

Bachelor oppgave - Ekbok Kjøreapp

**Prototype på GPS-basert mobilapp for
kjørebok for Adcom**



Forfatter(e):

Joakim Mork Landsverk

Fredrik Høvik Sørøy

Totalt antall sider inkludert forsiden: 46

Molde, Innleveringsdato: 16.05.2022

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

| Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6: | | |
|---|---|---|
| 1. | Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen. | X |
| 2. | Jeg/vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none"> ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. ikke refererer til andres arbeid uten at det er oppgitt. ikke refererer til eget tidligere arbeid uten at det er oppgitt. har alle referansene oppgitt i litteraturlisten. ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse. | X |
| 3. | Jeg/vi er kjent med at brudd på ovennevnte er å <u>betrakte som fusk</u> og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§16 og 36. | X |
| 4. | Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert, jf. høgskolens regler og konsekvenser for fusk og plagiat | X |

| | | |
|----|--|---|
| 5. | Jeg/vi er kjent med at høgsolen vil behandle alle saker hvor det forligger mistanke om fusk etter høgsolens retningslinjer for behandling av saker om fusk | X |
| 6. | Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider | X |

Personvern

Publiseringsavtale

Studiepoeng: 15 poeng

Veileder: Ketil Danielsen

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).

Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage HiM med forfatter(ne)s godkjenning.

Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved Høgskolen i Molde en vederlagsfri rett til å

| | | |
|--|-----------|-----|
| gjøre oppgaven tilgjengelig for elektronisk publisering: | <u>JA</u> | nei |
|--|-----------|-----|

| | | |
|---------------------------------------|----|------------|
| Er oppgaven båndlagt (konfidensiell)? | ja | <u>NEI</u> |
|---------------------------------------|----|------------|

(Båndleggingsavtale må fylles ut)

- Hvis ja:

| | | |
|---|----|-----|
| Kan oppgaven publiseres når båndleggingsperioden er over? | ja | nei |
|---|----|-----|

Dato: 16.05.22

Innholdsfortegnelse

| | |
|----------------------------------|----------|
| Innholdsfortegnelse | 4 |
| Sammendrag | 7 |
| Kapittel 1: Innledning | 7 |
| 1.1: Problemstilling | 7 |
| 1.2: Visjon for fullført oppgave | 7 |

| | |
|--|-----------|
| Oppgaven | 8 |
| Kapittel 2: Om Prosjektet | 8 |
| 2.1: Organisasjon og styring av prosjektet | 8 |
| GitHub | 8 |
| Trello | 8 |
| Kategorier | 9 |
| Lister | 9 |
| Epics | 10 |
| Scrum for trello | 11 |
| 2.2: Rammeverk / Biblioteker | 11 |
| React Native | 12 |
| Firebase | 12 |
| Kapittel 3: Utførelse | 13 |
| 3.1: VSCode | 13 |
| Node.JS | 13 |
| Utvidelser | 13 |
| 3.2: Prosess | 14 |
| 3.3: Testing | 15 |
| Android studio og Expo Go | 15 |
| Kapittel 4: Informasjonsarkitektur | 15 |
| 4.1: Introduksjon | 15 |
| 4.2: Skissering/prototyping | 16 |
| 4.3 Applikasjonens Oppbygging | 20 |
| 4.4: Heuristikk | 21 |
| 4.5: Brukertester | 23 |
| Oppgaver: | 24 |
| Tilbakemeldinger etter brukertesting: | 24 |
| 4.6: Resultat Informasjonsarkitekturanalyser | 24 |
| Kapittel 5: Informasjonssikkerhet | 25 |
| 5.1: Verdivurdering | 25 |
| Kartlagte verdier for applikasjonen | 25 |
| 5.2: Trusselmodellering | 25 |
| 5.3: Risikoanalyse | 27 |
| Konsekvenser av hendelsene i trusselmodelleringen | 27 |
| Sannsynligheten for hendelsene i trusselmodelleringen | 28 |
| Tabellen for ROS-analysen | 29 |
| Konklusjon basert på ROS-analysen | 30 |
| 5.4: Tiltaksliste og prioritering | 30 |
| Kostnad for tiltakene | 30 |
| Oversikt over effektiviteten tiltakene har totalt sett | 32 |
| 5.5 Resultat fra Informasjonssikkerhetanalyser | 32 |
| Kapittel 6: Programmering | 33 |

| | |
|---|-----------|
| 6.1: Prosjekt Oppstart | 33 |
| Firebase oppsett | 33 |
| Videre utvikling av prototype | 33 |
| Styling | 34 |
| Registrering | 34 |
| Innlogging | 35 |
| Glemt passord | 35 |
| Kommentering | 35 |
| Refaktoring | 36 |
| 6.2: Videreføring av prosjektet | 36 |
| Hovedsiden | 36 |
| Innstillinger | 37 |
| Kjøring | 38 |
| Historikk | 40 |
| Historikk Underside | 40 |
| Utlogging | 41 |
| Akseptasjonstest - Løste gruppen kravene Adcom gav til applikasjonen - Bekreftelse fra Adcom | 41 |
| Litteraturliste | 42 |
| Lenker | 42 |
| Node.JS: | 42 |
| VSCode: | 42 |
| React Native: | 42 |
| Firestore: | 42 |
| Ekbok: | 42 |
| Expo Go: | 42 |
| Expo tracking: | 42 |
| Stack Overflow: | 43 |
| Heuristikk: | 43 |
| Dribbble: | 43 |
| NSM | 43 |
| Vedlegg | 44 |
| Vedlegg 1: Brukertest samtykke skjema Ekbok - knut olav. | 44 |
| Vedlegg 2: Brukertest samtykke skjema Ekbok - Håvard | 44 |

Sammendrag

I løpet av denne bachelor oppgaven har gruppen forsøkt å belyse hvordan man kan gå fram dersom man ønsker å lage en mobil applikasjon i hovedsak for Android, men med bruk av React Native er det mulig å utvide støtten til iOS også.

Vi har i denne oppgaven kapitler som inneholder hvordan prosjektstyringen ble gjort, hvilke rammeverk og bibliotek vi benyttet oss av for å nå målet, utdypelse om hva som spesifikt ble brukt i utviklingen for å dra mest mulig nytte ut av perioden vi har disponibel til programmeringsfasen, informasjonsarkitektur ble brukt til å analysere hvordan applikasjonen eventuelt burde se ut basert på skissering, heuristikk og brukertesting.

Vi inkluderte også et kapittel som tar for seg informasjonsikkerhetsdelen av prosjektet, hvor det ble forsøkt å kartlegge hvilke sikkerhetskritiske verdier applikasjonen inneholder samt trussel modellering og risikoanalyser. Til slutt i kapittel 6 kommer delen hvor vi går igjennom de forskjellige komponentene vi lagde, og framgangsmåtene vi brukte for å få appen til å fungere i samsvar med kravspesifikasjonen.

Gruppen endte med en applikasjon som ble godkjent av oppdragsgiveren Adcom etter utført arbeid, og en generelt positiv erfaring fra arbeidet.

Kapittel 1: Innledning

1.1: Problemstilling

Adcom har et ønske om å utvikle en mobilapp som skal fungere som en demoversjon av en tjeneste som heter Ek bok, som de tilbyr via nettside per i dag.

Denne mobilapplikasjonen skal være en elektronisk kjørebok som lar brukeren se ruten som ble tatt for å nå destinasjonen, og den skal inneholde:

- Innlogging
- Registrering
- Glemte passord
- Hjem
- Kjøring (Start / Stopp)
- Historikk(Viser oversikt over turer kjørt, med start og stopp adresse)

1.2: Visjon for fullført oppgave

Formålet er å gi Adcoms nåværende og prospektive kunder flere alternativer for bruk av deres tjeneste, samt nå en bredere brukerbase ved å tilby et bredere utvalg av tjenester.

Oppgaven

Adcom hadde sett for seg denne mobilapplikasjonen som en gratis variant av deres webbaserte kjøredagbok tjeneste Ekbok, og ønsket å ha denne applikasjonen som en form for en demo for deres nåværende produkt, som skal ha mer utvidet funksjonalitet i forhold til applikasjonen som lages i denne oppgaven.

Kapittel 2: Om Prosjektet

2.1: Organisasjon og styring av prosjektet

For å utvikle en slik mobilapplikasjon har vi valgt å bruke noen verktøy for prosjektstyring, for å ha en bedre oversikt over hvor langt man har kommet i prosessen og hva som er planlagt for fremtiden.

GitHub

Under utviklingen av dette prosjektet har gruppen valgt å benytte GitHub for å dele koden med hverandre, og å samarbeide under utviklingsprosessen.

For å unngå problemer som kan oppstå når flere personer jobber med den samme koden, ble det opprettet flere grener, slik at hver person i gruppen hadde mulighet til å kunne jobbe med sine løsninger uten å skape problemer for hverandre.

I dette prosjektet er "Main" hoved grenen hvor gruppemedlemmene laster opp sitt arbeid mot etter at det er testet i det gitte medlemmets gren.

Etter at grenene er kombinert til "Main" testes koden av de andre medlemmene på gruppen for å kunne dobbeltsjekke at alt fungerer som det skal, før man begynner på neste gjøremål.

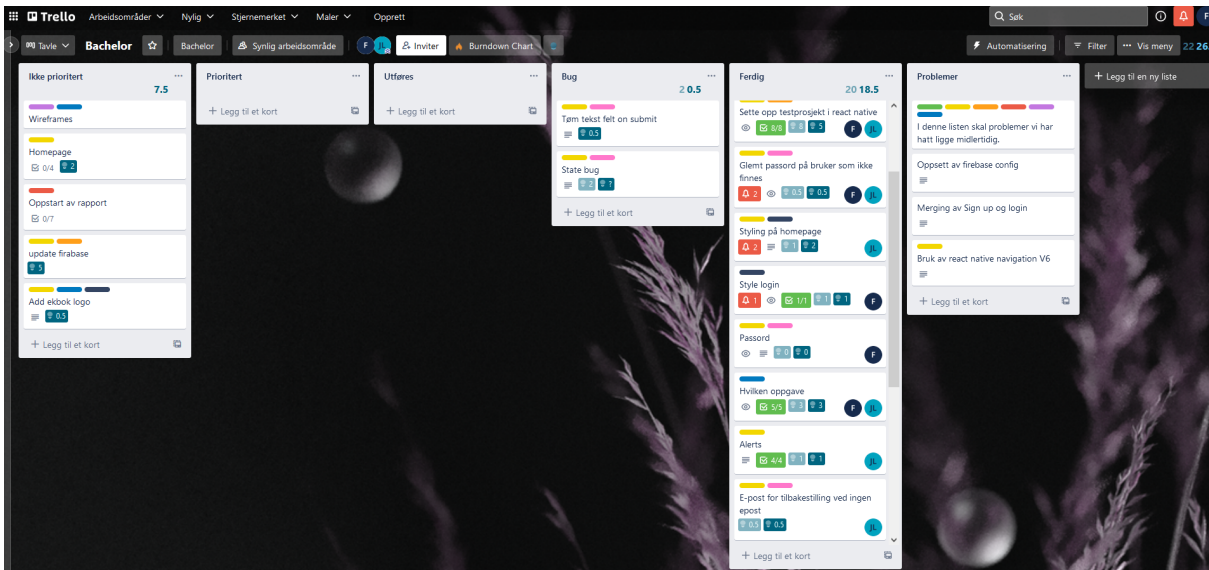
Trello

[Trello](https://trello.com/) er et verktøy for bruk av Kanban som er en form av Lean/Agile metoder for å notere gjøremål.¹

Trello kan brukes til å holde oversikt over alt som skal gjøres og sammenligne oppgavene som ligger i de forskjellige listene for å bestemme hva som burde gjøres i neste steg av utviklingsperioden.

I resten av dette delkapitlet vil vi gå igjennom hvordan gruppen har valgt å ta i bruk Trello.

¹ <https://trello.com/>



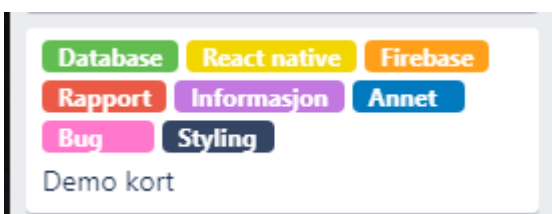
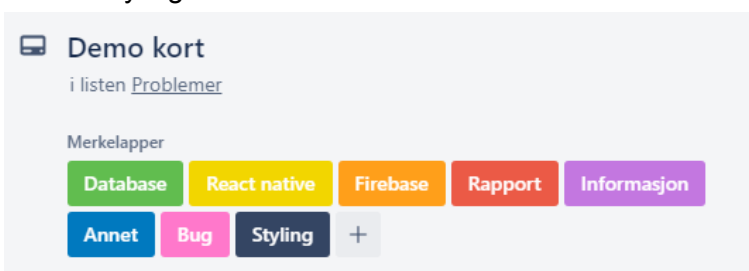
I det inkluderte bildet viser vi hvordan gruppen valgte å sette opp trello Brettet.

Kategorier

Trello kan settes opp med merkelapper som kategorier for å enklere kunne se hva de forskjellige oppgavene inneholder.

Kategorier benyttet:

- Database
- React Native
- Firebase
- Rapport
- Informasjon
- Annet
- Bug
- Styling



Lister

I Trello opprettes en side hvor man legger inn lister for å sortere arbeidsoppgavene. Dette gjøres for at man enklere skal kunne skille hvilken prosess de forskjellige oppgavene hører til.

Under kan man se hvilke lister vi har valgt å bruke i dette prosjektet.

Ikke prioritert

I denne listen legges oppgaver som ikke er prioritert enda. Disse oppgavene blir flyttet til prioritert ut fra rekkefølgen av hvordan oppgavene bør løses og hvor mye tid som er tilgjengelig.

Prioritert

I denne listen legges oppgavene som har høy prioritert, og som bør løses før arbeid på andre oppgaver kan startes/fortsette.

Utføres

I denne listen legges oppgaver som er under aktivt arbeid, slik at det skal være enkelt å se hva gruppemedlemmene arbeider med, og man slipper at det blir overlapp av arbeid ved at to personer arbeider på samme oppgave uten at det er avtalt.

Bug

Dersom et problem oppdages under testing av programmet, så lages en oppgave i bug listen slik at gruppen vet at dette er et problem som må løses.

Ferdig

I denne listen legges ferdige oppgaver, slik at det blir lettere å se igjennom hva som har blitt gjort, hvem som har gjort hva og hvor lang tid det ble brukt på denne oppgaven.

Epics

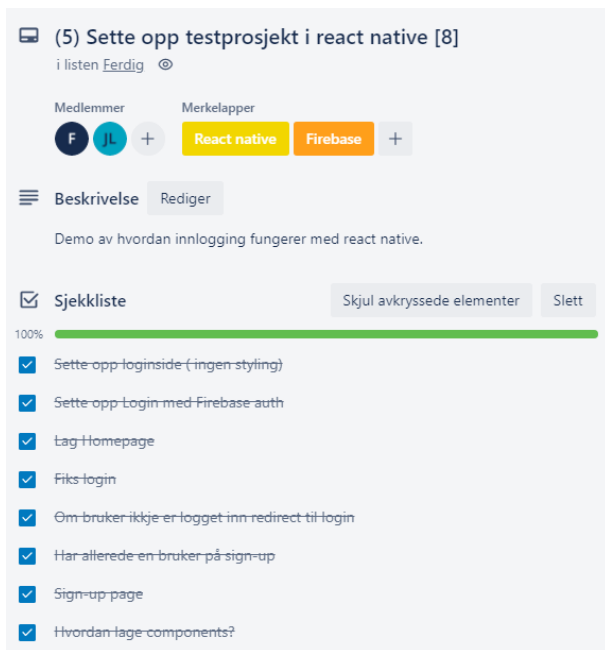
I denne oppgaven brukes noe som heter "[epics](#)"²

En epic er en stor oppgave, hvor man har under-oppgaver som heter "[user stories](#)"³. En oppgave er ikke ferdig før alle story points er utført og markert som gjennomført. Når det settes opp en user story under en epic, kan man skrive en kort beskrivelse av hva som skal gjøres, og det legges inn punkter i sjekklister slik at det er enkelt å markere en user story som utført, ved å klikke på punktet.

Grunnen til at vi ønsker å ta i bruk epics er for å redusere antall små oppgaver vi har, og heller samle dem under epics.

² <https://www.productplan.com/glossary/epic/>

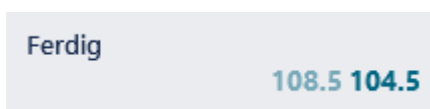
³ <https://www.productplan.com/glossary/user-story/>



Scrum for trello

For å ha kontroll på hvor mye som skal gjøres og hvor lang tid som blir brukt på hver oppgave, har det i dette prosjektet blitt brukt en nettleser utvidelse som heter "[Scrum for Trello](#)"⁴. Denne utvidelsen lar brukerne velge hvor mye tid man tror en oppgave kommer til å ta, og hvor mye tid oppgaven faktisk tok.

Scrum for trello summerer også opp hvor mange antatte og faktiske timer som ligger i hver liste, slik at man kan få en oversikt over hvor mange timer som det er antatt at de prioriterte oppgavene kommer til å ta, og hvor mange timer som faktisk har blitt brukt på prosjektet til enhver tid som vises i figuren under. Det lyseblå tallet gjenspeiler antatt tid brukt, mens det mørkeblå tallet viser faktisk tid brukt.



2.2: Rammeverk / Biblioteker

For å effektivisere utførelsen av en slik oppgave finnes det rammeverk man kan bruke for å gjøre utviklingsprosessen enklere. I dette prosjektet tok vi noen vurderinger av forskjellige tjenester man kan bruke for å redusere tiden gruppen måtte bruke på ting som det finnes alternativer for enn å lage det fra bunnen av selv. Et eksempel på dette er bruken av Googles Firebase til å drifte alle backend behov i oppgaven noe som flytter en del av det nødvendige arbeidet over til Googles tjenester.

⁴ <http://scrumfortrello.com/>

React Native

Denne applikasjonen ble hovedsaklig utviklet med tanke på Android, men siden vi benytter React Native, så har vi mulighet til å også utvide støtten til iOS også, dette var en av grunnene til at vi valgte React Native.

[React native](#)⁵ og [ReactJS](#)⁶ er begge utviklet av Facebook, men hvor ReactJS er et bibliotek mer rettet mot å lage interaktive nettsider, så er React Native et rammeverk for å gjøre utviklingen av mobilapplikasjoner enklere, ved at det inneholder funksjoner som allerede er laget, slik at utviklerne ikke trenger å lage dette på nytt, og kan ta i bruk dette for en raskere start.

React Native bruker "views" som er bakt inn i språket som oversetter Javascript til et språk som IOS og Android forstår respektivt. Dette gjør at man kun trenger å ha en kodebase og det vil være kompatibelt med begge de to nevnte mobil operativsystemene.

Firestore

[Firestore](#)⁷ er en tjeneste opprettet av google som hjelper utviklere med å sette opp en "backend", som gjør at utviklere ikke trenger å opprette en egen backend for å få programmene sine til å fungere og snakke med databaser.

Firestore hjelper utviklere ved at de tilbyr en rekke tjenester utviklere kan velge å ta i bruk:

Google authentication

[Firestore authentication](#)⁸ hjelper utviklerne med hele prosessen av brukerkontoer, og håndteringen av disse.

Dette inkluderer:

- Innlogging (med bruker, google konto, mobil, facebook konto, og flere)
- Kryptering av passord
- Deaktivering av brukere
- 2 faktor autentisering
- Glemt passord
- Bytte av e-postadresse
- Sending av e-post til brukere

[firebaseauth](#)⁹

Firestore / Realtime database

Firestore inkluderer to forskjellige cloud baserte databaser. Begge databasene blir synkronisert i samtid til alle klienter, og er tilgjengelig også dersom programmet ikke er koblet til internett.

1. [Realtime database](#)¹⁰ er en cloud-basert database, som lagrer dataene i JSON filer.

⁵ <https://reactnative.dev/>

⁶ <https://reactjs.org/>

⁷ <https://firebase.google.com/>

⁸ <https://firebase.google.com/docs/auth/>

⁹ <https://firebase.google.com/docs/auth/>

¹⁰ <https://firebase.google.com/docs/database/>

2. [Firestore](#)¹¹ er en fleksibel og skalerbar database for mobil, web, og server utvikling.

Kapittel 3: Utførelse

3.1: VSCode

[VSCode](#)¹² (<https://code.visualstudio.com/>) eller Visual Studio Code er en gratis kode editor som er utviklet som open source fra Microsoft, og har støtte for mange [utvidelser](#)¹³ som kan gjøre livet til en utvikler lettere når man skal kode. Det finnes mange utvidelser for nesten alle programmeringsspråk man kan tenke seg, og dette gjør VSCode til et naturlig valg av kode editor for mange utviklere.

Node.JS

I dette prosjektet har brukes [Node.JS](#)¹⁴ (<https://nodejs.org/en/>) for kjøre applikasjonene som lages i React Native. Det er også Node JS som gjør at man enkelt kan installere de rammeverkene og bibliotekene man har behov for prosjektet. Et eksempel kan være om man vil installere React Native, så kan du skrive det følgende i den innebygde terminalen:

“npm install react”

Så gjør den jobben med å hente det man trenger for å bruke React. Etter man har installert alle pakkene med biblioteker eller rammeverk starter man prosjektet sitt ved å skrive **“npm start”**.

Utvidelser

Utvidelsene finnes under “Extentions” fanen i VSCode.

Noen av utvidelsene som har blitt brukt i løpet av utviklingen av denne mobilappen er:

ES7 React/Redux/GraphQL/React-Native snippets

- [Snippets](#)¹⁵

Fullfører funksjoner basert på kodeord som er predefinerte av utvidelsen

For eksempel: Om man skriver “imrc” i en fil så får du valget om å automatisk importere react og components fra react biblioteket.

```
imrc→ import React, { Component } from 'react'
```

Prettier - Code formatter

- Fra siden på Visual Studio marketplace forklarer de utvidelsen slik:
[Prettier](#)¹⁶ is an opinionated code formatter. It enforces a consistent style by parsing

¹¹ <https://firebase.google.com/docs/firestore/>

¹² <https://code.visualstudio.com/>

¹³ <https://marketplace.visualstudio.com/VSCode>

¹⁴ <https://nodejs.org/en/>

¹⁵ <https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets>

¹⁶ <https://prettier.io/>

your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.”

Auto rename tag

- [Auto rename tag](#)¹⁷ hjelper ved å endre skriften i den korresponderende slutt taggen om du endrer skriften i start taggen.

ESLint

- [ESLint](#)¹⁸ hjelper med å rette på syntax feil den merker mens du skriver koden, slik mengden feil som oppstår av skriverfeil kan reduseres.

Auto close tag

- [Auto close tag](#)¹⁹ hjelper utviklere å kode raskere ved å fullføre tags de bruker i html eller i react ved å automatisk generere en slutt tag når man skriver åpningstaggen.

Auto import

- [Autoimport](#)²⁰ hjelper med importeringsprosessen i Javascript.

React Native Tools

- [React Native Tools](#)²¹ gir tilgang til utvidet funksjonalitet for React Native applikasjoner man utvikler.

3.2: Prosess

I planleggingsfasen av prosjektet arbeider gruppen sammen, og skriver ned ideer, problemer og utfordringer slik at man ikke skal trenge å ta en pause fra utviklingen for å bli enige om noe, som kan diskuteres før man begynner.

Etter at gruppen har bestemt hvordan en oppgave skal løses, lages en epic i trello hvor det blir bestemt hvem som skal gjøre oppgaven.

I utviklingsprosessen arbeider gruppen for det meste selvstendig, men hjelper hverandre med testingen etter at et gjøremål er fullført, for å finne flest mulig feil som kan rettes.

Dersom gruppen jobber med en spesielt stor eller utfordrende oppgave, kan man bruke pair-programming for å løse denne oppgaven lettere. Dette betyr at man jobber sammen, ofte ved at en person deler skjermen sin, samtidig som de programmerer og den andre ser hva som blir gjort og kan komme med innspill til hva som bør gjøres eller inkluderes. Denne løsningen gjør at man enklere kan finne løsninger på større problemer, og implementere disse.

¹⁷ <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-rename-tag>

¹⁸ <https://eslint.org/>

¹⁹ <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-close-tag>

²⁰ <https://marketplace.visualstudio.com/items?itemName=NucleaR.vscode-extension-auto-import>

²¹ <https://marketplace.visualstudio.com/items?itemName=msjsdiag.vscode-react-native>

3.3: Testing

Android studio og Expo Go

For å teste koden som har blitt skrevet, ble [Android Studio](#)²² med [Expo Go](#)²³ brukt. Dette var for å emulere en android mobil slik at man kan teste hvordan ting fungerer, og hva en bruker vil oppleve dersom de kjører programmet på sin mobil.

Expo Go er en Mobilapplikasjon som lar bruken teste programmet den utvikler på en mobiltelefon, ved at brukeren kan laste opp kodefilen sin med en kommando for å kunne kjøre denne på en mobiltelefon.

Android Studio lar gruppen bruke virtuelle telefoner med android installert. Disse fungerer likt som en fysisk mobil, og lot gruppen arbeide mer effektivt.

Kapittel 4: Informasjonsarkitektur

4.1: Introduksjon

For å få en bruker til å velge å bruke din applikasjon i motsetning til konkurrerende applikasjoner, kan man bruke en god oppbygging av informasjonsarkitektur for å gjøre applikasjonen så enkel som mulig å bruke, uavhengig av brukerens tidligere erfaringer med mobilapplikasjoner.

I designfasen av prosjektet har gruppen satt stort fokus på de fem kvotene som er nevnt under, og benyttet disse som en pekepinn på hva som skulle være i fokus under designingen av applikasjonen.

1. “Krug’s first law of usability: Don’t make me think” ([1] s. 24)
2. “Krug’s second law of usability: It doesn’t matter how many times I have to click, as long as each click is a mindless, unambiguous choice.” ([1] s. 54)
3. “Nothing important should ever be more than two clicks away” ([1] s. 24)
4. “Get rid of half the words on each page, then get rid of half of what’s left.” ([1] s. 59)
5. “The name of the page will match the words I clicked to get there.” ([1], s. 80)

²² <https://developer.android.com/studio/>

²³ <https://expo.dev/expo-go>

Alle fem kvoter er tatt fra boken: "[Don't Make Me Think](#)"[1] av Steve Krug. Ved å benytte Krugs bok under utviklingen kan man sjekke om man har fulgt tipsene i etterkant, og stille seg spørsmålet om man har fulgt tipsene under designfasen.

Noen av kvotene vi valgte å forholde oss til kan virke motstridene, som nr.1 og nr.4, men her har vi valgt å bruke begge, og forsøkt å bruke dem til å treffe et midtpunkt mellom de to ved hjelp av å gjøre at ting ligger færrest mulig klikk unna samtidig som at de ikke får brukeren til å undre hva valgene gjør.

Gjennom denne prosessen kan man utføre brukertesting for å få tilbakemeldinger man kan referere til i design arbeidet. I tillegg kan man utføre brukertester og heuristikker. Dette kan være nyttig fordi brukere og utviklere kan ha forskjellige perspektiv når man tenker på applikasjonen.

På grunn av dette kan en utvikler også ha partiskhet som kan gjøre det enklere for utvikleren å forklare bort hvorfor noe er som det er istedenfor å tenke på hvordan det kan gjøre på en mer intuitiv måte.

Ved bruk av tester, tilbakemeldinger og analyser, vil man kunne se bort ifra denne partiskheten og se applikasjonen med "nye øyne" som vil kan føre til at man enklere kan oppdage ting som ikke er så selvforklarende som antatt før man begynner prosessene.

4.2: Skissering/prototyping

Dette er noen tidlige skisser på hvordan vår applikasjon så ut før vi utførte noe testing av brukere eller heuristiske analyser, som vil bli utført i de neste del kapitlene.

| | |
|-------------------|---------------------|
| Innlogging | Registrering |
|-------------------|---------------------|



Innlogging

E-post

Passord

[Glemt Passord](#)

Logg inn

[Har du ikke en bruker enda?](#)

Glemt Passord

← Registrering



E-post





Passord



Gjenta Passord

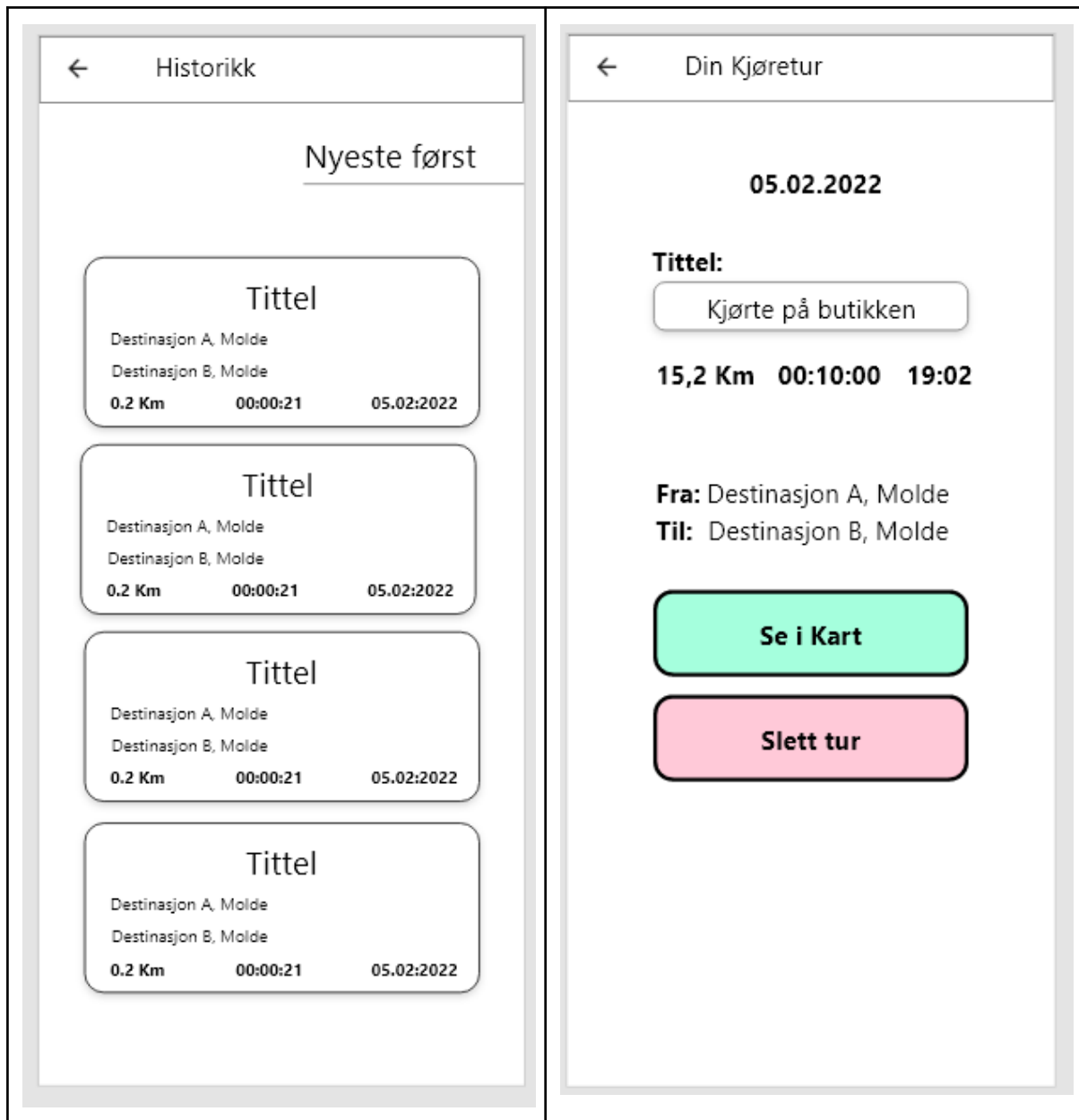
Jeg er enig med [Bruksvilkårene](#)

Registrer

Hjem

| | |
|--|--|
| <p>← Glemt Passord</p>  <p>E-post</p> <input data-bbox="304 719 691 801" type="text" value="E-post"/> <p>Bekreft</p> |  <p>Hjem</p> <p>Kjøring</p>  <p>Historikk</p>  <p>Logg Ut</p> |
| <p>Kjøring</p> | <p>Kjøring</p> |
| <p>Stoppet</p> | <p>Kjører</p> |

| | |
|--|---|
| <p>← Kjøring</p> <p>EKBOOK</p> <p>Kjøring</p> <p>Start</p>  <p>00:00:00</p> | <p>← Kjøring</p> <p>EKBOOK</p> <p>Kjøring</p> <p>Stopp</p>  <p>00:00:00</p> |
| <p>Historikk</p> | <p>Din Kjøretur</p> |



4.3 Applikasjonens Oppbygging

Dette er en beskrivelse på hvordan denne applikasjonen er oppbygd hvor overskriften er hvilken side bruken er på, og listen under er hvilke alternativer som befinner seg på denne siden. Dette er gjort for å kunne vise frem hvordan navigeringen fungerer.

Innlogging:

- Registrering
- Glemt Passord

Hjem

- Kjøring
- Historikk
- Innstillinger

Historikk

- Dine kjøreturer

Dine Kjøreturer:

- Kartvisning

Innstillinger

- Endre passord
- Aktiver / Deaktiver 2 faktor autentisering

4.4: Heuristikk

Heuristiske analyser kan bli brukt for å finne problemer i applikasjonen. I dette prosjektet har vi valgt å gå igjennom [Jakob Nielsens 10 heuristikker](#) ²⁴

for finne potensielle Problemområder i applikasjonens første prototype

Fordi noen av punktene i heuristikken er relevante i flere heuristikker har vi valgt å skrive de samme punktene hvor de passer, selv om de allerede er listet opp tidligere

1: Visibility of system status

- Noen av tilbakemeldingene ved feilmelding i innlogging og registrering kan gjøres bedre, ved å forbedre tegnbruk, og spesifisering.
- Brukeren får tilbakemelding om hva som skjer, et eksempel kan være når en bruker klikker bekreft sending av nytt passord, vil man få en beskjed om at e posten blir sendt.
- Dersom en side må vente på å hente noe fra databasen vil brukeren få en tilbakemelding om at siden lastes inn.

2: Match between system and the real world

- Historikk kan byttes til: "Dine kjøreturer", noe som kan gjøre det enklere for en bruker å forstå hva som ligger der.
- Applikasjonen benytter ikon som samsvarer med betydningen av funksjonen den leder til, et eksempel på dette kan være "Kjøring" har en bil som ikon.

²⁴ <https://www.nngroup.com/articles/ten-usability-heuristics/>

- I applikasjonen er det forsøkt å bruke enkelt språk, slik at en bruker skal forstå hva som skal gjøres.

3: User control and freedom

- Det kan være en ide å implementere en angreknapp, dersom en bruker sletter en kjøretur.
- Dersom en bruker klikker på slett kjøretur, vil en advarsel dukke opp for å spørre en bruker om man virkelig vil slette denne kjøreturen.
- Applikasjonen bruker en topplinje som forklarer hvor en bruker er, og hvordan man kan gå tilbake.
- Dersom en bruker starter en kjøretur og stopper den med en gang, vil denne ikke bli lagret.
- En bruker har ikke mulighet til å endre passord, uten å bruke glemt passord funksjonen.

4: Consistency and standards

- Som i punkt 2 gjelder: Applikasjonen benytter ikon som samsvarer med betydningen av funksjonen den leder til, et eksempel på dette kan være "Kjøring" har en bil som ikon.
- Ved å legge tilbake knapp og tittel i en topplinje er det forsøkt å følge en standard som brukes i applikasjonsutvikling.
- Under utviklingen av innloggingen er det blitt sett på andre innloggingssider for å sammenligne denne applikasjonen med standardene.

5: Error prevention

- Applikasjonen har tilbakemeldinger til brukeren som er ment til å være forståelig, dersom man skriver inn feil brukernavn, eller passord.
- Brukeren vil få en beskjed dersom de ikke har skrudd på GPSen på mobilen sin.
- Dersom en bruker ikke gir applikasjonen tilgang til å bruke GPS vil brukeren få beskjed om at dette vil påvirke hvordan applikasjonen fungerer.
- Applikasjonen burde hindre en bruker fra å logge ut når de er i en kjøretur.

6: Recognition rather than recall

- Brukeren kan gi en tittel til en kjøretur slik at det blir enklere å huske hvilken kjøretur det var.
- Applikasjonen bruker bilder som er ment til å hjelpe brukeren å forstå hva som skal gjøres istedenfor å lese hva som skal gjøres.
- Brukeren kan sortere kjøreturene basert på forskjellige kategorier slik at de ikke trenger å huske informasjonen om sin kjøretur.
- Applikasjonen kunne hatt informasjonsbokser som forklarer hva en bruker skal gjøre.

7: Flexibility and efficiency of use

- I Historikk listen er det mulig å sortere rekkefølgen basert på informasjonen som ligger i kjøreturen f.eks(lengste tur, korteste tur, nyeste tur, eldste tur, osv)
- Det kunne vært mulig å la brukeren slette flere kjøreturer på en gang.
- Noen brukere kan foretrekke å legge inn en tittel til kjøreturen, før den blir lagret.
- Dersom en bruker har mange kjøreturer, vil det kunne være ønskelig med en søkefunksjon, som kan søke på tittel.

8: Aesthetic and minimalist design

- I denne applikasjonen har rådet "Get rid of half the words on each page, then get rid of half of what's left." fra Steve Krug blitt fulgt, med målet om å oppnå et mer minimalistisk design, som forsøker å ikke forvirre brukerne.
- I applikasjonen er det unngått å bruke fargerike elementer, som for eksempel bakgrunnen for å unngå å distrahere en bruker.

9: Help users recognize, diagnose, and recover from errors

- Applikasjonen sender en tilbakemelding som er basert på hvilken feilmelding som oppstår.
- Feilmeldingene blir vist til brukeren som en advarsels boks, slik at brukeren må trykke ok, for å fjerne denne, dette vil kunne gjøre at flere brukere velger å lese hva som står der.
- Feilmeldingene prøver å bruke et språk som er lett gjenkjennelig for folk flest.

10: Help and documentation

- Noen brukere vil kunne ønske en søkefunksjon, for å se igjennom sine kjøreturer.

Konklusjoner basert på heuristikken:

Noen av de enkle punktene i heuristikken var nødvendig funksjonalitet som vi valgte å implementere f.eks punktene:

- En bruker har ikke mulighet til å endre passord, uten å bruke glemt passord funksjonen.
- Applikasjonen burde hindre en bruker fra å logge ut når de er i en kjøretur.

Andre punkter inkluderer funksjonalitet som er greit å ha, men er ikke nødvendige i en første versjon av en slik oppgave f.eks:

- Søkefunksjon
- Multiselect av kjøreturer

4.5: Brukertester

Brukertester kan brukes til å teste applikasjonens brukervennlighet ved å opprette noen oppgaver som brukerne skal teste og gi tilbakemelding på, disse oppgavene kan gjerne være basert på punkter hentet fra heuristikken. Dette kan brukes til å bekrefte om problemene som ble oppdaget i heuristikken virkelig var et problem for en sluttbruker.

I følge Datatilsynet må man hente samtykke i et [samtykke skjema](#)²⁵ når man samler persondata. Samtykke skjemaene som ble benyttet for å utføre brukertestene har vi lagt ved som vedlegg til slutt i oppgaven.

Oppgaver:

Dette er oppgavene vi har valgt å bruke. Grunnen til dette er at dette er den viktigste funksjonaliteten som er implementert i applikasjonen, og det er viktig å vite om det er noe som kan bli forbedret.

- Opprett en bruker
- Ta en kjøretur, og se den i kartet
- Ta en kjøretur, og se om finne ut hvilken kjøretur var først
- Endre tittel på en kjøretur
- Slett en kjøretur

Tilbakemeldinger etter brukertesting:

Her er en liste av noen av tilbakemeldingene som ble sanket inn under de fem brukertester vi utførte med oppgavene over som gjøremål:

- Det står ingen steder i applikasjonen hvilken funksjonalitet som eksisterer, dette er noe som kunne blitt inkludert i en applikasjonsside dersom applikasjonen skulle blitt lagt til i en app store.
- Brukerne hadde ingen problemer med å finne frem til hvordan de skulle opprette en bruker, og hvordan de skulle logge inn på en bruker.
- Noe utydelig hvordan man kan endre tittel på en kjøretur man har utført. Denne tilbakemeldingen førte til at tittelfeltet ble forsøkt å gjort mer åpenbart.
- Det var enkelt for brukerne i testen å se hvilken kjøretur som var først, og å finne ut informasjon om kjøreturen de var ute etter.
- Brukerne fant raskt ut hvordan de skulle se en kjøretur i kartet.
- En bruker gav tilbakemelding om at de ønsket å inkludere start og slutt tid på en kjøretur. Dette ble inkludert etter vi fikk høre om det.

4.6: Resultat Informasjonsarkitekturanalyser

I kapittel 4.3: Heuristikk fikk vi fastslått at flere ting allerede var bra som for eksempel det minimalistiske designet, og at noen ting kunne gjøres bedre som å gjøre feilmeldingene mer tydelig for brukeren.

²⁵

<https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/behandlingsgrunnlag/veileder-om-behandlingsgrunnlag/samtykke/>

Under brukertestene i kapittel 4.4 ble brukervennligheten testet av flere brukere, og som resultat fikk vi noen tilbakemeldinger og noe informasjon om hva som var bra og hva som kunne gjøres bedre.

Ved å arbeide med Informasjonsarkitekturen parallelt med designprosessen har det ført til mindre dobbeltarbeid, ved at ting ble gjort med tanke på hvilken prosess vi var kommet til i informasjonsarkitekturen.

Kapittel 5: Informasjonssikkerhet

I dette kapitlet skal gå igjennom:

- Verdivurdering
- Trusselmodellering
- Risikoanalyse
- Tiltaksliste
- Resultatet

5.1: Verdivurdering

I en [verdivurdering](#) ²⁶ kan man kartlegge verdier som er ønskelig å beskytte og under skal gruppen vise et eksempel på hvordan dette kan gjøres ved å foreta en verdivurdering av applikasjonen som er laget.

Kartlagte verdier for applikasjonen

Verdiene i vår applikasjon:

- E-post adresse
- Passord
- Administrator tillatelser i Firebase databasen
- Telefon nummer
- Navn
- Kjøreturer med gps lokasjon

5.2: Trusselmodellering

I [trusselmodelleringen](#) ²⁷ har man brukt metoden hvor man ser for seg mulige måter ting kan gå galt på, og hva man ikke ønsker skal skje, og så tenke hvordan det allikevel kan skje. Her kan man trekke fram forutsetningene som må brytes og hvordan det kan gjøres.

²⁶

https://www.digdir.no/informasjonssikkerhet/begrepsliste/3230#identifisering_av_oppgaver_og_informasjonstyper_verdivurdering

²⁷

<https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/informasjonssikkerhet-internkontroll/risikovurdering/>

Hvor mange punkter man kommer på avhenger av hvor mange scenarioer som er mulige og hvor mye tid man ønsker å bruke på dette., og burde bli utført i en idemyldringssesjon med relevant fagpersonell og ledere for forskjellige perspektiver av programvaren, eller målet for trusselmodelleringen.

En ting vi har valgt å legge fokuset på i informasjonsikkerhetsarbeidet er [STRIDE](#)²⁸ STRIDE er en liste over forskjellige angrepstyper laget av Microsoft, som er et akronym for hvordan trusselaktører kan forsøke å ta seg inn i sikkerhetssystemer og dette sier SINTEF om STRIDE:

- **Spoofing** – en angriper gir seg ut for å være en annen bruker, komponent eller system
- **Tampering** – en angriper modifierer data
- **Repudiation** – en angriper (eller annen aktør) kan nekte å ha utført en handling eller sendt en melding hvis systemet ikke har tilstrekkelige mekanismer eller bevismateriale
- **Information disclosure** – en angriper kan få leseaksess til beskyttede data
- **Denial of Service (DoS)** – en angriper kan hindre legitime brukere i å bruke den vanlige funksjonaliteten til systemet
- **Elevation of privilege** – en angriper bruker ulovlige midler for å nå et tilgangsnivå med andre privilegier enn han har i øyeblikket

I denne sammenhengen navngir man hvert punkt med “H” og et nummer inkrementert for hver hendelse for å kunne benytte disse i en ROS-analyse senere i dette kapitlet. Under listes hva som kan tenkes å være problemer i denne spesifikke applikasjonen med tanke på sikkerhet og bruker punkter som ble definert i verdivurderingen:

- H1: E-post adressen skal ikke komme på avveie, dette kan skje dersom:
 - En bruker frivillig gir fra seg informasjonen til uvedkomne i forbindelse med social engineering angrep
 - E-post adressen blir hentet ut fra en liste over kompromitterte E-post adresser
 - Noen får tilgang til databasen
- H2: Passordet skal ikke komme på avveie, dette kan skje dersom:
 - En bruker frivillig gir fra seg informasjonen til uvedkomne i forbindelse med social engineering angrep
 - En bruker benytter likt passord på flere tjenester
 - Passordet blir hentet ut fra en liste over kompromitterte passord
 - Noen får tilgang til databasen og benytter denne til å hente ut passord
- H3: Telefonnummer kan komme på avveie, dette kan skje dersom:
 - Noen får tilgang til databasen
 - En feil i applikasjonen
- H4: Brukere skal ikke ha tilgang til annen data enn sin egen, dette kan skje dersom:

²⁸ <https://infosec.sintef.no/informasjonssikkerhet/2018/06/trusselmodellering/>

- Utvilkerne ikke tar høyde for å validere brukere mot databasen.
- En feil kan gjøre det mulig å logge inn på en bruker som ikke er din.
- H5: En kjøretur kan bli ufrivillig slettet, dette kan skje dersom:
 - Dersom det oppstår en feil i databasen/applikasjonen
 - Dersom brukerkontoen kommer på avveie
 - Brukeren kan slette den med et uhell

5.3: Risikoanalyse

Vi har valgt å ta i bruk en [ROS analyse](#)²⁹ for å kartlegge hva som skal vektlegges for å forhindre hendelsene som ble beskrevet i kapittel 5.4.

Her tar man konsekvensen og sannsynligheten for å forme risikoen en gitt hendelse “H” har for å inntreffe, og hva som prioriteres i sikkerhetsarbeidet for å forhindre disse.

Konsekvenser av hendelsene i trusselmodelleringen

I denne delen av kapitlet klargjør gruppen hva som er mulige konsekvenser av at hendelsene som ble definert i 5.2 faktisk forekommer, og hvilken gradering disse konsekvensene har.

H1: E-post adressen skal ikke komme på avveie.

Konsekvenser om denne hendelsen forekommer kan være:

- En e-post adresse på avveie kan brukes til å sende spam, og andre svindelforsøk til brukeren
- Dersom e-postadressen kommer på avveie, kan angripere benytte e-postlister over tidligere kombinasjoner av passord og e-poster som har vært utsatt for angrep, og via dette forsøke å komme seg inn på brukerens konto.
- Dette kan brukes til social engineering, som for eksempel å ta kontakt med brukeren via e-post korrespondanser og utgi seg for å være noen som er knyttet til applikasjonen, og via dette få hentet ut innloggingsinformasjonen. Dette kalles [“Spoofing”](#)³⁰
-

H1 Konsekvensgrad: Lav konsekvens

H2: Passordet skal ikke komme på avveie.

Konsekvenser om denne hendelsen forekommer kan være:

- Samme som på e-post, så kan trusselaktører benytte passordet til å sammenligne med passordlister, og forsøke å finne en match for å ta seg inn i systemet. Denne blir vurdert til ett hakk høyere konsekvensgrad fordi noen brukere kan bruke ett passord flere plasser.

H2 Konsekvensgrad: Moderat konsekvens

H3: Telefonnummer kan komme på avveie, og knyttes til applikasjonen

Konsekvenser om denne hendelsen forekommer kan være:

²⁹ <https://www.helsebiblioteket.no/kvalitetsforbedring/metoder-og-verktoy/ros-analyse>

³⁰ <https://www.telenor.no/sikkerhet/faq/hva-er-spoofing/>

- Dette kan brukes til social engineering, som for eksempel å ta kontakt med brukeren via telefonsamtaler, eller SMS og utgi seg for å være noen som er knyttet til applikasjonen, og via dette få hentet ut innloggingsinformasjonen.

H3 Konsekvensgrad: Lav konsekvens

H4: Brukere skal ikke ha tilgang til annen data enn sin egen.

Konsekvenser om denne hendelsen forekommer kan være:

- Personlig data leses av uvedkomne.
- Data i applikasjonen kan slettes av uvedkomne
- Dersom en adminbruker er utsatt kan dette ha som følge at funksjonellstruktur blir utsatt for sabotasje om uvedkomne har tilgang.
- Straffeløp for utviklerne for brudd av personopplysningsloven og GDPR

H4 Konsekvensgrad: Svært Høy konsekvens

H5: En kjøretur kan bli ufrivillig slettet

Konsekvenser om denne hendelsen forekommer kan være:

- Brukeren kan miste retten til å få tilbakebetalt utgiftene for en kjøreturen
- En bruker kan velge å gå over til en konkurrerende applikasjon.

H5 Konsekvensgrad: Moderat konsekvens

Sannsynligheten for hendelsene i trusselmodelleringen

For å anta sannsynlighetene for at noen av hendelsene faktisk skjer, har vi benyttet “[Nasjonalt Digitalt Sikkerhetsbilde 2021](#)”³¹ fra [NSM](#)³² hvor det nevnes at norske virksomheter har i større grad blitt utsatt for phishing angrep og svindelforsøk, og dermed lagt vekt på dette i vurderingene.

H1: E-post adressen skal ikke komme på avveie.

Sannsynligheten for at denne hendelsen forekommer:

- På grunn av at ansvaret her hovedsakelig ligger hos brukerne er sannsynligheten avhengig av hvordan brukerne selv hemmeligholder sin informasjon. Det kan derfor antas at for noen brukere er dette en svært høy sannsynlighet.

H1 Sannsynlighetsgrad: Svært Høy Sannsynlighet

H2: Passordet skal ikke komme på avveie.

Sannsynligheten for at denne hendelsen forekommer:

- Slik som i hendelse H1 er sannsynligheten av denne hendelsen basert på brukerens behandling av sikker informasjon, det kan derfor antas at sannsynligheten for denne hendelsen er høy. Begrunnelsen for at denne sannsynligheten er lavere enn e-post, er at mange brukere vet at de ikke bør gi bort sine passord.

H2 Sannsynlighetsgrad: Høy Sannsynlighet

³¹ <https://nsm.no/aktuelt/nasjonalt-digitalt-risikobilde-2021>

³² <https://nsm.no/>

H3: Telefonnummer kan komme på avveie, og knyttes til applikasjonen

Sannsynligheten for at denne hendelsen forekommer:

- Ettersom denne applikasjonen ikke displayer brukerens telefonnummer, vil denne sannsynligheten også være knyttet til brukervaner, ettersom sammenhengen mellom telefonnummer og vår applikasjon er liten, kan det tenkes at denne sannsynligheten er moderat.

H3 Sannsynlighetsgrad: Moderat Sannsynlighet

H4: Brukere skal ikke ha tilgang til annen data enn sin egen.

Sannsynligheten for at denne hendelsen forekommer:

- En av grunnene til at gruppen valgte å benytte Firebase er kontrollen utviklerne får over databasen, og dette gjør at feil basert på vår kode kan reduseres, og det er Google som utviklerne av Firebase som må ha en feil i koden sin. Dette gjør at sannsynligheten for denne hendelsen vurderes som lav.

H4: Sannsynlighetsgrad: Lav Sannsynlighet

H5: En kjøretur kan bli ufrivillig slettet

Sannsynligheten for at denne hendelsen forekommer:

- Den eneste måten for en bruker å slette en kjøretur på er dersom en bruker trykker spesifikt på slett kjøretur, dette kan derfor ansees som lav sannsynlighet

H5: Sannsynlighetsgrad: Lav Sannsynlighet

Tabellen for ROS-analysen

For å sette inn hendelsene i tabellen har man her brukt konsekvensgraden og sannsynlighetsgraden

| | Lav Konsekvens | Moderat Konsekvens | Høy Konsekvens | Svært Høy Konsekvens |
|-------------------------|----------------|--------------------|----------------|----------------------|
| Lav Sannsynlighet | | H5 | | H4 |
| Moderat Sannsynlighet | H3 | H2 | | |
| Høy Sannsynlighet | H1 | | | |
| Svært Høy Sannsynlighet | | | | |

Konsekvens * Sannsynlighet = Risiko

Grønn: Ingen tiltak nødvendig

Gul: Fortsatt akseptabel risiko

Rød: Bør håndteres med mottiltak

Konklusjon basert på ROS-analysen

5.4: Tiltaksliste og prioritering

I denne delen har vi listet opp eventuelle tiltak basert på ROS-analysen, og beskrevet disse under en hendelse der de reduserer konsekvensene eller sannsynligheten. Dersom et tiltak fungerer på flere hendelser kan dette også listes opp flere ganger i de hendelsene der hvor det er relevant. Vi bruker Tx for å referere til tiltakene som blir laget, hvor "x" blir inkrementert for hvert tiltak.

Det finnes også verktøy man kan bruke som hjelpemiddel i å utføre analysene vi har foretatt i løpet av dette kapittelet. Ett eksempel er Visure Solutions sitt [program](#)³³

H1: E-post adressen skal ikke komme på avveie.

- T1: Skjule alle forekomster av e-post adresse
- T2: Advare brukeren om hva som kan skje dersom de gir bort e-postadressen

H2: Passordet skal ikke komme på avveie.

- T3: Sette krav for hvilken struktur passord skal ha (Stor bokstav, lengde, spesialtegn)
- T4: Skjule passordfelt
- T5: Sette opp 2-faktor-autentisering

H3: Telefonnummer kan komme på avveie, og knyttes til applikasjonen

- T6 Ikke vis telefonnummer i applikasjonen

H4: Brukere skal ikke ha tilgang til annen data enn sin egen.

- T3: Sette krav for hvilken struktur passord skal ha (Stor bokstav, lengde, spesialtegn)
- T5: Sette opp 2-faktor-autentisering
- T7: Validere brukere mot databasen med unik identifikator, slik at brukerne kun ser sitt eget innhold

H5: En kjøretur kan bli ufrivillig slettet

- T3: Sette krav for hvilken struktur passord skal ha (Stor bokstav, lengde, spesialtegn)
- T5: Sette opp 2-faktor-autentisering
- T7: Validere brukere mot databasen med unik identifikator, slik at brukerne kun ser sitt eget innhold
- T8 : Legg inn en sjekk når en bruker forsøker å slette en kjøretur.
- T9: Fjern muligheten for å la brukeren slette kjøreturer.

Kostnad for tiltakene

I denne seksjonen vil vi se på hvilke kostnader de forskjellige tiltakene blir anslått å ha i utviklingsfasen. Siden denne bachelor oppgaven ikke inkluderer penger, så vil tid benyttes som måleenheten for kostnadene, og sette dette opp i en tabell for å få en visuell oversikt

³³ <https://visuresolutions.com/no/programvare-for-risikostyring/>

over de billigste tiltakene, og de dyreste. Senere kombineres dette med hvilken effekt tiltakene har, og så konkluderer vi hvilke som bør utføres.

T1: Skjule alle forekomster av e-post adresse

- E-postadressen vil i denne applikasjonen bli brukt som et brukernavn, som hjelper brukere å se hvilken brukerkonto de er pålogget, fjerning av dette vil derfor ha en høy kostnad.

T1 Kostnad: Høy Kostnad

T2: Advare brukeren om hva som kan skje dersom de gir bort e-postadressen

- Dette kan gjøres ved hjelp av en alert, eller modal. Det som tar tid i dette tiltaket ville være å lage teksten brukeren får se. Denne får derfor en moderat kostnad

T2 Kostnad: Moderat

T3: Sette krav for hvilken struktur passord skal ha (Stor bokstav, lengde, spesialtegn)

- Dette gjøres uten mye problemer, derfor er kostnaden lav.

T3 Kostnad: Lav Kostnad

T4: Skjule passordfelt

- Dette gjøres enkelt i koden ved å legge til et parameter, dette vil derfor ha en lav kostnad.

T4 Kostnad: Lav Kostnad

T5: Sette opp 2-faktor-autentisering

- Dette kan settes opp ved å bruke funksjonalitet som allerede er laget, men vil ta litt tid å få til. Kostnaden er derfor høy.

T5 Kostnad: Høy Kostnad

T6: ikke vis telefonnummer i applikasjonen

- Telefonnummeret vil ikke bli brukt til noe annet en 2 faktor autentisering, og er derfor ikke nødvendig å vise i applikasjonen. Kostnaden er derfor lav

T6 Kostnad: Lav Kostnad

T7: Validere brukere mot databasen med unik identifikator, slik at brukerne kun ser sitt eget innhold

- Dette gjøres enkelt ved å bruke e-post i en spørring mot databasen.

T7 Kostnad: Lav Kostnad

T8: Legg inn en sjekk når en bruker forsøker å slette en kjøretur.

- En alert kan enkelt legges inn, og spørre brukeren om den er sikker på at den vil slette kjøreturen, dette er derfor en lav kostnad.

T8 Kostnad: Lav Kostnad

T9: Fjern muligheten for å la brukeren slette kjøreturer.

- Å utføre dette ville vært relativt raskt å gjøre dette, men dette ville ha en ekstra kostnad ved at applikasjonen har mindre funksjonalitet. Vurderes derfor til Høy kostnad

T9 Kostnad: Høy Kostnad

| | Lav Kostnad | Moderat Kostnad | Høy Kostnad |
|----|-------------|-----------------|-------------|
| T1 | | | H |
| T2 | | M | |
| T3 | L | | |
| T4 | L | | |
| T5 | | | H |
| T6 | L | | |
| T7 | L | | |
| T8 | L | | |
| T9 | | | H |

Oversikt over effektiviteten tiltakene har totalt sett

Her brukes tiltakene som ble laget tidligere i dette kapittelet, for sjekke på en visuell måte om noen av tiltakene også reduserer risikoen for andre hendelser. I motsetning til de andre tabellene som har blitt laget hittil, så vil denne gå fra høy effektivitet til lav effektivitet, og det betyr at det som er positivt her er Høy, og negativt er Lav.

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| H1 | M | L | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| H2 | N/A | N/A | H | M | N/A | N/A | N/A | N/A | N/A |
| H3 | N/A | N/A | N/A | N/A | N/A | H | N/A | N/A | N/A |
| H4 | N/A | N/A | L | N/A | H | N/A | H | N/A | N/A |
| H5 | N/A | N/A | L | N/A | H | N/A | H | H | H |

5.5 Resultat fra Informasjonssikkerhetanalyser

Basert på analysen som har blitt gjort i løpet av dette kapittelet har vi valgt å anse disse tiltakene som de mest gjennomførbare fordi de har den sammenlagte største verdien når man tar i betraktning kostnaden og effektiviteten på hendelsene:

- T3: Sette krav for hvilken struktur passord skal ha.
- T4: Skjule passordfelt

- T5: Sette opp 2-faktor-autentisering
- T6: Ikke vis telefonnummer i applikasjonen
- T7: Validere brukere mot databasen med unik identifaktor
- T8: Legg inn en sjekk når en bruker forsøker å slette en kjøretur

Kapittel 6: Programmering

I denne delen av prosjektet skal vi forklare hvordan utviklingen av denne applikasjonen gikk, og hvilke valg som ble tatt.

6.1: Prosjekt Oppstart

Gruppen hadde ikke programmert i React Native tidligere, av den grunn ble det bestemt at det skulle lages en rask prototype, for å sjekke at de utvalgte rammeverkene og bibliotekene fungerte sammen.

Prototypen var en liten del av applikasjonen hvor det var mulig å opprette en bruker, og logge inn som denne brukeren. Ettersom gruppen ikke hadde brukt React Native tidligere ble det benyttet en guide for å sette opp starten av prosjektet. Vi benyttet oss av denne videoen til formålet: [programming with mosh](#) ³⁴.

Firestore oppsett

Ettersom gruppen ikke hadde noen forkunnskaper i bruk av firestore, var det nødvendig med veiledning på hvordan sammenkoblingen mellom firestore og dette prosjektet skulle fungere. For å løse dette ble det brukt en video guide: [React Native authentication with expo](#) ³⁵. Etter vi hadde fått satt opp et prosjekt, så ble det mulig for gruppen å kunne bruke dokumentasjonen: <https://firebase.google.com/docs/auth/where-to-start> ³⁶ for å forstå hvilke funksjonaliteter som eksisterer, og hvordan disse kan brukes.

Videre utvikling av prototype

I oppstartsprosessen av prosjektet ble det planlagt hvilke sider, og hvilken funksjonalitet som var nødvendig å ha på forhånd ved hjelp av møter med Adcom. Slik at det var mulig å begynne rett på utviklingen og sette opp en prototype.

Dermed fikk gruppen satt opp en prototype som inneholdt:

- Innlogging

³⁴ <https://www.youtube.com/watch?v=0-S5a0eXPoc>

³⁵ <https://www.youtube.com/watch?v=gl4J6SpLXZA&t=353s>

³⁶ <https://firebase.google.com/docs/auth/where-to-start>

- Glemt Passord
- Registrering

Deretter ble dette vist frem til adcom som et forslag på hvordan Innloggingen skulle se ut og fungere.

Styling

For styling av React-Native applikasjoner brukes et stylesheet som ligner på [CSS](#)³⁷. Denne stylingentypen heter “[Styled Components](#)”³⁸.

Hovedsaklig ble stylingen gjort under funksjonen som opprettet siden, men ettersom det finnes mange sider som gjenbraker deler fra andre sider, har gruppen valgt å sette opp stylingen i separate filer som importeres til sidene der den blir brukt.

For å følge vår egen standard ble det bestemt at komponenter som ikke trenger å være unike skal bruke den samme stylingen som lignende komponenter som er brukt tidligere. Dette gjør at brukere enklere kan bli kjent med applikasjonen og ikke trenge å tenke over hva de skal gjøre. Dette er informasjon vi fant i kapittel 4.

Registrering

Registreringen var hovedsakelig satt opp med bare ett felt for e-post, og ett felt for passord men, under testing, og gjennomgang av forskjellige andre innlogging systemer ble det bestemt at brukeren skal bli bedt om å bekrefte passordet, slik at brukeren ikke skal kunne skrive feil passord.

For at en bruker skal vite hva som skjer i applikasjonen, ble det bestemt at programmet skal gi brukeren grundig, og detaljerte feilmeldinger dersom en feil oppstår. For eksempel om brukeren har skrevet inn en epostadresse med feil format, vil brukeren få en feilmelding om dette.

Når en bruker har skrevet inn feltene riktig og trykket på registrerings knappen, vil applikasjonen kjøre en funksjon fra firebase som heter `signupwithemailandpassword`, som sender denne informasjonen til firebase og oppretter brukeren.

I et forsøk på å gjøre applikasjonen mer brukervennlig, ble det bestemt at når en bruker registrerer seg, skal den bli automatisk logget inn på denne brukeren etter opprettelsen er ferdig. Dette blir utført ved å sende epost og passord videre til funksjonen for innlogging.

³⁷ https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS

³⁸ <https://styled-components.com/>

Innlogging

Ved oppsettet av innloggingssiden var det mulig å bruke opp igjen mange av komponentene som ble brukt i registreringen, ettersom disse sidene er forholdsvis like.

Slik som i registrering ble det også inkludert feilmeldinger for brukeren i denne delen av prosjektet, hvor mye av de samme feilmeldingene som ble brukt i Registrering kunne brukes på nytt på denne siden.

Når brukeren prøver å logge inn vil applikasjonens logikk først sjekke om alle felter er riktig fylt inn, før det blir sjekket om brukernavn og passord stemmer. Brukernavn og passord blir sjekket igjennom en innebygd funksjon i firebase som heter `signinwithemailandpassword`.

Dersom brukeren har skrevet inn feil e-postadresse eller passord, vil applikasjonen gi en tilbakemelding som sier "brukernavn eller passord er feil". Grunnen til at denne tilbakemeldingen ikke er mer spesifikk er slik at det ikke skal være mulig å sjekke opp mange e postadresser for å sjekke hvilke brukere som er registrert.

Glemt passord

Det kan være vanskelig for brukere å huske ting som passord, dette gjør at det er lurt for en applikasjon å inkludere en funksjon for glemt passord, slik at en bruker fortsatt skal ha mulighet til å få tilgang til brukerkontoen uansett om de har glemt passordet.

Firestore har allerede en funksjon som kan håndtere dette, dette gjorde det enklere å sette opp et inntastingsfelt for e-post, og en knapp som kunne kjøre en funksjon for gjenoppretting av glemt passord.

I Firestore kan man endre hvordan e-posten skal se ut, hvilken adresse den er sendt fra og hva som skal være emnet.

Denne e-posten blir sendt via firestore, fra den e-posten som blir spesifisert i firestore til e-postadressen brukeren skrev inn.

Dersom brukeren har skrevet inn en epostadresse med feil format, eller ikke har skrevet noe i det hele tatt, vil brukeren få en feilmelding som forklarer hva som har skjedd.

Kommentering

Via tidligere erfaringer med prosjektarbeid er det blitt erfart at dersom koden ikke har gode kommentarer vil det være vanskelig å forstå hva som er gjort dersom noen andre skal se koden, eller når man skal fikse kode som ikke er helt nyskrevet.

Gruppen har derfor valgt å være ekstra nøye med å kommentere så mye som er nødvendig for at det skal være enkelt å se hvilke deler av koden som gjør hva, og hvordan de forskjellige kodesnuttene henger sammen.

Refaktorering

Fordi gruppen ikke hadde noen spesiell erfaring med React Native tidligere, ble det gjort noen feil tidlig i prosjektet som gjorde det nødvendig å refaktorere en stor del av prosjektet ganske tidlig.

[Refaktorering](#)³⁹ er når man går over kode som er blitt skrevet for å se om man kan redusere den på punkter ved å bruke opp ting man har laget fra før, istedet for å lage en ny ting som gjør omtrent det samme som kode man allerede har, i vårt tilfelle gjelder dette å gjenbruke komponenter som allerede er laget til et tidligere formål.

I starten i navigasjonen ble det brukt en noe utdatert måte å navigere mellom de forskjellige sidene. det hadde blitt brukt noe som heter [Modal](#)⁴⁰, men dette gjorde at mye av koden måtte ligge i den samme filen for at det skulle fungere. Dette gjorde at det ble vanskeligere enn nødvendig å finne frem noe i koden, og mye rot i selve prosjektet.

Gruppen fant derfor ut at man skulle prøve å finne en bedre måte å navigere mellom sidene slik at man kunne fikse på oppsettet av koden, og få den mer oversiktlig og lesbar.

Derfor leste gruppen seg opp på noe som heter [React-Native-Navigation](#)⁴¹ som ble valgt å implementere. Etter vi hadde satt opp de forskjellige sidene i React-Native-Navigation ble det enkelt å dele opp koden vi hadde i forskjellige "screens", hvor en skjerm kun trenger å inneholde koden fra den spesifikke siden for å fungere, dette gjorde at man lett kunne dele opp koden i de skjermene brukeren skulle navigere til, og koden ble mye mer oversiktlig.

6.2: Videreføring av prosjektet

Etter at gruppen hadde opprettet en fungerende prototype, vist frem denne og fått tilbakemeldinger fra oppdragsgiver, var vi klare til å begynne på den neste delen av prosjektet hvor hoveddelen av prosjektet skal utvikles.

Hovedsiden

For å gjøre det enklere for en bruker å finne frem til de forskjellige tingene som ligger i applikasjonen, har vi valgt å følge noe som er regnet som en standard for mange applikasjoner

Denne standarden går ut på å ha en hovedside hvor du kan navigere til alle de viktigste sidene som finnes på applikasjonen.

³⁹ <https://www.agilealliance.org/glossary/refactoring/>

⁴⁰ <https://reactnative.dev/docs/modal>

⁴¹ <https://reactnavigation.org/>

På hovedsiden har brukeren 4 forskjellige alternativer han kan gjøre, forklaring på alternativene kommer senere.

- 1) Navigere til kjøring
- 2) Navigere i Historikk
- 3) Logge ut av brukeren sin
- 4) Åpne innstillinger

For å få tilgang til hovedsiden i vår applikasjon må brukeren bruke innloggingsfunksjon, og blir deretter navigert videre til hovedsiden, dersom brukeren allerede er innlogget blir den automatisk videresendt til hovedsiden ved åpning av applikasjonen.

Innstillinger

Basert på informasjonssikkerhet kapitlet kom det frem at det var nødvendig med en side hvor bruken hadde mulighet til å:

- 1) Endre passord
- 2) Aktivere / Deaktivere 2 faktor autentisering

Endre passord

For å gjøre det enklere for brukeren å endre passord, samtidig som det skulle bli en lavere utviklingskostnad, ble det gjort en avgjørelse om å gjenbruke den samme funksjonaliteten som var brukt i glemt passord, hvor brukeren vil få en e-post der de kan trykke seg inn på en link for å endre passordet sitt.

Forskjellen på denne siden og på glemt passord siden er at ettersom brukeren allerede er pålogget med en epostadresse er det ikke nødvendig å spørre brukeren hvilken epostadresse som tilhører brukeren, men sender en epost direkte.

Aktivere 2 faktor autentisering

Under utviklingen av denne funksjonaliteten var det nødvendig med noe planlegging av hvordan dette skulle fungere.

Gruppen ble enig om at dersom en bruker allerede har satt på 2 faktor autentisering, skulle den samme knappen de brukte for å skru det på, brukes til å skru dette av.

Gruppen ble enig om at dersom brukeren allerede hadde aktivert 2 faktor autentisering var det ikke nødvendig å vise knappen som aktiverer 2 faktor autentisering, og dersom brukeren ikke hadde aktivert 2 faktor var det ikke nødvendig å vise knappen som deaktiverer 2 faktor autentisering. Innstillinger siden bruker derfor noe som heter [conditional rendering](https://reactjs.org/docs/conditional-rendering.html)⁴² for å bestemme hvilken av disse knappene som skal vises.

Dette betyr at dersom et telefonnummer er knyttet til brukerkontoen vil alternativet for å skru av 2 faktor autentisering vises for brukeren, men dersom brukeren ikke har knyttet et telefonnummer til brukerkontoen vil knappen for å aktivere 2 faktor autentisering vises.

⁴² <https://reactjs.org/docs/conditional-rendering.html>

Når brukeren skruer på 2 faktor autentisering vil de få opp en tekstboks hvor de kan skrive inn sitt telefonnummer. Etter at brukeren har bekreftet sitt telefonnummer vil en sms bli sendt til dette telefonnummeret.

Denne smsen inneholder en kode som brukeren skal skrive inn i en annen tekstboks. Når koden er bekreftet blir dette telefonnummeret lagt til deres bruker, og denne samme prosessen vil da bli brukt under innlogging fra det punktet.

Dersom brukeren ikke lenger ønsker å ha 2 faktor autentisering aktivert kan de gå inn på innstillinger og klikke på "deaktiver 2 faktor autentisering". Dette vil fjerne telefonnummeret fra bruker kontoen, slik at denne prosessen ikke lenger blir inkludert under innlogging.

Kjøring

Kjøring siden er hvor mesteparten av programmets kode vil kjøres. Denne siden er en enkel side med en Ekbok logo, og en startknapp som blir gjort om til en stoppknapp når den blir klikket på. Dette skjer ved en conditional rendering i react-native som nevnt tidligere gjør at dersom noen spesielle forhold blir møtt vil en annen knapp, som bruker samme styling erstatte start knappen som ligger der når du ikke har startet applikasjonen.

De to forskjellige knappene har forskjellige onClick metoder som blir kjørt avhengig av hvilken knapp du trykker på.

Startknapp

Når du trykker på startknappen vil programmet først begynne å telle hvor mange sekunder programmet har kjørt. Deretter vil brukeren få en forespørsel om samtykke for å bruke mobiltelefonens GPS for å kunne sjekke brukerens posisjon. Dersom brukeren ikke godtar dette vil funksjonen bli stoppet, og brukeren kan ikke bruke denne applikasjonen uten å godta, fordi hele applikasjonen er basert på at vi kan hente brukerens GPS lokasjon for å kunne vite hvor langt de har kjørt etter telleren har startet.

Dersom brukeren aksepterer samtykke forespørselen vil en funksjon starte. Denne funksjonen lagrer når kjøreturer er blitt startet, hvor brukeren var når funksjonen ble startet og begynner å ta brukerens GPS lokasjon hvert andre sekund. Denne informasjonen vil bli lagret i en dictionary som inneholder:

- Navn
- Tidsbruk
- Avstand
- Tittel
- Koordinater
- Start Klokkeslett
- Slutt Klokkeslett
- Startsted
- Sluttsted

Når brukeren trykker på stoppknappen vil start funksjonen bli stoppet, og en stop funksjon vil bli kjørt. Denne funksjonen fyller inn de forskjellige feltene i et dictionary som ble nevnt tidligere, og lagrer dette i databasen.

Her er en forklaring på hva som blir fylt inn i punktene i en dictionary:

- Navn - Programmet plukker ut brukerens e-postadresse og fyller inn denne
- Tidsbruk - Det blir tatt et nytt tidspunkt, og start tidspunktet vil bli trukket ifra dette for å finne ut hvor lang tid kjøreturen tok.
- Tittel - Her fylles ingenting nå, men dette feltet kan endres av brukeren senere.
- Start Klokkeslett - Klokkeslettet kjøreturen ble startet.
- Slutt Klokkeslett - Klokkeslettet kjøreturen ble stoppet.

Avstand

For å regne ut dette har det blitt brukt en formel som heter [haversine formula](#) ⁴³

For inspirasjon til å løse denne delen av oppgaven ble [denne](#) ⁴⁴ siden brukt.

Startsted / Sluttsted

For å finne ut hvilket sted brukeren har startet eller stoppet, ble det brukt en funksjonalitet som heter [reverse geolocation](#) ⁴⁵. Denne funksjonaliteten bruker koordinatpunkter og finner ut hva dette stedet heter.

Koordinater

Koordinater vil bli hentet via mobilens GPS og deretter bli lagret i denne listen.

For å lagre informasjonen vi har funnet ut, bruker vi firebase sin firestore. Firestore er en noSql database, dette gjør at det er mulig å lagre mange flere typer verdier, for eksempel når vi skal lagre koordinat punktene hadde vi mulighet til å lagre alle sammen i en liste, uten at dette krever noe spesielt oppsett, derfor er lagringen av denne informasjonen ganske enkel, og blir satt direkte inn i databasen.

Stoppknapp

Utviklingen av stoppknappen hadde noen problemer, dette er fordi gruppen testet en del forskjellige ting under utviklingen, som f.eks dersom brukeren går ut av siden som inneholder knappene ville variablene som bestemmer hvilken knapp som skal vises, bli resatt.

Å løse dette problemet viste seg å ikke være så lett, gruppen prøvde mange forskjellige måter for å beholde state på variabelen, og til slutt bestemte man seg for å lage et [spørsmål](#) ⁴⁶ på stack overflow.

Gruppen fikk noen svar på stack overflow, men ingen som direkte løste problemet vårt, men svarene var nyttige til å finne en løsning som fungerte.

⁴³ https://en.wikipedia.org/wiki/Haversine_formula

⁴⁴ <https://www.movable-type.co.uk/scripts/latlong.html>

⁴⁵ <https://docs.expo.dev/versions/latest/sdk/location/>

⁴⁶

<https://stackoverflow.com/questions/70772288/how-do-i-run-the-setstate-function-from-another-screen-in-react-native-navigatio>

Løsningen ble å sette opp funksjonen i en annen fil, slik at denne ikke ble kjørt på nytt dersom siden ble åpnet på nytt.

Når stopp knappen blir trykket, stopper start funksjonen, slik at data kan bli lagret riktig i en variabel, som da kan bli sendt videre til firebase databasen.

Historikk

Når brukeren åpner Historikk siden vil den få en oversikt over alle tidligere kjøretider for deres brukerkonto. Disse vil være sortert med nyeste først, men brukeren har mulighet til å kunne sortere de på andre måter, som f.eks: eldste først, korteste først osv.

Kjøreturene vises som en liste av knapper som brukeren kan trykke på for å navigere seg videre til en underside om den spesifikke kjøreturen.

For å enklere kunne se hvilken kjøretur som er hvilken blir det vist frem informasjon om kjøreturen på knappen, hvor kjøreturens tittel blir brukt som en tittel.

Historikk Underside

Dette er en side som tar imot informasjonen fra den spesifikke kjøreturen brukeren har trykket på, hvor brukeren har tilgang å behandle informasjonen på forskjellige måter som å:

- Se kjøreturen i kartet
- Få tydeligere informasjon om den spesifikke kjøreturen
- Slette kjøreturen
- Navngi kjøreturen

Kartside

Dersom brukeren ønsker å se den valgte kjøreturen i kartet, vil en side med et kart åpnes. Denne siden er satt til å sentrere kartet på det første koordinat punktet som blir tatt under kjøreturen. Brukeren kan så velge å zoome inn eller ut, eller flytte på kartet. På selve kartet blir det tegnet en linje mellom alle koordinat punktene slik at det skal være synlig for brukeren å se hvilken rute den spesifikke kjøreturen tok.

Slette kjøreturen

Dersom brukeren ikke lenger ønsker å beholde en spesifikk kjøretur, er det mulig å slette denne. For å slette kjøreturen er det bare å klikke seg inn på den spesifikke kjøreturen og klikke på slett kjøretur. For å unngå at en brukere sletter en kjøretur ved et uhell, er det lagt inn en spørre boks som spør brukeren om den er helt sikker på at den vil slette denne kjøreturen.

Navngi kjøreturen

En kjøretur vil automatisk bruke brukerens e-postadresse som en tittel, men dersom brukeren ønsker dette er det mulig å endre navnet på en kjøretur, dette er implementert slik at en bruker skal ha mulighet til å gi en liten beskrivelse på kjøreturen sin.

For å navngi en kjøretur må brukeren gå inn på siden for den spesifikke kjøreturen, hvor det ligger et felt hvor brukeren kan skrive inn den ønskede beskrivelsen, for så å klikke lagre.

Utlogging

På hovedsiden kan en bruker logge ut av sin brukerkonto, dersom den skulle ønske å gjøre dette. Dette gjøres ved at brukeren trykker på en logg ut knapp som aktiverer en funksjon som sletter de lokalt lagrede verdiene som forklarer programmet at brukeren er logget inn, og brukeren blir så navigert videre til innloggingssiden, slik at den kan logge inn igjen senere dersom den skulle ønske dette.

Akseptasjonstest - Løste gruppen kravene Adcom gav til applikasjonen - Bekreftelse fra Adcom

Etter arbeidet fikk oppdragsgiver sett på resultatet i en live framvisning, og deretter fikk vi tilsendt det følgende skjemaet som bevis på at de har godkjent at arbeidet var i tråd med deres kravspesifikasjoner, samt noen av deres opplevelse av samarbeidet.

Godkjennelse av bachelor oppgave

Dette er en bekreftelse fra (bedrift): IT Data AS (Adcom Molde)

På atstudenten(e) : Fredrik Høvik Sørøy og Joakim Mork Landsverk

Har utført prosjektet:

Prototype på app for elektronisk kjørebok basert på mobilens gps.

For/med oss i perioden januar 2022 til mai 2022.

Vi er kjent med de viktigste resultat og funn fra prosjektarbeidet.

Studentene har vist hvilke muligheter som finnes i å benytte mobilens GPS til å lage en frittstående app for elektronisk kjørebok.

Dette er svært nyttig relatert til at vi vurderer å lage en «gratis»-mobilapp som skal trekke kunder til vår kommersielle elektroniske kjørebokløsning.

Kommentarer/erfaringer med samarbeidet:

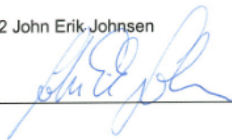
Studentene har vist arbeidsvilje og engasjement og har også vist at de har tilstrekkelig programmeringskunnskap til å presentere en fungerende prototype for Androidtelefoner på forholdsvis kort tid.

Vi har også opplevd studentene som meget interesserte i forhold til oppdraget.

Prosjektet var en positiv opplevelse for oss som oppdragsgiver.

Sted og dato. (signert av veileder/bedriftsrepresentant:)

Molde 21.03.2022 John Erik Johnsen



Litteraturliste

[1] Don't Make Me Think:

- Steve Krug, 2014, *Don't Make Me Think*

Lenker

Node.JS:

- <https://nodejs.org/en/about/>

VSCode:

- <https://code.visualstudio.com/>

- <https://code.visualstudio.com/docs/supporting/faq>
- <https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets>
- <https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>

React Native:

- <https://reactnative.dev/>
- <https://medium.com/@thinkwik/react-native-what-is-it-and-why-is-it-used-b132c3581df>

Firebase:

- <https://firebase.google.com/docs/database?authuser=0>
- <https://firebase.google.com/docs/firestore?authuser=0>

Ekbok:

- <https://www.ekbok.no/index.php?r=site%2Findex#ingress>

Expo Go:

- <https://expo.dev/>
- <https://expo.dev/client>

Expo tracking:

- <https://docs.expo.dev/versions/latest/sdk/location/>

Stack Overflow:

- Vårt spørsmål:
<https://stackoverflow.com/questions/70772288/how-do-i-run-the-setstate-function-from-another-screen-in-react-native-navigation>

Heuristikk:

- <https://www.nngroup.com/articles/ten-usability-heuristics/>

Dribbble:

- <https://dribbble.com/shots/popular/mobile?tag=card&timeframe=ever>

NSM

- <https://nsm.no/aktuelt/nasjonalt-digitalt-risikobilde-2021>

Vedlegg

Vedlegg 1: Brukertest samtykke skjema Ekbok - knut olav.

Bachelor oppgave – Ekbok mobilapplikasjon

Takk for at du tar deg tiden til å svare på samtykke skjemaet. Hvis du har lyst til å delta, vær så snill å fyll ut skjemaet under. Spør meg gjerne spørsmål hvis du lurer på noe.

Angående brukertest:

Emne: Ekbok mobilapplikasjon

Tid: 10-30 min

Det vil bli tatt notater under testen og testtakeren vil bli spurt oppfølgings spørsmål. All personlig data vil bli holdt konfidensielt og slettet etter at notater er tatt.

Ditt navn (fornavn og etternavn)

knut-olav

E-mail

SELLERKKO@LIVE.NO

Jeg har sagt ja til å delta og er enig i følgende:

Jeg deltar frivillig og kan trekke meg når som helst, uten å måtte oppgi en grunn. I tillegg, er det ett spørsmål jeg ikke har lyst til å svare på, så kan jeg si at jeg ikke ønsker å svare.

Mine svar vil bli holdt konfidensielt. Jeg forstår at navnet mitt ikke vil kunne bli koblet tilbake til meg og vil ikke kunne bli identifisert igjennom notatene som blir tatt under testen.

Informasjonen fra denne brukertesten vil bare bli brukt til notater og jeg vil ikke kunne bli identifisert fra dette.

Jeg har lest innholdet og deltar på brukertesten (Sett kryss for ja)

Jeg har lest innholdet og vil ikke delta på brukertesten (Sett kryss for nei)

Vedlegg 2: Brukertest samtykke skjema Ekbok - Håvard

Bachelor oppgave – Ekbok mobilapplikasjon

Takk for at du tar deg tiden til å svare på samtykke skjemaet. Hvis du har lyst til å delta, vær så snill å fyll ut skjemaet under. Spør meg gjerne spørsmål hvis du lurer på noe.

Angående brukertest:

Emne: Ekbok mobilapplikasjon

Tid: 10-30 min

Det vil bli tatt notater under testen og testtakeren vil bli spurt oppfølgings spørsmål. All personlig data vil bli holdt konfidensielt og slettet etter at notater er tatt.

Ditt navn (fornavn og etternavn)

Håvard SØRDAL

E-mail

HavardS@gmail.com

Jeg har sagt ja til å delta og er enig i følgende:

Jeg deltar frivillig og kan trekke meg når som helst, uten å måtte oppgi en grunn. I tillegg, er det ett spørsmål jeg ikke har lyst til å svare på, så kan jeg si at jeg ikke ønsker å svare.

Mine svar vil bli holdt konfidensielt. Jeg forstår at navnet mitt ikke vil kunne bli koblet tilbake til meg og vil ikke kunne bli identifisert igjennom notatene som blir tatt under testen.

Informasjonen fra denne brukertesten vil bare bli brukt til notater og jeg vil ikke kunne bli identifisert fra dette.

Jeg har lest innholdet og deltar på brukertesten (Sett kryss for ja)

Jeg har lest innholdet og vil ikke delta på brukertesten (Sett kryss for nei)