Fuel Finder

CS 4220 - Embedded Systems
Fall 2022
Final Report

Zhi Wei Chen
Garrett Cooley

Motivation

Refilling gas is an essential activity, however, unlike restaurants, there are not any dominating recommendation applications for refilling gas. In this project, we want to build a gas station recommendation application based on real time data similar to the restaurant recommendation from yelp. We felt that an application for gas station recommendation is long overdue. For big cities, there could potentially be many gas stations clustered together, our application can help users decide which one is the best for them.

There are many variables in play when it comes to choosing the right gas station. Our application seeks to consolidate information about surrounding gas stations. FuelFinder provides a concise interface for users to generate and access information about gas stations surrounding a desired location.

Related Works

There are a few applications out there that provide gas recommendations. Two very popular applications are Google maps and GasBuddy. Although Google maps also provides all three of the important factors listed before, it automatically recommends users to the nearest gas station, and does not take into consideration the price and rating. However, it provides a quick way to get directions to the station once a good station has been found.

GasBuddy, on the other hand, does take into consideration distance, expense, and rating. The application itself is incredibly difficult to use. The UI is very flashy with lots of unnecessary information, and the application itself is cluttered with advertisements and offers. Other applications have similar issues.

All of the applications on the market currently have both pros and cons, while Google maps provides users with a clean and simple UI, it fails to take into consideration other important factors when choosing a gas

station. Gas station focused applications like GasBuddy instead use flashy UI and advertisements to provide revenue for the developers.

System Architecture

We propose to build a solution that contains the best features of the two aforementioned alternatives. Our application will have the following attributes:

- Map
    - Using Google Maps API for Android, the map will initially show the user's current location. When the user has entered their desired coordinates, the map will display the results and the user will be able to zoom in and out to examine the resultant locations. They will also have the option to pull up the location within Google Maps on their device for directions.
- Results List
    - Under the map will be a table populated with the results of their search. This table will contain the name of the gas station, distance from input coordinates, address, phone number, and city. For this purpose, we will be using *Xavvy Gas Station POI Data USA,* which allows us to access the above information and more, such as latitude and longitude coordinates.

To achieve this purpose, we have implemented a mobile application for Android 8.0 Oreo. We will provide the user with an intuitive, easy-to-use frontend that will allow them to input their set of desired coordinates and navigate to the Map and List views, respectively. Overall, Fuel Finder is comprised of three principal views: landing page, map view and results list. Below is a high-level diagram of our system architecture. It depicts the interaction between the various components of our system.

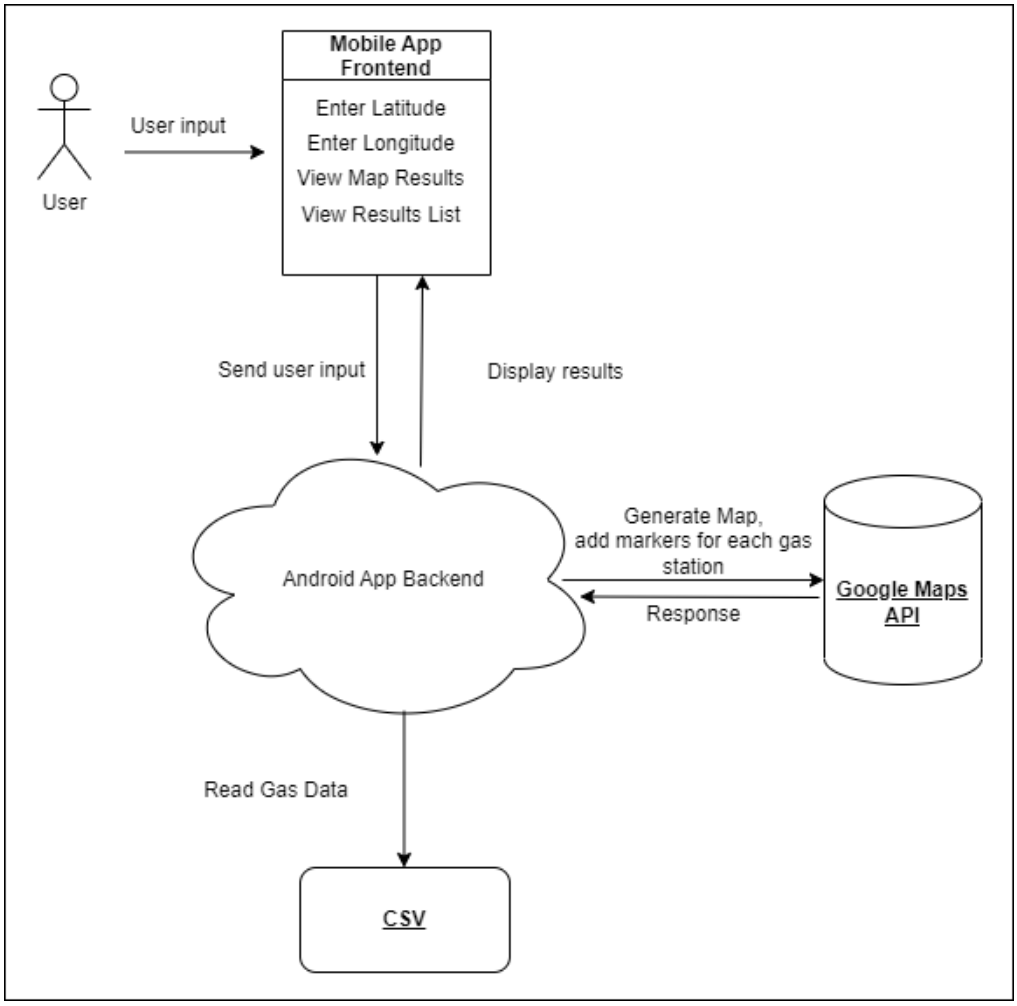

Figure 1. System Architecture.

Implementation

Google Maps API

Fuel Finder uses Google Maps API for the integrated map component of our application. We have created a project on the Google Cloud Platform (GCP) and generated an API key to gain access to the API's functionality. This allows us to instantiate a GoogleMap java object in the backend of the application, add markers for each nearby gas station, and display device

location on the map. It also allows us to enable map zoom and drag functionality, and click listening for each marker. Upon click, a pop up will appear showing the name of the gas station that corresponds to the clicked marker.

Database

We make use of the *Xavvy Gas Station POI Data USA* database of gas station information. This resource provides gas station data in the form of a comma-separated value (.csv) file. One important consideration for this project is the distinction between the free and premium versions of this database. We did not have the funds to access the premium version, so we made use of a free sample subset of this database. This sample only contains gas stations and corresponding information in the state of North Carolina. Therefore this version of Fuel Finder is localized to that state. However, our application is compatible with the premium version of this database, which has more data fields and live updates. Overall, the reason we do not provide more data is an issue of cost rather than capability.

Frontend

Our priority when designing the user interface of Fuel Finder was ease-of-use, clarity, and fluidity. There are three views that the user will interact with in our application. The first is the landing page. As discussed above, this is where users will input the coordinates around which the gas station search will be conducted. Further, it contains two buttons that display the map view and list view. Depicted below are screenshots of each of these three pages generated from a sample set of lat/long coordinates (35.2, 80.8) near Charlotte, NC. In the case of the map and list views, there are multiple screenshots for each one in order to demonstrate their full functionality.

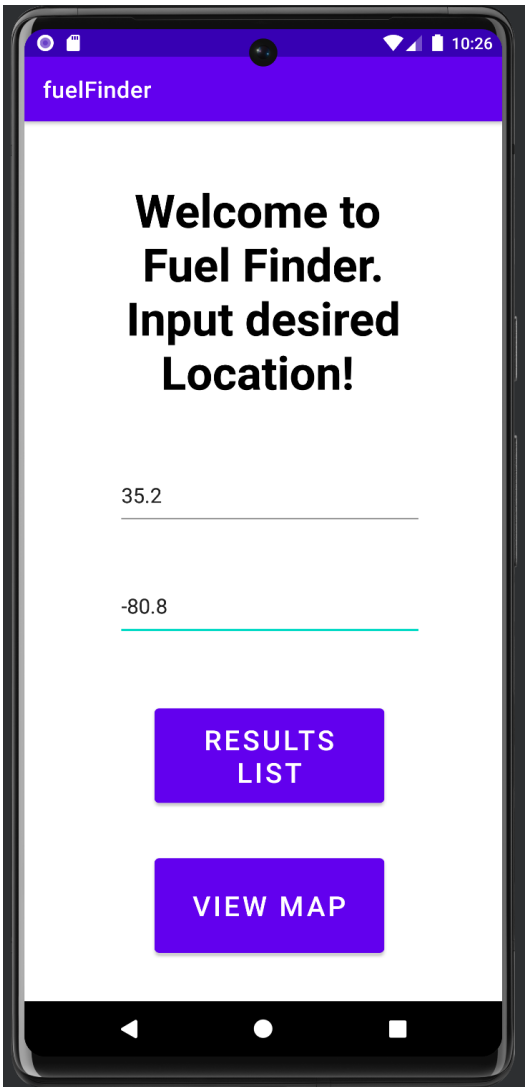

Figure 2. Landing page with sample coordinates in NC.

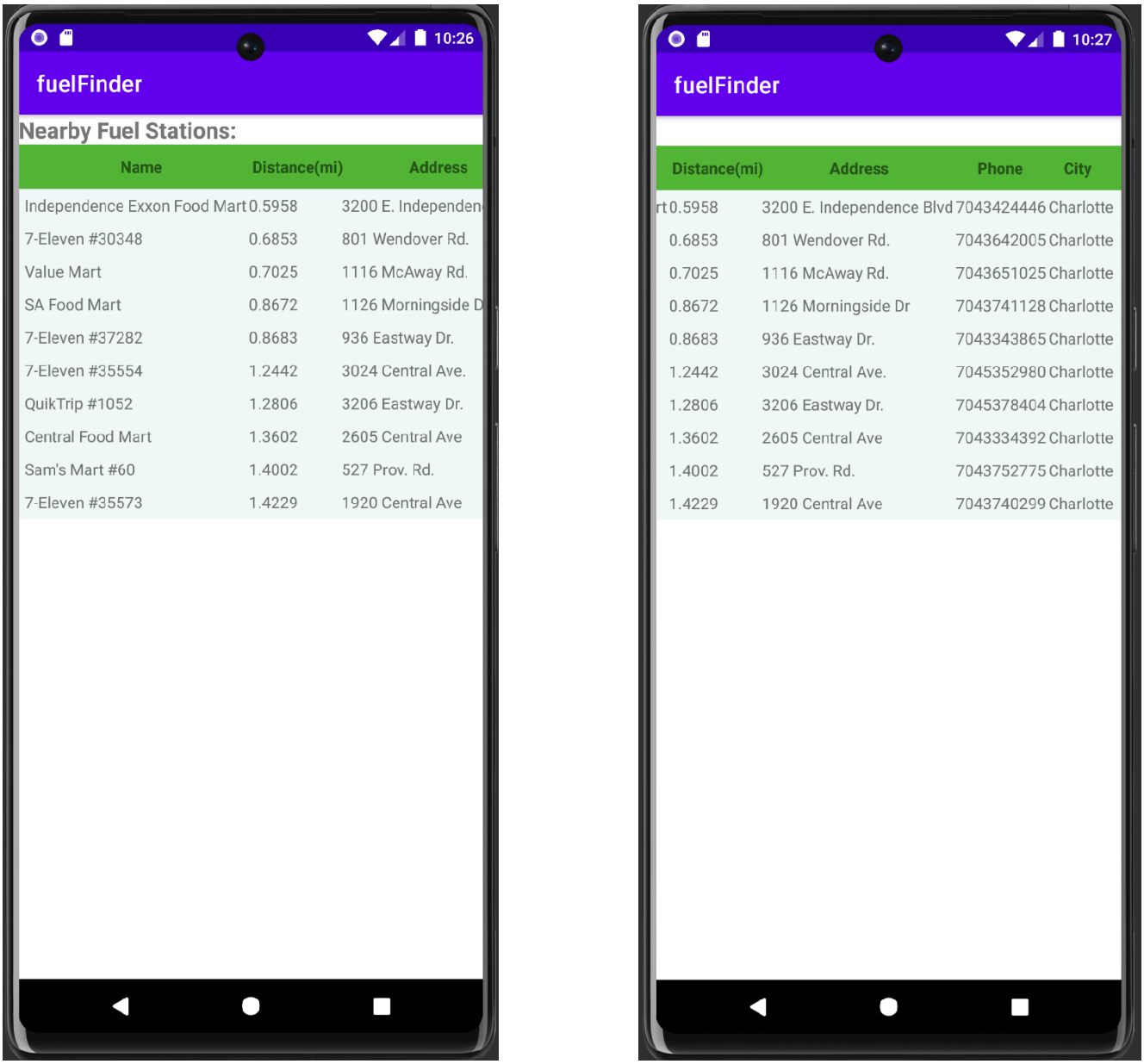

Figure 3. Listview displaying ten closest gas station names, distances away from coordinates, addresses, phone numbers, and cities.

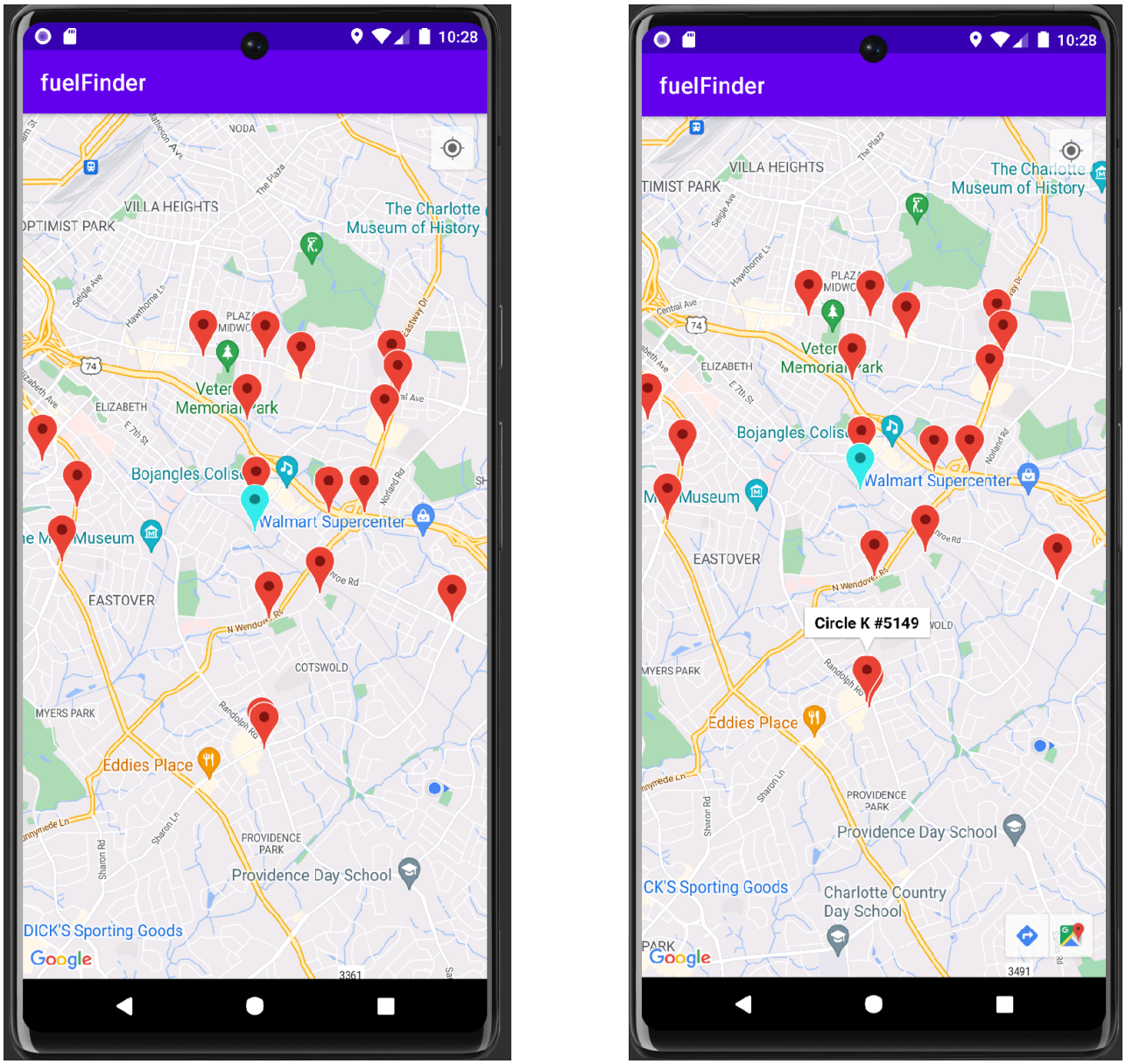

Figure 4. Map View before clicking marker (left) and after clicking marker (right). Teal marker corresponds to the search epicenter that user input at landing page.

Backend

Below is a class diagram of our system. The classes *MainActivity, MapActivity, and Result* contain the backend functionality for the landing page, map view, and list view respectively. *GasStation* is a model class used to extract gas station data into a java object for use in the application.

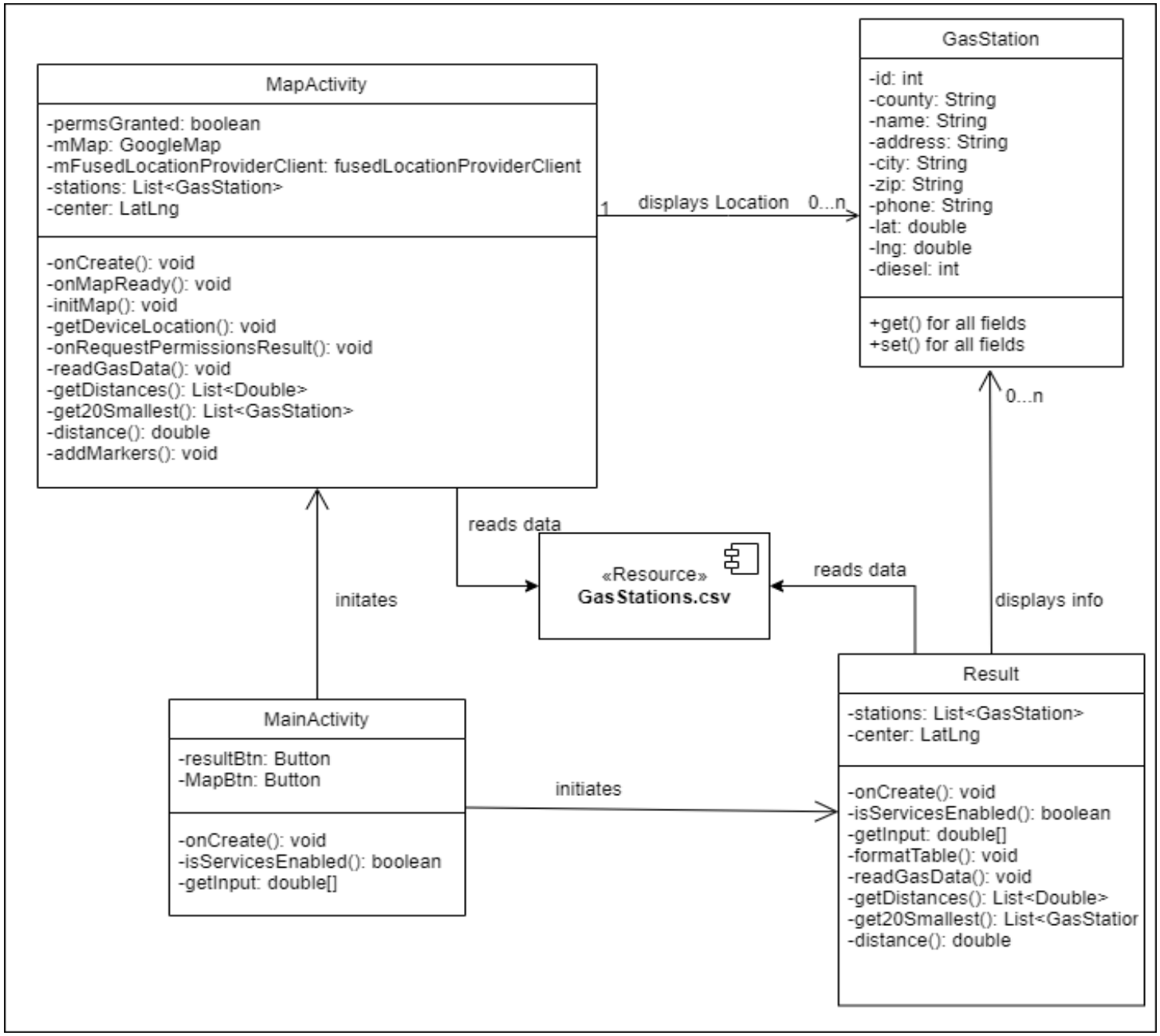

Figure 4. Class Diagram

MainActivity and *GasStation* are the simplest of the classes depicted above. The instance variables of *GasStation* correspond to each of the value types within the GasStations.csv file provided by Xavvy. *MainActivity* defines the functionality of the two buttons seen in figure 2. Additionally, it parses Strings from the latitude and longitude text fields and converts them to primitive double values. These double values are then passed to either *MapActivity or MainActivity* as a *LatLng* object.

MapActivity accepts the coordinates as a *LatLng* object called *center.* From here, it initializes the *GoogleMap* object called *mMap.* It also prompts the user for location permissions if not already granted, then displays user location on the map, as seen in the lower right area of figure 4. Upon creation of the map, *MapActivity* reads the *GasStations.csv* file and populates a *List<GasStation>* instance variable called *stations*. Then, it iterates through *stations* and calculates a corresponding distance value for each value in the list. These distances are calculated using the *distance()* function, which is based on the formula for the distance between two latitude and longitude coordinates. From these values, *MapActivity* finds the twenty smallest distances and displays each of those gas stations as markers on the map. One additional marker is created to denote the epicenter of the search.

Result reads *GasStations.csv* and calculates the distances in the same way as *MapActivity.* Once this step is completed, it formats the data from the ten closest gas stations to appear in the list view as shown in figure 3.

Challenges

While completing the project, our team encountered several difficulties. One of which is a strange error that occurs on google maps that causes the device to freeze after allowing permission. After thoroughly checking and searching for a solution, our team was able to finally proceed after switching to Android 8. This decision resulted in a far more stable and functional application.

One other major difficulty we encountered during working on the project is the exclusion of gas station prices on google maps. Google maps API only provides a price level on a scale from 1-5. This was not what we envisioned during the inception of the project. As a result, we chose to use a different source for our gas station data. We chose *Xavvy's Gas Station POI Data USA.* For the scope of this project, we use a free provided sample subset of data that only contains station information in the state of North Carolina.

Future Work

Our application can definitely still be improved. One update that the application needs is the option for users to input prices they encounter during their gas station. This information will be stored on an online database and allow our application to display gas station info without having to need to use any gas price api.

Another future update we plan on adding to our application is working to get a paid live version of gas station api, this would allow the application to update live and not rely on historic data and api with fixed data. We could also update the application to include more information about each gas station, such as the kind of pump and whether or not it has a convenience store and restrooms. We could also take into consideration more factors when giving recommendations, and perhaps even implement a filter system to filter out the gas stations with low ratings. In addition, we also need to develop the same application for iOS systems.

Deliverables

The deliverables for this project are:

1. Mobile application source code
2. Presentation materials
3. Presentation Video with Demo
4. README.md instructions for running application

References

[1]https://www.nytimes.com/wirecutter/blog/ways-to-find-cheapest-gas/
[2]https://www.11alive.com/article/traffic/gas-prices/gas-prices-georgia/85-0b0966e4-05f6-42b1-ab2c-8939168d90a3
[3]https://www.partoo.co/en/enterprise/blog/2021/07/02/gas-station-pricing-on-google-maps/