

# School of Computer Science and Engineering

**CZ2006 Software Engineering Software Project Documentation**

## Lab Group: SS3

**Supervisor: Prof Liu Cheng Wei**

**Instructor: Prof Liu Cheng Wei**

**Team Name: No Turning Bac**

**Project Name: Parking@SG**

**Team members: Lim Yong Xuan Isaac (U1821486G)**

**Ho Swee Ngee**

**(U1822701G)**

**Lew Jian Chung Kenny (U1821042H)**

**Zheng Zhenkai**

**(U1821564D)**

**Lynn Htet Aung**

**(U1821162A)**

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

# Introduction

## Purpose

Parking@SG is an open-source Android application aimed to provide convenience for local drivers. Parking@SG aim is to provide users with real time information of the car parks available in Singapore. Being able to offer this information to users enables them to make decisions on where to park based on the conditions with ease.

## Document Conventions

Font: Times

Font Sizes:

* Main heading 18 (Bold)
* Heading 14 (Bold)
* Sub-header 12 (Bold)
* Text 11

## Intended Audience

Parking@SG is developed based on a requirement-driven basis. It is designed for drivers in Singapore, solving daily problems difficulties like having to find a suitable parking location to their needs especially in cases where they have not visited the area before or when visiting an area during peak hours. Parking@SG app aims to help reduce this stress by being able to provide information the driver needs so that he is able to settle on a carpark location.

## Reading Suggestions

The documentation is a detailed guideline on the architecture of Parking@SG and provides different readers with information needed for them to fulfill their roles

1.4.1 Front-end engineers

Front-end engineers may use the documentation to review the user-interface, functional and non-functional requirements to suggest and implement features.

1.4.2 Back-end engineers

Back-end engineers may use the documentation to check on the use cases, implementation of classes and methods, APIs and databases.

1.4.3 Testers

Testers may use the documentation to plan for different testing strategies based on the use cases and dialog map.

1.4.4 User & Stakeholders

Users may use the documentation to familiarize themselves with the features found in Parking@Sg

1.4.5 Documentation writers

Documentation writers should review the document conventions before editing.

## Product Scope

The application provides all carpark information that are registered under the LTA of Singapore. The application is designed to provide 24/7 constant feedback of the carparks all around Singapore.

The application will depend on the user’s feedback to detect any faults on the information given, such as closure of carparks and misinformation.

## References

-Blank for now

# Overall Description

## Product Perspective

Parking@SG is a self-contained, open source product. This product is developed and deployed on Android and IOS platform. Its main responsibilities are to provide navigation and recommendations for parking lots in Singapore. The usability of this product allows users to reduce time spent on trying to find a suitable carpark to park by using our recommended list of carparks based on user’s preference. It also provides with more options for users to park as the application is capable to extrapolate information of numerous carparks around user’s preferred carpark locations.

## Product Functions

The following are the main features that are included in Parking@SG：

* Sign in system: The system allows users to create an account for the application. In fact, users must have an account to access the application in the first place.
* Feedback system: the system allows users to feedback to our developer team regarding the features of the application itself.
* Report system: the system allows users to report any faults in the application such as carpark information like location, availability, type of carpark lots.
* Search system: the system allows user to search for carpark around a specific radius of the targeted location that the user had selected. By default, the application will show carparks surrounding the user’s current location before search.
* Filter system: the system allows users to filter the carparks after searching by name, price and location.
* Route system: the system allows users to visualize the route in the map from the user’s current location to the designated carpark, by calling Google API.

## User Classes and Characteristics

We assume there are two main classes of our users. One is the user, who should have basic knowledge on how to navigate within a phone application. Tutorial on how to navigate through the application will be also given to users who are less familiar with the application to aid users to have a pleasant experience with the application. Another class is the admin, who is expected to be very familiar with the application, is very proficient in using the application and understands every functionality of the application as admins are required to change or edit snippets of the application when needed.

## Operating Environment

The Parking@SG application is built on Android Studio. It operates on mobile phones with android operating system installed. Since its Minimum API Level is API15: Android 4.0.3 IceCreamSandwich, the app will run on approximately 100% of android devices. Therefore, the app is well-adapted. The app will require users to turn on GPS on their mobile phones as Google API will be used to do the routing towards the carpark.

## Design and Implementation Constraints

Database used: MySQL, Firestore (Firebase)

Communication protocols: - Http requests: GET and POST

Security considerations: - (In-built encapsulation by firebase for authentication)

Design convention: - MVC (model-view-controller)

## User Documentation

Tutorials will be available for new users which includes teaching users on how to navigate through the application, how to report faults and where to go when they are lost while using the application. Users can also send in feedbacks about the application using the built-in function of the application. Tutorials will still be available for users to view at any point of time while using the app.

## Assumptions and Dependencies

The server must have MySQL running.

The users must connect to internet.

The users must turn on GPS.

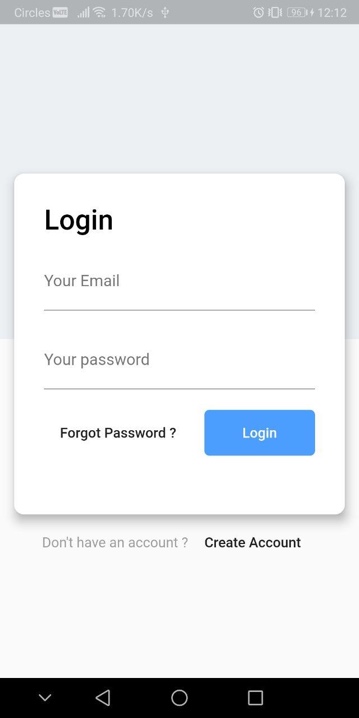
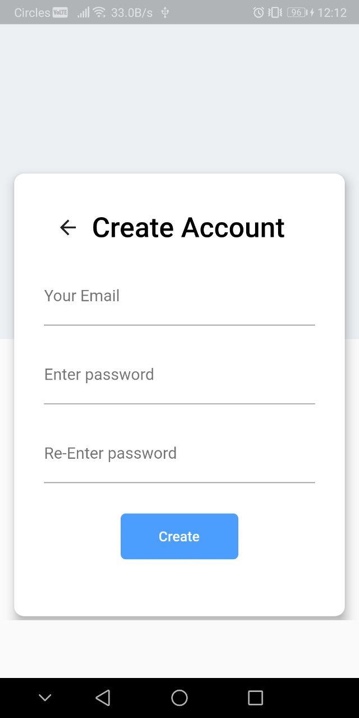
Google API must be working and running.

The carpark API that the government provides is accurate.

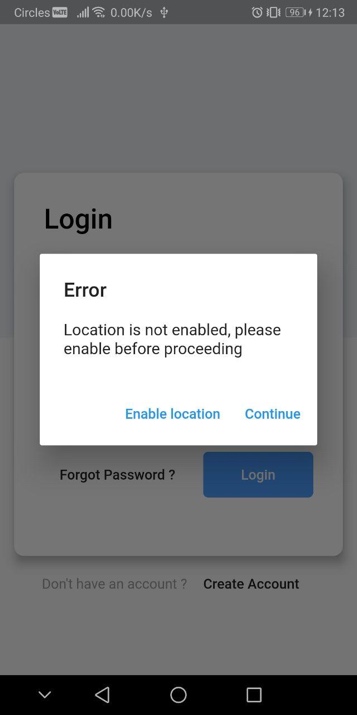
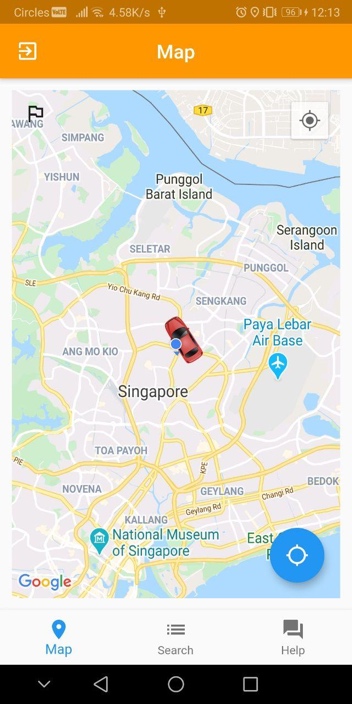
# External Interface Requirements

## User Interfaces

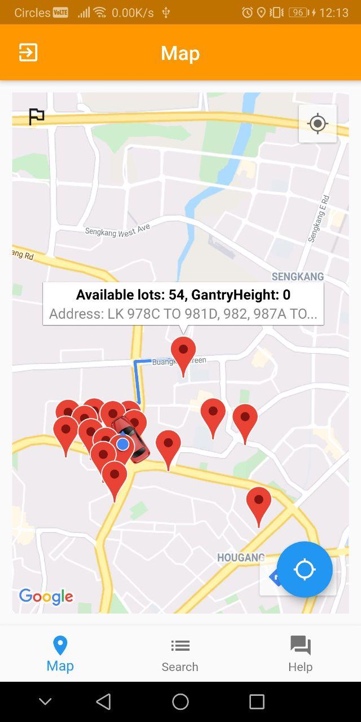
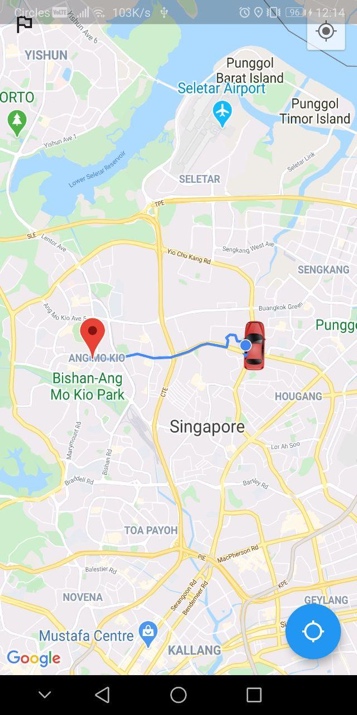
3.1.1 This is the first page users will see when they open the app. Users can choose to login or create a new account. The right image is the create account page

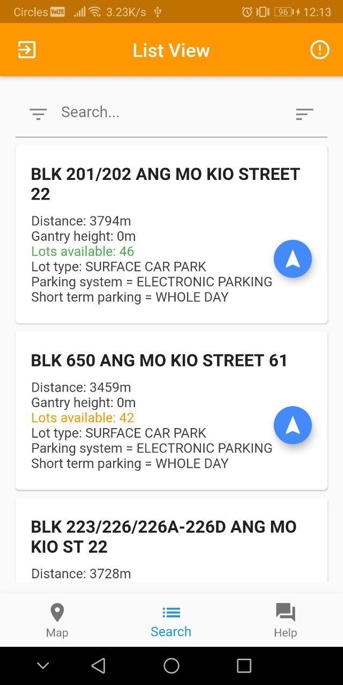
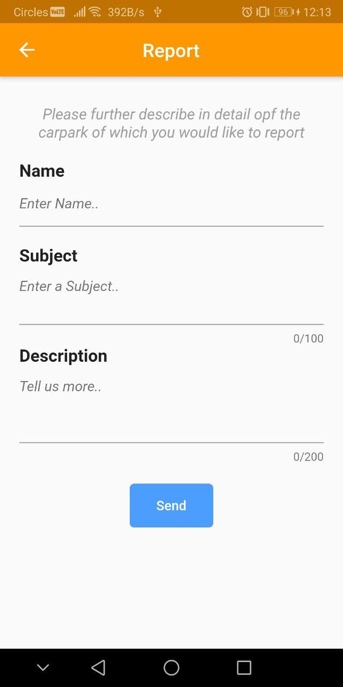
3.1.2 The left image is a pop-up message if location service is not enabled in the phone settings. The right image is the map view page which the user will first see after logging in. It shows the current location of the user on the map.

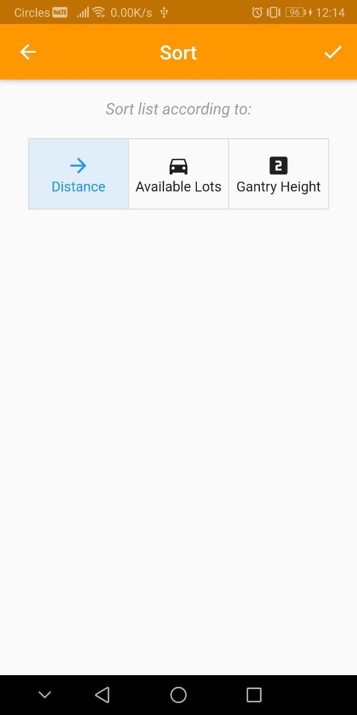
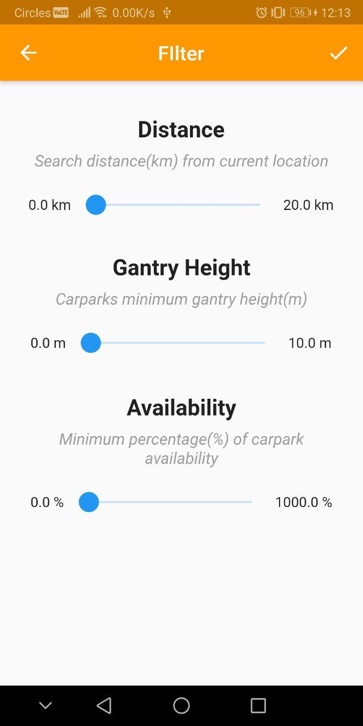
3.1.3 When the user taps on a carpark, the shortest route is generated to it.

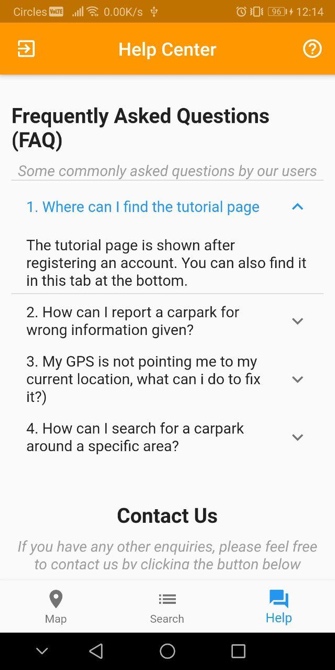
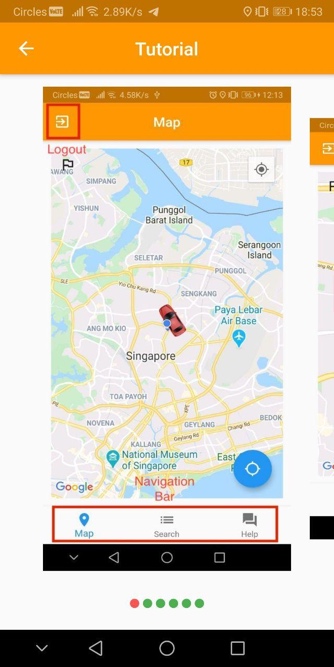
3.1.4 This is the list view page. Details of each carpark is displayed along with each carpark. The user can also report a carpark for any misinformation.

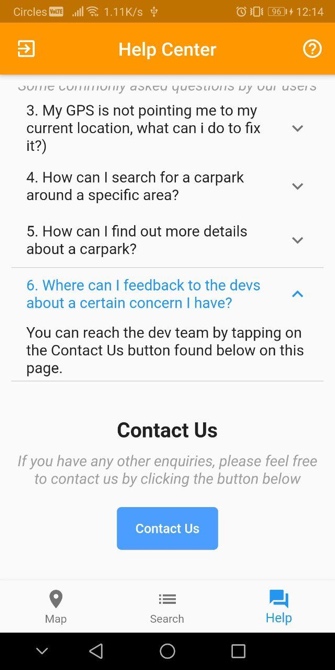
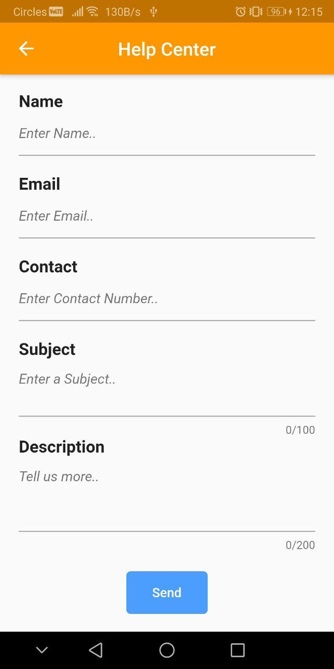
3.1.5 The user can also sort and filter their searches according to their liking.

3.1.6 The help center is accessible by tapping on its icon found in the bottom right in the navigation bar. Frequently Asked Questions (FAQ) can be found in the help center. Users can also access the tutorial by clicking on the icon on the top right.

3.1.7 The help center also has a Contact Us section for the user to reach out to the dev team.

## Hardware Interfaces

The hardware interface is mainly based on the touch screen. Users can interact with on-screen objects through direct manipulations on the touch screen using gestures. Gestures such as tapping, pinching and swiping are used as inputs to interact with the application. The application will respond to the user differently based on the different inputs. For the internal hardware, GPS, cellular and Wi-Fi modules are used.

## Software Interfaces

Android Studio in cohesion with Flutter and Dart plugins support the construction of the user interface. Flutter’s structured layout and widget UI system allows users to build on the graphical interface of the application. Google maps and APIs from Gov.sg are embedded as external data sources. Different versions of android are supported in Android Studios and are used in the test and developing stages in building our application.

## Communications Interfaces

SGParking backend servers are based on Laravel for database storage and Google’s firebase for user authentication.

# System Features and Functional Requirements 4.1 Starting the application 1.1 The application should have a login and registration function 1.1.1 The login function shall have a text box for email and a text box for password 1.1.1.1 The application must display successful or unsuccessful login 1.1.1.2 The application should store the user's setting and history after login 1.1.1.3 The application should encrypt the user's password 1.1.1.4 The ID should not exceed 20 characters 1.1.1.5 The password should not exceed 20 characters

**1.2** The user can create an account with our application

**1.2.1** The user needs to provide the following information

**1.2.1.1** The user's email

**1.2.1.2** The user's password less than 20 characters and more than 7

**1.2.2** The user will receive a verification code sent to their email

**1.2.2.1** User will have to enter the verification code in the application to verify their email address

**4.2 Search function**

**2.1** The user must on GPS before using the application

**2.1.1** The application will have a pop-up dialog to prompt the user to on GPS 2.1.1.1 The pop-up dialog will have a button to redirect user to phone settings to enable location services

**2.2** There will be a "map view" & "listing view" tab at the bottom navigation bar.

**2.2.1** The default tab will be "map view" when user log into the application

**2.2.2** Users can only be in 1 tab at a time.

**2.3** The search query can only be used in the search tab and must have at least one of the parameters

**2.3.1** The name of the car park must be in valid English characters and not exceeding characters

**2.3.2** The location of the carpark based on the GPS location of the device

**2.3.3** The budgeted price of parking, or a range of acceptable prices from a minimum to a maximum price

**2.4** Each search shall return the following information

**2.4.1** The carpark name not exceeding 50 characters

**2.4.2** Price in 2 decimal places in SGD per hour

**2.4.3** The distance of the car park from the current location in meters

**2.4.4** "No search result found" will be displayed below the search bar in the font size of 15.

**2.4.5** Absolute values of the carpark’s available lots will have a specific color to indicate the availability of the search results

**2.4.5.1** Green indicating at least 50% of the lots are occupied

**2.4.5.2** Yellow indicating at least 80% of the lots are occupied

**2.4.5.3** Red indicating 90% of the lots are occupied

**2.5** Each search can be sorted based on the following parameters either before or after search. The following parameters will be displayed in a set of toggled buttons

**2.5.1** Distance to the car park in meters (increasing order) in kilometers(km)

**2.5.2** Car park availability based on number of unoccupied slots in integer (decreasing order)

**2.5.3** Gantry height of the carpark

**2.6** The search results can be filtered based on the following parameters

**2.6.1** A maximum distance from the current location to the carparks, can be selected using the slider, the distance will be in meters.

**2.6.2** The minimum gantry height of the carpark

**2.6.3** The car park availability based on the percentage occupancy of the car park; the slider value must be between 1 - 1000.

**2.7** The user can get directions to the parking lot

**2.7.1** The nearest route to the parking lot will be displayed on a map with clear route path (blue)

**2.7.1.1** The application will automatically allocate the shortest route for user to use to navigate

**2.7.1.2** The application will automatically reroute the shortest route for the user if the user goes off course

**4.3 Feedback function**

**3.1** The user can report a car park

**3.1.1** A report button is displayed on the right side of the page header.

**3.1.2** The user can report any issues of the carpark

**3.1.3** The user has to enter the following information

**3.1.3.1** The users name not exceeding 20 letters

**3.1.3.2** Subject of the matter the user will like to bring up not exceeding 100 letters

**3.1.3.3** Description of the matter not exceeding 200 letters

**4.4 Help center**

**4.1** The help center will answer the following Frequently Asked Questions (FAQs):

**4.1.1** The help center will have a "Where can I find the tutorial page" description

**4.1.2** The help center will have a "How can I report a carpark " description

**4.1.3** The help center will have a "My GPS is not pointing me to my current location, what can I do to fix it?" description

**4.1.4** The help center will have a "How can I search for a carpark around a specific area" description

**4.2** The help center will have contact us button for users to reach out to the dev team for miscellaneous queries

**4.2.1** The user has to enter the following details

**4.2.1.1** User’s name not exceeding 20 characters

**4.2.1.1** User’s email

**4.2.1.2** User’s phone number with exactly 8 numbers

**4.2.1.3** The subject the user would like to bring up not exceeding 100 characters

**4.1.5.2** The description of the subject matter not exceeding 200 characters

**4.1.5.3** The dev team will respond at the earliest possible time

**4.5 Tutorial**

**5.1** Users can access the tutorial in the help page

**5.1.1** There will be a step-by-step guide about the application

**5.1.1.1** The tutorial will image to guide users on how to use the app and explain and explain the functions of the buttons

**5.1.1.2** The tutorial will contain guidelines on

**5.1.1.2.1** How to route to a carpark

**5.1.1.2.2** How to sort

**5.1.1.2.3** How to filter

**5.1.1.2.4** How to navigate through the app

**5.1.2** Users are not required to access the tutorial before using the application

**5.1.2.1** Users can access the tutorial at any point in time

# 5. Other Nonfunctional Requirements 5.1 Usability 1.1 The application must give useful help information when there are errors like 1.1.1 User has no internet access  1.1.2 API cannot be reached  1.1.3 Application undergoing maintenance 1.1.4 User’s GPS location cannot be accessed 1.2 The font size must be reasonably big enough for users to be able to read while on-board on the vehicle

**5.2 Reliability**

**2.1** APIs used in the application must be from a reliable source such as government websites

**2.2** The application will continue to function even when

**2.2.1** New carparks are built

**2.2.2** Carpark is under maintenance

**2.3** The application must not reveal any personal information

**2.3.1** Personal particulars must not be displayed on the main functionality page

**2.3.1.1** Phone number/address/email address/age/etc must not be shown

**2.3.3** User information will be deleted from the application when he logs off from the application

**2.3.3.1** If the user does not have an account, user information will be deleted from the application when he uninstalls the application

**5.3 Performance**

**3.1** The availability of carpark will be constantly updated every 3-5 minutes

**3.2** The application shall take at least 5 second to load for every page

**3.3** The application can at most handle 50 users at any point of time

**5.4 Supportability**

**4.1** The application shall be extended in the future

**4.1.1** New user interface and designs will be developed in future editions

**4.1.2** New APIs can be integrated

**4.1.2.1** Search function for the nearest petrol station

**4.1.2.2** Current and forecasted weather conditions to ease driving conditions

**4.1.2.3** Notification on road traffic accident/congestion

**4.1.2.4** Navigating within the car park itself

**4.2** The development team will maintain the application and debug when necessary

**4.3** The application shall be adapted to desktop versions when popularity increases

# Appendix A: Glossary

|  |  |
| --- | --- |
| GPS | A functionality that almost all smartphones have today. It uses the satellite to return the ground position of the smartphone. It is the acronym for Global Positioning System. |
| Map view | A Map displayed on the application, there will be other items displayed base on the different functionalities |
| Listing view | Objects/items show in sequence from top to bottom, it could be sorted. |
| Carpark | A facility designed to allow motorbikes/cars/vans/etc to park over long hours. |
| Slider | A horizontal scroll button, allowing users to drag to their desired amount |
| Car Park availability | Number of available parking lots in a car park which has not been occupied, this information can be obtained from the API |
| Directions/routes | This will show the users the possible routes to the car park depicted by a line from current location to the desired car park by a line |
| Application | The SG parking app developed by the No Turning Back team. |
| HDB carpark | Car parks which are located within the vicinity of a housing development board estate. |
| API | An application programmable interface where the application is able to get information or data from a third party. |
| Feedback | Users are able to report a problem and give textual responses which will be sent to the administrator of the application who will then proceed to try and rectify the problem. |
| FAQ | A screen which shows the frequently asked questions which will give basic information for users of the application. |
| Tutorial | A guide for first time users to know how to use our application, showing them basic functionalities |
| Report button | A red triangle with an exclamation mark in the middle, located next to the carpark name, is used to report the selected carpark for misinformation. |
| Sort | User arrange results based on the sorting criteria (Availability, Price, Distance) |
| Filter | An advanced search based on the selected parameters (Price, Distance, Carpark Type) |
| Distance | Searching distance in kilometres (in a radius) from a location or users' current location |
| Carpark Type | Indoor and outdoor type of car park |
| Help Centre | Located on the right end of the navigation bar, allowing users to seek additional help regarding the app. |
| Contact us | Users can reach out to the developer team directly by submitting their name, phone number and description through the app. |
| Email us | Users can email their concerns using the provided email. |
| Login | Email used during registration by the user |
| Password | Encapsulated information required to login to the user’s account, consisting of minimum of 8 characters with at least one caps and at least one special character |
| id | User’s id used for log in |
| password | User’s password used for log in |
| pageAccess | Page accessed by the user |
| timeStamp | Time stamp for the page accessed by user |
| creationDate | The date for which the user’s account was created |
| profilePicture | Image of the user’s profile picture |
| accID | User’s ID |
| email | Email that user used to register account with |
| gender | User’s gender (Male or Female or Others) |
| DOB | User’s Date of Birth |
| firstName | User’s first name |
| lastName | User’s last name |
| location | Location given to the gps to search for nearby carparks |
| carparkId | Carpark’s id |
| indoor | Boolean variable to be set to true if carpark is indoors and false if carpark is outdoors |
| weekend | Boolean variable to be set to true if there is free weekend parking and false if there is not |
| Price | Price the car park lot ($/per hour) |
| Availability | Number of empty lots in the carpark |