

# **Just Park It! Android app using Google Maps and SFPark API**

## **Executive summary**

The main goal of the project was to develop an Android parking app with the basic feature of displaying available parking locations using the Google Maps API and the SFPark API. We were also required to demonstrate that data was being stored to and retrieved from the SQLite database. Apart from these basic features, we have implemented a unique feature of displaying street cleaning data when the user clicks on a location. Once these high priority functionalities were completed, features like color coding of street cleaning data, search bar, radius filter to display parking locations, Add to favorites, GPS status check, Clear Markers, Layers, and Help were implemented. One of the distinguishing features of our app is the Street Cleaning database, which contains more than 37000 rows of data.

Agile methodology promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change. All these factors were clearly seen during the course of development of our application. Adaptive planning was required to meet with the change in specifications we had at the beginning of our app development. As we had to rapidly develop our app, evolutionary development was a part of the process. Since each set of milestones had to be met in a week's time, there was scope for early delivery and continuous improvement. We incorporated the feedback from our Professor after each milestone, and so the app was flexible to changes. We were also able to improve the performance of the app over time. Thus by following the agile methodology, we were able to practice teamwork, collaboration, and adaptability throughout the development of the application.

In the beginning of the semester, the whole class participated in brainstorming sessions and came up with many interesting ideas for our class project, and all 70 students were expected to work on one single project. The whole class was divided into 10 groups of 7 members each. Once it was decided that the project would be an Android parking app, each group came up with a set of specifications for the app. Professor helped us with the herculean task of combining the specifications of all 10 groups into a single set of specifications for the entire class. Before we actually began working on the project, we had a sample assignment to become acquainted with Subversion for code sharing, and it was then that we started to realize that it might not be a good idea for the whole class to work on a single project. Before it was too late, we agreed that each group would work on their own individual app with GitHub as the code repository. Considering the limited amount of time left, we scaled down our initial set of specifications by pushing all priority 1s to priority 2s and so on. Our 1st milestone was just a basic app with working interfaces to the Google Maps API and SFPark API. Each week we had some milestones set and everybody worked hard to meet the requirements. By the end of the 2nd week, we had a database connection established to handle street cleaning data from the SFSU area with approx 3000+ rows of data in the table. When a user clicks on the map, the street cleaning schedule for both sides of the street is displayed on the marker. Display of this street cleaning data is the unique feature of our app. We then added additional priority 2 features to make our app more interesting. We were successful in implementing some of the lower priority features during this project.

## Sketches and Screenshots

Figures 1 through 3 are our initial wireframes which were created right at the beginning of the project. Initially, we wanted to implement a login for users but due to time limitations, we had to exclude that from our project. We also planned to implement street type, start and end time, max price and max distance filters. Of these options, only the max distance filter was finally implemented. The initial wireframes also included other features such as 'Find Car' and User reviews, none of which are part of the final application. Note that display of street cleaning information was not listed in these initial wireframes.

Figures 4 to 17 demonstrate our final app features. We implemented an action bar with an action overflow that contains all the menus (Clear Markers, Layers, Help, View Favorites, and Parking Radius Settings). We kept the functionality simple by having users tap anywhere on the map to display both parking data from SFPark and street cleaning data from the database. If parking data is available within the selected radius, markers with the letter 'P' appear on the map. If the user taps on one of the 'P' markers, the marker will display parking information such as type of parking, price per hour, current garage availability and garage contact information.

### **Initial wireframes:**

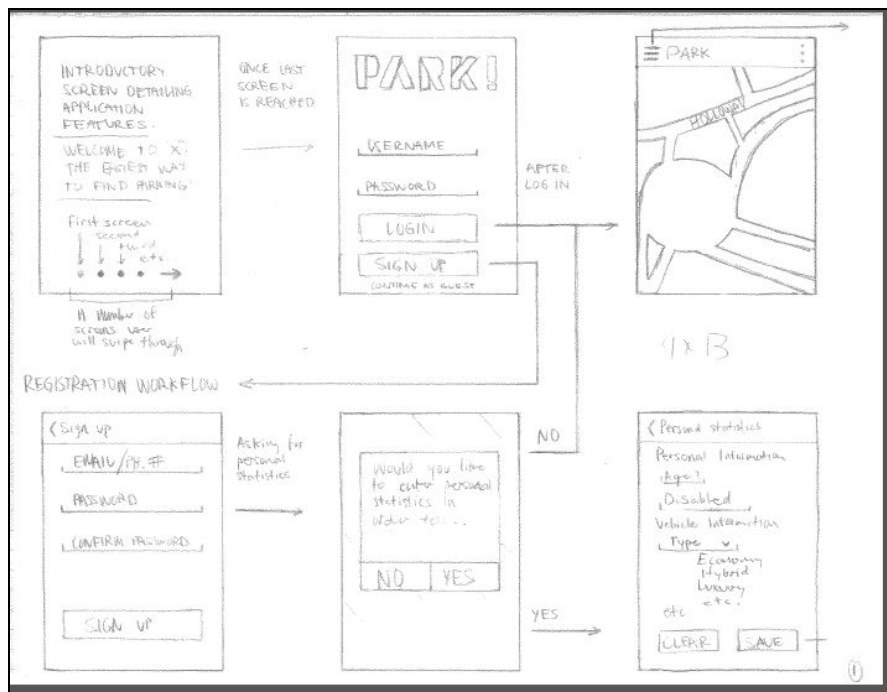


Figure 1: Initial Wire Frame Diagram 1

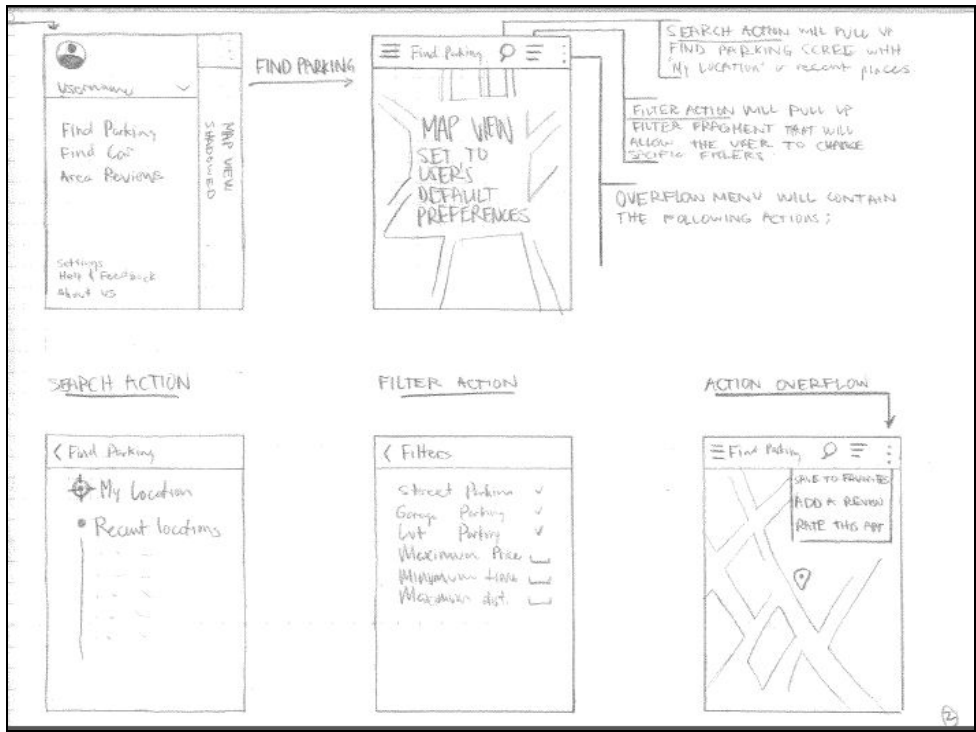


Figure 2: Initial Wire Frame Diagram 2

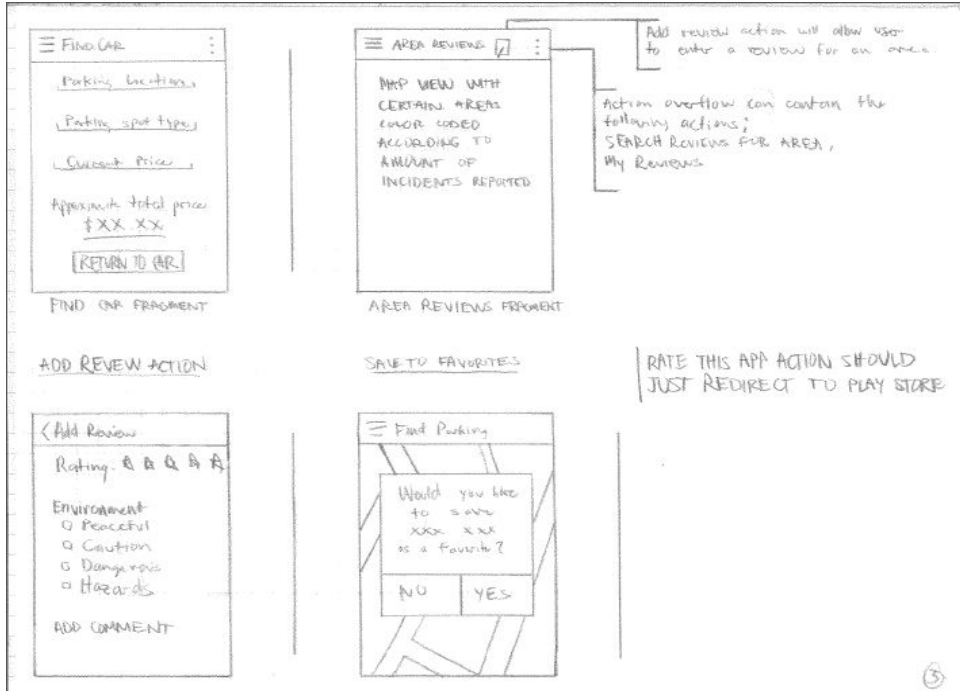


Figure 3: Initial Wire Frame Diagram 3

# Final screenshots from the app:

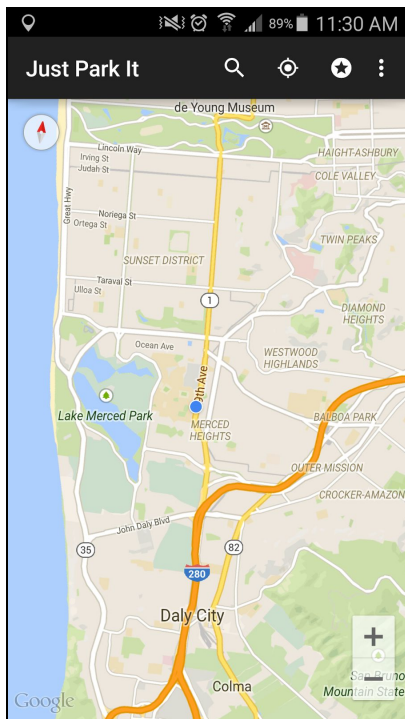


Figure 4: Main Activity

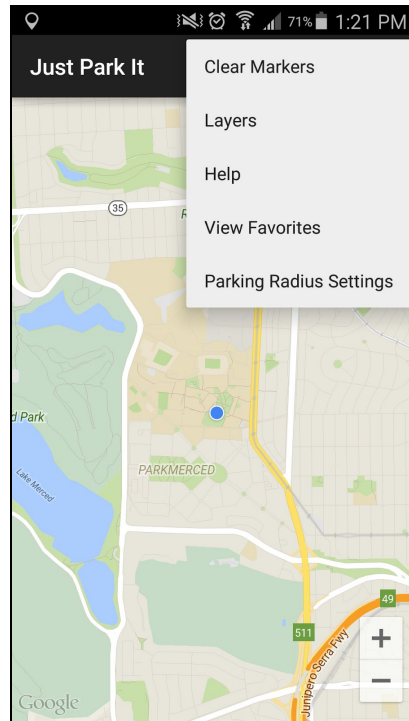


Figure 5: Action overflow menu

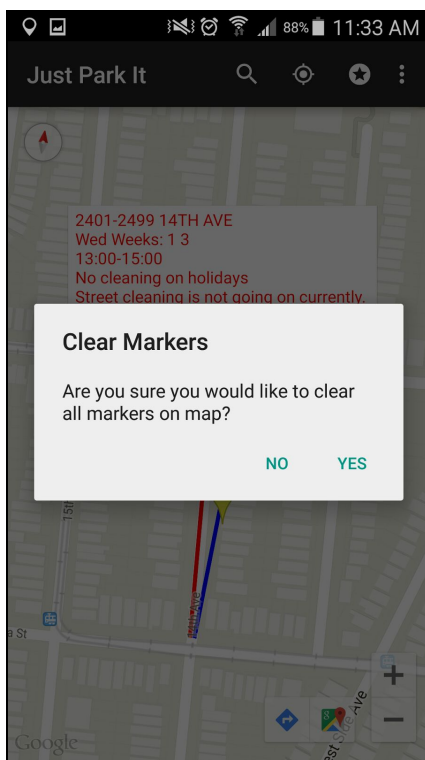


Figure 6: Clear Markers Dialog

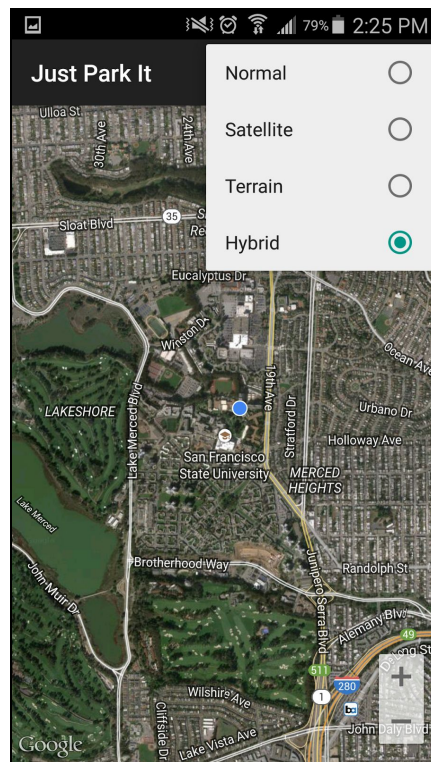


Figure 7: Layers menu

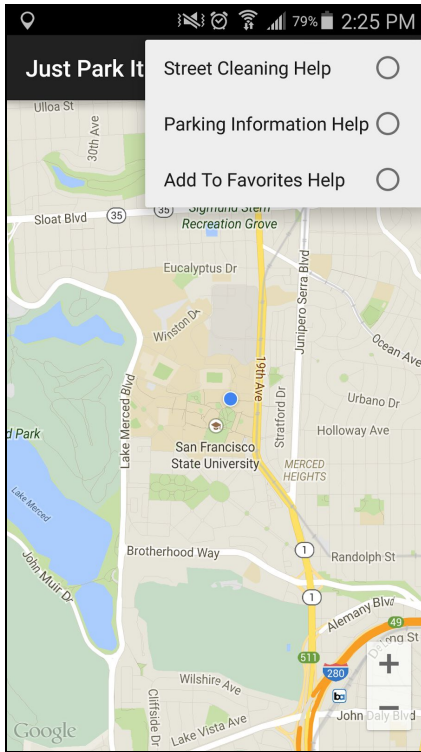


Figure 8: Help Menu

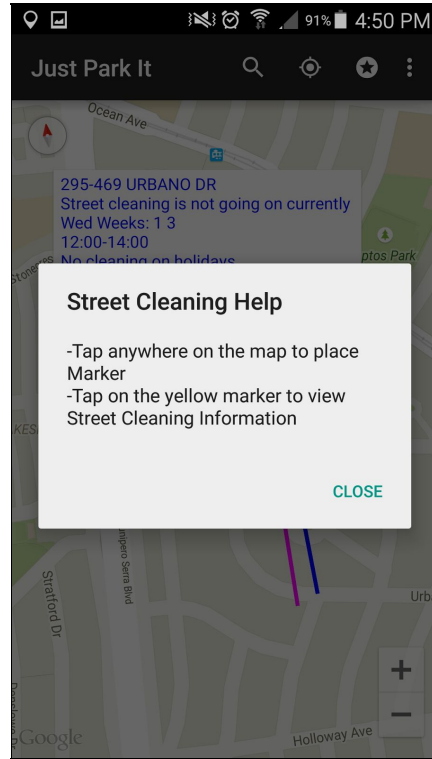


Figure 9: Street Cleaning Help

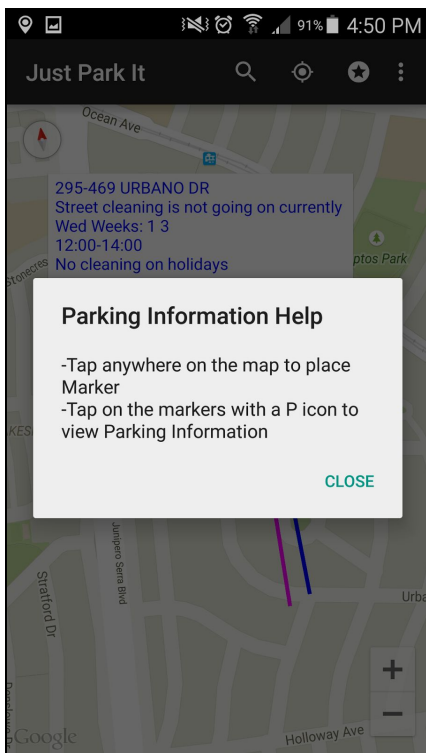


Figure 10: Parking Info Help

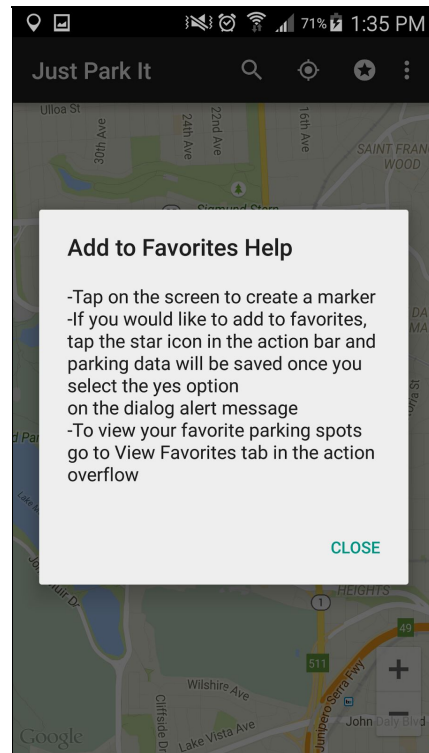


Figure 11: Add to Favorites Help

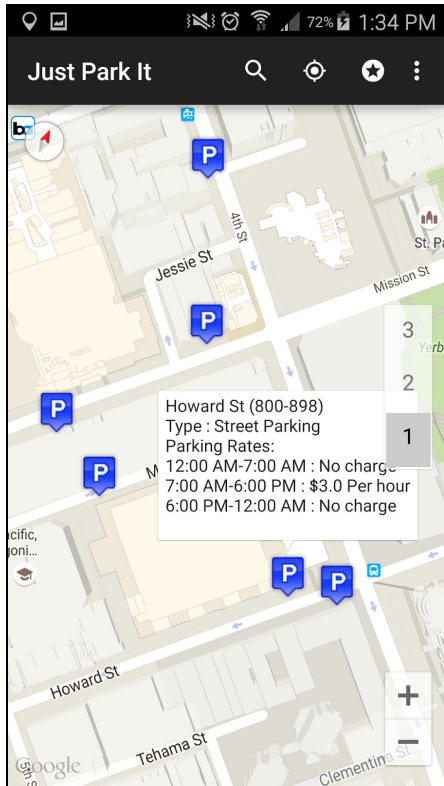


Figure 12: Finding parking information on map

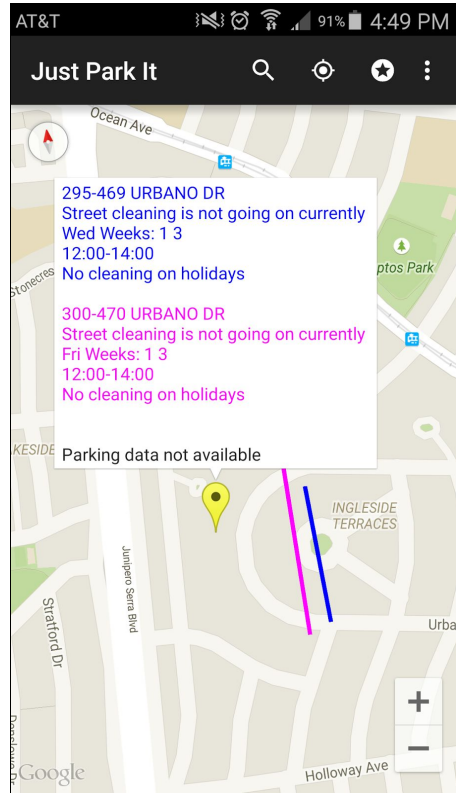


Figure 13: Street Cleaning Data

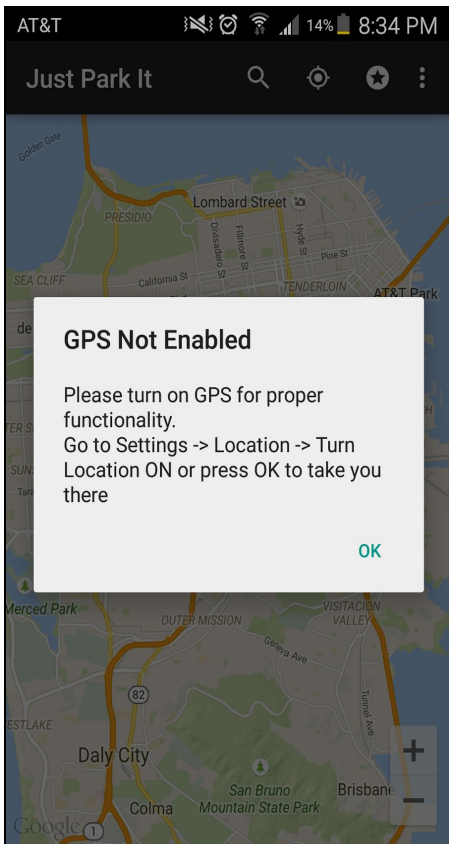


Figure 14: GPS Status Check

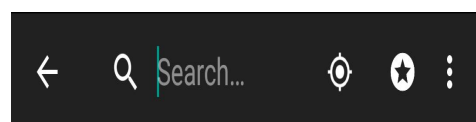


Figure 15: Search Option



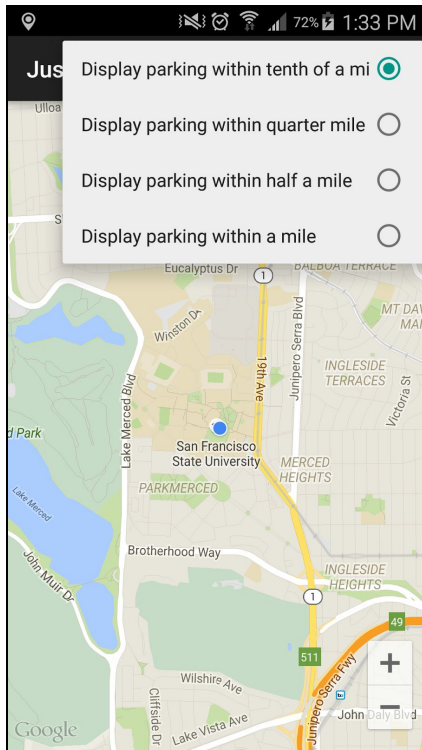


Figure 16: Radius Filter

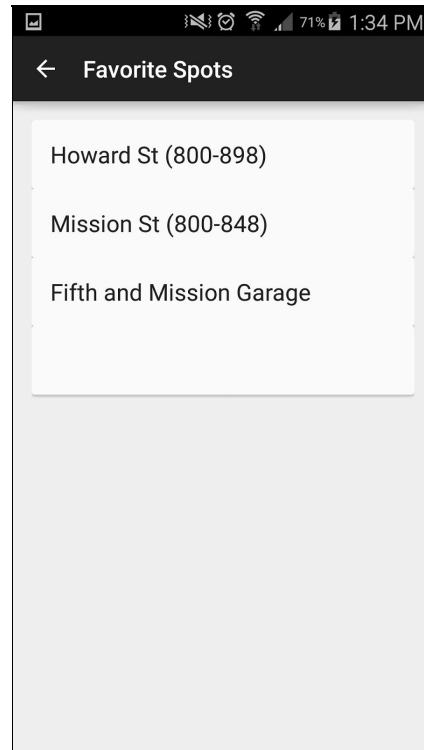


Figure 17: Add to Favorites

### API Usage

1. **SFPark API:** We used the availability service of SFPark API to find parking locations at the required latitude and longitude and within a given radius.
2. **Google Maps API :** We used geocoder API to convert address in text form into latitude and longitude coordinates while implementing the search box, and also to convert the latitude and longitude coordinates to address format while displaying the street cleaning data.

### Team Members

1. Aman Arora
2. Pooja Kanchan
3. Omar Sandoval
4. Siddhartha Sankasala
5. Raymond Thai
6. Nayana Kamath J.
7. Kavya Katipally