

## Opgaven

Jeg har valgt at integrere biblioteket *MapView* i Notebook applikationen MyNotebook i henhold til de krav stillet i Mobile Development React Native E24-Mandatory 1.

I den gamle udgave af MyNotebook kunne brugeren få vist noter på simpel listeform og oprette noter uden lokations data. Desuden kunne brugen højst tilknytte et billede til hvert enkel note.

I den nye version kan brugeren tilknytte flere billeder samt lokations data til hvert enkelt note som kan fremvises i et billedgallary i den detaljerede visning af noter.

Brugeren kan nu oprette en note ud fra en bestemt lokation som vælges på et kort. Brugeren vil med et langt tryk sendes til en ny side hvor titel, indhold og billeder kan indsættes.

Brugeren kan vælge at få vist gemte noter på et interaktivt kort eller på listeform.

## Design

Dette afsnit behandler selve design delen af MyNotebook, hvilket omfatter en kort beskrivelse af funktionaliten set ud fra brugerens perspektiv, og dennes rejse gennem applikationen.

### Kort funktions beskrivelse

MyNotebook er en applikation testet og designet til Android styresystemet. Applikationen tillader en bruger at oprette og se noter forudsat at denne er korrekt logget ind i henhold til de krav der er fastsat i den givne opgavebeskrivelse. Brugeren kan selv vælge om denne vil tilknytte lokations data til de enkelte noter, eller om disse skal vælges ud fra et kort eller nuværende lokation.

### Brugerens rejse

I det tilfælde hvor brugeren ikke er logget ind, vil brugeren mødes af en start-side hvor der er mulighed for at oprette ny bruger eller logge ind.

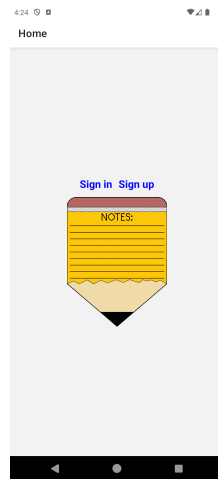


Figure 1: Startside

Herefter vil brugeren ledes til MyNotebook hovedside:

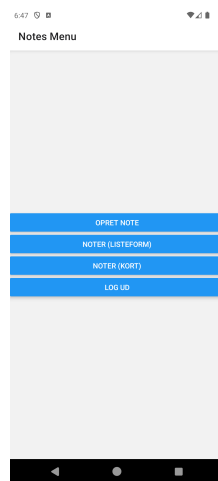


Figure 2: Hovedside

Den vigtige side i forbindelse med opgaven er siden der tilbyder en oversigt over alle noter ud fra deres lokation:



Figure 3: Kort



Figure 4: Detaljer

## Implementation

Dette afsnit giver et indblik i selve implementations detaljerne af MyNotebook. Der gives overblik og beskrivelse af benyttede biblioteker og eksempler af relevant kode i forhold til de krav angivet i opgavebeskrivelsen.

## Sprog og teknologier

MyNotebook er skrevet i React Native og benytter Google Firebase til håndtering og lagring af data (Firebase Firestore), brugerhåndtering (Firebase Authentication) og lagring af billedfiler (Firebase Storage).

## Biblioteker

Et udpluk af relevante biblioteker MyNotebook omfatter:

- `MapView`  
Står for fremvisning af kort og oprettelse af kort markørere.
- `expo/imagepicker`  
Som navnet antyder bruges dette bibliotek til billedhåndtering. Det er ansvarligt for at brugeren kan vælge billeder fra telefonen eller tage et billede ved hjælp af sit kamera.
- `react-native-firebase/firestore`  
Ansvarlig for håndtering af note data i forbindelse med Firebase Firestore.

Disse biblioteker spiller en væsentlig rolle i forhold til brugeroplevelsen og jeg vil vise nogle eksempler for deres brug i det næste afsnit.

## Kort visning

Selve kortet vises ved følgende uddrag:

```
1  <View style={styles.container}>
2    <MapView region={mapCords}
3      style={styles.map}
4      onLongPress={createNote}>
5      {markers}
6    </MapView>
7    <View style={styles.posWrapper}>
8      <IconButton width={64}
9        height={64}
10       uri={require("../assets/cPos.png")}
11       onPress={goToCurrentLocation}
12     />
13   </View>
14 </View>
```

Uddrag 1: Visning af kort

Siden har følgende variabler, som ved ændring, vil medføre gentegning af siden:

```

1  const [mapCords, setMapCoords] = useState(initialCoords)
2  const [markers, setMarkers] = useState([])
3  const [needsFetch, setNeedsFetch] = useState(true)

```

### Uddrag 2: Variabler

Siden gentegnes ved et *focus event*. Siden sørger først at fetche noter og gemme alle brugerens noter i en variabel. Derefter kaldes *updateMarkers* som vil knytte en markør til hver note:

```

1  useEffect(useCallback(() => {
2    fetchNotes().then((status) => {
3      if(!needsFetch)
4        return
5      const notesData = getAllNotes()
6      updateMarkers(notesData).then(noteMarkers => {
7        setMarkers(noteMarkers)
8        setNeedsFetch(false)
9      })
10   }).catch(handleFetchError)
11 })

```

### Uddrag 3: Fokus

```

1  async function updateMarkers(notesData) {
2    const noteMarkers = notesData.map(async (noteData) => {
3      const coordinates = toLocationCoords(noteData.latitude,
4        noteData.longitude)
5      const imageUrl = noteData.images[0]?.uri ?? undefined
6      const imageData = await imageFetch(imageUrl);
7      return NoteMarker(coordinates, noteData.title, noteData.
8        content, imageData)
9    })
10   return noteMarkers
11 }

```

### Uddrag 4: Opdatering af markører

Selve markør vises ved følgende *jsx* og benytter komponenten *Callout* bragt af *MapView* biblioteket til at vise en modificeret udgave:

```

1  <Marker key={index++} coordinate={coordinates}>
2    <Callout>
3      <View style={styles.markerContainter}>
4        <Text style={{fontSize: 20, fontWeight: "bold"}}>{title}</
5        Text>
6        <Text style={{fontSize: 14}}>{content}</Text>
7        {imageData ? renderImage(imageData) : null}
8      </View>
9    </Callout>
10  </Marker>

```

### Uddrag 5: Markør

Billedet vises ved følgende uddrag:

```

1  <View style={styles.imageWrapper}>
2    <Svg width={'100%'} height={'100%'}>
3      <ImageSvg

```

```

4      width={'100%'}
5      height={'100%'}
6      preserveAspectRatio="none"
7      href={{ uri: `${data}` }}
8    />
9  </Svg>
10 </View>

```

#### Uddrag 6: Billede markør

Der skal dog lige knyttes nogle bemærkninger i forhold til billede funktionen da denne ikke viste sig helt problemfri.

Første bemærkning omhandler den mangelfulde undersøgtelse af React Natives eget *Image* komponent der ikke tillod visning af billeder pakket ind i *Call-out*. Jeg måtte på nettet for til sidst at finde ovenstående løsning.

Den anden bemærkning omhandler forsinkelse i forhold til indlæsning af markører grundet asynkron indlæsning af billeder. Her ligger problemet i den måde billeder indlæses. Noter har hver et eller flere billeder tilknyttet via billede urls som først skal hentes. Det skaber noget forsinkelse som jeg har prøvet at løse, ved at hente billedet ned på enheden ved hjælp af *fetch API*, og derefter kode billedet om til BASE64. Dog med svingende resultat, da jeg ikke synes problemet er løst 100%.

## Nuværende lokation

Som en krølle på halen har jeg implementet en knap til at nulstille kortet til brugerens nuværende lokation. Til det har jeg implementeret en funktion *goToCurrentLocation* som er en asynkron funktion som henter nuværende lokations data fra *expo/geoLocation* og opdaterer *coords* variabelen, hvilket gennemtvinger en gengtegning af siden. Værd at bemærke her, er, at jeg først sætter variabelen til et tomt objekt i tilfælde af, at den nuværende lokation ikke har ændret sig i forhold til sidste gengtegning.

```

1  async function goToCurrentLocation() {
2    const cPos = await currentLocation()
3    const coords = {
4      latitude: cPos.latitude,
5      longitude: cPos.longitude,
6      latitudeDelta: 0.75,
7      longitudeDelta: 0.75
8    }
9    setMapCoords({})
10   setMapCoords(coords)
11 }

```

#### Uddrag 7: Gå til nuværende lokation

## Mangler og work to do

Denne opgave er en *minimal effort* opgave og der er plads til forbedring og finpudsninger. r

Jeg vil slutte af med et overblik over hvad jeg helst ser ændret eller modificeret i forhold til senere opdateringer:

- Hovedmenuen

Hovedsiden bestående af trykknapper er kun en midlertidig løsning. Idéen er, at disse skal udfases til fordel for et flise layout med ikon og tekst. Dette vil forøge det visuelle udtryk betragteligt og vil virke mere appellerende for eventuelle brugere.

- Note markør

Note markør viser kun titel og indholds samt et billede i et groft layout. Havde dog lidt udfordringer med selve billede funktionaliteten, og brugen af *svg* var den eneste fungerende løsning for mig. Der er dog stadig problemer i starten når brugeren indlæser kortet, hvor der virker til at være noget forsinkelse på indlæsning af billede<sup>1</sup>.

- Små justeringer

---

<sup>1</sup>Brugeren skal nogle gange klikke flere gange på markøren for successfull fremvisning af billede.