Software Requirements Specification

for

NAP

Version 1.0 approved

Prepared by:

HENG CHOON KANG,
MUHAMMAD IQSHAN BIN MOHD SA'AD,
PAN ZAIYU, RYAN,
TAN WEI HONG

NANYANG TECHNOLOGICAL UNIVERSITY, Team Foresight

2/9/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Ryan Pan	2/9/2023	Initial draft	1.0.0

1. Introduction

1.1 Purpose

Provide ios/android users an app to navigate and find available parking lots. Users would require ios/android devices with location service.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

Users who drives and are unsure of where to park / drivers who does not want to waste time finding parking lots.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

Our application has the following functions:

- Allows user to view carparks near him/ her in a map or as a list.
- Allows user to search for carparks by location.
- Allows user to check and filter carparks by availability, distance, and cost.
- Allows user to save carparks for convenient access.
- Allows user to navigate to a selected carpark.
- Allows user to save his/ her parking lot number.
- Allows user to make parking fee payments through Parking.sg application.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

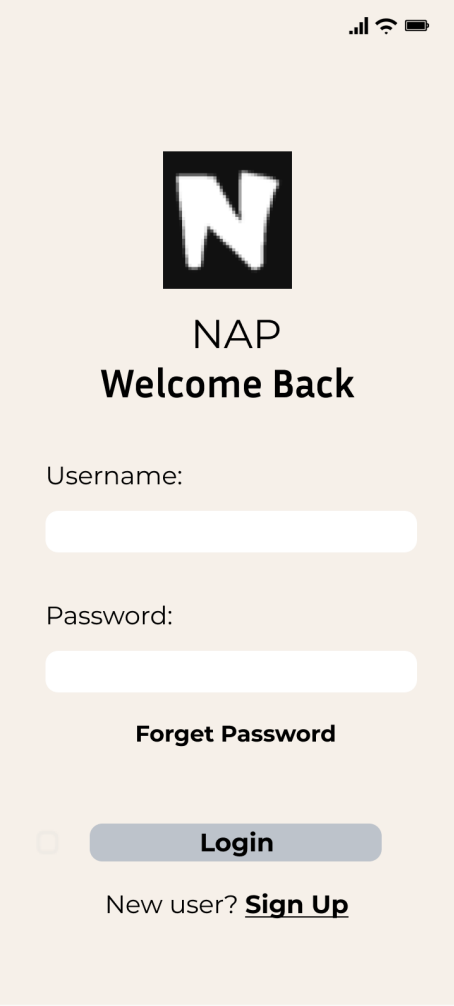
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

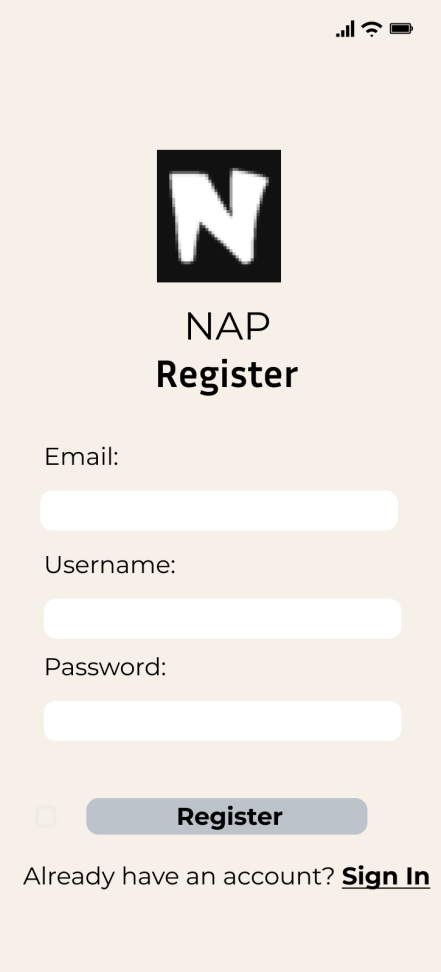
1. **Login/ Landing Screen**

   Existing users can login to the application. New users can click the link at the bottom to sign up.



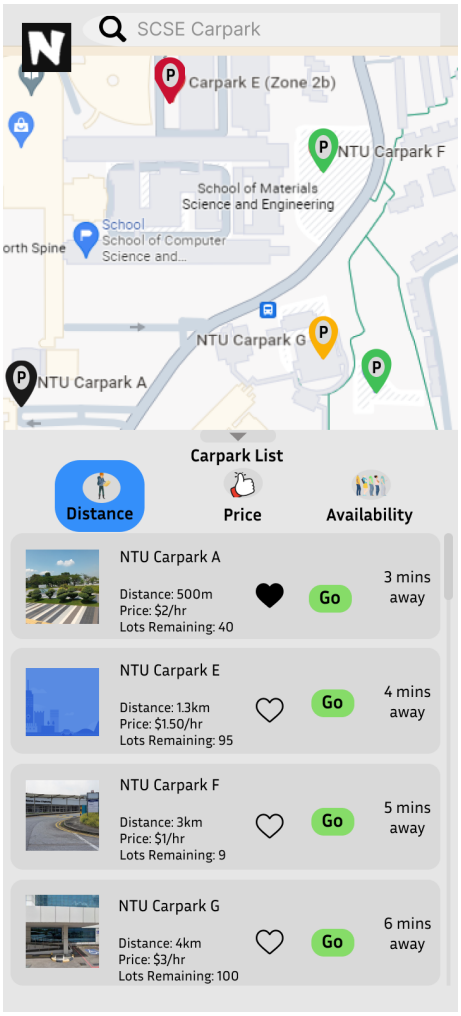
2. **Registration Screen**

   New users can register for an account.



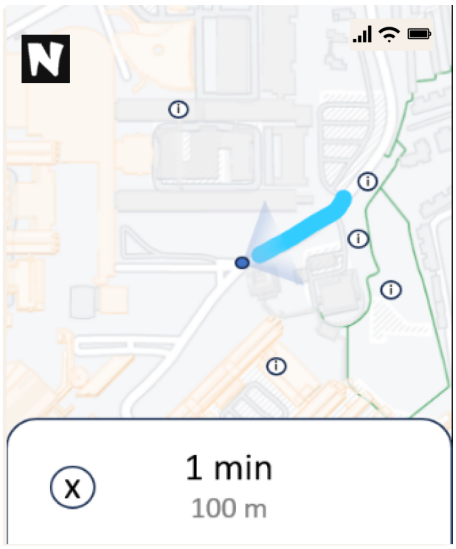
3. Home Screen

User can view carparks near him in a map or as a list. Carparks in the map are marked by colour-coded icons. User can also search for carparks using the search bar, and filter the list of carparks by distance, price, and availability.



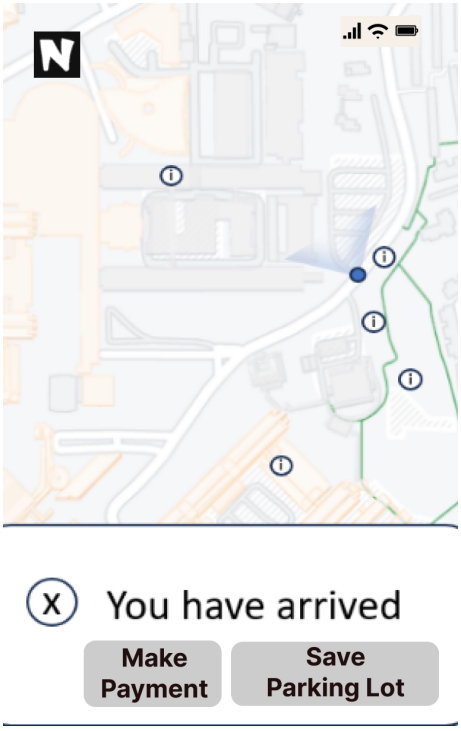
4. Navigation Screen

User gets GPS instruction to navigate to the selected carpark.



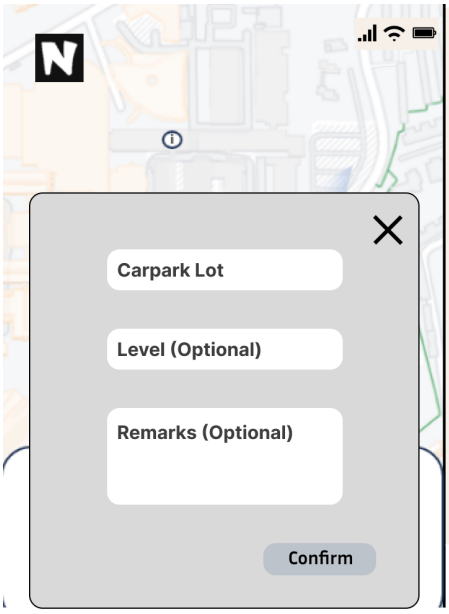
5. Navigation Completion Screen

User receives prompt to make payment or save his/ her parking information.



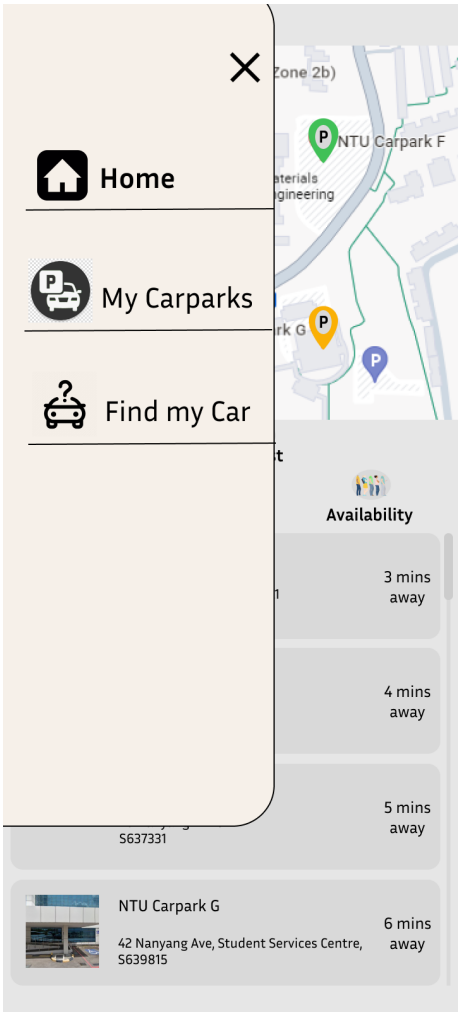
6. Save Parking Information Screen

User can save their carpark lot number, carpark level, or other remarks.



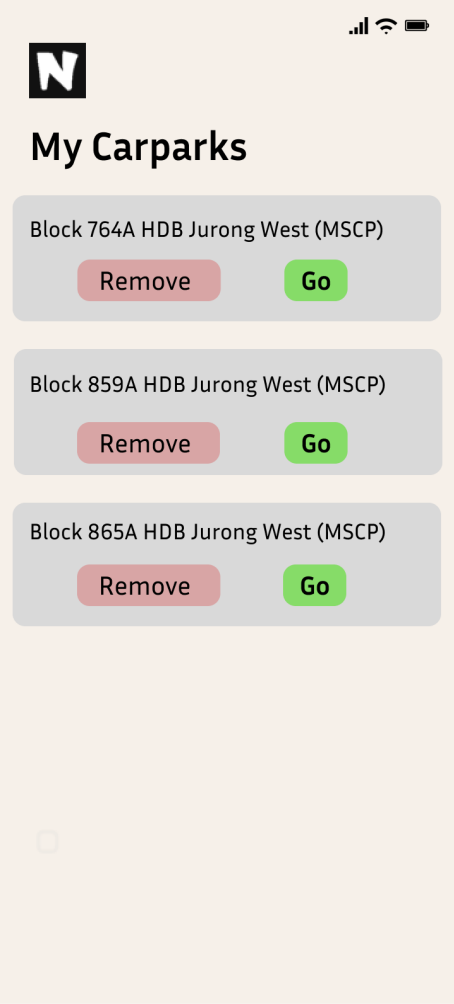
7. Application Menu

User can toggle between the available pages.



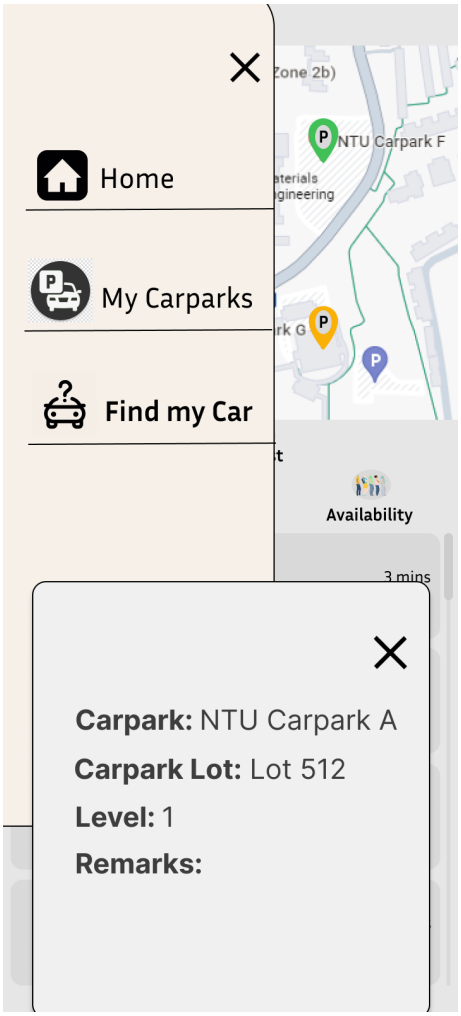
8. Saved Carparks Screen

User can view a list of their saved carparks.



9. Retrieve Parking Information Screen

User can view their latest parking information saved in the system.



3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 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.>

3.4 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.>

4. System Features

4.1 Registration

4.1.1 Description

New users can register for an account. Once they have successfully created an account, their login details will be saved to our system for future logins.

4.1.2 Use Case

Use Case ID:	REG 1		
Use Case Name:	Registration		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	First time users can register for an account. After registration, the user can save carparks that they frequent to
Preconditions:	The user should not have registered an account previously

Postconditions:	The user has successfully created the account and logged in
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. User clicks on 'Sign Up' in the 'Log In' page 2. System prompts the user to enter username, password and email address 3. User inputs the required information and clicks 'Register' 4. System checks whether the information submitted is valid and sufficient 5. System stores these information into a database 6. User is logged in
Alternative Flows:	AF-S3 If username is taken by another user: 1. System displays "Username already exists. Please enter another username" 2. System returns to Step 2 AF-S3 If password do not meets the requirements: 1. System displays "Passwords do not meet the necessary requirements" 2. System returns to Step 2 AF-S3 If email address is invalid 1. System displays "Please enter a valid email address" 2. System returns to Step 2
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	User should be connected to the internet when registering
Notes and Issues:	NIL

4.1.3 Functional Requirements

1. The user must be able to register for an account on the system.

1.1. The system must display text fields for the user to enter account information.

1.1.1. The text field must contain a username.

1.1.2. The text field must contain a password.

1.2. The user must fill in all fields before clicking the “Register” button.

1.3. The system must verify the information provided before registering the account.

1.3.1. The text fields must be filled.

1.3.2. The username must be unique in the system.

1.3.3. The password must contain one uppercase character.

1.3.4. The password must contain one number.

1.3.5. The password must contain at least 8 characters.

1.4. The system must create an account for the user upon verifying the account information.

1.5. The system must redirect the user to the main page upon creating an account.

4.2 Login

4.2.1 Description

After successfully registering for an account, the user can login to the application and access all features.

4.2.2 Use Case

Use Case ID:	L1		
Use Case Name:	Account login		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	User use the app to pay for parking fees for carpark without gantries
Preconditions:	User has an account
Postconditions:	User can use the app

Priority:	High
Frequency of Use:	Once for every user
Flow of Events:	1. User keys in username and password to log in 2. User is free to use the app
Alternative Flows:	AF-S1 If user does not have a registered account: 1. User can click on the 'Register' button to register an account AF-S1 If user forgots their username/password: 1. User can click on the 'Forgot Password' button to initiate a password reset process and receive a password reset link or instructions sent to their registered email address. AF-S1 If user selects 'Remember my login info' before logging in: 1. The system will securely store their username and password, ensuring smoother subsequent logins AF-S1 If username/password is incorrect: 1. System displays "Username/password is incorrect. Please try again" 2. Go back to Step 1
Exceptions:	NIL
Includes:	Database for User information
Special Requirements:	NIL
Assumptions:	1. User must be connected to the internet at all times
Notes and Issues:	NIL

4.2.3 Functional Requirements

1. The user must be able to log in to the system.
    1.1. The system must display text fields for the user to enter account information.
        1.1.1. The text field must contain a username.

1.1.2. The text field must contain a password.

1.2. The user must fill in all fields before clicking the “Log In” button.

1.3. The system must verify the information provided before logging in to the user.

1.3.1. The text fields must be filled.

1.3.2. The username must exist in the system.

1.3.3. The password must match the password of the user.

2. The system must redirect the user to the main page upon verifying the login.

4.3 View Carparks on Map

4.3.1 Description

The user can view carparks near him/ her and their availability on a map.

4.3.2 Use Case

Use Case ID:	UC1.1		
Use Case Name:	Search carparks (Map)		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do an automatic search around the user of 0.5km
Preconditions:	The user must launch the application
Postconditions:	The nearest carparks around the user will be displayed on the map
Priority:	High
Frequency of Use:	
Flow of Events:	1. User opens the application 2. The system will do a scan around the user to find the nearest carparks
Alternative Flows:	1. The user can choose to adjust the search radius
Exceptions:	NIL
Includes:	Carpark Availability

Actor:	User
Description:	The system will do an automatic search around the user of 0.5km
Preconditions:	The user must launch the application
Postconditions:	The nearest carparks around the user will be displayed on the map
Priority:	High
Frequency of Use:	
Special Requirements:	NIL
Assumptions:	1. The user must have enabled location service on their device 2. The user must be connected to the internet
Notes and Issues:	NIL

4.3.3 Functional Requirements

1. The user must be able to view at least one carpark on the map.
    1.1. The system must display a map of Singapore.
    1.2. The system must scale the map to show at least one carpark nearest to the user.
    1.3. The system must indicate each carpark using a button.
    1.4. The system must colour code the button to indicate the carpark's availability.
    1.5. The user must be able to click the button to select a carpark.
    1.6. The system must display the relevant information related to the selected carpark.
        1.6.1. The information must contain the exact distance of the carpark from the current location in kilometres/metres.
        1.6.2. The information must contain the carpark's availability.
        1.6.3. The information must contain the parking fee rates in SGD.
        1.6.4. The user must be able to navigate to the carpark by clicking on the "Go" button.
        1.6.5. The user must be able to add the selected carpark to the saved list by clicking "Save Carpark".
    1.7. The user must be able to view maps in a list by clicking "List".

4.4 View Carparks in List

4.4.1 Description

The user can view a list of carparks near him/ her and their relevant information.

4.4.2 Use Case

Use Case ID:	UC1.2		
Use Case Name:	Search carparks (List)		
Created By:	Ryan Pan	Last Updated By:	
Date Created:	2/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do an automatic search around the user of 0.5km
Preconditions:	The user must launch the application
Postconditions:	The nearest carparks around the user will be displayed in a list
Priority:	High
Frequency of Use:	
Flow of Events:	1. User opens the application 2. The system will do a scan around the user to find the nearest carparks 3. The system will display the results in a list
Alternative Flows:	2. The user can choose to adjust the search radius
Exceptions:	NIL
Includes:	Carpark Availability
Special Requirements:	NIL
Assumptions:	3. The user must have enabled location service on their device 4. The user must be connected to the internet
Notes and Issues:	NIL

4.4.3 Functional Requirements

1. The user must be able to view a list of carparks by clicking “List”.

1.1. The system must display a list of carparks ordered by distance.

1.2. The user must be able to filter the list by distance.

1.3. The user must be able to filter the list by the availability.

1.4. The user must be able to filter the list by parking fee rates.

1.5. The user must be able to return to map view by clicking “Map”.

2. The user must be able to click on any carpark in the list.

2.1. The system must display the relevant information related to the selected carpark.

2.1.1. The information must contain the exact distance of the carpark from the current location in kilometres/metres.

2.1.2. The information must contain the carpark’s availability.

2.1.3. The information must contain the parking fee rates in SGD.

2.1.4. The user must be able to navigate to the carpark by clicking on the “Go” button.

2.1.5. The user must be able to add the selected carpark to the saved list by clicking “Save Carpark”.

4.5 View Saved Carparks

4.5.1 Description

The user can view the list of carparks previously saved by him/ her and their relevant information.

4.5.2 Use Case

Use Case ID:	UC5.1		
Use Case Name:	View saved carpark		
Created By:	Ryan Pan	Last Updated By:	
Date Created:	2/9/2023	Date Last Updated:	

Actor:	User
Description:	The user will view a list of carparks that he/ her previously saved

Preconditions:	The user must click on "Saved"
Postconditions:	The system will display a list of carparks that the user saved
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	1. User enters the main page 2. User clicks on "Saved" 3. System retrieves a list of saved carparks from the user's account and displays them in a list
Alternative Flows:	AF-S1: If user has no saved carparks: 1. System displays "No saved carparks" 2. System returns to step 1
Exceptions:	NIL
Includes:	NIL
Special Requirements:	Database for User information
Assumptions:	1. The user must be connected to the internet
Notes and Issues:	NIL

4.5.3 Functional Requirements

1. The user must be able to view a list of saved carparks by clicking "Saved".
    1.1. The system must display a list of saved carparks.
2. The user must be able to click on any carpark in the list.
    2.1. The system must display the relevant information related to the selected carpark.
        2.1.1. The information must contain the exact distance of the carpark from the current location in kilometres/metres.
        2.1.2. The information must contain the carpark's availability.
        2.1.3. The information must contain the parking fee rates in SGD.
        2.1.4. The user must be able to navigate to the carpark by clicking on the "Go" button.

4.6 Search for Carparks

4.6.1 Description

The user can search for a carpark by location. Both the map and list will be updated with the search results.

4.6.2 Use Case

Use Case ID:	UC2		
Use Case Name:	Search carparks around an area		
Created By:	Choon Kang	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do a manual search around the specific area of 0.5km
Preconditions:	The user must launch the application.
Postconditions:	The nearest carpacks around the searched area will be displayed on the map or list.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. User opens the application 2. User search for a specific postal area 3. The system will scan around the area and find the nearest carpark within the set radius
Alternative Flows:	
Exceptions:	NIL
Includes:	Carpark Availability
Special Requirements:	NIL
Assumptions:	1. The user must have enabled location service on their device 2. The user must be connected to the internet
Notes and Issues:	NIL

4.6.3 Functional Requirements

1. The user must be able to search for carparks.
    1.1. The user must be able to enter a location to search in the search bar.
    1.2. The user must be able to view carparks around the location searched on map by clicking “Map”.
        1.2.1. The system must display the map of the location searched.
        1.2.2. The system must scale the map to show at least one carpark nearest to the location searched.
    1.3. The user must be able to view carparks around the location searched in a list by clicking “List”.
        1.3.1. The system must display a list of carparks ordered by distance to the location searched.

4.7 Navigate to Carpark

4.7.1 Description

After the user selects a carpark, the application will suggest a route to the selected carpark.

4.7.2 Use Case

Use Case ID:	UC4		
Use Case Name:	Navigating to carpark		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	Google Map API, User
Description:	Google Map will find the shortest route to the carpark that the user has decided
Preconditions:	User should have already made a selection regarding which car park they intend to visit
Postconditions:	User has successfully arrived at his destination

Priority:	High
Frequency of Use:	Once, each time the user wants to travel to their preselected car park
Flow of Events:	1. The user selects a carpark as their destination 2. The system utilizes Google Maps' API to determine the closest and most efficient route to the user's chosen car park
Alternative Flows:	AF-S1: If the user changes their mind about the car park destination enroute: 1. The system can recalculate the route based on its new selection
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User must be connected to the internet at all times 2. User has their location service enabled at all times for the Google Maps API to accurately determine their current location 3. User's device must be compatible with Google Maps API for it to function effectively 4. The system is properly integrated with Google Maps API which allows for seamless communication and route calculations
Notes and Issues:	NIL

4.7.3 Functional Requirements

1. The system must be able to navigate the user to a selected carpark.
    1.1. The user must be able to select a carpark to navigate to.
        1.1.1. The user must be able to select a carpark from the map by clicking the "Go" button.
        1.1.2. The user must be able to select a carpark from the list by clicking the "Go" button.

1.2. The system must be able to display the fastest route by travelling time to that carpark

4.8 Save Parking Information

4.8.1 Description

The user can save his/ her parking lot information in the application.

4.8.2 Use Case

Use Case ID:	UC5.2		
Use Case Name:	Save parking lot		
Created By:	1/9/2023	Last Updated By:	
Date Created:	Weihong	Date Last Updated:	

Actor:	User, System
Description:	User will enter the carpark lot that he parked his car at
Preconditions:	User has reached the car park After user has decided whether he wants to save the carpark for future use User has an account
Postconditions:	User saves the carpark lot
Priority:	Low
Frequency of Use:	Once
Flow of Events:	1. System will prompt the user to enter the carpark lot 2. User enters the carpark lot 3. System saves the information
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

4.8.3 Functional Requirements

1. The system must be able to store the user's parking information.
    1.1. The system must display a text field for the user to enter parking information.
        1.1.1. The text field must contain the carpark lot number.
    1.2. The user must fill in the text field before clicking on the "Save" button.
    1.3. The system must verify the information provided before saving the information.
        1.3.1. The text field must be filled.
    1.4. The system must save the parking information upon verifying parking information.
2. The system must be able to display the saved parking information when the user logs in.
    2.1. The parking information must contain carpark address.
    2.2. The parking information must contain carpark lot number.
    2.3. The parking information must contain time of parking.

4.9 Make Payment through Parking.sg

4.9.1 Description

User can be redirected to the Parking.sg application from our application to make payment for coupon parking carparks.

4.9.2 Use Case

Use Case ID:	UC6		
Use Case Name:	Pay Parking fees		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System, Parking.sg
Description:	User use the app to pay for parking fees for carpark without gantries

Preconditions:	User reached the car park
Postconditions:	User pays his fees
Priority:	Medium
Frequency of Use:	Each time User parks in a car park that has no gantry
Flow of Events:	1. User parks his car 2. User pays his parking fees using a link sent to him through System app
Alternative Flows:	AF-S1: User parks his car at carparks with electronic gantries: 1.
Exceptions:	NIL
Includes:	Integration with Parking.sg
Special Requirements:	NIL
Assumptions:	1. User visits a coupon parking carpark instead of the ones with electronic gantries 2. User must be connected to the internet at all times 3. User must have already created an account with Parking.sg 4. User must have already save their credit card information with Parking.sg 5. System is properly integrated with Parking.sg to allow seamless transition from one app to another
Notes and Issues:	NIL

4.9.3 Functional Requirements

1. The user must be able to make payment through Parking.sg
    1.1. The user must be able to select a carpark.
        1.1.1. The user must be able to select a carpark from the map by clicking the "Pay" button.
        1.1.2. The user must be able to select a carpark from the list by clicking the "Pay" button.
    1.2. The system must redirect user to their Parking.sg application.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The application must load within 5 seconds of launch.
2. The application must render the map based on the user's current location or search requests within 5 seconds.
3. The application must render the list of carparks based on the user's current location or search requests within 5 seconds.

5.2 Usability Requirements

1. 90% of users must be able to view and search for carparks through our application within 1 minute.
2. The system must be able to sort the carparks by distance.
3. 85% of users must be able to select a carpark to navigate to within 2 minutes.

5.3 Security Requirements

1. All user account passwords must be hashed and protected.
2. All information relating to the user's account must be stored adhering to data privacy regulations.
3. All live location must not be stored in the application's database.

5.4 Reliability Requirements

1. The application must be able to reload within 5 seconds of launch.

6. Other Requirements

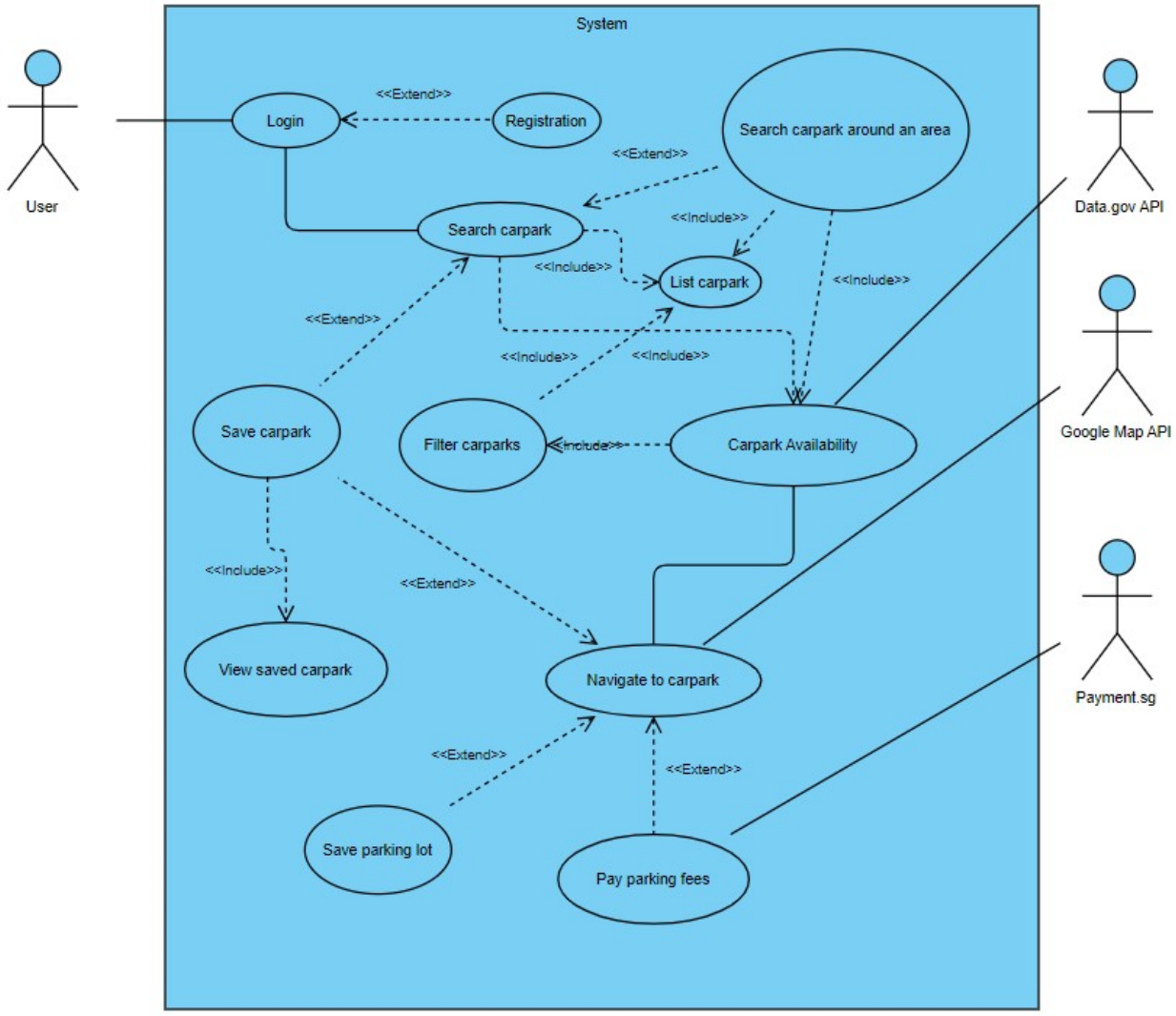
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Data Dictionary

Term	**Definition**
User	Refers to a person who needs to park a vehicle and uses the website itself
Location	Refers to a particular location
State	Refers to the state of a parking lot, whether it is occupied or not
Vacant Spaces	Refers to the number of empty carpark lots
Occupied Spaces	Refers to the current count of occupied carpark lots
Parking Lot	Refers to an area where a user can park their car
Parking Lot Name	Refers to the name of the parking lot
Parking Lot No.	Refers to the users' parked car lot number
Search Radius	Refers to the value decided by the user to query nearby car parks in the vicinity
Availability	Refers to the number of parking lots available for a carpark
Availability Indicator	Refers to the visual representation of a carpark indicating its' availability status
Parking Rates	Refers to the amount to be paid/hr for a carpark
Update Frequency	Refers to how often the data will be updated/refreshed

Green	0 - 24% of the carpark being filled
Yellow	25 - 49% of the carpark being filled
Red	50 - 74% of the carpark being filled
Black	75 - 100% of the carpark being filled
Filter	Refers to the sorting and display of requirements that the user selects to view his desired carpark
Distance	Refers to the distance from the user to the selected carpark
Username	Refers to the unique identifier for a user account
Password	The encrypted or hashed representation of a user's password
Email Address	Refers to the email associated with the user's account

Appendix B: Use Case Diagram



Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc