Requirements Specifications

1. Domain

The city of Chicago has a significantly high crime rate that is above the US average. In particular, the number of car thefts in the city has been on the rise since 2015. People want to avoid falling victim to these crimes and want to keep their vehicles as safe as possible. SafeParkingZones is an application that aims to help drivers find safe and authorized parking zones to park their vehicles.

Stakeholders

- Drivers: Drivers are the primary users of this application. Their goal is to find a safe place to park their car. Their expectation of the application is that it recommends them safe parking zones that are a reasonable distance from their current location.
- City of Chicago
- Owners of the app (Group 10)

Shareholders

- Professor- Dr. Reza Samavi
- Teaching Assistants for CS/SE 2XB3
- Group (10) members

Main Entities

- User: Represents the person interacting with the application. It will contain the user's current location.
- Parking Zones: Represents authorized parking zones in Chicago. It will contain exact locations (latitude and longitude) of authorized parking zones in Chicago.
- Vehicle Theft Areas: Represents the various vehicle theft area coordinates in Chicago.
- Map: Graphic representation of the city of Chicago. Its main relationships are with the user and parking Zones entities. Using the user and parking zone locations, it will display the nearby safe parking zones on the map in relation to the user.

2. Functional Requirements

Detect and update the user location

- The app should be able to continuously detect the searched location by the user and show it on the map using Google Maps API.

Check for availability of a parking spot on a street

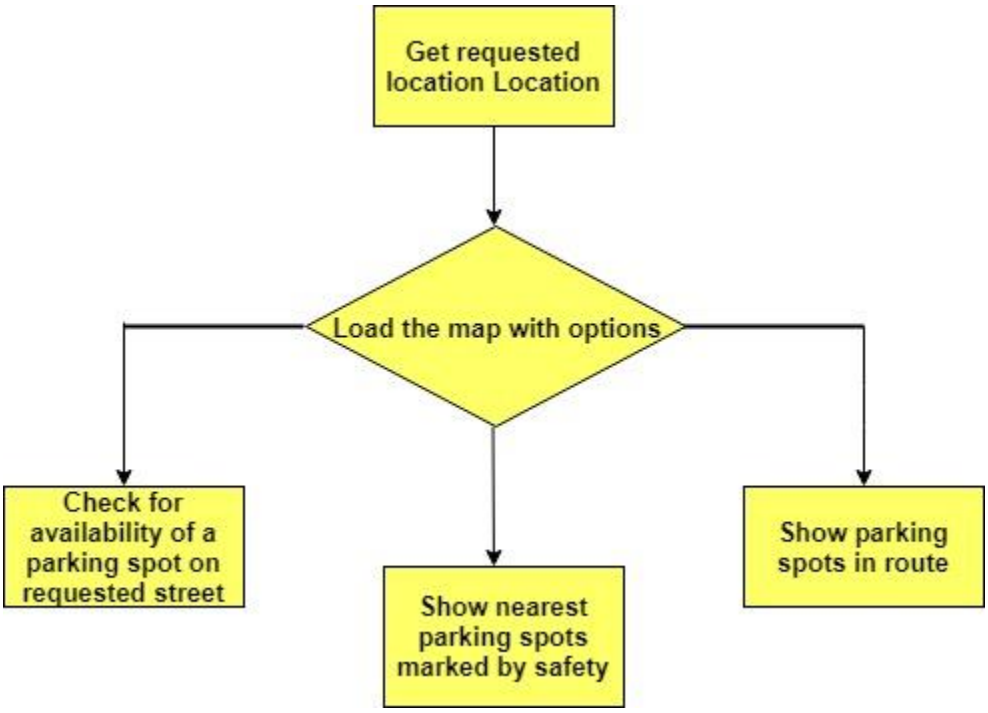
- The app should be able to update the user about the availability of a parking spot on the requested street.

Update Best Available Parking Spot

- The app should be able to update the user about the nearest parking zones marked by safety depending on their requested position by using the motor vehicle theft datasets that provides the locations where the crimes happened. It also provides a list view of addresses of those parking spots.

Parking Spots in route

- The app should be able to update the user about the parking spots in route from a requested source to a destination. In case there are no parking spots in route, the app shows available parking spots near the destination.



3. Non-Functional Requirements

Utility Requirements

- The app should be fast and easy to use (specifically when looking for an available parking spot on a street), as this app will mostly be used by drivers who are checking for a parking spot while in a safe stop area, this means they will usually be in a hurry and expecting a fast service. It is imperative that the app is reasonably fast in order for our users to be able to find their parking spot quickly.

Availability Requirements

- The app is available 24/7 as long as the user has a reliable internet connection.

Integrity Requirements

- The app does not track the user location, instead it asks the user for the locations.

Accuracy Requirements

- The app should be 98%+ accurate when it comes to finding parking spots within 2 km radius in order for users to easily park near their current/selected location.

Human-Interface and Portability Issues

- While the app is built to provide results as fast as possible, the user may be subject to short loading times in order for the app to be able to find the safest parking spots closest to them.

- Since the app will only be available on Android, users from other mobile platforms will be unable to access the app, but the backend of the application will be available to everyone as long as the user has eclipse IDE installed on their device

4. Requirements on Development and Maintenance

Quality Control Procedures

- This will be done by checking the integrity of the datasets and making sure that the database is consistent, clean and error free. All the incomplete and unnecessary values will be removed.
- Moreover, all the functional requirements need to be checked against the user requirements (Drivers in this case) to validate the fact that the app is doing what it is supposed to do. It is also crucial to verify that the app is being developed in the right manner, i.e. by making sure that it is following the specification correctly.
- The code needs to be tested against various cases to make sure that the app works for every user input.

Priorities of The Required Functions

- During the development cycle, it is critical to identify the focus of the app and to clarify the goals. The main aim of the app is to provide the safest parking zones therefore it is essential to make sure that the app provides that functionality to the user
- The app needs to be developed by keeping future changes in mind, which in this case can be adding a new feature or improving an existing feature.

Likely Changes to Maintenance Procedures

- There can be features that can have minor issues in future, therefore it is necessary to make sure that such issues are fixed. User reported issues also need to be fixed to ensure smooth functioning of the app.
- In the end, it is important to continuously review and revise the functioning of the app.