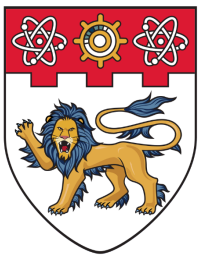


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 Software Engineering

FDAR Group 2 (Parked Up)

Final Report

Group Members:

Full Name	**Matriculation Number**
Ang Qile, Dora	U2421379G
Ethan Wong Kee Jann	U2323765K
Zhang Yuhe	U2422060C
Tan Zhi Rong, Marvin	U2422502D

Table of Contents

Software Requirements

Project Description

Parked Up is a web application that helps drivers in Singapore find available parking lots in real time, reducing the time they spend searching for carparks. Our application uses data from data.gov.sg APIs to provide carpark information and their real-time parking lot availability.

Upon opening the app, carparks around Singapore are displayed on a map as “bubbles”, with numbers in them representing the number of available lots. Users can toggle between Map View and List View, and click on carparks to view more details. They can also search for carparks by address, and apply filters to only view carparks that fit their requirements. Users can also sign up for an account to save carparks for easy access to the carpark’s info and directions.

Parked Up aims to improve users’ daily commute and parking experience, and help to create a more efficient and driver-friendly city.

Functional Requirements

1. The driver must be able to sign up for an account
    1.1. The driver must input their username
    1.2. The driver must input their phone number
    1.3. The driver must input their email address
    1.4. The driver must set a secure account password
    1.5. Users must be able to change their account information (username, phone number, email address, and password)

2. The system must be able to display carparks in a map view, and in a list view

3. The system must be able to display information of selected carpark
    3.1. The system must display the carpark’s address
    3.2. The system must display the carpark’s type
    3.3. The system must display the carpark’s lots availability
    3.4. The system must display the carpark’s height

4. Users can save specific carparks to their account for quick access, and remove saved carparks

5. The system must allow the driver to filter carparks in both map and list view
    5.1. The driver must be able to filter by address
    5.2. The driver must be able to filter by carpark type

5.3. The driver must be able to filter by lot availability
5.4. The driver must be able to filter by carpark height
5.5. In list view, users must also be able to filter carparks by their proximity

6. The system must display real time information of the carparks from the system
   6.1. The system must display parking lot availability in real time
   6.2. The system must display the current location of the driver in real time

Non-Functional Requirements

1. Performance & Reliability
   1.1. The system must load within 3 seconds under normal network conditions.
   1.2. The system must handle at least 1,000 concurrent users without performance degradation.

2. Usability
   2.1. The system must have a simple and intuitive UI for users of all ages.

3. Security
   3.1. Passwords must meet certain requirements (e.g. must be 10 characters long, must have one number)

4. Compatibility
   4.1. The system must work on major browsers (e.g. Chrome, Safari, Edge)

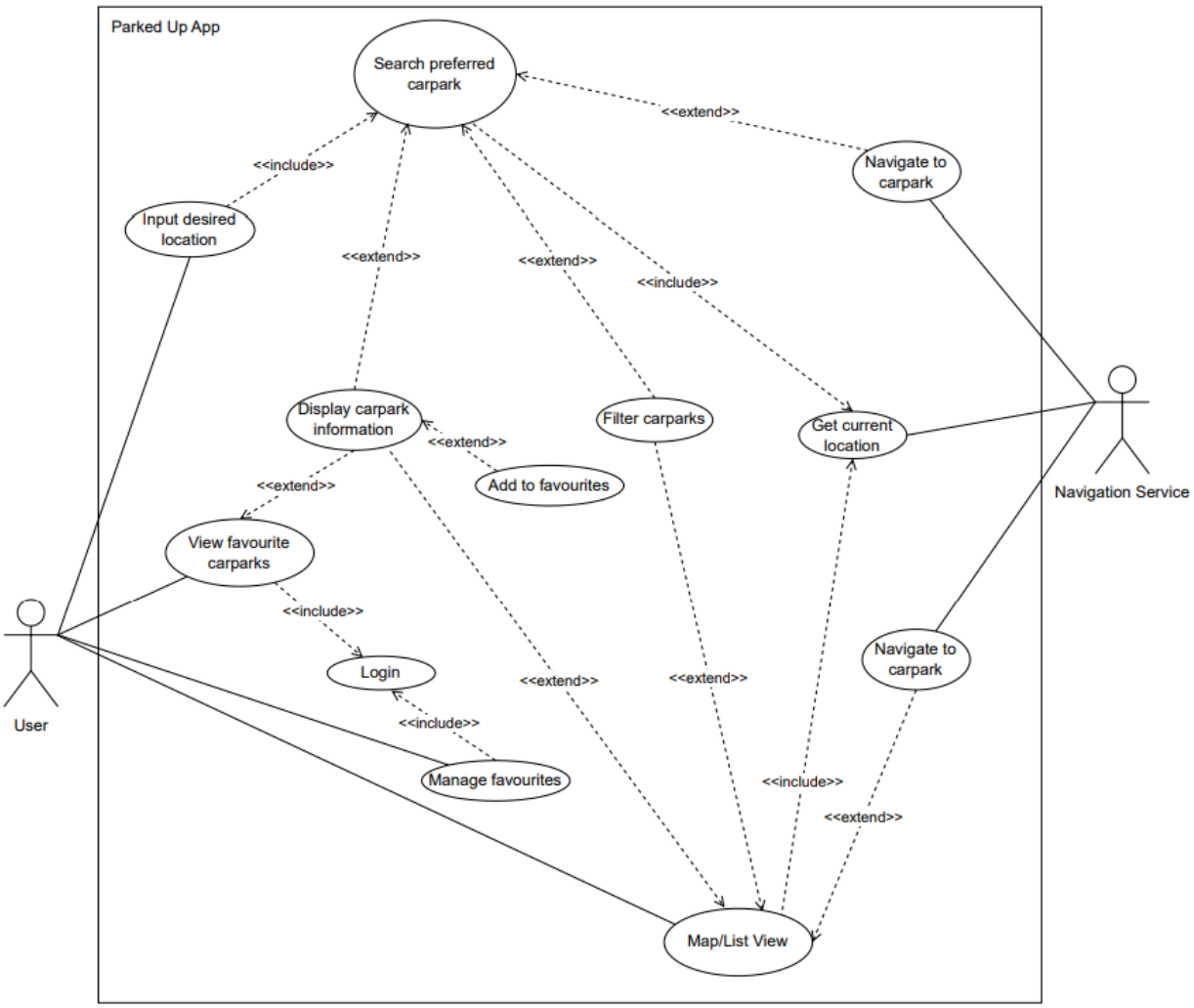
Data Dictionary

Term	Definition
Driver	A person operating a vehicle
Vehicle	A mode of transport (e.g., car, motorcycle, van)
Carpark	A place where drivers may temporarily park their vehicles
Parking lot	A slot where one vehicle may be parked
Carpark location	The specific geographic coordinates or address of a parking facility
Carpark number	A unique alphanumeric ID assigned to each HDB carpark, used to identify and reference it in datasets and systems.
Carpark type	Carparks are classified into different types. For our system there are only 5 types of carparks: "Basement Car Park", "Multi-Storey Car Park", "Surface Car Park", "Mechanised Car Park", and "Mechanised And Surface Car Park"

Carpark availability	The real-time status of vacant parking spaces in a given carpark
Carpark height	The maximum allowable vehicle height for entry into a parking facility
HDB Carpark	A parking facility managed by Singapore's Housing & Development Board (HDB), typically serving residential estates and public areas
Favourites List	A list of carparks saved by a user

Use Case Model

Use Case Diagrams



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Sign Up		
Created By:	Dora	Last Updated By:	Dora
Date Created:	20/02/25	Date Last Updated:	09/04/25

Actor:	User, Database
Description:	User signs up for an account.
Preconditions:	1. The user's device must have access to the internet. 2. The system must be connected to the database. 3. The user must not already have an account.
Postconditions:	User is redirected to login page.
Priority:	Medium
Frequency of Use:	Once for new users
Flow of Events:	1. The user clicks on "Signup" from the navigation bar. 2. The system displays the Signup page. 3. The user is required to enter: a. Username b. Email address c. Phone number d. Password 4. User enters their information and clicks the "Signup" button to sign up. 5. The system validates the information: a. Ensures all fields are filled b. Checks for valid and unused email c. Checks for valid and unused phone number d. Checks password strength based on the following conditions: i. Minimum 8 characters. ii. At least 1 alphabet should be of Upper Case [A-Z] iii. At least 1 number or digit between [0-9]. iv. At least 1 symbol. 6. The system saves their information into the database. 7. The system displays "Signup successful! Please login.". 8. The system displays the login page when user clicks "OK".
Alternative Flows:	1. AF-S5: Email already registered a. The system displays "Email already exists!" until the user clicks "OK" b. Return to step 2 2. AF-S5: Phone number already registered a. The system displays "Failed to create account!" until the user clicks "OK" b. Return to step 2 3. AF-S4: Weak password a. The system displays the message "Failed to create account!" until the user clicks "OK" b. Return to step 2

Exceptions:	1. EX1: System unavailable a. The app displays that the system is unavailable.
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	2		
Use Case Name:	Login		
Created By:	Dora	Last Updated By:	Dora
Date Created:	20/02/25	Date Last Updated:	09/04/25

Actor:	User, Database
Description:	User logs in to the app using their email and password.
Preconditions:	1. The user's device must have access to the internet. 2. The system must be connected to the database. 3. The user must already have an account.
Postconditions:	The app displays the homepage in map view.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	1. User clicks on "Login" from the navigation bar. 2. The system displays the login page. 3. The user is required to enter: a. Email b. Password 4. User clicks the "Login" button to login. 5. The system authenticates the user's credentials with the database. 6. User is logged in to their account.
Alternative Flows:	1. AF-S4: Email not in database a. The system displays "Invalid email or password" until the user clicks "OK" b. Return to step 3 2. AF-S4: Wrong password a. The system displays "Invalid email or password" until the user clicks "OK" b. Return to step 3
Exceptions:	1. EX1: System unavailable a. The app displays that the system is unavailable.
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	3		
Use Case Name:	Carpark Search in Map View		
Created By:	Marvin	Last Updated By:	Dora

Date Created:	20/02/25	Date Last Updated:	09/04/25

Actor:	User, Google Maps
Description:	User searches for carparks in map view and navigates to a carpark
Preconditions:	1. The user's device must have access to the internet. 2. User must have location services on their device, and given permission to the system to access it.
Postconditions:	1. User is redirected to Google Maps if they choose to navigate to a carpark. 2. Default map view is displayed if the user exits the search without selecting a carpark.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User opens the app. Default map view is displayed. 2. The system requests the user's current location from the device's GPS and displays it in the map. 3. The system retrieves carpark data from the API: a. Carpark name b. Address c. Available Lots d. Gantry Height e. Carpark Type 4. The system displays a map of the users' surroundings. The map includes numbers in bubbles. The bubbles represent a single carpark, and the numbers represent the number of available lots in that carpark. 5. User selects a carpark bubble from the map. 6. The system displays carpark details. 7. Users clicks on the "Directions" button to navigate to carpark. 8. User is redirected to Google Maps website for navigation to the selected carpark.
Alternative Flows:	1. AF-S2: System cannot retrieve users' current location a. System displays the message "Error fetching geolocation". b. Return to step 2
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	4		
Use Case Name:	Carpark Search in List View		
Created By:	Ethan	Last Updated By:	Dora

Date Created:	20/02/25	Date Last Updated:	09/04/25

Actor:	User, Google Maps
Description:	User searches for carparks in list view and navigates to a carpark
Preconditions:	1. The user's device must have access to the internet. 2. User must have location services on their device, and given permission to the system to access it
Postconditions:	1. User is redirected to Google Maps if they choose to navigate to a carpark. 2. Default list view is displayed if the user exits the search without selecting a carpark.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User opens the app. Default map view is displayed. 2. The system requests the user's current location from the device's GPS and displays it in the map. 3. The system retrieves carpark data from the API: a. Carpark number b. Address c. Available Lots d. Gantry Height e. Carpark Type 4. User clicks on "List View" from the navigation bar. 5. The system displays a list of nearby carparks sorted from nearest to furthest, and their available lots. 6. User selects a carpark from the list to view detailed information. 7. The system displays carpark details. 8. Users clicks on the "Directions" button to navigate to carpark. 9. User is redirected to Google Maps website for navigation to the selected carpark.
Alternative Flows:	1. AF-S2: System cannot retrieve users' current location a. System displays the message "Error fetching geolocation". b. Return to step 2
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	5		
Use Case Name:	Filtering in Map View		
Created By:	Ethan	Last Updated By:	Dora
Date Created:	20/02/25	Date Last Updated:	08/04/25

Actor:	User, Google Maps
Description:	User filters carparks by available lots and navigates to a carpark
Preconditions:	1. The user's device must have access to the internet. 2. The user must have location services available on their device, and give permission to the system to access them.
Postconditions:	1. User is redirected to Google Maps if they choose to navigate to a carpark. 2. Default map view is displayed if the user exits the search without selecting a carpark.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User opens the app. Default map view is displayed. 2. The system requests the user's current location from the device's GPS and displays it in the map. 3. The system retrieves carpark data from the API: a. Carpark number b. Address c. Available Lots d. Gantry Height e. Carpark Type 4. The system displays a map of the users' surroundings. The map includes numbers in bubbles. The bubbles represent a single carpark, and the numbers represent the number of available lots in that carpark. 5. User clicks the filter icon on the page. 6. The system displays the carpark filters panel, allowing the user to filter by available lots, gantry height, or carpark type. 7. The user filters carparks based on available lots (e.g. at least 30 available lots) 8. The system applies the filter and updates the map accordingly, only displaying carparks that fulfill the requirements. 9. User selects a carpark bubble from the map. 10. The system displays carpark details. 11. Users clicks on the "Directions" button to navigate to carpark. 12. User is redirected to Google Maps website for navigation to the selected carpark.
Alternative Flows:	1. AF-S2: System cannot retrieve users' current location a. System displays the message "Error fetching geolocation". b. Return to step 2
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A

Notes and Issues:	Instead of filtering by available lots, the user may also filter by carpark type, or gantry height.

Use Case ID:	6		
Use Case Name:	Add Favourite Carpark in Map View		
Created By:	Yuhe	Last Updated By:	Dora
Date Created:	20/02/25	Date Last Updated:	09/04/25

Actor:	User, Google Maps, Database
Description:	User adds a favourite carpark to their account from the map.
Preconditions:	1. The user's device must have access to the internet. 2. The user must be logged into their account.
Postconditions:	1. The carpark is saved to the user's account and appears in the "Favourites" tab if the user chooses to add it. 2. The carpark is not saved and does not appear in the "Favourites" tab if the user exits before adding it.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User opens the app. Default map view is displayed. 2. The system requests the user's current location from the device's GPS and displays it in the map. 3. The system retrieves carpark data from the API: a. Carpark number b. Address c. Available Lots d. Gantry Height e. Carpark Type 4. The system displays a map of the users' surroundings. The map includes numbers in bubbles. The bubbles represent a single carpark, and the numbers represent the number of available lots in that carpark. 5. User selects a carpark bubble from the map. 6. The system displays carpark details. 7. User clicks on the heart icon. Carpark is saved to their favourites. 8. User clicks on "Favourites" from the navigation bar. 9. The system displays the user's list of favourite carparks.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	7		
Use Case Name:	Manage profile		
Created By:	Marvin	Last Updated By:	Dora
Date Created:	19/03/25	Date Last Updated:	09/04/25

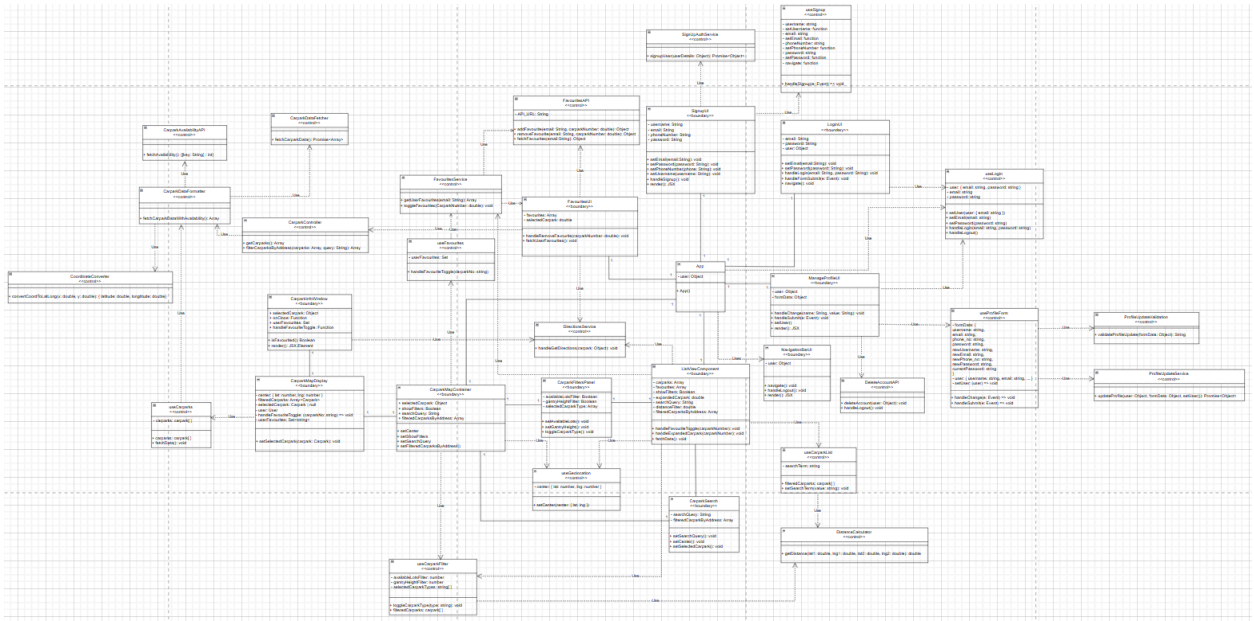
Actor:	User, Database
Description:	Users views their profile and may update their information (i.e. email, password, username and/or phone number)
Preconditions:	1. The user's device must have access to the internet. 2. User must be logged into their account.
Postconditions:	1. The system displays the updated profile page if the user successfully updates their information. 2. The system displays the old information if the user exits before successfully updating their information.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	1. User clicks on the profile icon on the navigation bar. 2. The system displays the current user's information. 3. User inputs any updated fields. 4. User inputs their current password for verification. 5. User clicks on "Update Profile" to confirm the update. 6. The system validates the information: a. Checks for valid and unused email b. Checks for valid and unused phone number c. Checks password strength based on the following conditions: i. Minimum 8 characters. ii. At least 1 alphabet should be of Upper Case [A-Z] iii. At least 1 number or digit between [0-9]. iv. At least 1 symbol. d. Checks that the current password is correct. 7. The system updates their information into the database.
Alternative Flows:	1. AF-S3: The user cancels the update process a. The system returns to the previous page without making any profile changes 2. AF-S4: The user enters an incorrect password a. The system displays "Current password is incorrect!" until the user clicks "OK" b. Return to step 4 3. AF-S6: The user's input is invalid a. The system displays "Failed to update profile!" until the user clicks "OK". b. Return to step 3
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	8		
Use Case Name:	Remove a Favourite Carpark		
Created By:	Yuhe	Last Updated By:	Dora
Date Created:	19/03/25	Date Last Updated:	09/04/25

Actor:	User
Description:	User removes a favourite carpark from the “Favourites” page.
Preconditions:	1. The user’s device must have access to the internet. 2. The user must be logged into their account. 3. The user must already have a carpark in their “Favourites” list
Postconditions:	1. The system displays the updated “Favourites” page if the user successfully removes the carpark from their favourites. 2. The “Favourites” page remains the same if the user exits before removing the carpark from their favourites.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user clicks on "Favourites" from the navigation bar. 2. The system displays the list of the user’s favourite carparks. 3. The user selects the carpark they want to remove. 4. The system displays the carpark details. 5. The user clicks the "Remove" button. 6. The system removes the carpark from the user’s favourite carparks in the database.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

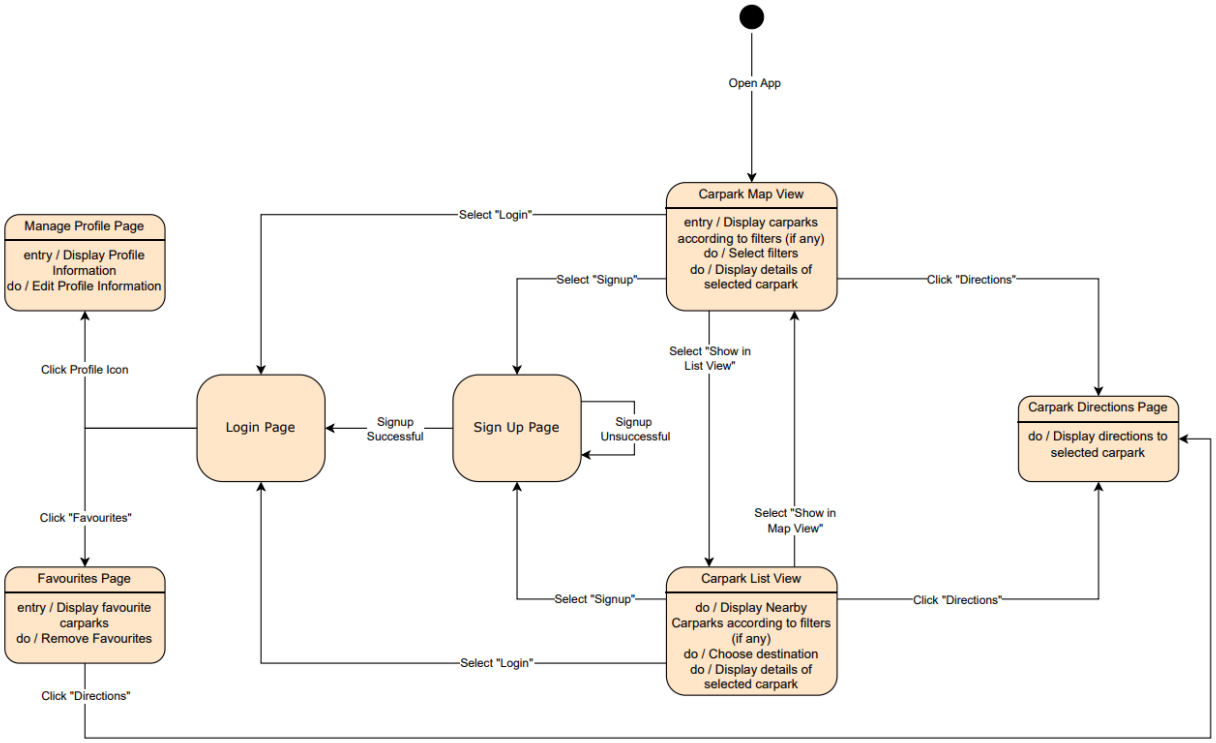
UML Class Diagram

Refer to the pdf attached for a clearer view.



Dialog Map

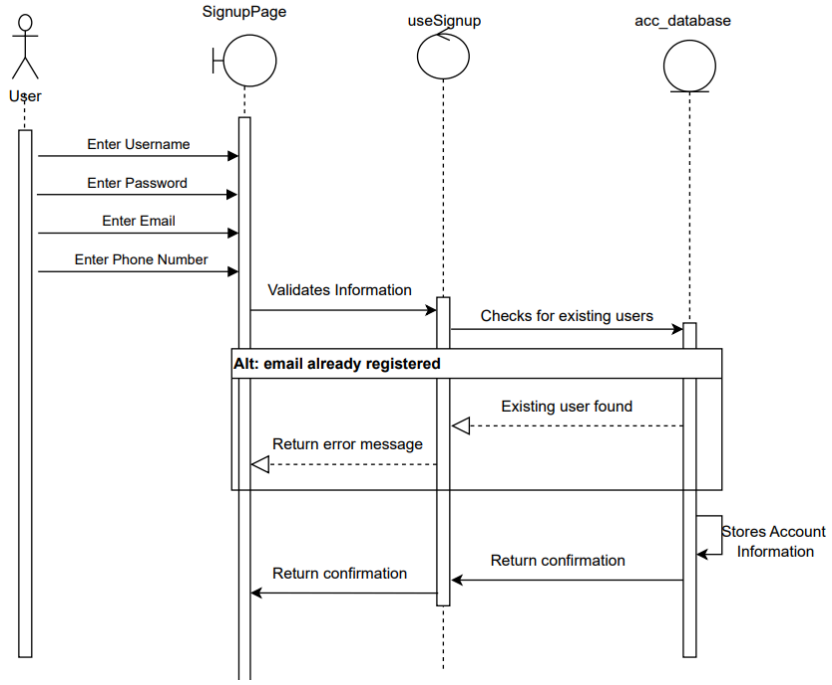
Refer to the pdf attached for a clearer view.



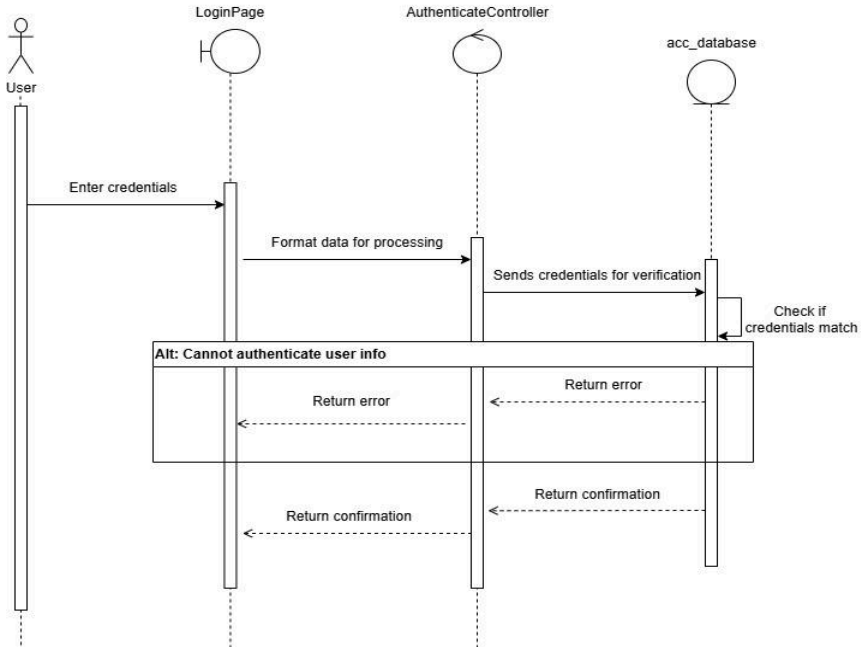
Sequence Diagrams

Refer to the pdf attached for a clearer view.

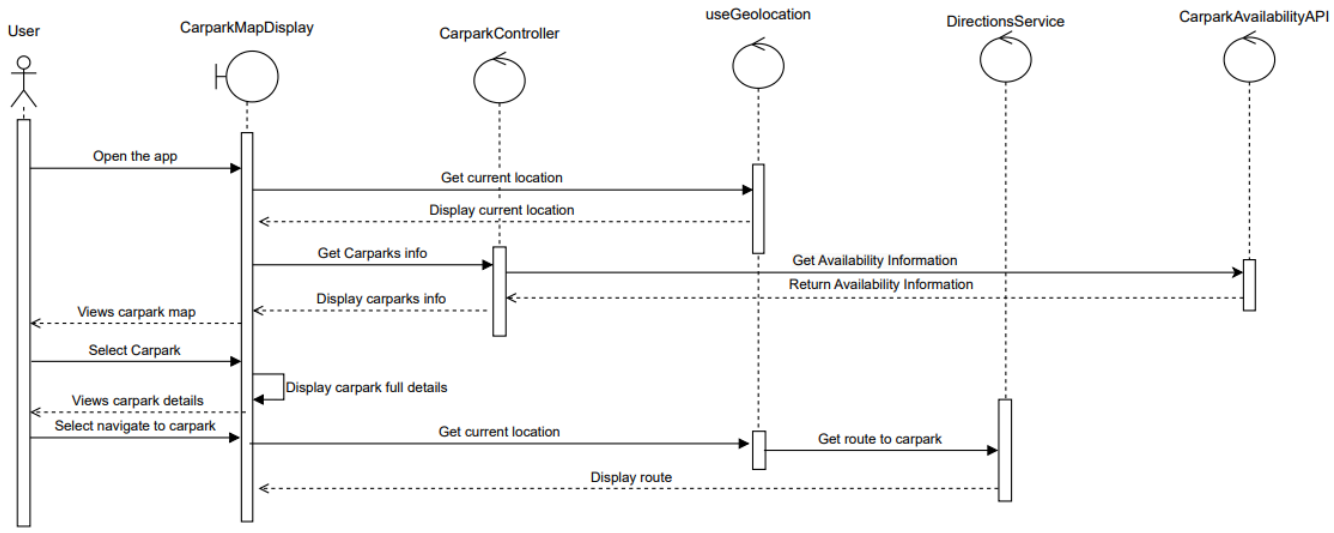
Use Case 1 - Sign Up



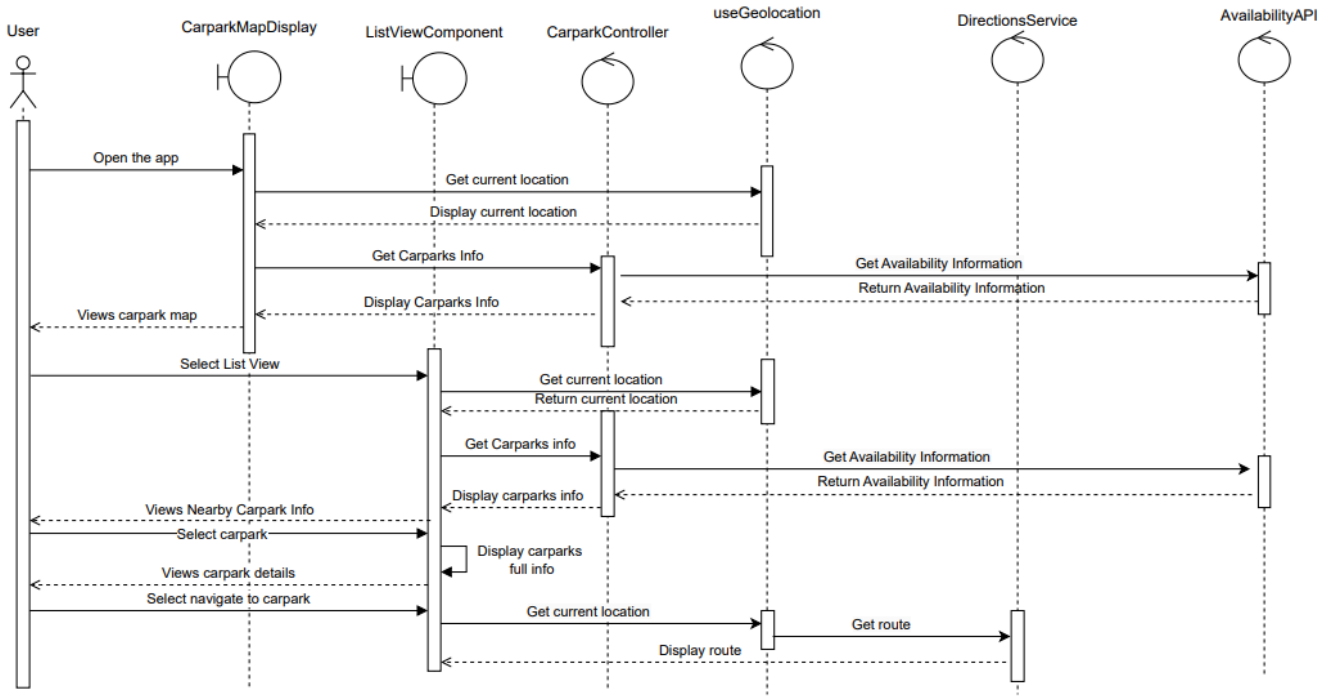
Use Case 2 - Login



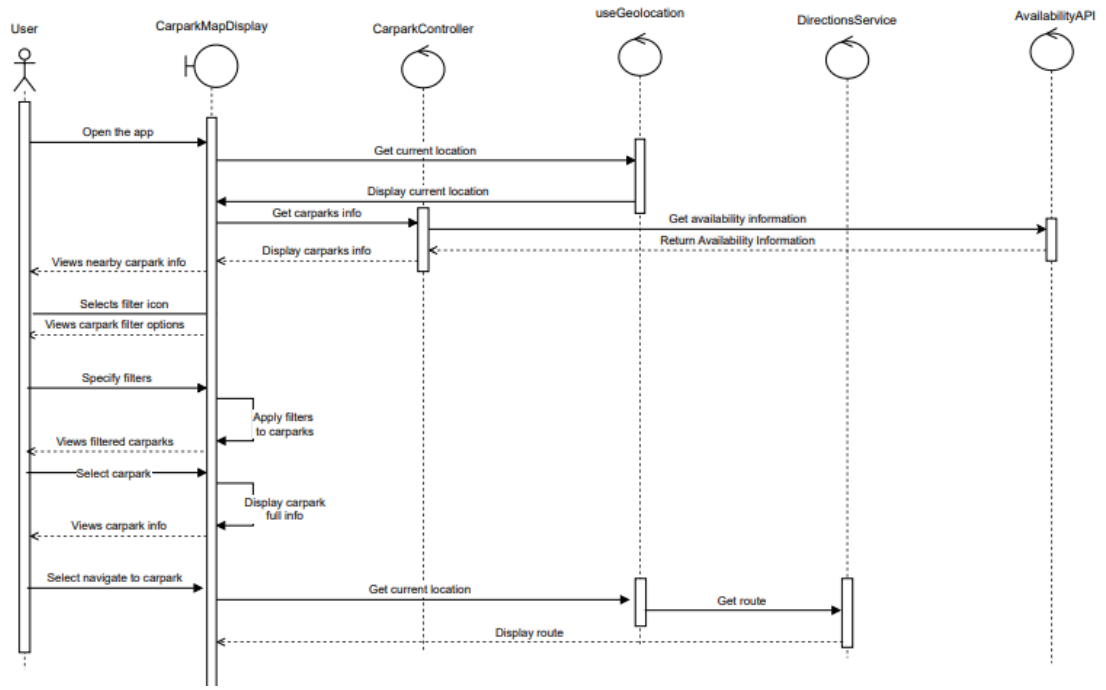
Use Case 3 - Carpark Search in Map View



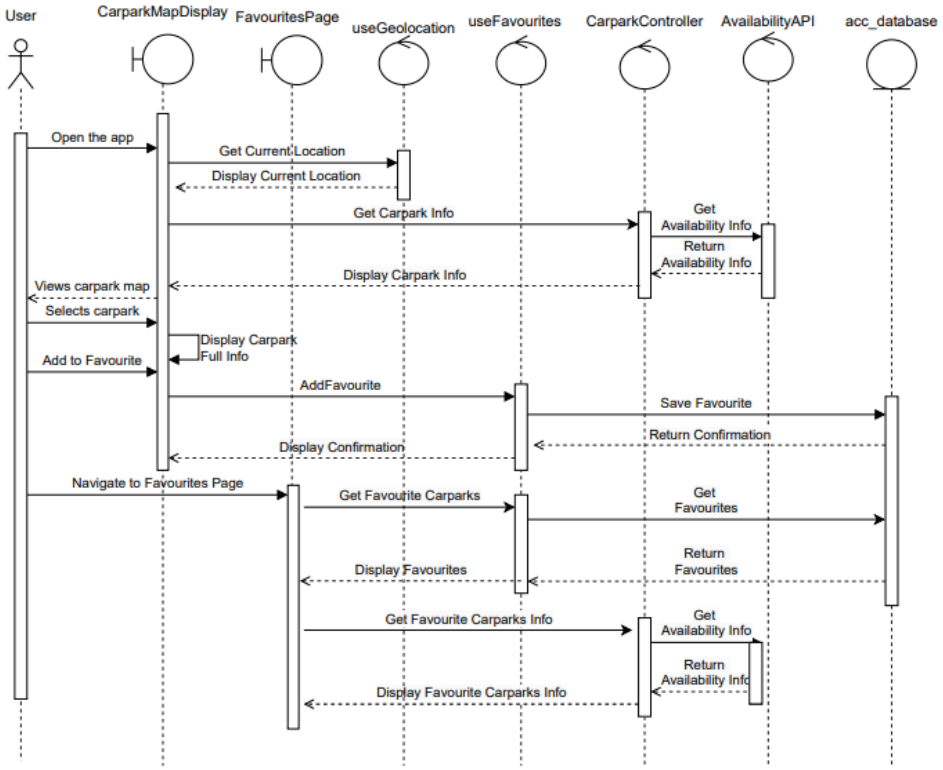
Use Case 4 - Carpark Search in List View



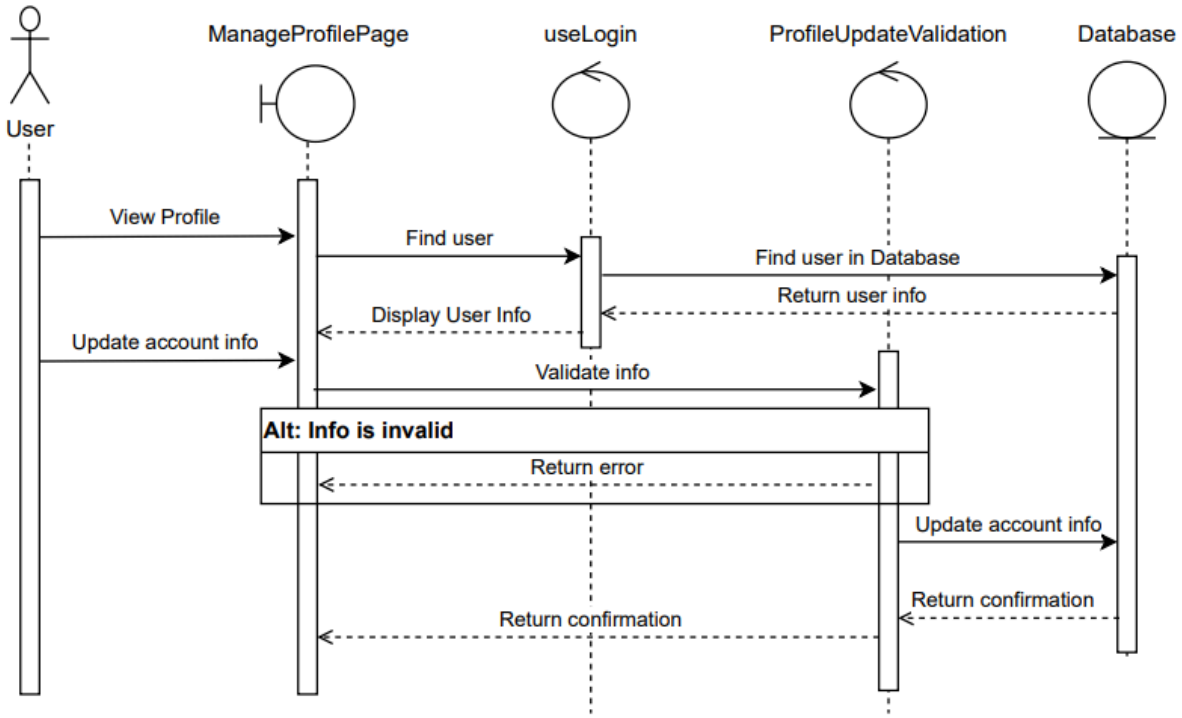
Use Case 5 - Filtering in Map View



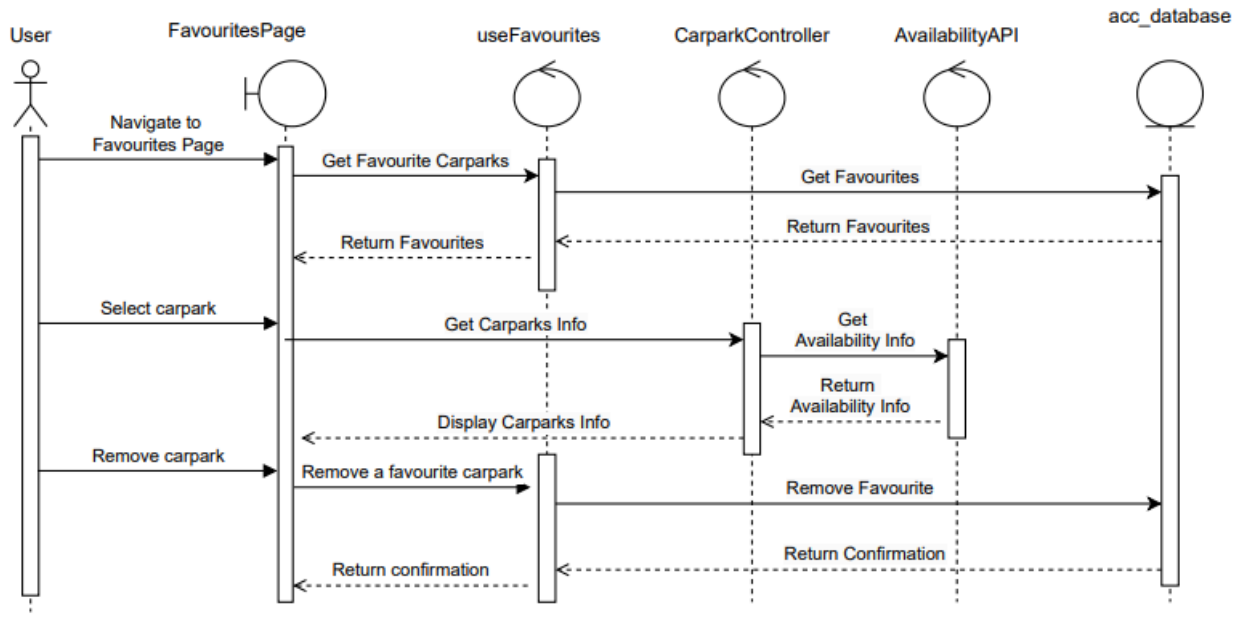
Use Case 6 - Add Favourite Carpark in Map View



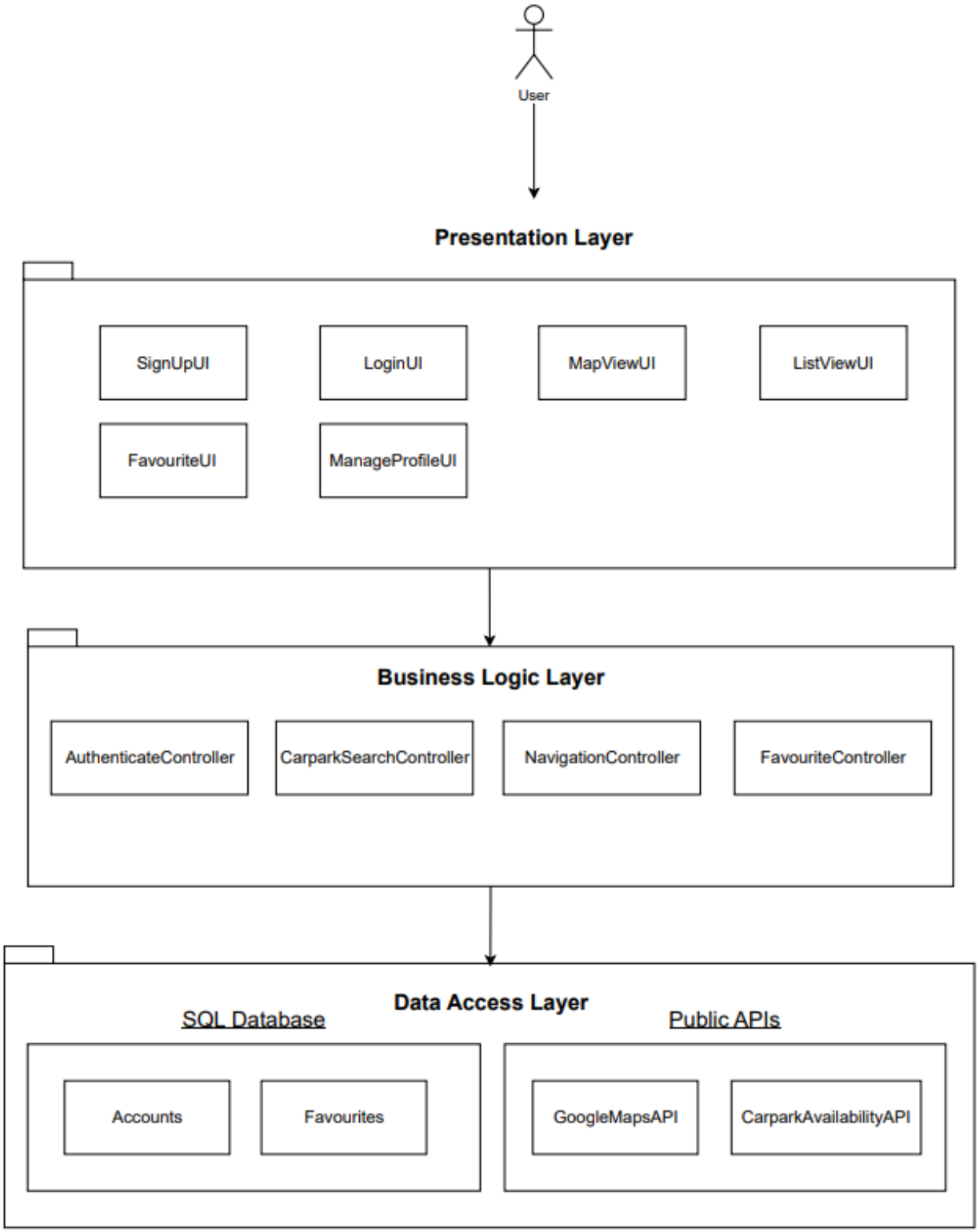
Use Case 7 - Manage Profile



Use Case 8 - Remove a Favourite Carpark



System Architecture



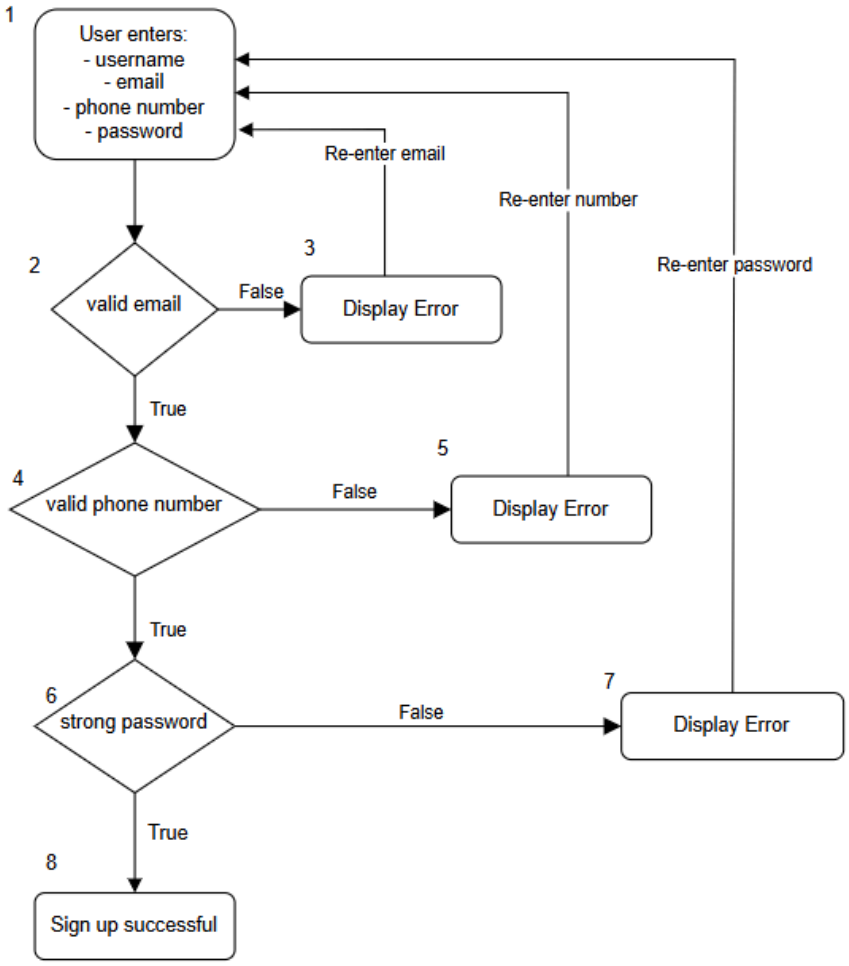
Test Cases

Test case 1: Sign Up

Test case name	Sign Up
Description	User signs up for an account
Preconditions	1. The user's device must have access to the internet. 2. The system must be connected to the database.

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User signs up successfully	1. User clicks the "Signup" button on the homepage. 2. User enters: a. Username b. Email c. Phone number d. Password 3. User clicks the "Signup" button to sign up.	- Person1 - person1@gmail.com - 88888888 - Person1PW@123	"Signup successful! Please login."	"Signup successful! Please login."	Passed
2	User enters an email that is already registered.	1. User clicks the "Signup" button on the homepage. 2. User enters: a. Username b. Email c. Phone number d. Password 3. User clicks the "Signup" button to sign up.	- dora2 - dora@gmail.com - 81238123 - Dora2PW@123	"Email already exists!"	"Email already exists!"	Passed
3	User enters a phone number that is already registered.	1. User clicks the "Signup" button on the homepage. 2. User enters: a. Username b. Email c. Phone number d. Password 3. User clicks the	- dora2 - dora2@gmail.com - 88888888 - Dora2PW@123	"Failed to create account!"	"Failed to create account!"	Passed

		“Signup” button to sign up.				
4	User enters a weak password	1. User clicks the “Signup” button on the homepage. 2. User enters: a. Username b. Email c. Phone number d. Password 3. User clicks the “Signup” button to sign up.	- dora2 - dora2@gmail.com - 82828282 - Dora2PW	“Failed to create account!”	“Failed to create account!”	Passed



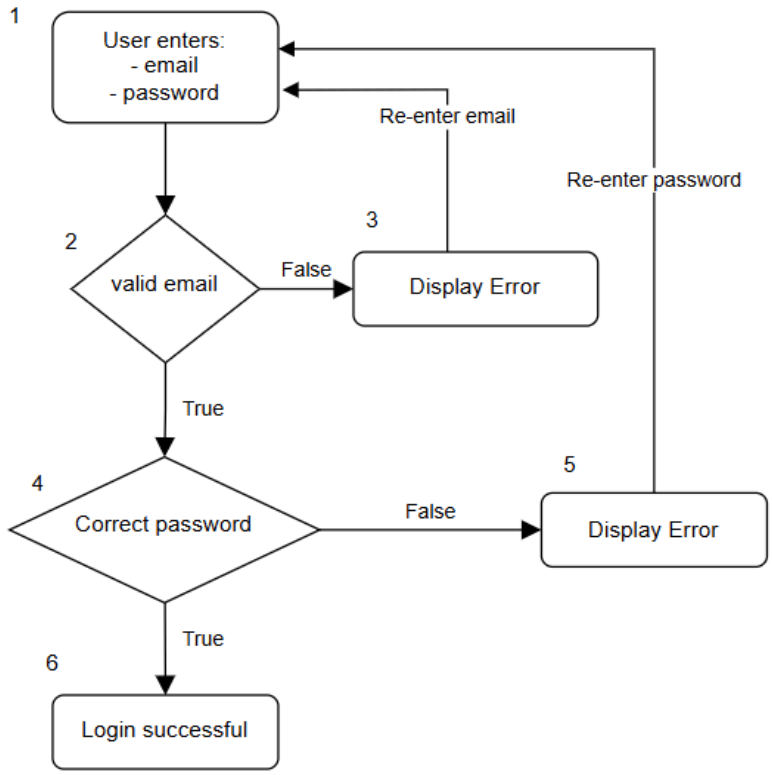
Basis Path #1 (Successful Login): 1, 2, 4, 6, 8

Basis Path #2 (Invalid Email): 1, 2, 3, 1, 2, 4, 6, 8
Basis Path #3 (Invalid Phone Number): 1, 2, 4, 5, 1, 2, 4, 6, 8
Basis Path #4 (Invalid Password): 1, 2, 4, 6, 7, 1, 2, 4, 6, 8

Test case 2: Login

Test case name	Login
Description	User logs in to their account
Preconditions	1. The user's device must have access to the internet. 2. The system must be connected to the database.

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User successfully logs in to their account	1. User clicks the "Login" button on the homepage. 2. User enters: a. Email b. Password 3. User clicks the "Login" button to login.	- dora@gmail.com - DoraPW@123	"Login successful!"	"Login successful!"	Passed
2	User enters an email that is not registered	1. User clicks the "Login" button on the homepage. 2. User enters: a. Email b. Password 3. User clicks the "Login" button to login.	- wrongemail@gmail.com - DoraPW@123	"Invalid email or password"	"Invalid email or password"	Passed
3	User enters the wrong password.	1. User clicks the "Login" button on the homepage. 2. User enters: a. Email b. Password 3. User clicks the "Login" button to login.	- dora@gmail.com - wrongpw@123	"Invalid email or password"	"Invalid email or password"	Passed



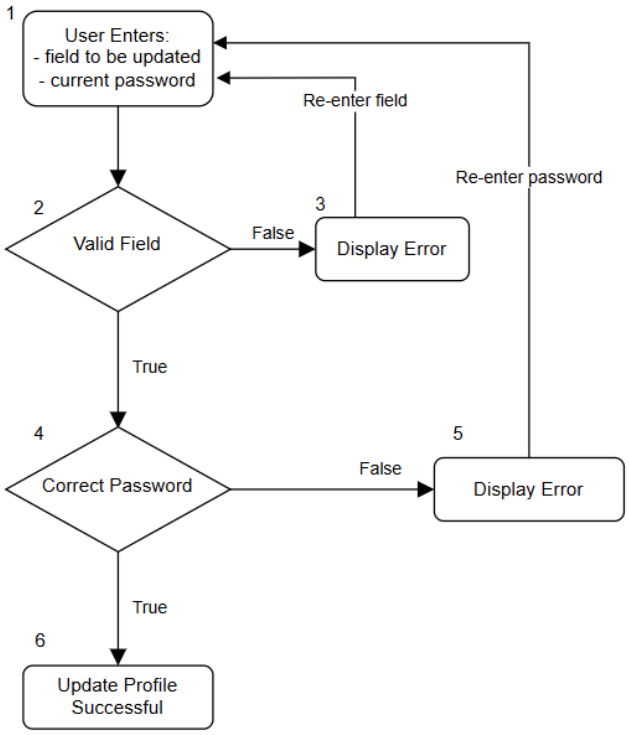
Basis Path #1 (Successful Login): 1, 2, 4, 6
Basis Path #2 (Invalid Email): 1, 2, 3, 1, 2, 4, 6
Basis Path #3 (Incorrect Password): 1, 2, 4, 5, 1, 2, 4, 6

Test case 3: Update Account Info

Test case name	Update account information
Description	User updates their account information
Preconditions	1. The user's device must have access to the internet. 2. The system must be connected to the database.

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User updates information successfully	1. User clicks on profile icon. 2. User enters field to be updated 3. User enters	- Username: Jane - Email: jane@gmail.com - Current password:	"Profile updated successfully!"	"Profile updated successfully!"	Passed

		current password. 4. User clicks on “Update Profile” button to update profile.	DoraPW@123			
2	User enters invalid field to update	1. User clicks on profile icon. 2. User enters field to be updated 3. User enters current password. 4. User clicks on “Update Profile” button to update profile.	- Username: Jane - Email: jane@gmail.com - New password: badpassword - Current password: DoraPW@123	“Failed to update profile!”	“Failed to update profile!”	Passed
3	User enters wrong password and fails to update information.	1. User clicks on profile icon. 2. User enters field to be updated 3. User enters current password. 4. User clicks on “Update Profile” button to update profile.	- Username: Jane - Email: jane@gmail.com - Current password: wrongpassword	“Current password is incorrect!”	“Current password is incorrect!”	Passed



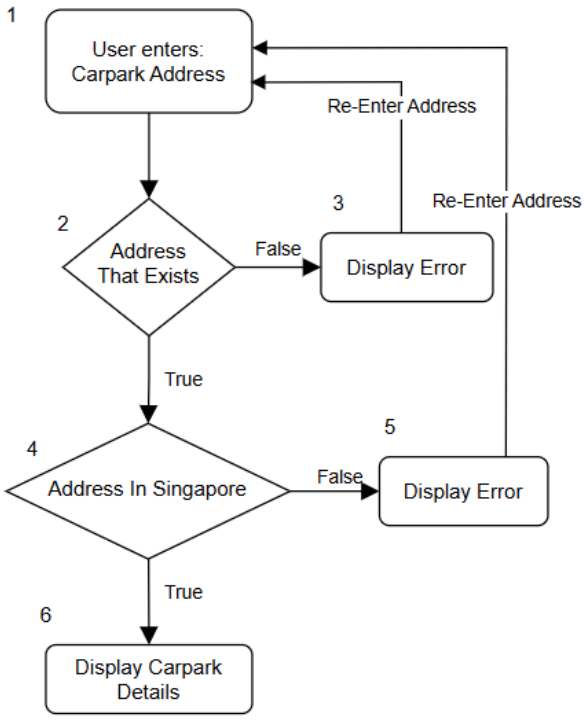
Basis Path #1 (Update Profile Successful): 1, 2, 4, 6
Basis Path #2 (Invalid Field): 1, 2, 3, 1, 2, 4, 6
Basis Path #3 (Incorrect Password): 1, 2, 4, 5, 1, 2, 4, 6

Test case 4: Search Carpark (Map View)

Test case name	Search Carpark (Map View)
Description	User searches for a carpark in map view
Preconditions	1. The user's device must have access to the internet. 2. User must have location services on their device, and given permission to the system to access it

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User searches for an existing carpark	1. User clicks on search bar. 2. User enters carpark to search	- BLK 517B JURONG WEST STREET 52	Carpark details displayed from bubble	Carpark details displayed from bubble	Passed

		for. 3. User clicks on desired carpark result.				
2	User searches for a carpark that does not exist	1. User clicks on search bar. 2. User enters carpark to search for.	- Jurong North	No results displayed	No results displayed	Passed
3	User searches for a carpark outside of Singapore	1. User clicks on search bar. 2. User enters carpark to search for.	- Mid Valley Southkey	No results displayed	No results displayed	Passed

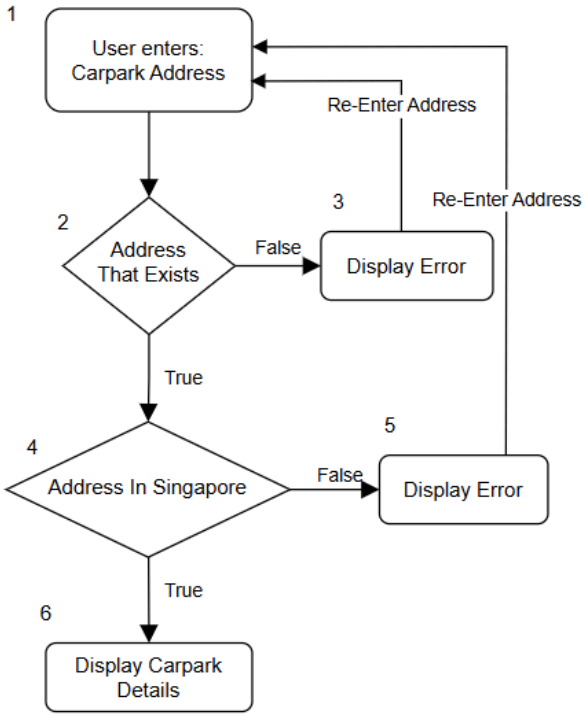


Basis Path #1 (Successful Search): 1, 2, 4, 6
Basis Path #2 (Address Does Not Exist): 1, 2, 3, 1, 2, 4, 6
Basis Path #3 (Address Not In Singapore): 1, 2, 4, 5, 1, 2, 4, 6

Test case 5: Search Carpark (List View)

Test case name	Search Carpark (List View)
Description	User searches for a carpark in list view
Preconditions	3. The user's device must have access to the internet. 4. User must have location services on their device, and given permission to the system to access it

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User searches for an existing carpark	4. User clicks on search bar. 5. User enters carpark to search for. 6. User clicks on desired carpark result.	- BLK 517B JURONG WEST STREET 52	Carpark details displayed as list	Carpark details displayed as list	Passed
2	User searches for a carpark that does not exist	3. User clicks on search bar. 4. User enters carpark to search for.	- Jurong North	No results displayed	No results displayed	Passed
3	User searches for a carpark outside of Singapore	3. User clicks on search bar. 4. User enters carpark to search for.	- Mid Valley Southkey	No results displayed	No results displayed	Passed



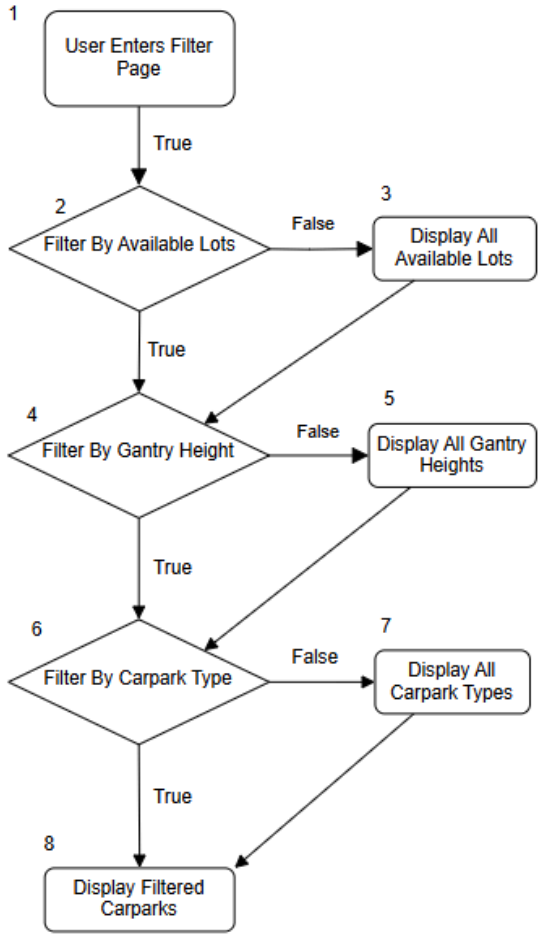
Basis Path #1 (Successful Search): 1, 2, 4, 6
Basis Path #2 (Address Does Not Exist): 1, 2, 3, 1, 2, 4, 6
Basis Path #3 (Address Not In Singapore): 1, 2, 4, 5, 1, 2, 4, 6

Test case 6: Filter

Test case name	Filter
Description	User applies filter on map view
Preconditions	1. The user's device must have access to the internet. 2. The user must have location services available on their device, and give permission to the system to access them.

#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User filters by available lots	1. User drags slider to "10+"	NIL	Only carparks with more than 10 available lots are displayed.	Only carparks with more than 10 available lots are displayed.	Passed

2	User filters by gantry height	1. User drags slider to “1.6m+”	NIL	Only carparks with gantry height above 1.6m are displayed.	Only carparks with gantry height above 1.6m are displayed.	Passed
3	User filters by carpark type	1. User selects “SURFACE CARPARK” option	NIL	Only surface carparks are displayed.	Only surface carparks are displayed.	Passed

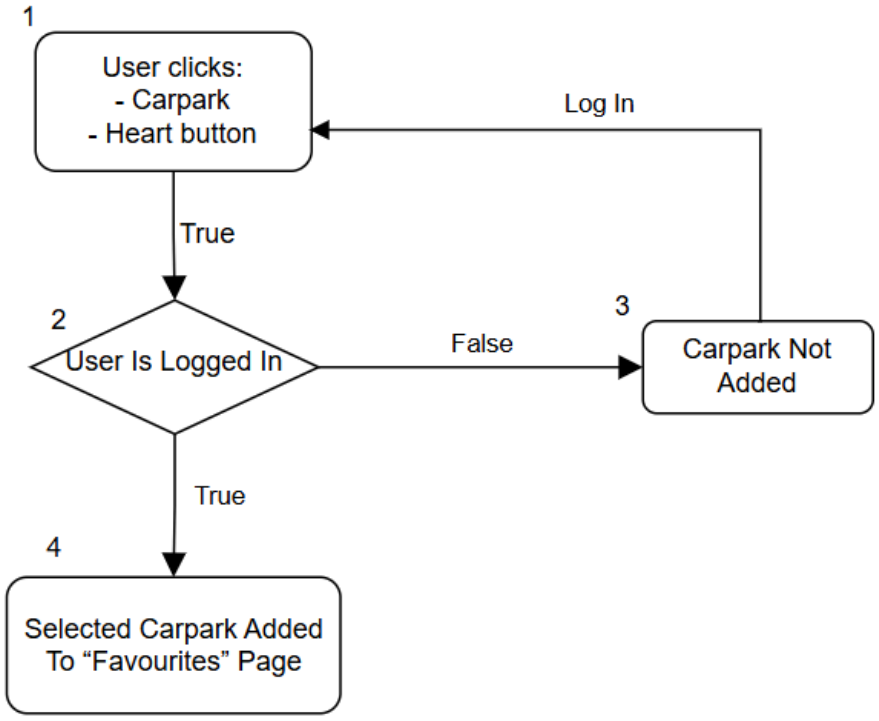


Basis Path #1 (Filter All): 1, 2, 4, 6, 8
Basis Path #2 (Filter Available Lots Only): 1, 2, 4, 5, 6, 7, 8
Basis Path #3 (Filter Gantry Height Only): 1, 2, 3, 4, 6, 7, 8
Basis Path #4 (Filter Carpark Type Only): 1, 2, 3, 4, 5, 6, 8

Test case 7: Add Favourite Carpark

Test case name	Add Favourite Carpark
Description	User adds a carpark to their favourites
Preconditions	1. The user's device must have access to the internet.

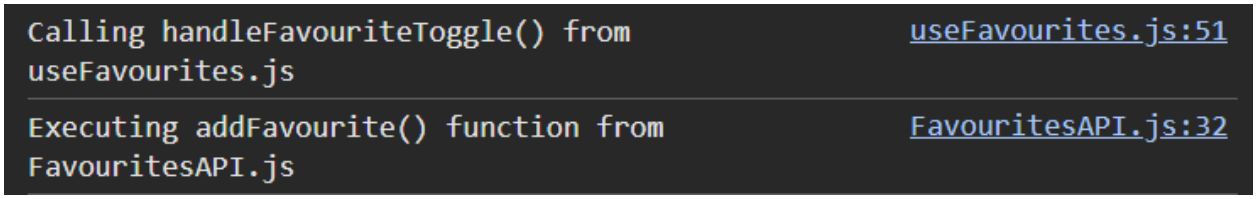
#	Description	Actions	Inputs	Expected Outputs	Actual Outputs	Test Result
1	User successfully adds a carpark to their favourites.	1. User clicks on a carpark 2. User clicks on the heart button.	NIL	Selected carpark added to "Favourites" page.	Selected carpark added to "Favourites" page.	Passed
2	User is not logged in	1. User clicks on a carpark 2. User clicks on the heart button.	NIL	Selected carpark is not added anywhere.	Selected carpark is not added anywhere.	Passed



Basis Path #1 (Carpark Successfully Added): 1, 2, 4

Basis Path #2 (User Not Logged In): 1, 2, 3, 1, 2, 4

Console log:
Successfully added



User is not logged in