Mappe 3 - DAVE3600 Apputvikling

S333329 - Therese Trollbu

1.0 Introduksjon

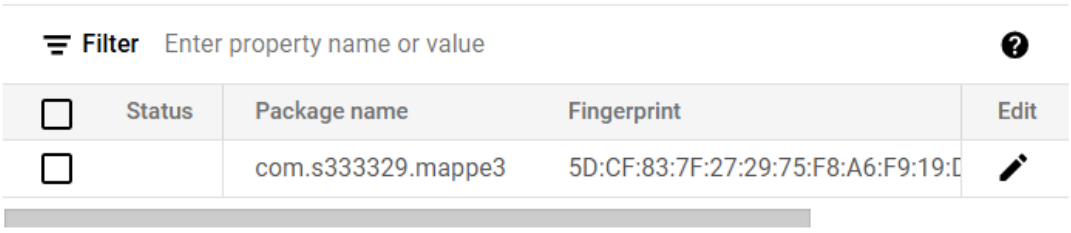
I denne applikasjonen kan man som bruker se og lagre sine favorittsteder ved bruk av et Google Maps kart. Appen er på engelsk og henter lagrede data fra en tabell kalt Locations på serveren s333329@dave3600.cs.oslomet.no. Bruker har muligheten til å laste opp og lagre nye lokasjoner i tillegg til å skrive inn sine egne beskrivelser av dem. Android Studio sender disse dataene ved bruk av to metoder med hver sin http lenke som trigger to PHP filer på serveren. Disse filene er locationsJsonin.php og locationsJsonout.php.

2.0 Koble Android studio til Google Maps

2.1 Api-nøkkel:
For å kunne legge til Google Maps i appen så var det behov for en api-nøkkel. Denne ble generert på Google Cloud og er filtrert til å inkludere Maps SDK for Android og Geolocation API. Maps SDK for Android er for å implementere selve kartet og Geolocation API er for å kunne hente ut en adresse tilhørende markøren på kartet.

Figur 1: API-key

2.2 Sha1hash:
For å legge til en restriksjon på nøkkelen så kopierte jeg inn Sha1hash nøkkelen fra Android Studio sine filer, inn i api-nøkkelinnstillingene på Google Cloud. Slik at den kun er tilknyttet det programmet.



Figur 2: SHa1hash

2.3 Implementering av kart:
Selve kartet er puttet i et fragment som importeres inn i activity_main.xml. Fragmentet er justert til å dekke halve skjermen og resten av skjermen blir brukt til informasjon og funksjonaliteter.

3.0 PHP kode på server

For å kunne lagre data på serveren så måtte det lages en tabell med riktige verdier som passer til input. Siden appen er på engelsk så fikk dataene de engelske ordene; *address, latitude, longitude* og *description.* Selve tabellen ble laget med koden:

CREATE TABLE Locations (Id INT AUTO_INCREMENT PRIMARY KEY, address VARCHAR(255), latitude DOUBLE, longitude DOUBLE, description VARCHAR(255));

Ved bruk av locationsJsonin.php finner serveren den aktuelle tabellen som er opprettet og legger inn verdiene i riktig kolonne.

```php
<?php
$con=mysqli_connect('localhost', 's333329',"", 's333329');
$address=$_REQUEST['address'];
$latitude=$_REQUEST['latitude'];
$longitude=$_REQUEST['longitude'];
$description=$_REQUEST['description'];

$address=(String)$address;
$latitude=(float)$latitude;
$longitude=(float)$longitude;
$description=(String)$description;

$sql=mysqli_query($con, "insert into Locations (address, latitude, longitude, description) values ('$address', $latitude, $longitude, '$description')");
mysqli_close($con);
?>
```

Figur 3: locationsJsonin.php

For å gi appen muligheten til å hente alt som er lagret i Location tabellen, så brukes locationJsonout.php som henter alt innhold ved bruk av SELECT * FROM Locations.

```php
<?php mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$con=mysqli_connect("localhost", "s333329","","s333329");
$sql=("SELECT * FROM Locations");
$tabell=mysqli_query($con,$sql);
mysqli_close($con);
while($row=mysqli_fetch_assoc($tabell)){
$output[]=$row;}
print(json_encode($output));
?>
```

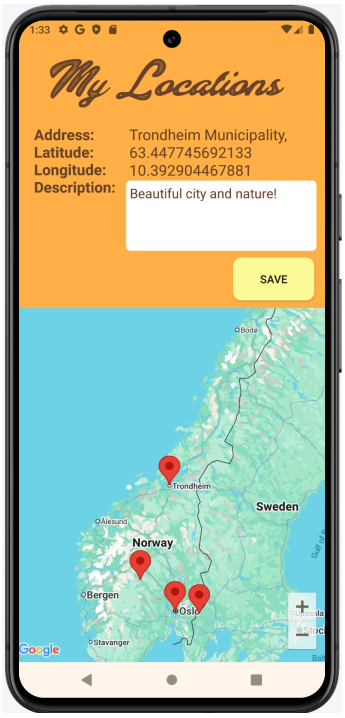
Figur 4: locationsJsonout.php

4.0 My Locations (MainActivity)

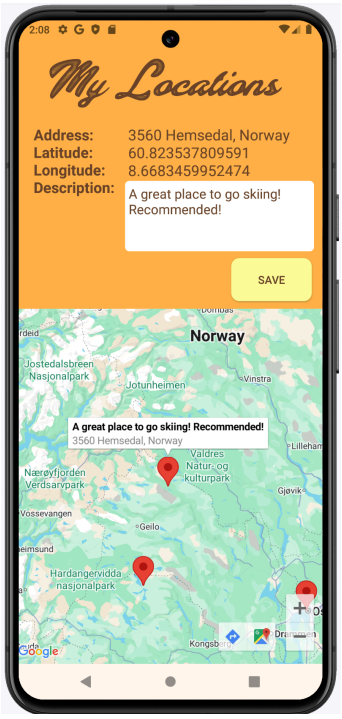
4.1 Hente tabell

Ved oppstart av programmet kjøres koden makeHttpRequest som henter tabellen fra serveren og legger innholdet i en liste. Deretter loopes det gjennom listen og plasserer hver lokasjon på kartet ved å gi hver av dem en markør. Den siste lokasjonen i tabellen er den som blir vist først når du åpner appen. Øverst på siden får man en oversikt over address, latitude, longitude og description til markøren og dette oppdateres når du trykker på de andre markørene eller et annet sted på kartet.

For å kunne gjøre det lettere å se hvor en har lagret en markør, så er det lagt til en automatisk zoom for å komme nærmere markøren og ikke vise hele verdenskartet. I tillegg er det lagt til et pluss og minus tegn for å gi muligheten til å zoome inn og plassere pins på et mer detaljert nivå eller kunne zoome ut og få en større oversikt over alle pins.



Figur 5: Forsiden



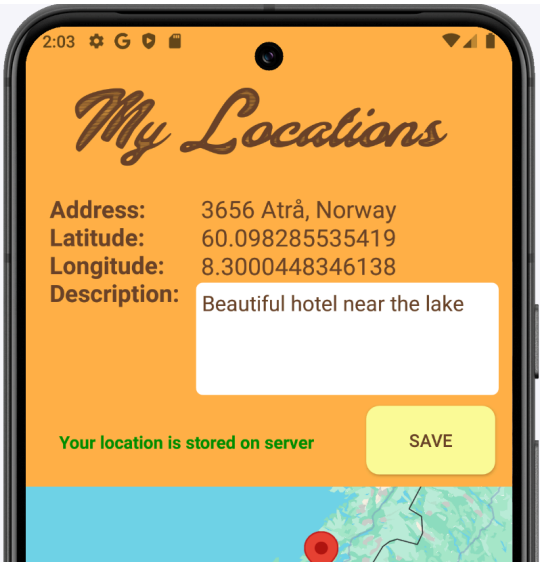
Figur 6: Zoomet inn

```
// retrieve data
private void makeHttpRequest() {  2 usages  ▲ TheresePinkLizard
    executor.execute(new Runnable() {  ▲ TheresePinkLizard
        @Override  ▲ TheresePinkLizard
        public void run() {
            try {
                URL url = new URL( spec: "https://dave3600.cs.oslomet.no/~s333329/locationsJsonout.php");
                HttpURLConnection httpUrlConnection = (HttpURLConnection) url.openConnection();
                httpUrlConnection.setRequestMethod("GET");
                httpUrlConnection.connect();
                int responseCode = httpUrlConnection.getResponseCode();
                if (responseCode == HttpURLConnection.HTTP_OK) {
                    InputStream inputStream = httpUrlConnection.getInputStream();
                    locations = readInputStreamToString(inputStream);
                    handler.post(new Runnable() {  ▲ TheresePinkLizard
                        @Override  ▲ TheresePinkLizard
                        public void run() {
                            float zoomLevel = 5.0f;
                            if(!locations.isEmpty()) {
                                for (Location location : locations) {
```

Figur 7: Deler av koden som henter tabelldata

4.2 Lagre markør på server

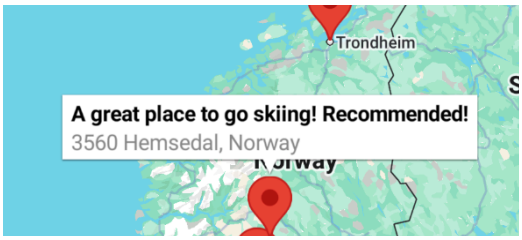
Når man ønsker å lagre en ny markør, så trykker man først på plasseringen på kartet man ønsker å lagre. Deretter skriver man inn en beskrivelse hvis man ønsker det og trykker lagre. Da vil koden makeWebServiceRequest kjøre som sender informasjonen i tekstfeltene til serveren for å bli lagret. Hvis dette er i orden så vil man få beskjed om at lagringen var vellykket i grønn farge, hvis det ikke ble lagret så får man beskjed om at det var mislykket i rød farge. Begge disse tilbakemeldingene vises på samme sted ved siden av lagre knappen og fordi de har en grønn og rød farge, så skaper det en tydeligere assosiasjon til dens betydning. (Grant, 2018, s. 99).



Figur 8: Tilbakemelding

4.3 Tekstboks på kartet

Hvis du trykker på markørene som er lagret så vil tekstfeltene øverst oppdatere seg, men i tillegg vil det vises en tekstboks over markøren med kommentaren tilhørende markøren og adressen under.



Figur 9: Tekstboks med informasjon

```
// send data
private void makeWebServiceRequest(String theaddress, double thelatitude, double thelongitude, String thedescription) {
    executor.execute(new Runnable() {  TheresePinkLizard
        @Override  TheresePinkLizard
        public void run() {
            try {
                URL url = new URL( spec: "https://dave3600.cs.oslomet.no/~s333329/locationsJsonin.php");
                HttpURLConnection httpUrlConnection = (HttpURLConnection) url.openConnection();
                httpUrlConnection.setRequestMethod("POST");
                httpUrlConnection.setDoOutput(true);

                // Prepare the data to be sent
                String data = "address=" + URLEncoder.encode(theaddress,  enc: "UTF-8")
                        + "&latitude=" + thelatitude
                        + "&longitude=" + thelongitude
                        + "&description=" + URLEncoder.encode(thedescription,  enc: "UTF-8");
```

Figur 10: Deler av koden som sender data til server

5.0 Design

For å indikere at en bruker kan endre beskrivelse av markøren, så har tekstfeltet du kan skrive i fått en hvit farge med en tekstindikator som forteller deg at verdien kan skrives inn. De andre verdiene er oransje og matcher bakgrunn for å skape en separasjon fra beskrivelsen og fremheve budskapet om at disse verdiene er låst (Crudu, 2024). Disse verdiene er låste fordi de er basert på Google Maps sine data.

Siden appen er ment for å kunne lagre sine favorittlokasjoner så ønsker jeg å gi appen et litt personlig preg. Derfor har overskriften en font som minner om en håndskrift. For å få resten av designet til å virke helhetlig og passe overskriften så er tekstboksen og knappen endret til å få runde kanter.

Figur 11: Tekstlig beskrivelse

6.0 Kilder

Font:

Colinas Personal Use Font. (n.d.). 1001Fonts.

https://www.1001fonts.com/colinas-personal-use-font.html

Teori:

Grant, R. (2019). *Data Visualization: Charts, Maps and Interactive Graphics*. CRC Press, Taylor & Francis Group.

Crudu, A. (2024, March 15). *Creating user-friendly interfaces for software applications*. MoldStud. Retrieved November 17, 2024, from https://moldstud.com/articles/p-creating-user-friendly-interfaces-for-software-applications