TikTok Youth Camp 2022

Android Development Group Project:

iSeeTaxi4u

Group Members	
Bruce Wang Zhuang Jie	2102506@sit.singaporetech.edu.sg
Ng Lay Kiat	2101035@sit.singaporetech.edu.sg
Seah Meng Chung Juleus	2100589@sit.singaporetech.edu.sg
Lim Ryan	2101702@sit.singaporetech.edu.sg

1.Technical Overview

1.1 Frameworks & Libraries used

The data being used in our application is retrieved from Data.gov.sg (https://data.gov.sg/dataset/taxi-availability). In our application we have made use of the Retrofit library to get this json file. We also used the Maps SDK from google to display our data out on a map. (https://developers.google.com/maps/documentation/android-sdk/start#maps_android_mapsactivity-kotlin)

1.2 Key technical selection

1.2.1 Retrofit

Retrofit is a type-safe HTTP library that allows us to extract the GEOson data we need from the government provided API. We decided to use it as we have read up on the multitudes of benefits Retrofit has. One of them is the ease of use. Retrofit treats API calls as simple Java method calls. Hence, we can define the URL to hit and the types of requests as Java classes. Furthermore, the network call and JSON/XML parsing is done using GSON. The aforementioned are reasons why we have chosen Retrofit to retrieve API data.

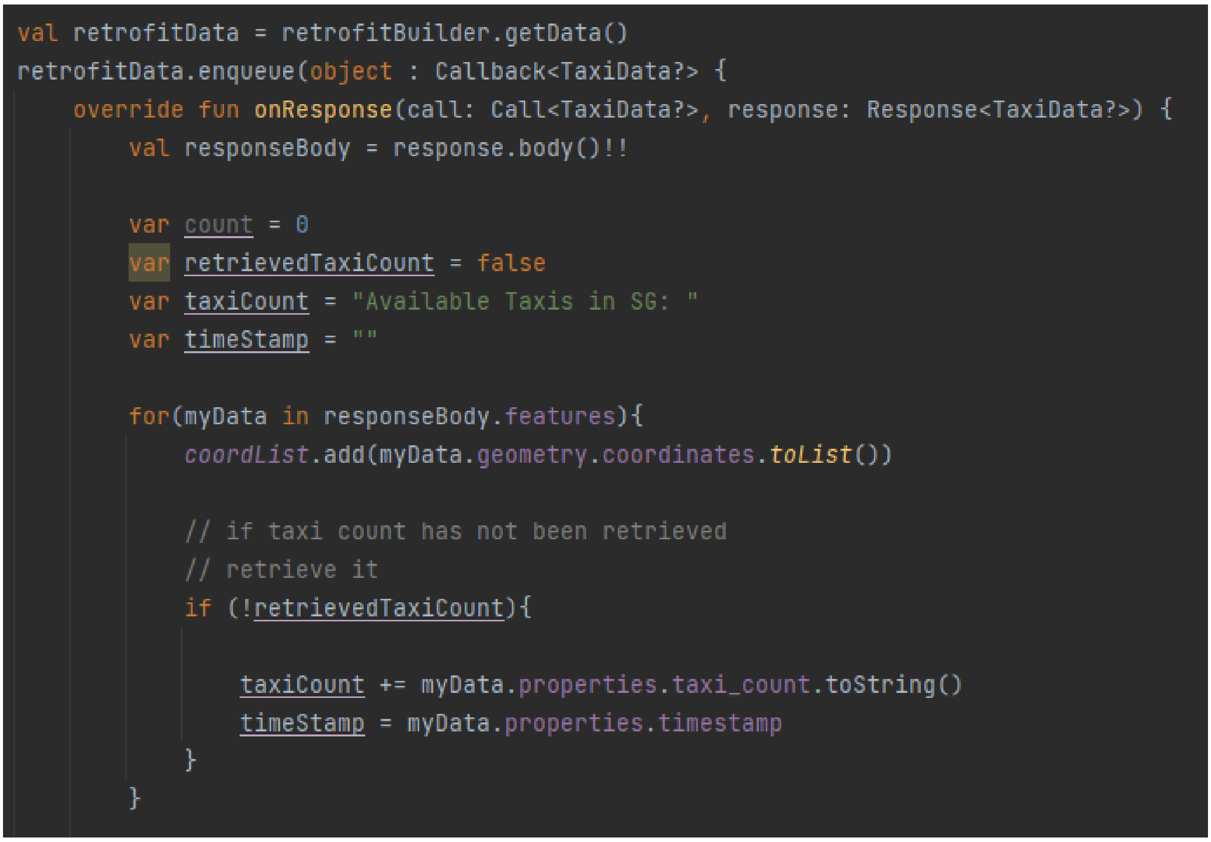
1.2.2 Google Location Services API

As we dived deeper into the project, we realised that we had to implement a map view for our application. As such, we looked for resources online and found out that the Google Location Services API was needed if we wanted to accomplish our task. By implementing a dependency and a MapsActivity, we were able to visualise a Google Maps view for the user to navigate Google Maps and spot taxis nearby his/her current location.

1.2.3 JSON to Kotlin Class Plugin (Android Studio)

While looking for resources online for Kotlin/Android Studio quality of life plugins, we stumbled upon the JSON to Kotlin Class plugin. This plugin allows us to input the contents of any JSON file and Android Studio would automatically convert all the properties to data classes. The data classes would later be utilised by the developer for fetching data via API calls to the Government data website.

1.3 Snippet of interesting codes



```
val retrofitData = retrofitBuilder.getData()
retrofitData.enqueue(object : Callback<TaxiData?> {
    override fun onResponse(call: Call<TaxiData?>, response: Response<TaxiData?>) {
        val responseBody = response.body()!!

        var count = 0
        var retrievedTaxiCount = false
        var taxiCount = "Available Taxis in SG: "
        var timeStamp = ""

        for(myData in responseBody.features){
            coordList.add(myData.geometry.coordinates.toList())

            // if taxi count has not been retrieved
            // retrieve it
            if (!retrievedTaxiCount){

                taxiCount += myData.properties.taxi_count.toString()
                timeStamp = myData.properties.timestamp
            }
        }
```

Fig 1. Retrieval of data from JSON via Retrofit API Call

In Fig 1, we are retrieving data from Gov Data's JSON via the Retrofit API call. Once retrieved, properties such as the coordinates of available taxis in Singapore and the total taxi count in Singapore are stored into variables and used for future purposes. The main purpose of retrieving the coordinates of each available taxi is so that we can utilise these coordinates to create markers on the Google Map view that the user will discover in the application.

2. Product Overview

2.1 Application landing page

The application landing page consists of several components. Firstly, the total number of taxis available in Singapore would be displayed along with the total number of taxis available close to the user's current location. With that, the user is able to set the distance visibility away from available taxis by adjusting a seek bar at the bottom of the landing page. The distance is defaulted to 3 km and can be adjusted from 1 km to 10 km. Following that, there are two buttons that the user can press. The first button is a “View Map” button to view a Google Map which indicates the user’s current location denoted by a blue marker with direction, and taxis around the user denoted by car icons. The other button is the “Refresh” button, should the user adjust the distance bar, the user is required to use this button to update both taxi counts and Google Map view.

2.2 Features

The main feature of our application is to allow users to make better decisions based on the surrounding number of taxis around them. As the price of oil and gas has hiked recently due to the Russian-Ukrainian war and the economies continuing to emerge from the pandemic, taxi prices have gone up to help drivers defray the higher operating costs. These costs are also translated to us consumers who also have to curb the higher cost of living. By creating a mobile application that allows users to see the surrounding number of taxis, users can see if there are sufficient taxis in his/her area to avoid price surge due to demand or maybe even travel a few km away from his/her current location where there are more taxis to get a better pricing.

3. Highlights

3.1 Challenges faced

Our team struggled with understanding how to use Retrofit to get the data and translate it into our application at first, we had to go through countless videos on youtube and research before getting it to work. Due to the API format being in GeoJSON, we were stuck for a period of time to extract the coordinates of the taxis. Although everyone in our team has a background in IT, this was something new to us as we didn’t have any experience with using Kotlin. This project has given us exposure to new knowledge that we might not have experienced in school.

3.2 Takeaway

Throughout this whole project our team learned how to use Kotlin to develop a mobile application from scratch and translating the data from JSON onto Google Map. We also got familiarised with Android Studio and its emulators, while also learning how we can request for location permissions from users to display their location into our application.

4. Project Management

4.1 Tasks allocation

Getting JSON from data.gov.sg (Bruce)

Google Location Permissions (Ryan)

User Interface (Juleus)

Getting the user's current location (Lay Kiat)

Translating the JSON data into GSON and integrating with Google Map (All)