Disabled
<input type="checkbox"/> Facebook	Disabled

Figure 11: viser tjenester firebase tilbyr og Sign-in providers

Identifiser	Providers	Created	Signed In	User UID ↑
test@test.com	✉	Feb 27, 2020	Apr 30, 2020	8f2UujbNVdZw3p5EbdXOw5TkXs1
maher.bbb@hotmail.com	✉	Feb 28, 2020	Feb 29, 2020	coPtHEMqxobpbViQsUdHhW7ph6...
shahedsa@stud.ntnu.no	✉	Feb 27, 2020	Feb 29, 2020	gz68Zxhn5OcKaMRNCBxpa6BEjo82

Rows per page: 50 ▾ 1-3 of 3 < >

Figure 12: viser liste over registrerte brukere

Som man ser i figur ? er det første bildet tjenester som firebase tilbyr og det andre bildet viser måter brukere kan bli autorisert på og i vårt tilfelle brukte jeg Email/password og som man ser på bildet det står enabled foran dem. Figur ?? viser oversikt over alle registrerte brukere, når de har registrert seg og når de logget seg inn siste gang og bruker ID.

- Navigering mellom ulike skjermer
 Navigering mellom ulike skjermer ble fullført ved å bruke den siste oppdaterte versjon av React Navigation, nemlig React Navigation v5 (delkapittel 4.5.5.5)
- profil implementering
 Denne funksjonaliteten var lavt prioritert og ble satt til slutt å implementere. Jeg har prøvd på denne men jeg hadde ikke mer tid å jobbe med den.
- Funksjon som kan oppdage alle beacons
 Denne funksjonen ble fullført, men det tok veldig lang tid før jeg fant den beste løsningen. Jeg måtte søke veldig mye og prøve å forstå hvordan beacon fungerer. Det var flere biblioteker som tilbød beacon oppdagelse, men de fleste var utdatert. Nedenfor er kodesnutt som oppdager alle beacons og tar den unike id-en til hver beacon og setter det i en liste:

```

79 // Add beacon listener
80 if (isAndroid) {
81   DeviceEventEmitter.addListener(
82     'beaconsDidUpdate',
83     ({beacons, region}) => {
84       console.log('beaconsDidUpdate', beacons, region);
85
86       let list = [];
87       for (let i = 0; i < beacons.length; i++) {
88         //if (!beacons.includs(beacons[i].uniqueId))
89         list.push(beacons[i].uniqueId);
90
91         console.log(beacons.length);
92       }
93
94       this.setState(
95         {Asset: beacons, beaconsList: [...this.state.beaconsList, ...list]},
96         () => {
97           console.log('beaconsList:', this.state.beaconsList);
98         },
99       );
100     },
101   );
102 } else {
103   kontaktEmitter.addListener('didDiscoverDevices', ({beacons}) => {
104     console.log('didDiscoverDevices', beacons);
105   });
106 }
107 };

```

kodesnutt 1: En måte på å oppdage beacons og sette dem i en liste

samme teknikk fra linje 86 til og med 101 kan implementeres etter linje 104 for å teste det i iOS enhet, men siden jeg hadde android mobil, så jeg testet bare i min android mobil. Nedenfor er bilder på resultatet fra terminal og hva som er pusket til mobilskjermen:

```

LOG beaconsDidUpdate [{"accuracy": 0.058799371490447845, "address": "D6:F1:53:6A:A8:6B", "batteryPower": 100, "firmwareVersion": "4.2", "isShuffled": false, "major": 60768, "minor": 59106, "name": "SodvinSystemer1", "proximity": "IMMEDIATE", "rssi": -58, "txPower": -77, "uniqueId": "Au9Z", "uuid": "f7826da6-4fa2-4e98-8024-bc5b71e0893e"}] [{"identifiser": "Everywhere", "major": -1, "minor": -1, "secureUuid": "null", "uuid": "f7826da6-4fa2-4e98-8024-bc5b71e0893e"}]
LOG 1
LOG beaconsList: ["Au9Z"]

```

Figure 13: viser resultat fra konsoll på oppdagede beacons

Siden jeg testet bare på en beacon, så beaconsList viser bare en beacons id.



Figure 14: viser en del av beacons informasjon på skjermen til applikasjon

- Funksjon som sjekker om de oppdagede beacons er registrert i database
Denne funksjonen ble fullført ved å opprette en liten tabell i databasen som jeg fikk fra ntnu i forbindelse med andre prosjekter og koble den til node js server. Her måtte man lage et endepunkt i serveren som henter alle beacons med id. Men i klientsiden ble det brukt axios bibliotek(delkapittel 2.5.5.4) for å bruke endepunktet for å hente registrerte beacons
- Funksjon som henter beacons navn fra database ved å bruke oppdaget id om den er registrert i database
Denne funksjonen ble fullført. Denne ble gjort ved at
- Funksjon for å åpne en spesifikk dør: Denne funksjonen ble fullført ved at man sender http-forespørsel til Sodvin Systemer sin server og når man får status 200, så dette betyr at døra ble åpnet.

```

LOG beacon id Au9Z
LOG res from http://10.0.2.2:3000/Beacons/Au9Z [{"doorName": "MainOuterDoor",
id": "Au9Z"}]
LOG opening door: MainOuterDoor
LOG sending HTTP GET request to https://xaktapi.sodvinsystemer.no/v1/AccessCon
rolArx/OpenDoor/MainOuterDoor
LOG open door api respose 200

```

Figure 15: viser resultat når det sende en http-request til serveren for å åpne døra

4.4 Administrative Resultater

I prosjekthåndbok i vedlegg A ligger det både framdriftsplan og timeliste som viser tidsbruk i dette prosjektet.

Kapittel 5 Drøfting av Resultater

5.1 Vitenskapelig resultater – diskusjon

5.1.1 Design

Designet som ble utviklet ved å bruke React Native CLI hadde ikke så store avvik fra den som først ble utviklet ved å bruke expo. Det var et liten avvik i forhold til å bruke ikoner som menu button og de ikonene som er brukt i sidebar menyen. De ikonene som var importert fra expo vector-icon var det lettere å gi dem den ønskede farge. Men i den siste versjonen som ble utviklet av react-native cli var det vanskelig å gi de ikonene som ble brukt den ønskede fargen og grunnen til dette er at det ble brukt “image-icon” fra flaticon nettside. Jeg prøvde å bruke ikoner fra react-native-element bibliotek men den fungerte ikke, ikonene vises som spørsmålsteget eller en firkant, selv om jeg har installert alle nødvendige avhengigheter. Til slutt var jeg bare nødt til å bruke vanlige bilder som sagt fra flaticon nettside. Disse ikonene gjorde at sidebar menyen ble mindre konsistent, men samtidig er det ikke så stor avvik i designet. Hvis det har vært litt mer tid for å utforske mer, så kunne jeg finne ut hva som er problemet med React-Native-Element bibliotek, fordi det kan hende at det var mangel på en avhengighet som gjorde at ikonene fra React-Native-element viste seg som spørsmålsteget

5.1.2 Produktet

Etter at designet ble fullført, så begynte jeg å programmere autentisering, innlogging og registrering. Appen som firmaet vil utvikle krever ikke registreringsside og grunnen til dette at når en melder seg i treningssenteret så får medlemmene innloggingsinformasjon fra treningssenteret sin administrasjon. Men i denne prototypen var jeg nødt til å lage registrering side for at man skal kunne logge seg inn.

Når jeg begynte å lage funksjonalitet for å oppdaget beacons, så fikk jeg vite at man kan ikke lagre denne funksjonaliteten på et expo basert prosjekt fordi expo støtter ikke Bluetooth, og det samme gjelder for Simulator og Emulator. Sistnevnte har ikke noen innebygd Bluetooth funksjonalitet og dermed måtte man teste denne funksjonaliteten i en virkelig enhet. I dette tilfellet var jeg nødt til å overføre prosjektet fra expo CLI til vanlig React Native CLI. Videre ble alle beacon funksjonaliteter testet på en fysisk enhet, nemlig min android mobil. Videre ble det utviklet en funksjon som sjekker om den oppdagede beacons id er registrert i database, men opplevde et problem, nemlig at jeg trengte en server som hostes på mobil. Problemet var at jeg brukt database kontoinformasjon som jeg fikk fra ntnu og dette er private, det vil si at serveren ligger lokalt på min Mac OS og dermed ikke i mobilen. Da jeg diskuterte det med oppdragsgiver, så fikk jeg beskjed at på grunn av coronavirus situasjon hadde de fleste utviklere blitt permittert og dermed hadde ikke oppdragsgiver anledning til å lage en server. Jeg prøvde å søke veldig mye i google, men fant ut at det å verte en server må man betale for og jeg fant ikke gratis alternativer. Jeg fikk forslag fra veilederen at jeg kan ta kontakt med ntnu laben. Fra laben fikk jeg en virtuell ubuntu server og de tilbyr ikke for mac. Jeg har dessverre veldig lite erfaring med virtuell server. Jeg prøvde å google og fant ut at jeg

måtte installere ubuntu virtuell maskin, men hadde lite minne i min mac. Til slutt laget jeg en funksjon som oppdager beacon virtuelt. Denne løsningen ble godtatt av oppdragsgiver. Etter at beacon oppdages virtuelt, så kalles en metode som sjekker om denne beacon id tilhører til et dørnavn, så hentes dørnnavnet på skjermen med en knapp. Siste fase var å programmere en metode som åpner døra og det er bare å sende HTTP-request til serveren som jeg fikk fra oppdragsgiveren. Jeg måtte vente ca to uker før jeg fikk serveren og dette gjorde at man mistet litt tid. Et avvik som også har skjedd er at det var vanskelig å få autentiseringsinformasjon på profilsiden til brukeren fordi disse ikke er lagret i en database, og jeg hadde ikke rukket å prøve på det mer.

5.2 Ingeniørfaglig resultater – diskusjon

- Innlogging og registrering: Siden dette var en prototype, så oppgavestiller hadde ikke noen spesifikke krav til denne funksjonaliteten, og derfor ble den løst på en måte som tok mindre tid, som sagt ved å bruke en google firebase autentisering. På den andre siden hvis jeg skulle lage en applikasjon som skal være komplett, så kunne man bruke vanlig database for å lagre brukere og hashe passordet. Ellers firebase autentisering måten var akseptert av oppdragsgiver.
- profil implementering: Som nevnt i resultatkapittelet at denne funksjonen ble ikke implementert og grunnen til dette at studenten var ikke så erfarne å hente brukeren sine informasjon fra firebase authentication til profil skjermen ellers ville det bli lettere å kunne hente bruker sine informasjon til profil skjermen hvis disse informasjon ble lagret i en database.
- Funksjon som kan oppdage alle beacons: Denne metoden ble fullført på den beste måten. Man kunne bruke react-native-beacon-manager bibliotek og da kan man spesifiserer det området som man vil oppdage alle beacons i, men denne biblioteket er utdatert og derfor ble løsning å bruke Kontakt.io modul nyeste versjon.
- Registrere alle beacons i en database: oppdragsgiver hadde et krav at alle beacons skulle registreres i en database, men på grunn av den ekstraordinær situasjon(Corona-situasjon) og at man må jobbe hjemmefra, så jeg hadde bare en beacon å teste på. Ellers man måtte test på flere beacons for å sikre bedre kvalitet på koden man har laget.

5.3 Evaluering av Systemet som helhet

Generelt sett er de fleste funksjonalitet som er påkrevd ble implementert. Systemet tilfredsstillende både interaksjonsdesign og brukskvalitet bortsett fra det avviket som oppstått i designprosessen

Kapittel 6 Konklusjon og videre arbeid

6.1 Konklusjon og videre arbeid

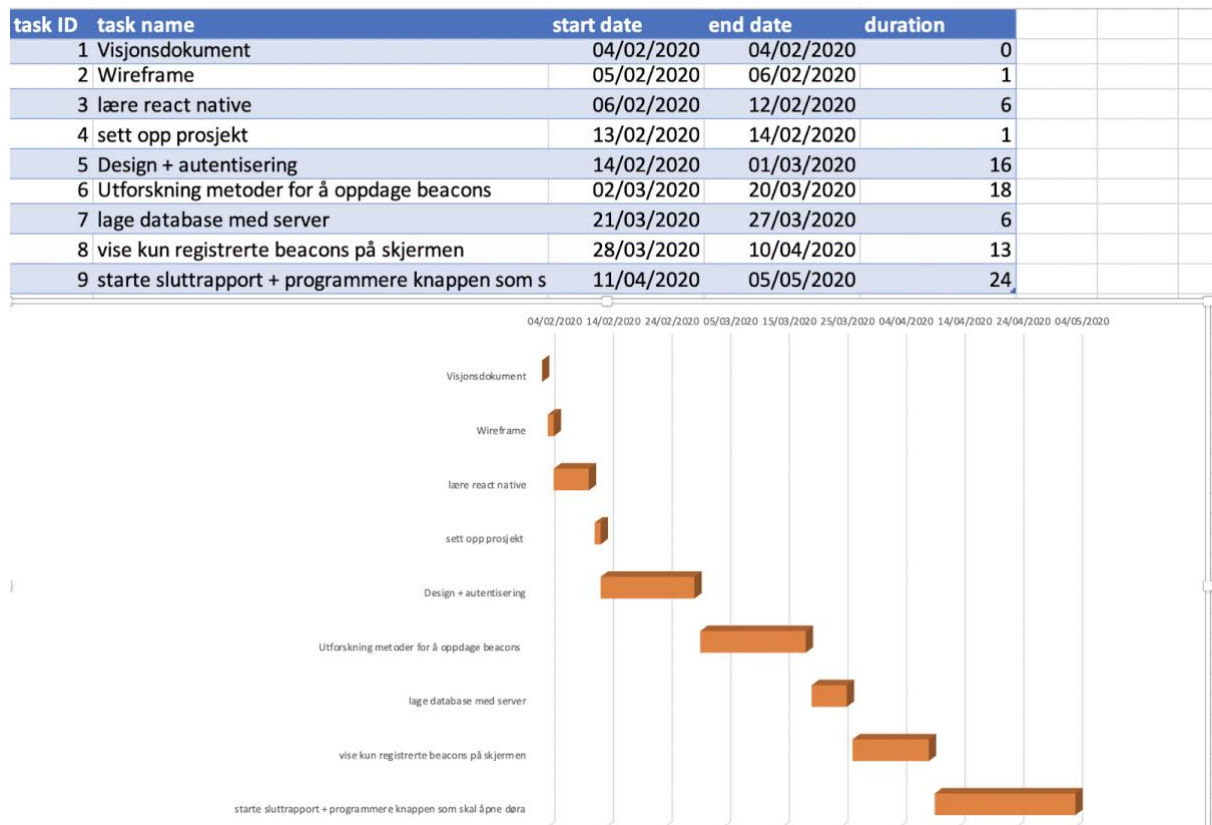
Ut fra resultater ser det ut at det er mulig å implementere funksjonaliteter som både ivareta både interaksjonsdesign og brukskvalitet på en hybrid app. Dessuten det er mulig å implementere ibeacon på en måte slik at den åpner treningssenteret sine dører. Denne applikasjonen kommer til å digitaliserer treningssenteret sin plastkort eller brikke og dermed slipper medlemmer å tenk på om de har tatt med kortet når de drar på treningssenteret. Som nevnt i teorikapitlet at ibeacon sin funksjon kan åpne flere muligheter og gjør samfunnet mer digitalisert.

Siden det som ble utviklet er kun en prototype, så kan Sodvin Systemer implementere ibeacon funksjonalitet i deres eksisterende applikasjon for å digitalisere treningssenteret sin kort/brikke for å åpne døra.

Vedlegg A Prosjekthåndbok

Dette dokumentet inneholder både framdriftsplan som ble laget i begynnelsen av semesteret, innkallings møter, referater og timeliste.

A.1 Framdriftsplan



Figur A 1: viser framdriftsplan

A.2 Møteinnkallinger

Innkalling til oppstartsmøte den

16.01.2020 Trondheim, 13.01.2020

Møteinnkallingen går til:

Veileder: Nils Tesdal

Oppgavestiller: Tore Kjørsvik

Tid : Torsdag 16.01.2020 klokken 09:00.

Sted: Kantega, Bassengbakken 4, 7042 Trondheim, Skype

Agenda:

Saknr.	Saker	Tid	Ansvarlig
01/2020	Godkjenning av møteinnkalling	1 min	Nils Tesdal
02/2020	Underskrive avtale.	5 min	Nils Tesdal, Tore Kjørsvik, Shahed Abbas
03/2020	Drøfting av prosjektet	15 min	Nils Tesdal, Tore Kjørsvik, Shahed Abbas
04/2020	Forklare eksterne oppgavestiller NTNU sine krav og forventninger til studentenes arbeid og innlevert dokumentasjon.	10 min	Nils Tesdal
05/2020	Sett opp krav til dokumentasjon	3 min	Nils Tesdal
06/2020	Andre ting		

Møteinnkalling går til : Nils Tesdal og Tore Kjørsvik.

Innkalling til møte den 03.03.2020

Trondheim,02.03.2020

Møteinnkallingen går til:
Veileder: Nils Tesdal

Tid : Torsdag 03.03.2020 klokken 13:00.
Sted: Kantega, Bassengbakken 4, 7042 Trondheim

Agenda:

Saknr.	Saker	Tid	Ansvarlig
06/2020	Signering av avtalet	1 min	Nils Tesdal
07/2020	Arbeidet utført hittil	10 min	Shahed Abbas
08/2020	Spørsmål og tips	15 min	Nils Tesdal, Shahed Abbas

Møteinnkalling går til : Nils Tesdal

Innkalling til møte den 23.03.2020

Trondheim, 18.03.2020

Møteinnkallingen går til:
Veileder: Nils Tesdal

Tid : Torsdag 23.03.2020 klokken 11:00.
Sted: Skype

Agenda:

Saknr.	Saker	Tid	Ansvarlig
09/2020	Arbeidet hittil	10 min	Shahed Abbas
10/2020	Videre arbeid	10 min	Shahed Abbas

Møteinnkalling går til : Nils Tesdal

Innkalling til møte den 15.04.2020

Trondheim, 14.04.2020

Møteinnkallingen går til:
Veileder: Nils Tesdal

Tid : Torsdag 15.04.2020 klokken 11:00.
Sted: Skype

Agenda:

Saknr.	Saker	Tid	Ansvarlig
11/2020	Server problem	10 min	Shahed Abbad

Møteinnkalling går til : Nils Tesdal

Innkalling til møte den 30.04.2020

Trondheim,28.04.2020

Møteinnkallingen går til:
Veileder: Nils Tesdal

Tid : Torsdag 30.04.2020 klokken 12:00.
Sted: Skype

Agenda:

Saknr.	Saker	Tid	Ansvarlig
12/2020	Arbeidet hittil	10 min	Shahed Abbad

Møteinnkalling går til : Nils Tesdal

A.3 Referater

Referat 1:

Møtereferat fra møte den 17.01.20.
Veiledningsmøte ved Kantega og skype

Tilstede:

Veilederen: Nils Tesdal

Oppgavestiller: Tore

Bachelor student: Shahed Abbas

Saker gjennomgått under møte:

Sake 02/2020 signering av bachelor avtale:

Vi ble enig om at oppgavestiller signere avtalen og skanne det og sende på mail slik at studenten og veilederen signere den til neste møte.

Sak 03/2020 Drøfting av prosjektet:

Oppgavestiller forklarte at prosjekt oppgave går ut på å lage en mobilapplikasjon som kan fungere som nøkkel for å låse opp dør. Applikasjonen skal i vårt konkrete tilfelle kunne erstatte nøkkelbrikke/plastkort for medlemmer som ønsker å trene på et treningssenter.

prinsipper:

Vi ønsker å få laget en hybrid app som benytter Native funksjonalitet for å kommunisere med Bluetooth Beacons. En Bluetooth Beacon vil identifisere stedet/døren som skal låses opp. Brukeren er autentisert i appen, og vi vet dermed hvilket medlem som ønsker å låse opp døren. Appen vil kommunisere med tjenester på våre servere via webservice (REST api), og det er disse serverside tjenestene som faktisk validerer kombinasjonen medlem/dør og som evt. sørger for å låse opp døren. I utgangspunktet ser vi for oss at oppgaven begrenses til klientsiden (hybrid app på telefon), men det kan være aktuelt å utvide til å inkludere serverside oppgaver dersom det viser seg å være tid og rom for det.

Sak 04/2020 Forklare eksterne oppgavestiller NTNU sine krav og forventninger til studentenes arbeid og innlevert dokumentasjon.

Vi ble enige at jeg begynner å laget visjon dokumentet og sende den til oppgavestiller via mail for å se om jeg går på riktig vei. Det ble sagt at i firma kjører de sprint hver 14 dager og jeg skal jobbe etter dette prinsippet.

Referat 2:

Møtereferat fra møte den 03.03.20.
Veiledningsmøte ved Kantega

Tilstede:

Veilederen: Nils Tesdal

Bachelor student: Shahed Abbas

Saker gjennomgått under møte:

Sake 06/2020 signering av bachelor avtale:

Jeg fikk et signert avtale fra oppgavestiller og den ble printet ut og signert av meg og veilederen .

Sak 07/2020 Arbeidet utført hittil:

Hittil ble det utviklet designet til prototypen og autentisering av login via firebase authentication. Neste steget er å utforske ulike metoder for å oppdage beacons.

Sak 08/2020 Spørsmål og tips

Vi ble enige at jeg sender visjonsdokumentet til veilederen på mail. Vi avtalt at hvis jeg ikke får til metoden som oppdager beacon, så jeg måtte si fra frem til neste møte for å kunne få hjelp til dette.

Referat 3:

Møtereferat fra møte den 23.03.20.

Veiledningsmøte via Skype

Tilstede:

Veilederen: Nils Tesdal

Bachelor student: Shahed Abbas

Saker gjennomgått under møte:

Sake 09/2020 Arbeidet hittil:

Jeg fikk til funksjonalitet som oppdager beacon og vise informasjon om den i konsoll.

Sak 10/2020 Arbeidet videre:

Jeg må lage database for å registrere beacons og vise kun de registrerte beacons på skjermen. Jeg ble anbefalt av veileder å redigere litt på visjonsdokumentet, for eksempel legge til ikke funksjonelle egenskaper og legge til første utkast av wireframe.

Referat 4:

Møtereferat fra møte den 15.04.2020.

Veiledningsmøte via skype

Tilstede:

Veilederen: Nils Tesdal

Bachelor student: Shahed Abbas

Saker gjennomgått under møte:

Sake 11/2020 Server problem:

Vi diskuterte hvorfor de registrerte beacons ikke vises på skjermen til mobilen min, og problemet var at jeg trengte en offentlig server som kan kjør på mobil. Veilederen foreslå at jeg prøver første med google engine og hvis jeg ikke får det til, så kan jeg kontakte NTNU lab.

Referat 5:

Møtereferat fra møte den 30.04.2020.

Veiledningsmøte via skype

Tilstede:

Veilederen: Nils Tesdal

Bachelor student: Shahed Abbas

Saker gjennomgått under møte:

Sake 12/2020 Arbeidet hittil:

Jeg har løst server problemet på en annen måte, på min egen måte selv om jeg fikk en virtuell server fra NTNU lab men jeg brukte ikke fordi da jeg googlet hvordan man bruker den, så fant jeg ut at jeg måtte installere ubuntu vm på min egen mac og da hadde jeg ikke nok minne til det. Veilederen syntes at jeg måtte ikke det, men siden jeg løste problemet på en måte som ble godtatt av oppdragsgiver, så det var greit. Vi ble enig om at jeg skal fortsette sluttrapport og sende en disposisjon om den til veileder for å se om jeg er på riktig spor.

A.4 Timeliste

Utført arbeid i perioden

Navn på studenter:

Stud1: Shahed Saeed Abbas

Navn på bedrift/organisasjon: Sodvin Systemer As

Navn på veileder ved bedrift/organisasjon: Niles Tesdal

Uka	dato	antall timer	kommentar
-----	------	--------------	-----------

6	2020-02-04	8	Møte med oppdragsgiver + jobbet med visjonsdokumentet
	2020-02-05	8	diskuterer visjonsdokumentet med oppdragsgiver + redigerer visjonsdokumentet + begynne med wireframe
	2020-02-06	8	lage andre utkast av wireframe + begynne med et kurs for react native
	2020-02-07	8	react native kurs
	2020-02-08	5	react native kurs
	2020-02-09	5	react native kurs
SUM		42	
7	2020-02-10	4	react native kurs
	2020-02-11	4	react native kurs
	2020-02-12	4	react native kurs
	2020-02-13	5	react-native kurs, sette opp prosjektet ordnet timeliste
	2020-02-14	8	begynne å lage komponenter
	2020-02-15	4	jobbe med design
	2020-02-16	4	jobbe med design
SUM		33	
8	2020-02-17	10	lære react-native gjennom kurs + laget logg inn komponent(design)
	2020-02-18	8	begynte å lage drawer navigation
	2020-02-19	5	drawer navigation
	2020-02-20	5	fortsette med design
	2020-02-21	10	starter et nytt prosjekt med den nyeste react native versjon uten å opprette en expo prosjekt

	2020-02-22		
	2020-02-23	9	Jobbet med design, dele kode i små komponenter.
SUM		47	
9	2020-02-24	3	jobbe med å fikse en bug i design
	2020-02-25	4	fikse en del ting i designet
	2020-02-26	8	fortsette med design
	2020-02-27	11	jobbet med authentication for logg inn og registrering ved å bruke firebase
	2020-02-28	10	fortsette å jobbe med authentication og fikse noen bug i design
	2020-02-29	2	rydde opp kode
	2020-03-01	7	fullføre design
SUM		45	
10	2020-03-02	8	Møte med oppdragsgiver for å vurdere designet + starte å google for å finne ulike metoder for å oppdage ibeacons
	2020-03-03	8	Møte med veileder + lese mer grundig om ibeacon og hvordan den fungerer
	2020-03-04	3	utforske ulike metoder for å oppdage ibeacon
	2020-03-05	3	utforske ulike metoder for å oppdage ibeacon
	2020-03-06	2	prøve react-native-beacons-manager bibliotek
	2020-03-07	2	fortsette prøve react-native-beacons-manager bibliotek
	2020-03-08	3	prøve å endre litt på code
SUM		29	
11	2020-03-09	3	prøver å søke andre metoder for å løse problemet med beacon oppdagelse.

	2020-03-10	2	prøver på React native modul Kontakt.io
	2020-03-11	2	overføre react native prosjektet til en React Native CLI prosjekt istedenfor expo
	2020-03-12	7	prøve kontakt.io på det nye react native prosjektet
	2020-03-13	3	prøve den gamle versjon av kontakt.io
	2020-03-14	2	koble prosjektet mot virkelig enhet(mobil telefon)
	2020-03-15	2	prøve den gamle versjon av kontakt.io, endre litt på kode
Sum		21	
12	2020-03-16	3	prøver videre på kontakt.io
	2020-03-17		
	2020-03-18		
	2020-03-19	8	teste kontakt.io sin oppdatert versjon
	2020-03-20	9	møte med oppdragsgiver + endre på kontakt.io kode for å sette de oppdagede beacons i en liste
	2020-03-21	8	Jobbe videre for å sette opp en database
	2020-03-22	3	sette opp en database og server og prøver å koble database til visual studio code editor
Sum		31	
13	2020-03-23	10	møte med veileder + jobbe videre med backend
	2020-03-24	9	prøver å finner andre alternativer for å sette opp en database
	2020-03-25	10	telefonsamtale med oppdragsgiver + jobbe videre med server
	2020-03-26	7	sette opp en database ved å bruke firebase
	2020-03-27	7	jobbe videre for å prøve å koble database

	2020-03-28	8	sette opp en database ved å bruke ntnu sin database
	2020-03-29	8	utforske andre metoder for database og server
SUM		59	
14	2020-03-30	8	jobbe videre med database
	2020-03-31	10	jobbe videre for å prøve å vise bare de beacons som er registrert i database i entrance skjermen
	2020-04-01	8	bygge videre på å teste å vise de registrerte beacons på entrance skjerm
	2020-04-02	9	begynne å skrive litt i sluttrapport + fortsette å teste ulike metoder for å vise kun registrert beacon på skjermen
	2020-04-03	8	trenger å hoste server på mobilen, prøver å finne en måte to host the server
	2020-04-04	8	prøver å utforske andre metoder for mobil server
	2020-04-05	8	prøver å utforske andre metoder for mobil server
SUM		59	
15	2020-04-06	8	prøver å bruke google sine tjenester for lage en server som kjører på mobil.
	2020-04-07	9	prøver å bruke google sine tjenester for lage en server som kjører på mobil + skriver litt på sluttrapport
	2020-04-08	9	prøver å installere server på mobil
	2020-04-09	9	søk andre alternativer for mobil server
	2020-04-10	9	søk andre alternativer for mobil server
	2020-04-11	7	søk andre alternativer for mobil server
	2020-04-12	6	søk andre alternativer for mobil server
SUM		57	
16	2020-04-13	9	kode en virtuell måte å oppdage beacon

	2020-04-14	9	jobbe med koding for inngang skjermer
	2020-04-15	9	møte med veileder + prøve andre metoder som kan erstatte virkelig server i mobil
	2020-04-16	9	fortsett jobbe med koding og fikse en del feil
	2020-04-17	9	Skrive på sluttrapport
	2020-04-18	6	Skrive på sluttrapport + endre litt på design
	2020-04-19	6	Skrive på sluttrapport
SUM		57	
17	2020-04-20	9	Skrive på sluttrapport
	2020-04-21	9	Skrive på sluttrapport + fikk server fra oppdragsgiver for å lage http-request
	2020-04-22	9	Skrive på sluttrapport + rydde hele kode på et nytt prosjekt + prøver å lage http-request som åpner dør
	2020-04-23	9	Skrive på sluttrapport + fikse feil i http-request
	2020-04-24	7	Skrive på sluttrapport + fikse feil i http-request
	2020-04-25	6	Skrive på sluttrapport
	2020-04-26	6	Skrive på sluttrapport + fikse på designet
SUM		55	
18	2020-04-27	9	Skrive på sluttrapport
	2020-04-28	8	Skrive på sluttrapport
	2020-04-29	8	Skrive på sluttrapport
	2020-04-30	8	møte med veileder + Skrive på sluttrapport
	2020-05-01	8	Skrive på sluttrapport

	2020-05-02	6	Skrive på sluttrapport
	2020-05-03	6	Skrive på sluttrapport
SUM		53	
19	2020-05-04	6	Skrive på sluttrapport + redigerer visjonsdokumentet
	2020-05-05	5	Skrive på sluttrapport + lage systemdokumentasjon
	2020-05-06	5	Skrive på sluttrapport + lage systemdokumentasjon
	2020-05-07	5	Skrive på sluttrapport
	2020-05-08	5	Skrive på sluttrapport
	2020-05-09	5	Skrive på sluttrapport
	2020-05-10	5	Skrive på sluttrapport
SUM		36	
20	2020-05-11		
	2020-05-12		
	2020-05-13		
	2020-05-14		
	2020-05-15	10	Jobbe med sluttrapport
	2020-05-16	6	Jobbe med sluttrapport
	2020-05-17	10	lage ppt presentasjon + legge til sikkerhet kapittel i systemdokumentasjon + lage prosjekthåndbok + ferdigstille sluttrapport
SUM		26	
Total timer		650	

Vedlegg B Visjonsdokument

Visjonsdokument – Nøkkelløs adgangskontroll(mobil app)

Anvendt Informasjonsteknologi, IDI, NTNU

Revisjonshistorikk

Dato	Versjon	Beskrivelse	Forfatter
04.02.2020	1.0	Første utkast	Shahed Abbas
05.02.2020	1.1	endre litt på user story og prototype	Shahed Abbas
05.05.2020	1.2	lagt til ikke funksjonelle krav og første utkast av wireframe prototype Det ble lagt til ekstra funksjonelle egenskaper	Shahed Abbas

B.1 Innledning

Dette dokumentet beskriver overordnede krav til prosjektoppgaven i emnet TDAT 3001 Bacheloroppgave for dataingeniør. Oppgaven består i å utvikle en mobilapplikasjon. Prosjektet skal utføres av 3. års bachelorstudent i vårsemesteret 2020.

B.2 Sammendrag – problem og produkt

2.1 Problem Sammendrag

Utfordringen er å lage en mobil applikasjon som kan åpne døra til et treningssenter. Hensikten er å finne en digital løsning som kan erstatte den tradisjonelle løsningen nemlig nøkkelbrikke eller plastkortet. Dette vil bli en løsning for alle som trener og som glemmer eller mister sin nøkkelkortet sine. Denne appen skal ikke fungere kun for å åpne døra, men brukere har også anledning til å navigere til “Min side” og se profil siden deres.

2.2 Produktsammendrag

Mobilapplikasjonen er ment for bruk av treningssenter medlemmene for åpne døra for treningssenteret. Dette skjer ved at medlemmene logger seg inn med e-mail og password. Når medlemmene kommer nærmere til døra som har bluetooth beacon, så får brukere en varsel på appen at det er bluetooth enhet i nærheten med navnet til selve beacon. Brukeren får melding om dem ønsker å åpne døra, og de trykker på “Ja” alternative så sendes det beskjed til backend for å sjekke om denne brukeren er registrert som medlem. Hvis brukeren er medlem så lyser det grønt på terminalen til døra og da får brukeren å komme seg inn eller lyser det rødt noe som tyder at personen har ikke tilgang til treningssentre eller til det spesifisert arealet.

B.3 Overordnet beskrivelse av interessenter og brukere

Her beskriver vi de ulike interessentene og hvilke roller de har i dette prosjektet, samt hvilken rolle de skal ha under utviklingen av systemet.

Navn	Utdypende beskrivelse	Rolle under utviklingen
Sluttbruker	Sluttbruker av systemet	Bistår med innspill og er sentrale under testing.
Veileder	Faglærer	Bistår prosjektgruppa hvor det er nødvendig.

Prosjektgruppe	Utviklere av systemet	Står for selve utviklingen av systemet.
Oppdragsgivere	Produkteier Gruppe	Har rollen som «kunde» og kommer med krav og ønsker som utføres av prosjektgruppa.

B.4 User stories

- Som bruker ønsker jeg å logge inn i app-en for å få tilgang til treningssenter
akseptansekriterier :
 - Brukeren må være medlem i treningssenteret
 - Brukeren må være registrert med gyldig e-post og og passord
 - brukeren bruker sin e-post og passord for å logge inn.
- Som bruker ønsker jeg å komme inn i treningssenteret i de arealene som jeg har lov til
akseptansekriterier :
 - Brukeren må være logget inn
 - brukeren går til menu og trykker på den
 - brukeren får en sidebar menu og velger inngang
 - brukeren må ha på bluetooth og må stå nærmere til beacon
 - brukeren få alternativer om hvilket arealet han/hun vil gå inn
 - brukeren velger arealet fra liste og det ble sendt en forespørsel om valget
 - Brukeren valideres om han/hun har tilgang til det spesifiserte arealet.

B.5 Produktets funksjonelle egenskaper

En liste over funksjonelle egenskaper som produktet må ha:

Behov	Prioritet	Vedrører	Foreslått løsning
Funksjon for Innlogging	Lav	Innlogging	Innlogging med e-post og password

Funksjon for å søke om beacon signaler som tilhører treningssenteret	høy	Bluetooth beacons	en liste over tilgjengelige beacon signaler som tilhører treningssenteret
Funksjon for å vise bare beacons som er registrert i database	høy		registrerer beacons som tilhører treningssenteret i en database
Funksjon som åpner treningssenter sin dør	høy		sende http-request til en server lages av oppdragsgiver

B.6 Ikke funksjonell krav

Siden dette er kun en prototype produkt, så det ingen krav for testing.

B.7 Prototype

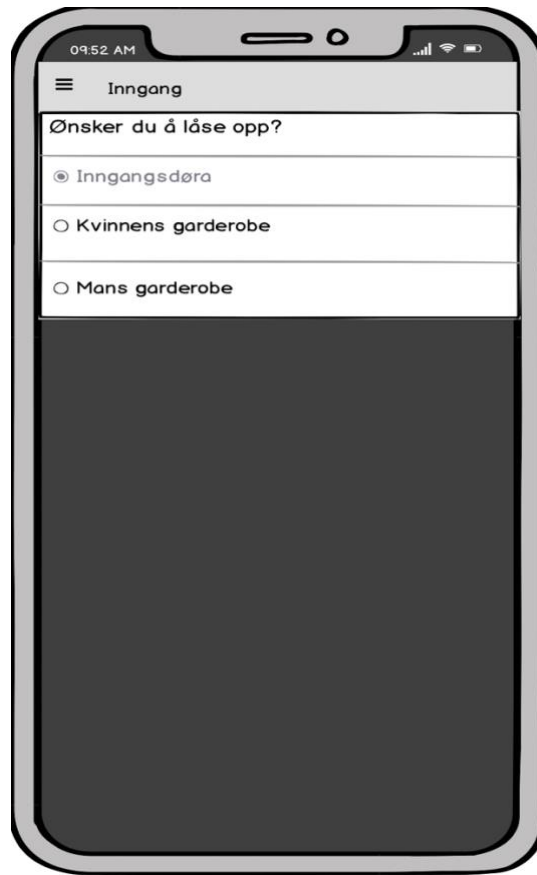
Det ble laget en enkel prototype av applikasjonen ved å bruke Wireframe. Nedenfor er det to versjoner wireframe bilder av designet. Den første versjon ble ikke godtatt og måtte redigere litt på den. Den andre versjon var mer akseptabelt. Det er også en .bmpr filer som har selve prototype som er interaktiv slik at man kan kjøre det og se lite funksjonalitet. Den første versjon:



Figur B. 1: første utkast prototype

Den andre versjon:





Figur B. 2: andre utkast prototype

Vedlegg C Systemdokumentasjon

Prosjektnr 109

Prosjektnavn: “Nøkkelfri adgangskontroll”
Systemdokumentasjon

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
04.05.2020	1.0	Systemdokumentasjon	Shahed Abbas
17.05.2020	1.1	legget til kapittel sikkerhet	Shahed Abbas

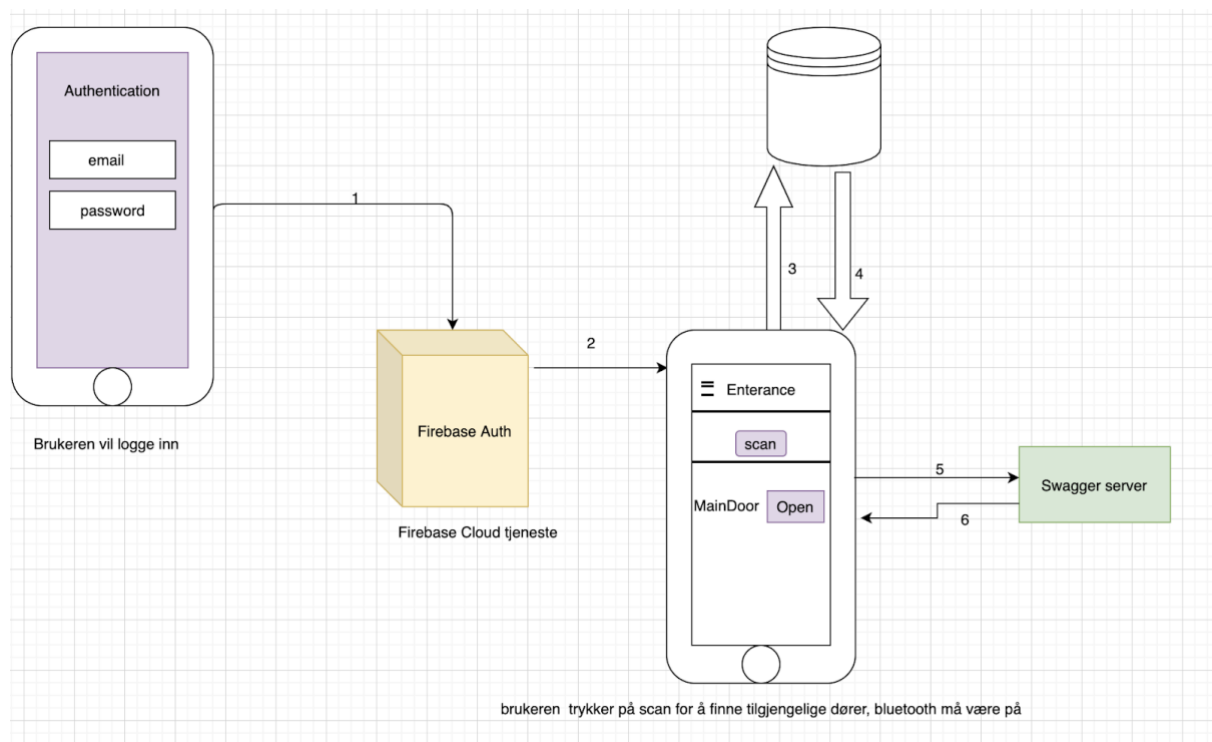
1. Introduksjon

C.1 Hensikt

Dette dokumentet skal beskrive systemet Nøkkelfri adgangskontroll som helhet. Dette dokumentet skal ta for seg alt om systemet som ble utviklet. Dokumentet skal inneholde alt fra prosjekt arkitektur, prosjektstruktur, klient og server sidene. Dette dokument vil ta i utgangspunktet visjonsdokumentet som ble lagt i tidlig fase av prosjektet.

C.2 Arkitektur

2.1 Arkitekturskisse



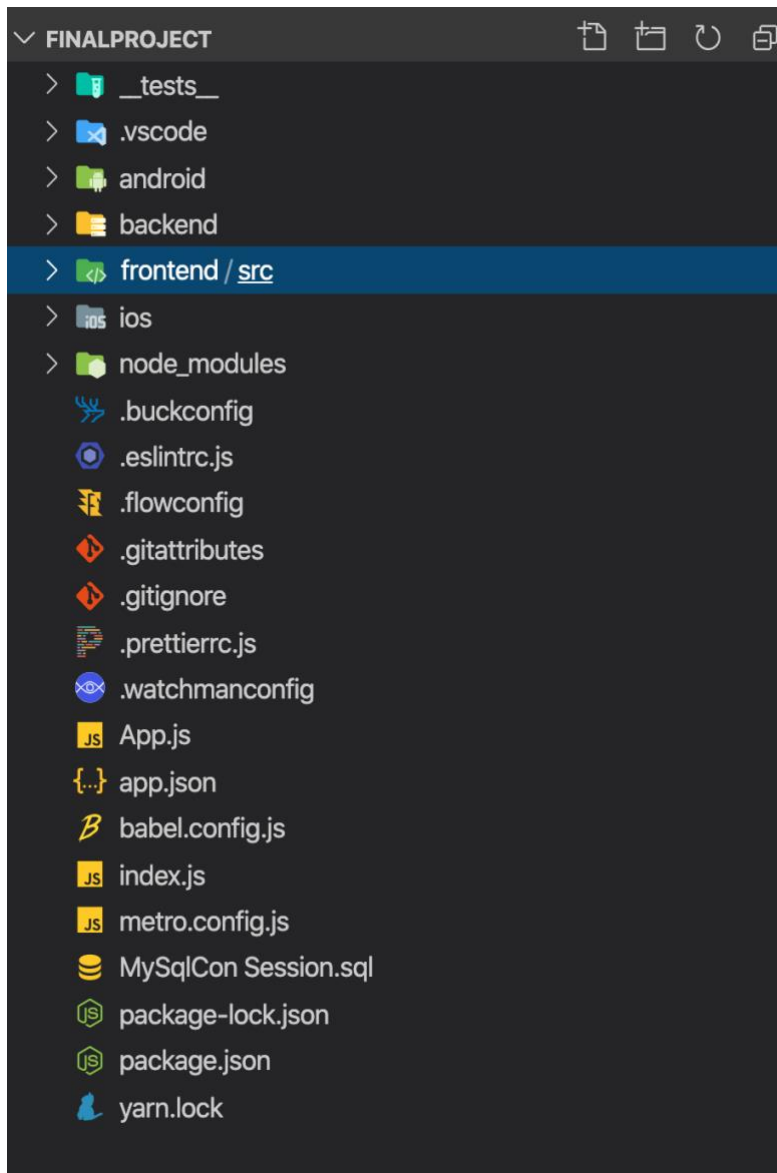
Figur C. 1: Arkitekturskisse

2.2 Tekstlig beskrivelse

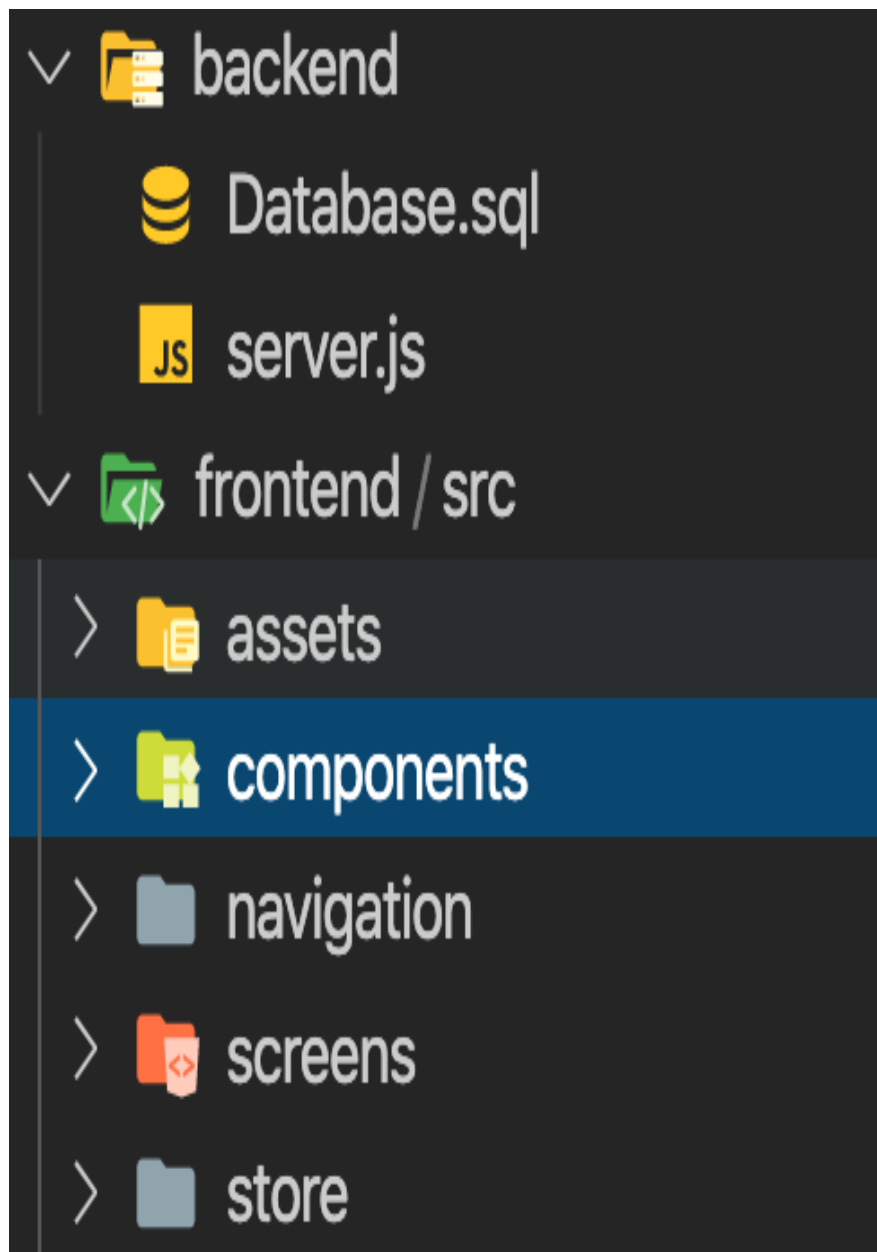
1. Først brukeren logger seg inn ved å skrive e-mail og passord, så sendes disse informasjonen til firebase serveren.
2. Serveren respons vil enten tillate brukeren å logge seg inn hvis brukeren bli autentisert eller feilmelding hvis brukeren skrev feil informasjon. Hvis brukeren er autentisert, får han/hun Home skjerm og derfra navigerer til Entrance skjerm.
3. Her trykker brukeren på "Scan" knappen og må ha bluetooth på slik at den fanger opp alle beacons og deretter sendes beacons id til database for å sjekke om den aktuell id har noe tilsvarende dør navn.
4. Database(sql) returnerer tilbake dør navn til registrert beacons id og da viser det på "Entrance" skjerm navnet til registrert dør og en knapp foran.
5. Brukeren trykker på "Open" knappen, så sender det en HTTP-request til serveren.
6. Serveren sender status 200 ok som innebære at brukeren er autentisert og døra åpnes.

C.3 Prosjektstruktur

Figuren nedenfor viser prosjekt strukturen som helhet. Jeg opprettet backend og frontend mappene. MySqlCon Session.sql ble opprettet i forbindelse med servertilkobling. Resten av filer og mapper genereres automatisk når man oppretter en react native CLI prosjekt.

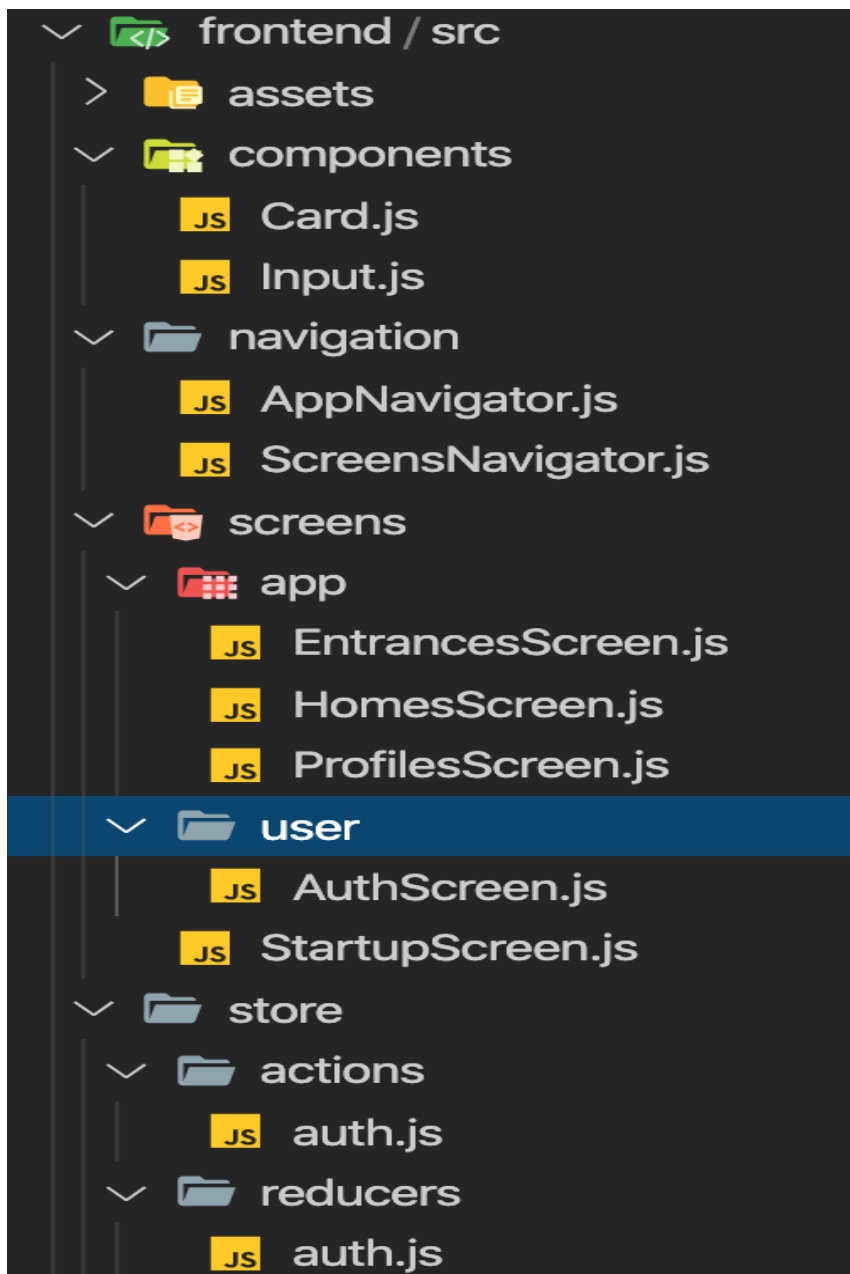


Figur C. 2: Prosjektstruktur



Figur C. 3: Frontend og backend struktur

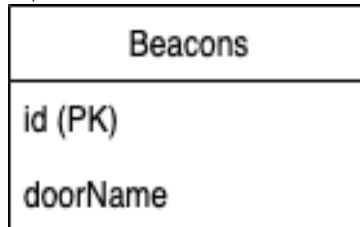
Figuren ovenfor viser strukturen på både frontend og backend mappene. Backend mappen er ikke så stor, den består av en sql-fil og en enkel server fil. Frontend mappen består av flere undermapper og hver undermappe inneholder js-filer.



Figur C. 4: Struktur på Frontend mappe

C.4 Databasemodell

Selv om dette prosjektet var frontend begrenset prosjekt, men måtte bruk mysql database som var en del av problemløsning. Det ble laget en enkel modell eller tabell for å registrere beacons, Hvor id er den primære nøkkelen.



Figur C. 5: beacons tabell i database

C.5 server-tjeneste

Det ble laget en enkel server og denne brukes både av Android og iOS og denne serveren har kun en endepunkt som henter alle beacon via id. Det ble brukt axios bibliotek for å kunne kalle dette endepunktet fra klient side.

```
33 //dette endpunktet for å hente beacons via id
34 app.get('/Beacons/:id', (req, res) => {
35   con.query('Select * from Beacons where id=?', [req.params.id], function (
36     error,
37     rows,
38     fields,
39   ) {
40     if (error) {
41       console.log(error);
42     } else {
43       console.log(rows);
44       res.send(rows);
45     }
46   });
47 });
```

kodesnutt 2: illustrerer endepunktet i serveren

Det ble også benytte Firebase Auth REST API for å autentisere brukere. For eksempel kodesnutt nedenfor viser hvordan man sende http-request til firebase for at brukere skal logge seg inn

```

66 // Denne metoden for å sende forespørsel til firebase hvis en bruker vil logge inn
67 // kilde som ble brukt: https://firebase.google.com/docs/reference/rest/auth#section-sign-in-email-password
68 export const login = (email, password) => {
69   return async (dispatch) => {
70     const response = await fetch(
71       'https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=AIzaSyCr-Yz7QlXhy33YQScIuFLA06YviuvGc1c',
72       {
73         method: 'POST',
74         headers: {
75           'Content-Type': 'application/json',
76         },
77         body: JSON.stringify({
78           email: email,
79           password: password,
80           returnSecureToken: true,
81         }),
82       },
83     );
84
85     if (!response.ok) {
86       const errorResData = await response.json();
87       const errorId = errorResData.error.message;
88       let message = 'Something went wrong!';
89       if (errorId === 'EMAIL_NOT_FOUND') {
90         message = 'This email could not be found!';
91       } else if (errorId === 'INVALID_PASSWORD') {
92         message = 'This password is not valid!';
93       }
94       throw new Error(message);
95     }
96
97     const resData = await response.json();
98     console.log('res', resData);
99     dispatch({
100       authenticate(
101         resData.localId,
102         resData.idToken,

```

kodesnutt 3: illustrerer http-forespørsel til firebase server for innlogging

C.6 Sikkerhet

I forhold til å sikre password, det ble brukt firebase authentication som tilbyr passord sikring og den hasher passwordet når en bruker registrerer seg.

C.7 Dokumentasjon av kildekode, Installasjon og kjøring

. Android

For å kjøre prosjektet på android emulator, trenger man å installere Android studio og gjøre en del setup og da kan man finne nyttig instruksjon for å sette opp android studio her: <https://reactnative.dev/docs/0.61/getting-started>.

For å kjøre prosjektet på fysisk enhet(android mobil), da kan man følge instruksjon her for tilkobling: <https://reactnative.dev/docs/0.61/running-on-device>

1. Først må prosjektet clones lokalt.
2. kjøre kommando **npm install**

3. Selv om Android Setup for kontakt.io beacon er satt, men i tilfelle blir noe problem med å fange opp beacons, da er det bare å dobbeltsjekke om **Auto-linking (React Native v0.60+)** ble gjort riktig som i bildet nedenfor: kilde <https://github.com/Driversnote-Dev/react-native-kontaktio/blob/HEAD/docs/setup.android.md>

det er bare å følge auto-linking fordi react native versjon som er brukt er "React-Native 0.61.5"

Android Setup

Some [Kontakt.io beacons prerequisites](#) have to be met.

Auto-linking (React Native v0.60+)

1. Add package

```
$ yarn add react-native-kontaktio
```

2. In `android/app/src/main/AndroidManifest.xml` inside `<application />` register the Proximity service:

```
<application
  android:name=".MainApplication"
  android:label="@string/app_name"
  android:icon="@mipmap/ic_launcher"
  ...
  <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" />
+ <service android:name="com.kontakt.sdk.android.ble.service.ProximityService" android:exported="false"/>
</application>
```

(... denote omitted lines)

3. Android Marshmallow (6.0) or higher: Don't forget to ask the user to grant the `android.permission.ACCESS_COARSE_LOCATION` permission - see the example in `README`).

Figur C. 6: viser android setup instruksjon for kontakt.io beacons

I dette prosjektet bruker jeg npm og ikke yarn, så for å legge kontakt.io package, man burde kjøre : `npm i react-native-kontaktio`.

Men siden jeg testet prosjektet på android, så de ovenfor nødvendig setup instruksjon ble gjort.

4. For å kjøre prosjektet på android , man må kjøre kommando: `npx react-native run-android`

• iOS

For å kjøre prosjektet på iOS simulator, trenger man å installere Xcode og gjøre en del setup og da kan man finne nyttig instruksjon for å sette opp Xcode her:

<https://reactnative.dev/docs/0.61/getting-started>. For iOS støttes av kun macOS og ikke windows operativsystemet.

For å kjøre prosjektet på fysisk enhet(android mobil), da kan man følge instruksjon her for tilkobling: <https://reactnative.dev/docs/0.61/running-on-device>.

Merk at dette prosjektet ble ikke testet på iOS enheten pga. mangel på iOS mobil og mangel på min macOS sin minnet som gjøre det var vanskelig å oppdatere Xcode.

1.Først må prosjektet clones lokalt.

2.kjøre kommando **npm install**

3. iOS setup instruksjon som er på bildet eller her: <https://github.com/Driversnote-Dev/react-native-kontaktio/blob/HEAD/docs/setup.ios.md>

iOS Setup

Some [Kontakt.io beacons prerequisites](#) have to be met.

Auto-linking (React Native v0.60+)

1. Add package

```
$ yarn add react-native-kontaktio
```

2. Update pods

```
$ cd ios  
$ pod install
```

3. Add permissions as described [below](#).

Figur C. 7: viser iOS setup instruksjon for kontakt.io beacons

4. For å kjøre prosjektet på iOS , man må kjøre kommando: `npx react-native run-ios`

Referanser

- [1a] [Jan Petter Moen](#) “Hvorfor velger adgangskontroll?” 30. januar 2018 [Accessed 03.04.2020]
tilgjengelig online: <https://blogg.assaabloyopeningsolutions.no/virksomhet/hvorfor-velge-adgangskontroll>
- [1b] Leksjon 2 - Introduksjon til brukergrensesnitt, brukskvalitet, brukeropplevelser og Interaksjonsdesign, hentet fra TDAT2003 Systemutvikling 2 med applikasjoner sin MMI leksjoner
- [2] [Håkon Sandmark](#) “Hva erstatter nøkkelen i et nøkkelfritt adgangssystem?” desember.2018 [Accessed 03.04.2020] tilgjengelig online: <https://blogg.assaabloyopeningsolutions.no/virksomhet/sikkerhet-hva-er-et-nokkelfritt-system>
- [3a] Nielsen, Jakob. 1993. Usability Engineering. San Francisco, CA, USA: Morgan Kaufmann. ISBN-10: 0-12-518406-9 ISBN-13: 978-0-12-518406-9
- [3b] Denne lille dingsen kan endre hverdagen din [Accessed 08.04.2020] tilgjengelig på <https://www.tek.no/nyheter/feature/i/wP0Jvd/denne-lille-dingsen-kan-endre-hverdagen-din>
- [4] Er dette det kuleste treningsarmbåndet? [Accessed 08.04.2020] tilgjengelig på <https://www.tek.no/test/i/JoRqqR/polar-loop-activity-tracker>
- [5] av Helge Olaussen “iBeacon - til glede eller besvær?” 15.08.2018 [Accessed 07.04.2020] tilgjengelig online på <https://blogg.oktanoslo.no/ibeacon-til-glede-eller-besvaer>.
- [6] <https://www.beaconstac.com/apple-ibeacon-technology>
- [7] <https://electricnews.fr/over-100-use-cases-and-examples-for-ibeacon-technology/>
- [8] Sommerville, I. Software Engineering Ninth Edition. 2009, [Accessed 10.04.2020] tilgjengelig online på https://dinus.ac.id/repository/docs/ajar/RPL-Sommerville_-_Software_Engineering_9ed.pdf
- [9] <https://agilemanifesto.org/iso/no/principles.html>
- [10] <https://agilemanifesto.org/iso/no/manifesto.html>
- [11] FUNDAMENTALS OF Database Systems Seventh Edition of Ramez Elmasri && Shamkant B. Navathe [Accessed 11.04.2020]
- [12] [https://no.wikipedia.org/wiki/React_\(webrammeverk\)](https://no.wikipedia.org/wiki/React_(webrammeverk))
- [13] Learning React Native: Building Native Mobile Apps with JavaScript Second edition av Bonnie Eisenman
- [14] Mastering React Native av Eric Masiello && Jacob Friedmann
- [15] <https://alligator.io/react/react-native-elements/> 3.feb.2020
- [16] [https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library))
- [17] av [Sebastian Eschweiler](#) den 7.03.2017 [Accessed 12.04.2020] og tilgjengelig på : <https://medium.com/codingthesmartway-com-blog/getting-started-with-axios-166cb0035237>
- [18] <https://reactnavigation.org/docs/getting-started>
- [18] Firebase documentation [Accessed 13.04.2020] og tilgjengelig online på : <https://firebase.google.com/docs/auth>
- [19] <https://github.com/Driversnote-Dev/react-native-kontaktio>
- [20] <https://getgist.com/difference-between-native-vs-web-vs-hybrid-apps/>
- [21] <https://www.flaticon.com/home>
- [pic1] <https://images.app.goo.gl/sofBaAx5qEj61oGp6>
- [pic2] <https://images.app.goo.gl/LgSw6ntBzVY5if3dA>

