Software Requirements Analysis and Design

COMP3059 – Capstone I

Justin Rolnick

Lenny Ramroop

Albert Nguyen

Jeremy Yang

# 

# 

# 1.0 Introduction

We are creating an application that will help students and visitors find parking when arriving to George Brown College Casa Loma campus. We have several requirements for the app that will define its success upon completion.

This document provides detailed information about our app 3Park. The document will explain why we will create this app, who are involved in it, and what features our app will contain. The document will also help clarify any questions and queries that stakeholders may possess. The speed of our project will be based on our success on what is listed below.

## Purpose

The purpose of this document is to present a detailed description of the 3Park hybrid mobile application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to user input. This document is intended for both the stakeholders and the developers of the system and will be proposed to Anjana Shah - professor at George Brown College - for its approval.

## Scope

The purpose of this app is to find parking spots for students and visitors who are driving to Casa Loma campus at George Brown College. Upon opening the application, users will be prompted to login to an account either through Facebook or Google, but it will not be required in order to use the application. Our application will include the ability to locate paid and non-paid parking spots around the Casa Loma campus. It will display this through the use of the Google Maps API contained in a simple, clean UI/UX from a top-down aerial view which will make it easy for users to scan areas. Upon map scans, we will have a dynamically updated list of available parking locations displayed based on the area of the map that is visible on the user’s device. The users have the ability to search for addresses/streets in our application and then navigate to their desired location through the use of their own preferred navigation application on their smartphone. The application will implement a history below the search bar of all past searches by the user.

Furthermore, the application will allow users the ability to view allowable parking times during anytime of the day, display the parking information for the streets around the Casa Loma campus and save the location to a favourites section. The application will have a current location button that will center the map view to your current location based on the user’s device GPS. If users do not have access to a mobile data plan or wi-fi, they will be able to download offline maps and still be able to use the map without an internet connection.

The processes and system that are not affected or influenced by this document are accessibility parking spaces for handicap people because there are very little accessibility parking spaces around Casa Loma. Users will not be able to input data regarding parking locations as it may lead to abuse of municipally-owned parking spaces as well as inaccurate and misleading information to other users. The use of dynamic suggestive history based off user’s most frequent searches will also not be included due to time and technical constraints as well as it not providing much value to our application.

# System Overview

## Project Perspective

3Park is a brand-new self-contained system. Utilizing a self-contained-system approach, we will eliminate any possibility of the application growing too large to maintain, and keep it simple to continue future development. It will link to external sources, such as the host/database, Google Maps API, and receive input from the user.

## System Context

The system in which we are building our application will be in React Native, with the purpose of deploying for both Android and iOS operating systems. There will be multiple actors in the operation of the application. The first, and most important, would be the end user. This is who the application is developed for. Next will be the administrator, who will be overseeing issues pertaining to the accuracy of information and any bug reporting submitted by the end users. The database will served using Amazon Web Services and written in mySQL, eliminating the need for any paid licenses for commercial use of a database language.

## General Constraints

**API and Datasets**

The application relies heavily on the use of Google Maps API, and the datasets provided by the City of Toronto Data Catalogue.

**Datasets Updates**

The data contained in all of the XML and JSON files must be updated regularly to reflect the current parking availability.

**Reliable Server**

There must be a reliable server in place in order for us to properly test our core functionalities of this app, namely the search feature.

## Assumptions and Dependencies

We assume the Google Maps API that we will call on will be updated periodically by Google and the API will be simple to implement into our application. We are dependant on these updates in order to provide reliability to our end users.  
  
 We are also assuming that the data contained in the datasets from the City of Toronto Data Catalogue are up to date and holds the majority - if not all - of the required data. This application is dependant on this data; without this data the workload will increase significantly.

We must create a function that will allow us to organize the string fields contained in the datasets logically in order to be put into our own database tables and interpreted by our app.

We believe that the handicap do not need any special accommodations to use our app, as it is a miniscule demographic that we choose not to target.

## 

## 3.0 Functional Requirements

### 3.1 Features

### Search

### The search function will allow the user to search for a specific location. The user will input where they would like to park. The app will then process and look for the street or address entered. The app will then display on the map the location searched by the user and display additional information about that parking location.

**Navigation**

The navigation function will allow the user to choose whether or not they wish to be navigated after a search. The user will be prompted to choose a navigation application of their choice. The app will look for the smartphone’s default navigation application, and if one does not exist, will prompt the user to download an application. If the application does exist it will take the user to the navigation application and display the directions to the searched location.

**Dynamic map listing**

This function will allow the user to see available parking areas for any location displayed within the current map window. The user will drag the map to a specific area. The map will then process the area and all the streets currently displayed on the screen. The map will then display all available parking locations.

**Favourites**

This function will allow the user to save regularly visited streets and locations to a Favourites tab in the application. The user will search for the street they wish to favourite and click on the Favourite button. The application will then register the location chosen by the user to their Favourites tab. The user will now be able to refer to their Favourites section in order to see their regularly visited locations.

**History**

This function will pull up a history of all the past searches by the user. The user will click on the search bar at the top of the application. The application will proceed by prompting the user to search a location as well as display the history of past searches below the search bar. The user will now be able to either search for a new location or select a previous search from the history list below.

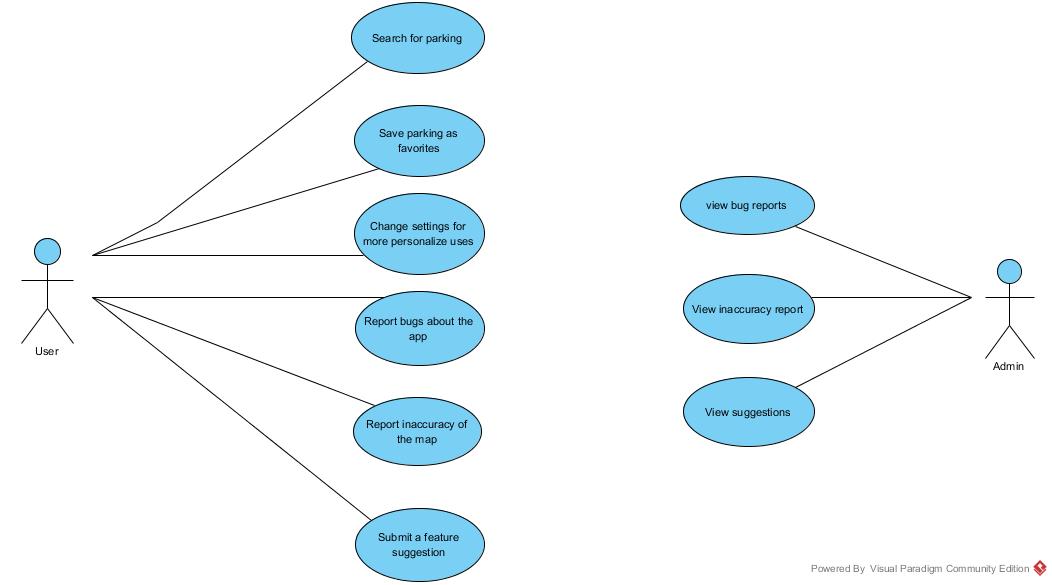
**Viable Parking Times**

This function will allow users to click on specific streets and see available free parking times alongside parking restrictions. The user will search for a street they wish to view or scroll along the map to find it. They will then press on the street. The application will search the database for the street selected and pull the information from the database to be displayed. The user will now be able to view the parking information for the street in a pop-up dialog box.

**Google/Facebook Login**

This function will allow users to login to our application using their Facebook or Google accounts. Upon opening the application they will be prompted to login. They can either select to login or opt to skip this feature. If selected, the application will redirect to either application’s respective login pages and ask for permission to use their information. If successful, the application will redirect back to our application and display the map, ready for use.

## 3.2 Use Cases



### 

### 

### 3.2.1 Use Case #1

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Search for parking | **ID:**  UC1 | **Important Level:** High |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** User can search for parking near the destination | | |
| **Trigger:** User can find parking by typing the street location. This will show the users the parking availability nearby.  **Type:** Internal | | |

### 

### 

### 3.2.1 Use Case #2

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Save parkings as favorite | **ID:**  UC2 | **Important Level:** Low |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe the favorite feature | | |
| **Trigger:** When a user find a parking of their liking they can click on a button that will save their parking location under a database.  **Type:** Internal | | |

### 

### 

### 3.2.1 Use Case #3

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Change settings | **ID:**  UC3 | **Important Level:** low |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe the available setting options the user can do | | |
| **Tigger:** User can personalize their app by changing the settings. The available option will have theme change, custom search settings, parameter preferences, and font change  **Type:** Internal | | |

### 3.2.1 Use Case #4

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Bug Reporting | **ID:**  UC4 | **Important Level:** Medium |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe bug reporting by the user | | |
| **Trigger:** User can report bugs about the app by sending a descriptive message to our servers. It will then later be checked by the admins  **Type:** Internal | | |

### 

### 

### 3.2.1 Use Case #5

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Inaccuracy Reporting | **ID:**  UC5 | **Important Level:** Medium |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe map inaccuracy reporting by the user | | |
| **Tigger:** User can report the inaccuracy of the map by sending a descriptive message and its geolocation to our servers. It will then later be checked by the admins  **Type:** Internal | | |

### 

### 

### 3.2.1 Use Case #6

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  Feature suggestions | **ID:**  UC6 | **Important Level:** Medium |
| **Primary Actor:** User | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe how user can suggest a feature for this app | | |
| **Tigger:** User can suggest a feature that he/she would like to have on this app. The admin can view the suggestion  **Type:** Internal | | |

### 3.2.1 Use Case #7

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  View Inaccuracy Reports | **ID:**  UC7 | **Important Level:** Low |
| **Primary Actor:** Administrator | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe inaccuracy report viewing | | |
| **Trigger:** Administrator can view inaccuracy reports sent by the user. It will be put into a list which admin can then select any report for more details  **Type:** external | | |

### 

### 

### 3.2.1 Use Case #8

|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  View Bug Reports | **ID:**  UC8 | **Important Level:** Low |
| **Primary Actor:** Administrator | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe bug report viewing | | |
| **Tigger:** Administrator can view bug reports sent by the user. It will be put into a list which admin can then select any report for more details  **Type:** External | | |

### 

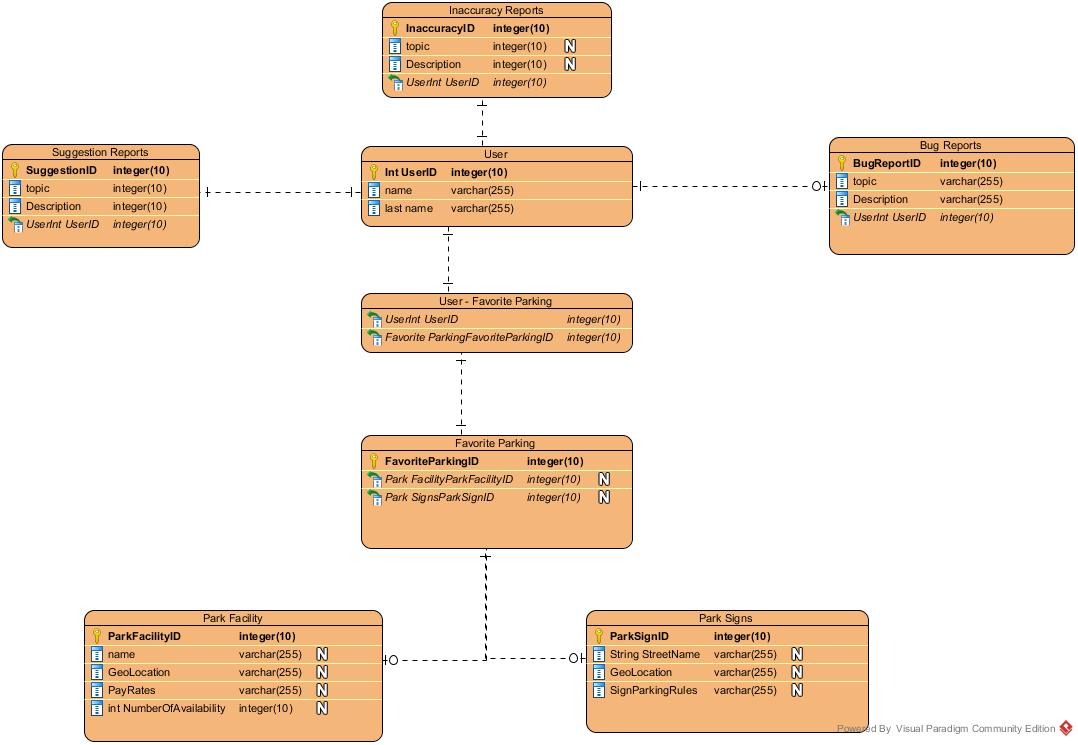
### 

### 3.2.1 Use Case #9

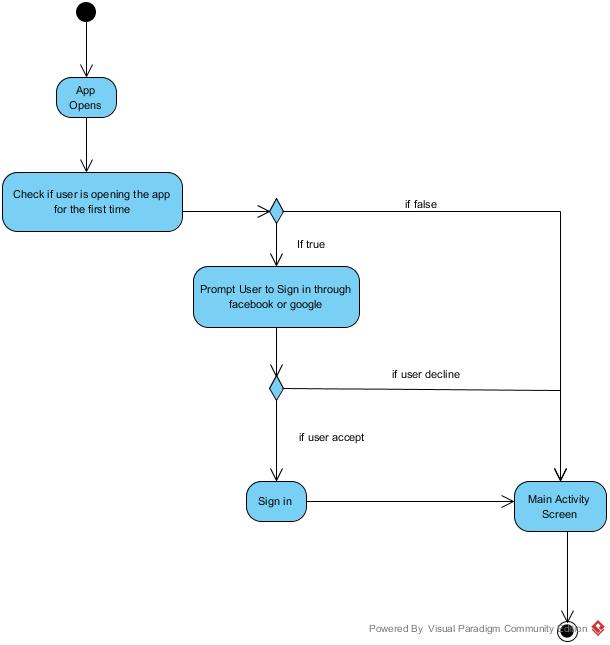
|  |  |  |
| --- | --- | --- |
| **Use Case Name:**  View Suggestions | **ID:**  UC9 | **Important Level:** Low |
| **Primary Actor:** Administrator | **Use Case Type:** Overview, Essential | |
| **Brief Description:** This use case describe suggestions view. | | |
| **Tigger:** Administrator can view suggestions sent by the user. It will be put into a list which admin can then select any suggestion messages for more details  **Type:** Internal | | |

**3.3 Data Modelling and Analysis**

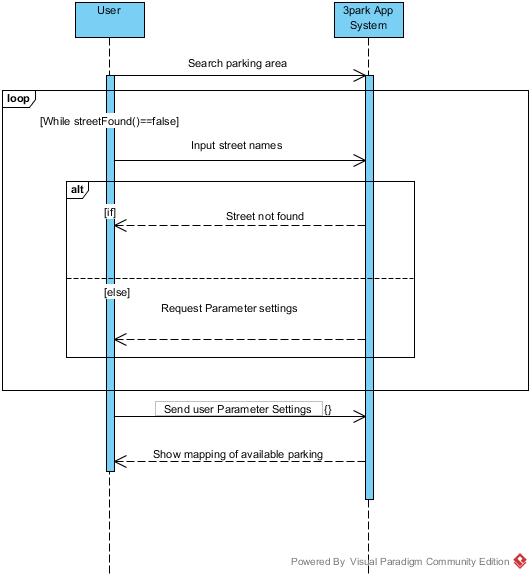
**3.3.1 Normalized Data Model Diagram**



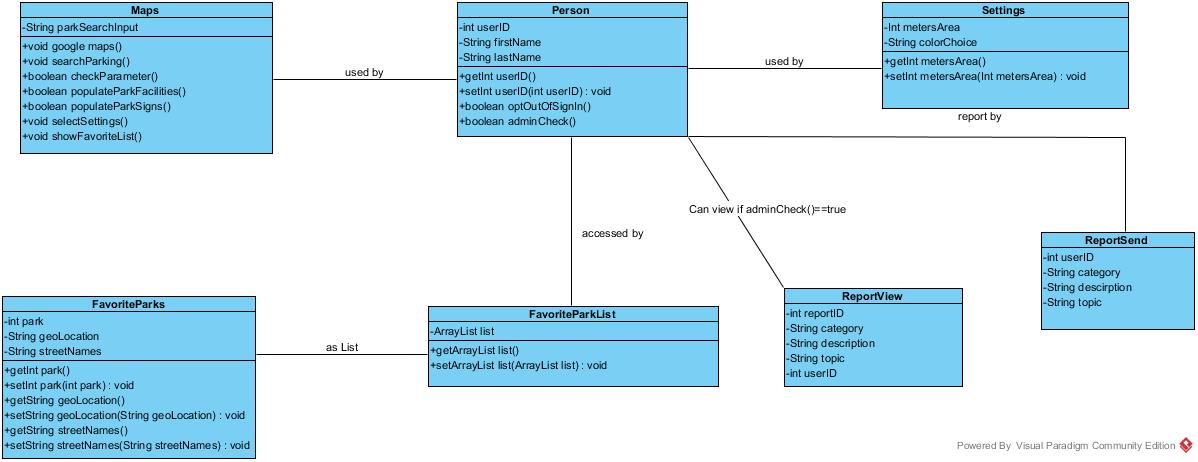
**3.3.2 Activity Diagram**



**3.3.3** **Sequence Diagram**

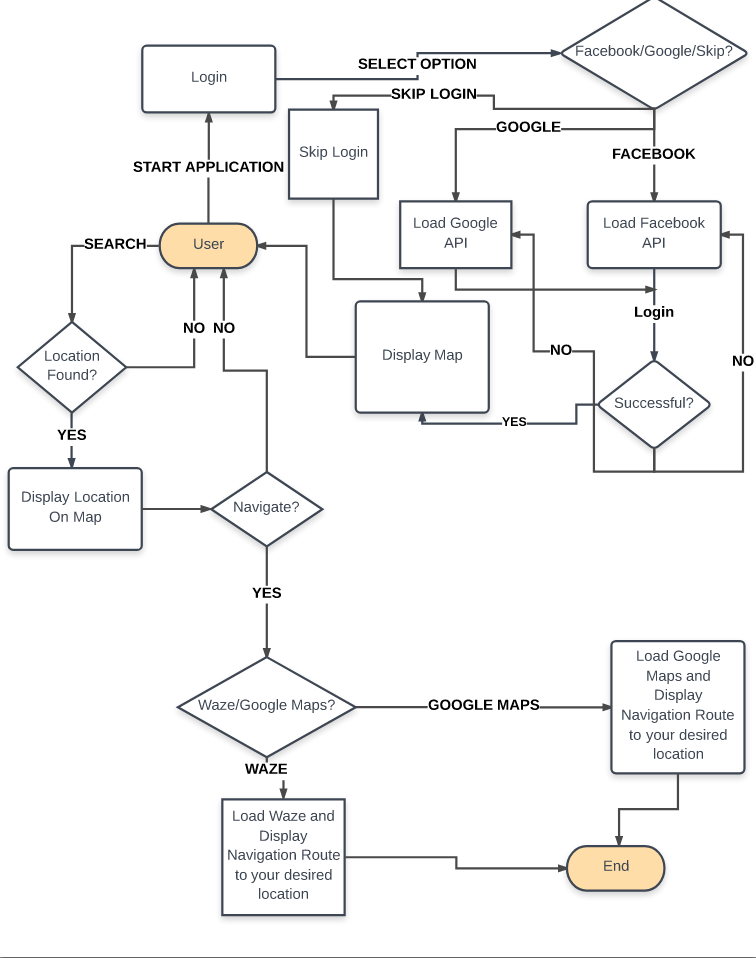


**3.3.4 UML Class Diagram**



**3.4 Process Modelling**

Data Flow Diagram



## 4.0 Non-Functional Requirements

The application 3Park will reside on individual user’s mobile smartphones; either on Android or iOS operating systems. The software developed here assumes the use of a smartphone with high speed Internet and GPS connectivity. They will be designed to run efficiently and make use of the user’s high speed data connection, mainly LTE, with GPS to support critical location features. The database will be hosted and served from Amazon Web Services, which is extremely reliable and fast. The speed of the database calls will depend on the user's Internet connection, and the host that the data will be served from (AWS).

**System Downtime**

* System downtime may not exceed one minute per day

**Latency**

* There must be an extremely low latency when making calls to the server, as obtaining the data for this application is critical to the performance and ease of use for users

**Internet Connection**

* User’s smartphone has access to high-speed Internet, preferably an LTE or 3G data connection through their respective service provider

**GPS Connection**

* User’s smartphone equipped with GPS is critical to support location features and services

**Search Return**

* User searches must be returned and rendered on the user interface within 3 seconds of a user clicking “Search” for a location

## 5.0 Logical Database Requirements

A database will be critical to the 3Park application. It will call on datasets from the City of Toronto’s data catalogue that are stored in JSON, XML, and CSV files. The application will have to execute CRUD actions to these files and incorporate them into the database. We will be hosted on Amazon Web Services. Having the reliability of AWS to host our servers means that we do not have to worry about data integrity or any reliability issues, as it is an extremely reliable and trustworthy server host.

**Create**

* The application’s backend will check datasets on the City of Toronto’s Data Catalogue website for an updated version of any given JSON / XML file. If a newer copy is found, the application will CREATE a new table(s) with newer information, while the active tables are still live and used by users.

**Read**

* Most used and important feature of our app. The database will be READ by users who submit queries through the user interface

**Update**

* The administrator will be tasked with maintaining the database, and making any necessary changes to the tables, which include needing to UPDATE the information contained in them.

**Delete**

* The application will host older copies of the database for redundancy purposes. However, there will be an auto-delete feature for table(s) contained that have at least 3 working backups.

## 6.0 Other Requirements

None.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Team Leader, Frontend developer | Justin Rolnick | JR | Dec 8, 2017 |
| Junior backend developer | Albert Nguyen | AN | Dec 8, 2017 |
| Backend developer, Database administrator | Jeremy Yang | JY | Dec 8, 2017 |
| Senior backend developer | Lenny Ramroop | LR | Dec 8, 2017 |
| Professor | Anjana Shah |  | Dec 8, 2017 |