GitHub Username: powernusa

PowerGlide

Description

Android has hit a near record 90% of mobile market share and therefore embracing android apps is a must for individuals who want to make their presence or services known to the market. This app will empower people to make a living in the transportation industry. By using this app, individuals can make a living by becoming a driver. Also by using this app, individuals can travel from one point to another in a city at their fingertips. Just by clicking their android apps, there are provided with transport service.

Intended User

This app is intended for individuals who want to make a living in a transport industry. Also intended for individuals who are looking for transport/taxi service at their fingertips.

Features

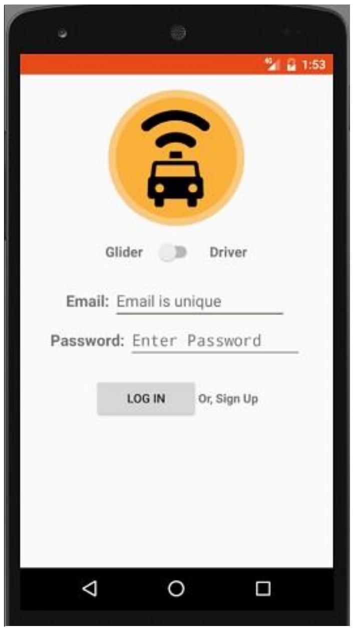
Main features of this app.

- Calling taxi/transport online on a mobile platform
- It helps drivers to view rider/glider requests based on distance in km
- Once accepted glider's request, it will help drivers to navigate to destination address on a google map
- It helps rider/glider to track the location of a driver
- It helps rider/glider to view nearby places around them
- Increases the productivity and service of drivers because riders/gliders must rate drivers. Rating is from 1 to 5. If rider/glider has not given a rating on his/her previous ride, no taxi/transport request is allowed

User Interface Mocks

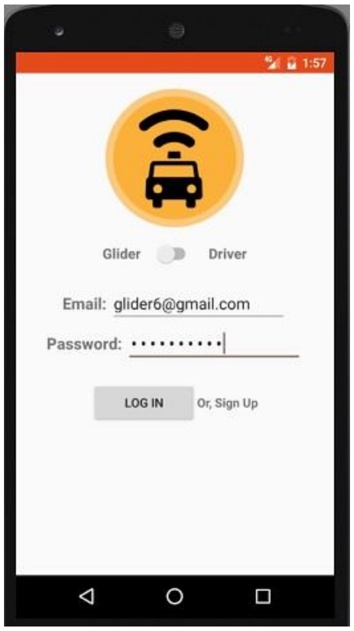
<u>Glider/Rider Mocks</u>

A. Home Screen



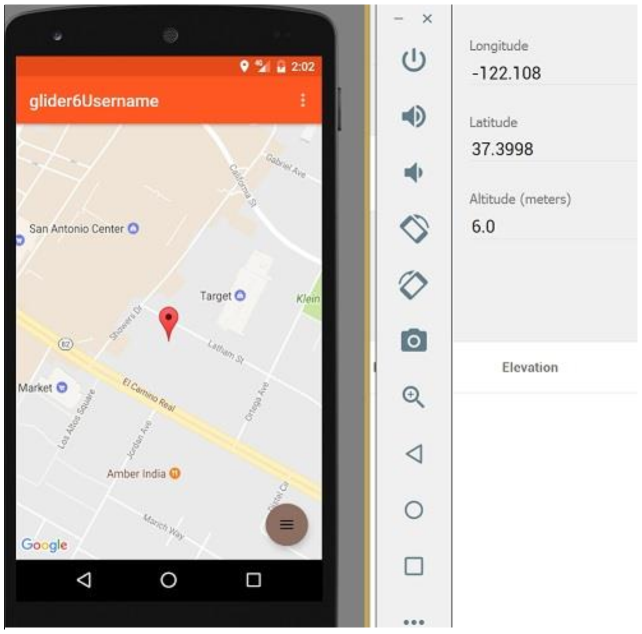
This is a home screen for both glider/rider and driver. You can change between them by sliding the switch. I combine glider/rider and driver into one app for practical purposes. When the app is ready for production, the app will be separated accordingly – one app for glider/user and one app for driver.

B.Glider/Rider logging in



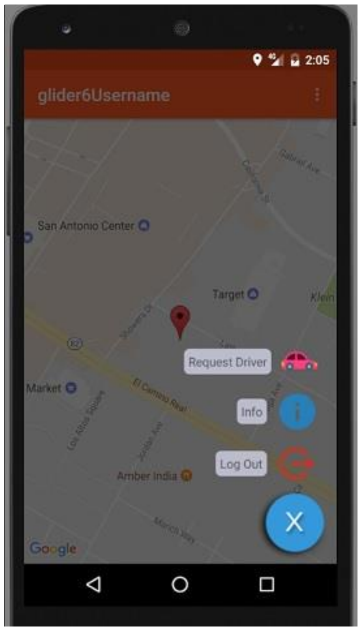
Enter glider/rider and password. Then click login.

C.Glider/Rider Activity



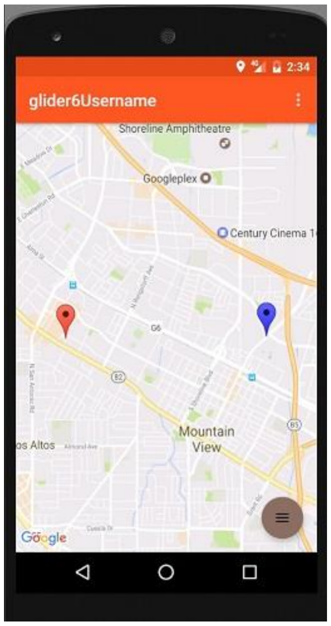
After a successful login, the app would take you to a map. You need to enter your GPS location into the emulator shown above.

D.Glider/Rider requesting driver



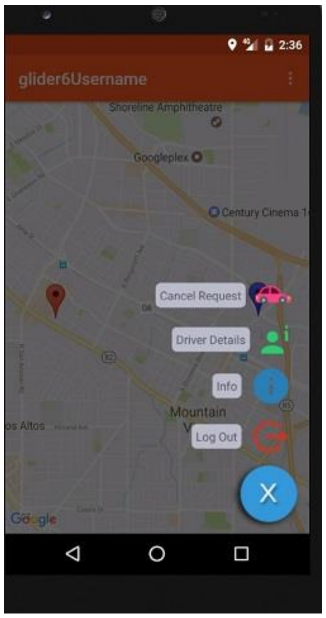
Then click on the FAB button, a dialog will appear. Then click on the “Request Driver” button.

E.Request accepted by Driver



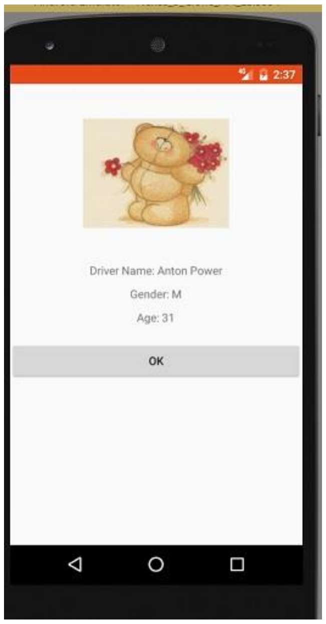
The red marker is the glider/rider location. The blue marker is the driver location. If there is a blue marker on the map, it means that a driver has already accepted glider’s request. Refer to point **“5. Driver Accepts Request”** below.

F. Click Driver details



Click on the FAB button and then click on “Driver Details” to view driver info.

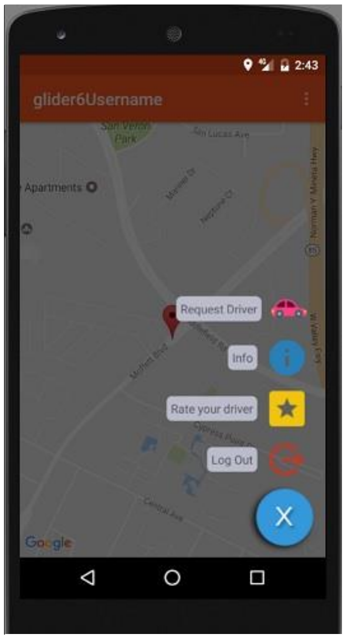
G. Driver profile



Driver Name: Anton Power
Gender: M
Age: 31

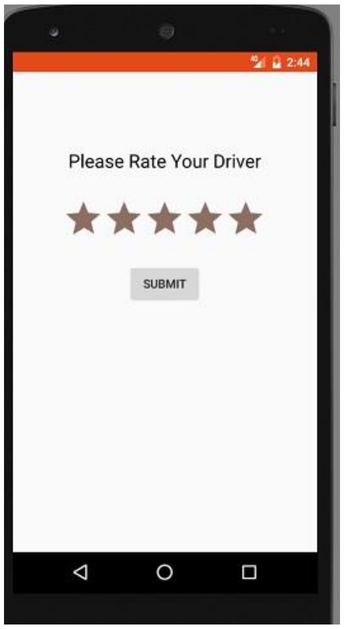
This is the Driver Info. Click “OK” to return to previous activity.

H. Rate Driver



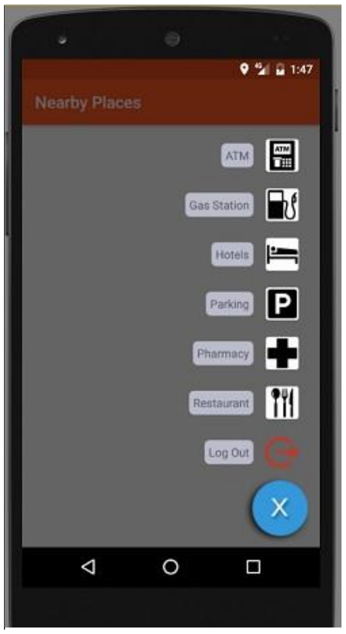
When a transaction takes place between glider and driver, “Rate Your Driver” (yellow star icon) appears on the dialog as shown above. Refer to 7. Click Transaction below. You need to rate your driver before you can request for a new driver.

I. Rate Driver



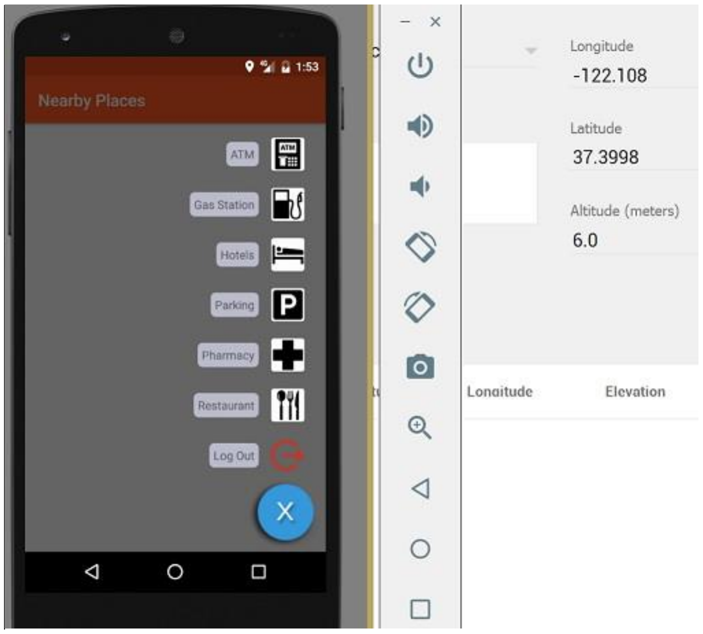
This is the activity for rating glider's driver. The ratings by glider/rider will appear on the drivers' apps. Refer to point "9.Driver Widget Rating" below.

J. Glider Nearby Places



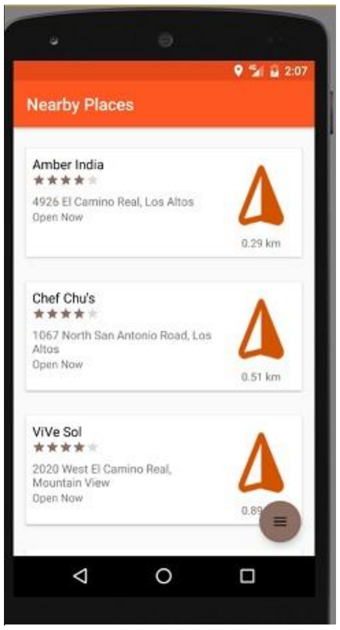
Glider/rider can also use this app to list nearby places.

K. Glider Current Location



Type current location into emulator.

L. List of Nearby Places



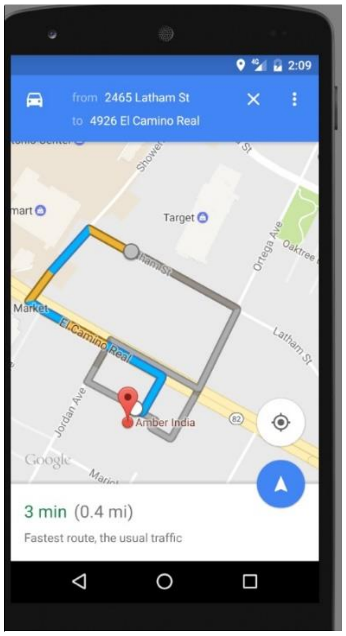
Amber India
4926 El Camino Real, Los Altos
Open Now
0.29 km

Chef Chu's
1067 North San Antonio Road, Los Altos
Open Now
0.51 km

ViVe Sol
2020 West El Camino Real, Mountain View
Open Now
0.89

By using Google Places Web Services API, a list of nearby places will be displayed on screen.

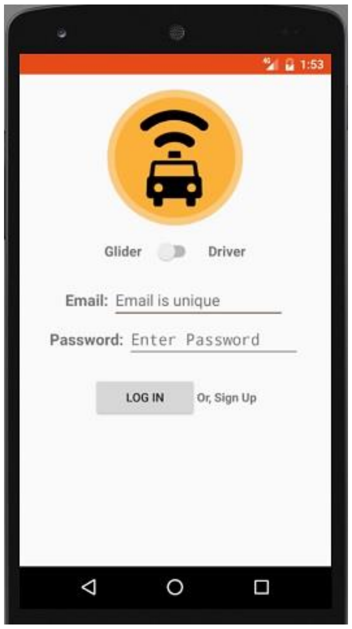
M. Start Navigation Intent



By clicking on the “GPS/Direction” button, an intent will be fired that can be used for navigation.

Driver Mocks

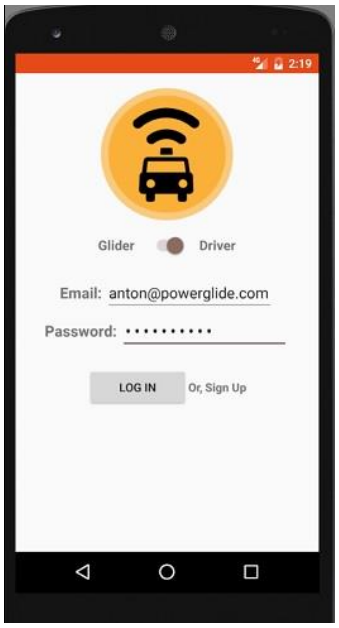
1. Home Screen



Email: Email is unique
Password: Enter Password

This is the home screen for both glider and driver.

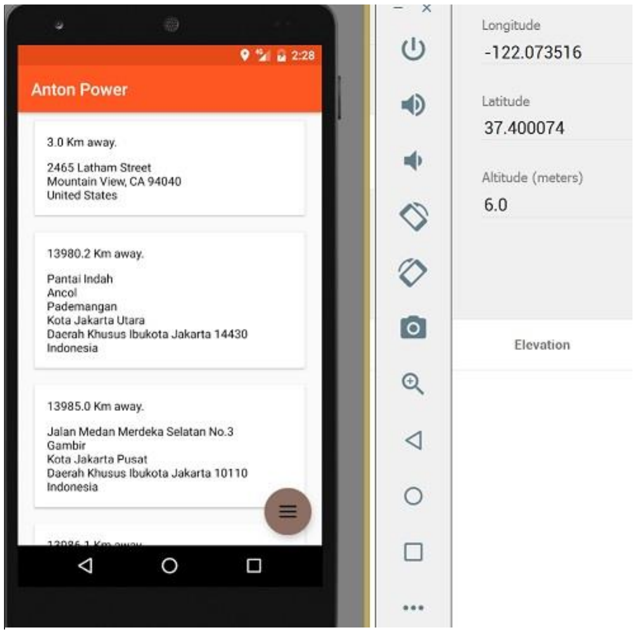
2. Login as driver



Email: anton@powerglide.com
Password: ••••••••••

Slide the switch to the right and login as a driver.

3.Driver view requests



2:28

Anton Power

3.0 Km away.
2465 Latham Street
Mountain View, CA 94040
United States

13980.2 Km away.
Pantai Indah
Ancol
Pademangan
Kota Jakarta Utara
Daerah Khusus Ibukota Jakarta 14430
Indonesia

13985.0 Km away.
Jalan Medan Merdeka Selatan No.3
Gambir
Kota Jakarta Pusat
Daerah Khusus Ibukota Jakarta 10110
Indonesia

Longitude
-122.073516

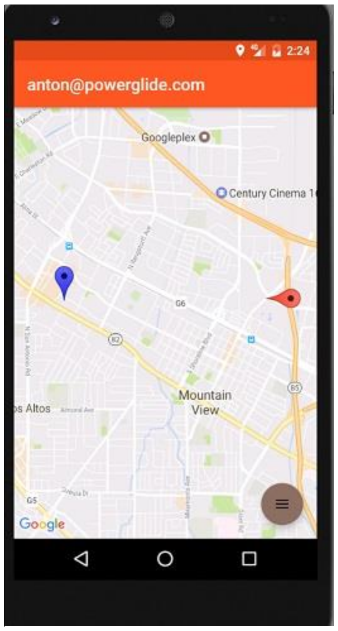
Latitude
37.400074

Altitude (meters)
6.0

Elevation

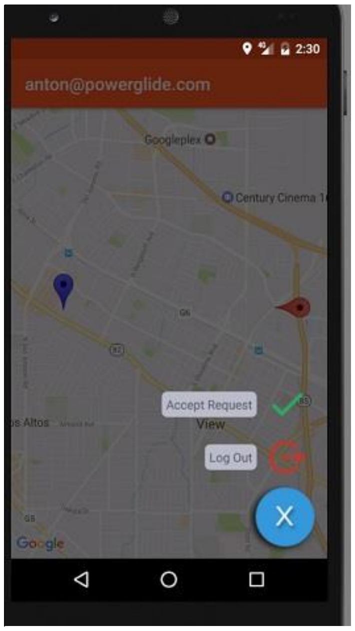
Enter a mock up GPS location into the emulator. A list of requests will appear on the screen. Choose the nearest glider/rider, which is 3 km away.

4.Driver view selected request



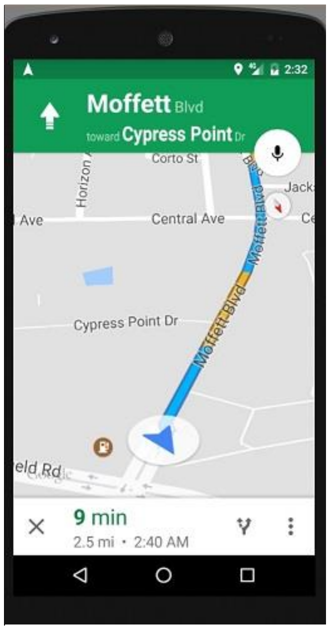
You will be redirected to a new map activity. The blue marker is the glider/rider location while the red marker is the driver.

5.Driver accepts request



Click on the FAB button "Accept Request". After a driver accepts request, the glider's map activity will be automatically updated with the information of this driver. Refer to point "E.Request accepted by Driver" above.

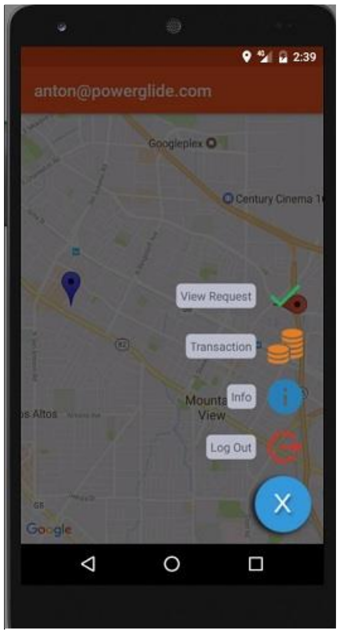
6.Driver navigates to glider



Then a new intent started by powerglide app will guide the driver to glider’s location. The code is given below start a new Intent.

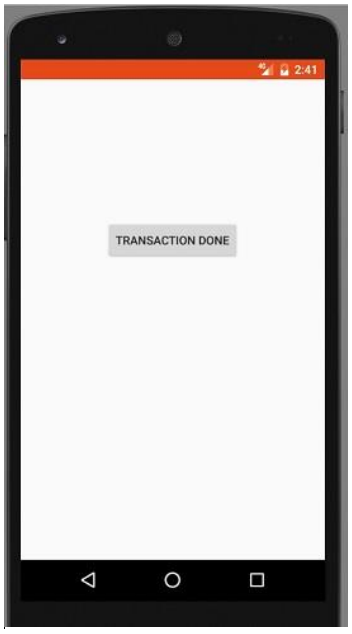
```
Uri directionUri = Uri.parse("http://maps.google.com/maps?saddr=" + driverLatitude + "," + driverLongitude
                                           + "&daddr=" + requestLatitude + "," + requestLongitude);

Intent directionIntent = new Intent(android.content.Intent.ACTION_VIEW, directionUri);
```

7.Click Transaction



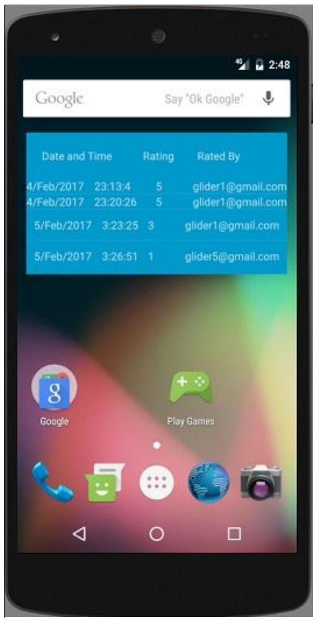
When the driver has driven the rider/glider to his/her destination, the driver needs to finalize the transaction by clicking on the “Transaction” button.

8.Transaction done



Click on the “Transaction Done” button. Then a new “Rate your driver” button will appear on the glider/rider apps. Refer to point H.Rate Driver above.

9.Driver Widget rating



Date and Time	Rating	Rated By
4/Feb/2017 23:13:4	5	glider1@gmail.com
4/Feb/2017 23:20:26	5	glider1@gmail.com
5/Feb/2017 3:23:25	3	glider1@gmail.com
5/Feb/2017 3:26:51	1	glider5@gmail.com

Ratings by glider/rider can be put on the drivers’ home screen.

Key Considerations

How will your app handle data persistence?

Following forum discussion https://discussions.udacity.com/t/satisfying-project-rubric/218671/27, all forum mentors including myself are wondering if I'm able to use a Parse Server as my backend. Therefore in order to satisfy the Data Persistence requirement, I use Content Providers in my driver's app and glider's app. Please tell me if my capstone satisfy the Data Persistence requirement.

```
App stores data locally either by implementing a ContentProvider OR using Firebase Realtime
Database. No third party frameworks may be used.
Must implement at least one of the three
If it regularly pulls or sends data to/from a web service or API, app updates data in its cache
at regular intervals using a SyncAdapter or JobDispacter.
OR
If it needs to pull or send data to/from a web service or API only once, or on a per request
basis (such as a search application), app uses an IntentService to do so.
OR
It it performs short duration, on-demand requests(such as search), app uses an AsyncTask.
App uses a Loader to move its data to its views.
```

Driver's data persistence
The app will pull data from the parse backend. The data pulled is related to Ratings given to Driver by glider/rider. This data will be stored into local SQLiteDatabase via ContentProvider and will be updated on demand. This data will also be shown in app widget. Refer to **9. Driver Widget Rating** screen above. Loader will be used to pull data from SQLiteDatabase into User Interface.

Glider's data persistence
JSON data extracted from Google Places Web Service API will be stored in local database via ContentProvider. AsyncTask will be used to perform JSON data fetching on demand. LoaderManager.LoaderCallback will be used to pull data from SQLitedatabase into User Interface. Also appWidget will be used to display nearby places in glider's homescreen.

Describe any corner cases in the UX.

The app can launch google map intent for navigation. Then the user can return to the app by hitting the back button.

Describe any libraries you'll be using and share your reasoning for including them.

The app will be dependent on parse libraries.

```
compile 'com.parse.bolts:bolts-tasks:1.3.0'
 compile 'com.parse:parse-android:1.13.0'
```

These parse libraries are chosen because I've worked on this library since I was working on Udacity Ubiqutous project and I haven't read the capstone project rubric. A lot of my time has been invested in working with parse library API and setting up the backend in Amazon cloud. I spend time with Parse Server backend because I know I wanted to work on UBER–inspired app for the capstone. This app requires a backend. I chose Parse Server.

To satisfy the requirement below, I also make use of Google Places Web Service API. The JSON data returned by the API will be stored in local database. This data is nearby data of glider/rider. ContentProvider will be used to store data. Loader will be used to pull data from database into User Interface.

```
App stores data locally either by implementing a ContentProvider OR using Firebase
Realtime Database. No third party frameworks may be used.
```

Describe how you will implement Google Play Services.

This app uses Google Play location and map services extensively. Therefore Google Maps Android API will be used. Also Google Places Web Service API is used. This will enable google to return JSON nearby placed data to my app.

Next Steps: Required Tasks

Task 1: Project Setup

Learn the Google Web Service API
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=37.3997983,-122.108&radius=5000&type=restaurant&key=<enter_api_key_here>

- Learn what JSON Data to parse
- Setting up what column data to use in SQLiteDatabase
- Write up Content Provider contract and provider

Learn how to use Google Location and maps
- Learn how to set up permission to use location and maps

- Learn to use GoogleApiClient
- Learn how to use Location callbacks and map callbacks

Learn how to use Parse Server
- Set up an amazon server to host Parse Server Backend
- Learn how to login and save data to Parse Server Backend

Task 2: Implement UI for Each Activity and Fragment

Setting up various dialog and fragments
- Because Nearby places for restaurants, parking, café, pharmacy, gas stations need to be displayed, fragments need to be set up.
- Search for icons, logos to be used for each fragment

Create Login page
- Switch widget is used to choose between Glider/Rider and User
- Search a nice logo to use for the app

Create a Rating and Driver Info Page
- Glider/Rider will be able to view driver info
- Glider/Rider will be able to give ratings to driver

Task 3: User Friendly and Intuitive User Interface

Create a user friendly user-experience
- Inform user if the is no internet available
- If there is no internet available, user still can view nearby places data
- Handle errors when location is not available

Create an intuitive UI for user
- Glider/Rider can only request one driver at a time
- Glider/Rider can cancel request on demand
- Glider/Rider must give a rating before he/she can request for another driver.
- Glider/Rider can view timely driver’s location. The map will be updated every 20 seconds