

## KartApp rapport – s364765

I de tidligere rapportene har jeg fokusert på designvalg og designprinsipper. Med tanke på at vi ikke har implementert noe spesielt brukergrensesnitt i denne appen vil jeg heller fokusere på fremgangsmåte og kode som er implementert i applikasjonen.

### PHP Scripts

Jeg startet med å lage to php scripts for å legge inn (figur 1) og hente ut (figur 2) verdier for steder. Her brukte jeg scriptene fra forelesningen hvor jeg endret verdiene til verdiene som må lagres for hvert steg

#### Jsonin.php

```
<?php
$con=mysqli_connect('localhost','s364765','','s364765');

$navn = $_REQUEST['navn'];
$beskrivelse = $_REQUEST['beskrivelse'];
$gateadresse = $_REQUEST['gateadresse'];
$latitude = $_REQUEST['latitude'];
$longitude = $_REQUEST['longitude'];

$sql = mysqli_query($con, "INSERT INTO Steder (navn, beskrivelse, gateadresse)
VALUES ('$navn', '$beskrivelse', '$gateadresse')");

mysqli_query($con, "insert into Food (Name) values ('$name')");
mysqli_close($con);
```

Figur 1

#### Jsonout.php

```
<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$con=mysqli_connect("localhost","s364765","","s364765");
$sql=("select * from Steder");
$tabell=mysqli_query($con,$sql);
mysqli_close($con);
while ($row=mysqli_fetch_assoc($tabell)) {
$output[]=$row;
}
print(json_encode($output));
?>
```

Figur 2

## URL

For å sjekke at scriptene fungerer på riktig måte testet jeg dette ved bruk av url'er for å legge inn (figur 3) og hente ut (figur 4) verdier (figur 5).

Legge inn:

```
https://dave3600.cs.oslomet.no/~s364765/jsonin.php?navn=Kebab&beskrivelse=Helt grei kebab til grei pris&gateadresse=Hegdehaugsveien 2, 0167 Oslo&latitude=59.920427614668654&longitude=10.733472593128683
```

Figur 3

Hente ut:

```
dave3600.cs.oslomet.no/~s364765/jsonout.php
```

Figur 4

```
[{"id": "1", "navn": "Sushi", "beskrivelse": "Veldig god sushi", "gateadresse": "Pilestredet 1c", "latitude": "50", "longitude": "32"}, {"id": "2", "navn": "Kebab", "beskrivelse": "Helt grei kebab til grei pris", "gateadresse": "Hegdehaugsveien 2, 0167 Oslo", "latitude": "59.920427614668654", "longitude": "10.733472593128683"}, {"id": "3", "navn": "Test", "beskrivelse": "h\u00f8h", "gateadresse": "oslo123", "latitude": "1111", "longitude": "3333"}, {"id": "4", "navn": "Kj\u00f8pesenter", "beskrivelse": "Mange fine butikker", "gateadresse": "Stenersgata 1B, 00500 Oslo, Norway", "latitude": "59.912812964785934", "longitude": "10.752383172512054"}, {"id": "5", "navn": "Kunst museum", "beskrivelse": "Fin kunst", "gateadresse": "Oda\u00f8 bro, Operagata 1, 0194 Oslo, Norway", "latitude": "59.90644061960834", "longitude": "10.754133984446524"}, {"id": "6", "navn": " ", "beskrivelse": " ", "gateadresse": "null", "latitude": "61.28846922083849", "longitude": "12.593684273938206"}, {"id": "7", "navn": "Nice view", "beskrivelse": "Very nice view", "gateadresse": "29 Rue Pasteur, 92140 Clamart, France", "latitude": "48.77937778076", "longitude": "2.2289887815713882"}, {"id": "8", "navn": "Nice city", "beskrivelse": "very nice city", "gateadresse": "Marstalsgade 1c, 2100 K\u00f8benhavn, Denmark", "latitude": "55.70356759797532", "longitude": "12.583986595273016"}, {"id": "9", "navn": "Island", "beskrivelse": " ", "gateadresse": "Vaksangatan 14, 387 93 Borgholm, Sweden", "latitude": "56.736492430811246", "longitude": "16.688640788197517"}, {"id": "10", "navn": "Finland", "beskrivelse": " ", "gateadresse": "Linnantie 61, 63350 Alavus, Finland", "latitude": "62.44725247780278", "longitude": "23.552483394742012"}, {"id": "11", "navn": "Finland2", "beskrivelse": " ", "gateadresse": " ", "latitude": " ", "longitude": " "}, {"id": "12", "navn": "Billing godteri", "beskrivelse": " ", "gateadresse": "Sj\u00f8tugatan 6, 673 31 Charlottenberg, Sweden", "latitude": "59.889604646367815", "longitude": "12.310299351811409"}, {"id": "13", "navn": "Kj\u00f8pesenter", "beskrivelse": "Test for \u00e5 se hvor mye \u00e5v tekstboksen som vises for et lagret sted", "gateadresse": "u00d8st\u00e4ngsju00f8sveien 74, 0667 Oslo, Norway", "latitude": "59.9480451360658", "longitude": "10.821804106235504"}, {"id": "14", "navn": "j", "beskrivelse": " ", "gateadresse": "Podg\u00f3rny 63, 88-400, Poland", "latitude": "52.8206838442673", "longitude": "17.76767671108246"}, {"id": "15", "navn": " ", "beskrivelse": " ", "gateadresse": "Asutustie 124, 78710 Varkaus, Finland", "latitude": "62.26315944852211", "longitude": "28.20554774254561"}]
```

Figur 5

## Kode for å hente og vise steder på kartet fra DB

### Hente data fra DB i webserver

Jeg startet med å få koden som henter ut objekter ved bruk av AsyncTask til å fungere med verdiene jeg har brukt for å lagre steder i DB (figur 6). Informasjon om disse stedene måtte lagres som et objekt i Java som jeg har kalt «Sted». Disse stedene har jeg deretter lagret i en liste med steder.

```
// Løkke som går gjennom hvert sted hentet og legger det til i en liste
for (int i = 0; i < steder.length(); i++) {
    JSONObject jsonobject = steder.getJSONObject(i);
    String navn = jsonobject.getString( name: "navn");
    String beskrivelse = jsonobject.getString( name: "beskrivelse");
    String gateadresse = jsonobject.getString( name: "gateadresse");
    double latitude = Double.parseDouble(jsonobject.getString( name: "latitude"));
    double longitude = Double.parseDouble(jsonobject.getString( name: "longitude"));

    // Sted
    Sted sted = new Sted(navn, beskrivelse, gateadresse, latitude, longitude);

    // Legger til sted i liste med steder
    stedListe.add(sted);

    Log.d( tag: "StedInfo", msg: "Sted added: " + sted.getNavn().toString());

    retur = retur + "Navn: " + navn + "\n" +
        "Beskrivelse: " + beskrivelse + "\n" +
        "Gateadresse: " + gateadresse + "\n" +
        "Latitude: " + latitude + "\n" +
        "Longitude: " + longitude + "\n\n";
}
StedListe.setStedListe(stedListe);
return retur;
```

Figur 6

### Legge til markører for data som er lagret i liste

Neste steg var å kunne bruke en løkke for inne i onPostExecute å gå gjennom stedene som er lagret i listen og legge til markører med informasjon for hvert lagret sted (figur 7).

```
@Override
protected void onPostExecute(String ss) {
    // Henter liste med steder
    List<Sted> stedListe = StedListe.getStedListe();

    // Løkke som går gjennom hvert sted og legger det inn på kartet med info
    for (Sted sted : stedListe) {

        // Finner koordinater
        LatLng gpsCords = new LatLng(sted.getLatitude(), sted.getLongitude());

        // Legger til markør
        mMap.addMarker(new MarkerOptions()
            .position(gpsCords)
            .title(sted.getNavn() + " - " + sted.getAdresse())
            .snippet(sted.getBeskrivelse()));
    }
}
```

Figur 7

## Kode for å legge inn steder i DB og deretter oppdatere kartet

### Markør

For å sette markør på kartet har jeg brukt `onMapClickListener` som legger inn en markør der man trykker på kartet, dersom man flytter markøren vil den forrige bli slettet med mindre man lagrer den som et sted (figur 8). Ved klikk på kartet henter jeg longitude og latitude verdier for stedet og bruker disse verdiene for å finne adresse ved hjelp av metoden «`adresseKoordinater`» (figur 9).

```
mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {  
    no usages  
    @Override  
    public void onMapClick(LatLng latLng) {  
        // Fjerner siste markør  
        if (sisteMarkør != null) {  
            sisteMarkør.remove();  
        }  
  
        // Legger til markør med tittel  
        sisteMarkør = mMap.addMarker(new MarkerOptions().position(latLng).title("Clicked Location"));  
  
        // Henter latitude/longitude  
        double latitude = latLng.latitude;  
        double longitude = latLng.longitude;  
  
        // Henter full adresse  
        String adresse = adresseKoordinater(latitude, longitude);  
        String latitudeString = Double.toString(latitude);  
        String longitudeString = Double.toString(longitude);  
  
        // Dialog åpnes ved klikk på kart  
        LagreDialog lagreDialog = new LagreDialog();  
  
        // legger verdier i bundle som kalles på i dialogen  
        Bundle bundle = new Bundle();  
        bundle.putString("adresse", adresse);  
        bundle.putString("latitude", latitudeString);  
        bundle.putString("longitude", longitudeString);  
        lagreDialog.setArguments(bundle);  
  
        lagreDialog.show(getSupportFragmentManager(), tag: "LagreDialog");  
    }  
}
```

Figur 8

```

/*-----Metode som regner ut adresse koordinater-----*/
1 usage
private String adresseKoordinater(double latitude, double longitude) {
    Geocoder geocoder = new Geocoder( context: this, Locale.getDefault());
    try {
        List<Address> adresser = geocoder.getFromLocation(latitude, longitude, maxResults: 1);

        if (adresser != null && adresser.size() > 0) {
            Address adresse = adresser.get(0);

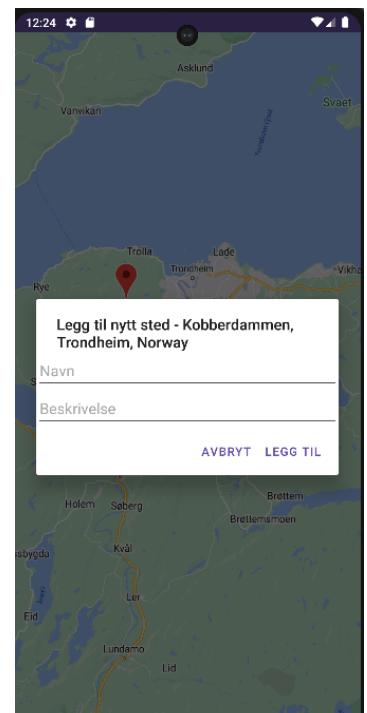
            // Henter full adresse
            String fullAdresse = adresse.getAddressLine( index: 0);
            Log.d( tag: "Adresse", msg: "Full Adresse: " + fullAdresse);
            return fullAdresse;
        } else {
            Log.d( tag: "Adresse", msg: "Ingen adresse funnet");
        }
    } catch (IOException e) {
        e.printStackTrace();
        Log.e( tag: "Adresse", msg: "Error ved henting av adresse", e);
    }
    return null;
}

```

Figur 9

## Dialog vindu

Da brukeren klikker på kartet vil det i tillegg til en markør komme opp ett dialog vindu (figur 10/11) hvor brukeren kan skrive inn navn på stedet og beskrivelse. Dersom brukeren velger å trykke på «legg til» vil disse verdiene bli lagret til webserveren ved hjelp av metoden «postJSON» som executer post url'en, hvor verdiene brukeren skrev inn om stedet og verdier som latitude/longitude og adresse blir lagt til i url'en slik at disse lagres om stedet (figur 12/13). Deretter vil metoden «getJSON» bli kalt igjen for å oppdatere kartet med det nye stedet som ble lagt til i DB. Dersom brukeren trykker avbryt vil markøren bare bli fjernet (figur 14).



Figur 10

```
// Dialog åpnes ved klikk på kart
LagreDialog lagreDialog = new LagreDialog();

// Legger verdier i bundle som kalles på i dialogen
Bundle bundle = new Bundle();
bundle.putString("adresse", adresse);
bundle.putString("latitude", latitudeString);
bundle.putString("longitude", longitudeString);
lagreDialog.setArguments(bundle);

lagreDialog.show(getSupportFragmentManager(), tag: "LagreDialog");
```

Figur 11

```
// Legger inn sted på webserver DB
1 usage
public void postJSON(String url) {
    getJSON post = new getJSON();
    post.execute(url);
}
```

Figur 12

```
// Lagre knapp i dialog
1 usage
@Override
public void onLagreClick(String navn, String beskrivelse, String adresse, String latitude, String longitude) {
    String url = "https://dave3600.cs.oslomet.no/~s364765/jsonin.php?navn=" + navn + "&beskrivelse=" + beskrivelse;
    // Lagrer dialog informasjon om stedet på DB i webserver
    postJSON(url);

    // Oppdaterer kartet med det nye stedet
    getJSON get = new getJSON();
    get.execute(new
        String[]{"https://dave3600.cs.oslomet.no/~s364765/jsonout.php"});
}
```

Figur 13

```
// Avbryt knapp i dialog
1 usage
@Override
public void onAvbrytClick() {
    // Fjerner markøren dersom avbryt trykkes
    if (sisteMarkør != null) {
        sisteMarkør.remove();
        sisteMarkør = null;
    }
}
```

Figur 14