

# School of Computer Science and Engineering

**CZ2006 Software Engineering Software Project Documentation**

## Lab Group: SS3

**Supervisor:**

**Instructor:**

**Team Name: No Turning Bac**

**Project Name: Parking@SG**

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

**Ho Swee Ngee (U1822701G)**

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

Times

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.

1.3.1 Drivers in Singapore.

It is noticed that drivers can find it difficult 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 first developed and deployed on Android 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.
* Bookmark system: the system allows users to bookmark their favorite carpark lots so that the application will record down the carpark and feature it in the main page for user to see whenever user exits and re-enter the application
* 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

Communication protocols: -

Security considerations: -

Design convention: -

## User Documentation

Tutorials for first time users will be given when users log into their account to use the application. The tutorial will include 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.

## 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

To put in our UI

## 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. As for internal hardware GPS, cellular and Wi-Fi modules are used.

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

HOS back-end servers are hosted by Spring Boot. The transmission, storage and transaction records of data is supported by HTTP protocols. Google Chrome is applied to support the external links embedded in HOS.

# 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 name and last name 10 characters

**1.2.1.2** The user's email address

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

**1.2.1.4** The user's contact number

**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 not be removed until GPS is ON

**2.2** There will be a "map view" & "listing view" tab at top left-hand corner of the display.

**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 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 50 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** A box of specific color will be used 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

**2.4.5.4** A cross overlapping the box will indicate that all of the lots are occupied

**2.4.6** Absolute values of each carpark’s vacancy slots will be shown

**2.5** Each search can be sorted based on the following parameters either before or after search. The following parameters will be displayed as a slider

**2.5.1** The carpark names in alphabetical order within 50 alphabetical order

**2.5.2** Price of the car park per hour (increasing order) in SGD

**2.5.3** Distance to the car park in meters (increasing order) in KM/M

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

**2.5.5** Indoor or outdoor car parks, this option will be displayed as a tab. The default will be no tab selected and users can only select 1 tab at any moment.

**2.5.6** After using the sorting function, a message “Applied successfully” will be shown for 2 seconds

**2.6** The search results can be filtered based on the following parameters either before or after search results are displayed

**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** A price range giving the maximum cut-off price in SGD, users can choose their input using the slider.

**2.6.3** The car park availability based on the percentage occupancy of the car park; the slider value must be between 1 - 100.

**2.6.4** After using the filter function, a message “Applied successfully” will be shown for 2 seconds

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

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

**2.7.1.1** Three routes with order will be suggested to user

**2.7.1.1.1** Highest ranking route being the fastest route towards parking lot

**2.7.1.1.2** The application will automatically allocate the highest-ranking route for user to use to navigate

**2.7.1.1.2.1** User can change route at any point of time during travelling

**2.7.1.1.3** Time needed to travel will be included

**2.7.1.1.4** User can choose to avoid tolls by activating the option of "avoid tolls" in the interface

**2.8** The user can also get the direction for the shortest exit from the HDB carpark to the main road

**2.8.1** The directions to the nearest exit will be displayed on a map with clear route path (green/blue)

**2.8.2** There will only be one suggested route, indicating the shortest route to the main road

**2.8.3** This feature will only be offered to users if user is within the vicinity of any registered HDB on the government HDB API.

**4.3 Feedback function**

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

**3.1.1** A report button will be shown after a car park has been selected from the map or from the search results

**3.1.2** The user can report the listed attributes of the car park

**3.1.3** The user has the option to suggest new attribute values of the following:

**3.1.3.1** The user can suggest the address and the GPS coordinates of the car park

**3.1.3.2** The user can suggest the type of car park as either ‘Multi-storey’, ‘Basement’ or ‘Surface’

**3.1.3.3** The user can suggest the short-term parking duration as either ‘No’, ‘Whole day’ or input a valid start and ending timing

**3.1.3.4** The user can suggest the free parking duration as either ‘No’ or input a valid start and ending timing

**3.1.3.5** The user can suggest the night parking as either ‘Yes’ or ‘No’

**3.1.3.6** The user can suggest the car park decks as a valid integer number, given the type of car park is listed as ‘Multi-storey’

**3.1.3.7** The user can suggest the height of the gantry as a valid floating-point number

**3.1.4** The carpark will have a default status of ‘Open’

**3.1.4.1** The user can flag the status of the car park as ‘Closed’

**3.1.4.2** The user can flag the car park as ‘Renovating’

**4.4 Help center**

**4.1** The help center should be a floating action button that is visible throughout the main application

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

**4.2.1** The help center will have a "how to plan route" description

**4.2.2** The help center will have a "how to view history" description

**4.2.3** The help center will have a "how to view car park price" description

**4.2.4** The help center will have a "how to filter search results" description

**4.2.5** The help center will have an option to contact the dev team for miscellaneous queries

**4.2.5.1** The user can contact the team through an in-app messaging channel by entering the following details

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

**4.2.5.1.2** User’s phone number not exceeding 8 numbers

**4.2.5.1.3** Description not exceeding 100 characters

**4.2.5.2** The user can contact the team through the given email address

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

**4.5 Tutorial**

**5.1** First-time user will be directed to the tutorial page

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

**5.1.1.1** The tutorial will contain guidelines on how to navigate through the application

**5.1.1.2** The tutorial will contain guidelines on how to search for car park

**5.1.1.2.1** Based on location

**5.1.1.2.2** Based on price

**5.1.1.2.3** Based on availability

**5.1.2** It is not a must to finish the tutorial before using the application

**5.1.2.1** Users can exit the tutorial at any point of time

**5.2** Users will be able revisit the tutorial while using the application, even if they had used the application before

# 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

**2.3.2** User's historical record can only be accessed by the user himself

**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

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

## Performance Requirements

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# 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) |
| Availability | Amount of carpark lots available in a carpark |
| Filter | An advanced search based on the selected parameters (Price, Distance, Carpark Type) |
| Price | Price the car park lot ($/per hour) |
| 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. |