

Carpark NF

Documentation of Carpark NF

School of Computer Science and Engineering

Course: SC2006 Software Engineering

Lab Group: SCS1

Team Pineapple:

Boonkitrungsaisarn Potala

Fan Jin Min

Lee Kian Han, Nicholas

Leong Wei Hong

Lim Haozheng

Thor Jia Ying

Document last revised on 6 Nov 2023 and is final.

Contents

Introduction	3
Purpose.....	3
Product scope	3
UI Mockups	4
Requirements Elicitation.....	8
Functional Requirements	8
Non-functional Requirements.....	10
Data Dictionary	11
Use Case Model	12
Use Case Diagram	12
Use Case Descriptions	13
Requirements Analysis.....	30
Class Diagram	30
Sequence Diagrams	31
Dialog Map.....	38
System Design	39
System Architecture	39
Software Engineering Practices	40
Test Cases & Testing Results	41
Black Box Testing.....	41
White Box Testing (Control Flow Testing).....	44
Video	47
Source Code	47

Introduction

Carpark NF is a web application, optimised for mobile, designed to enhance the convenience and efficiency of drivers' lives by offering real-time carpark information and navigation. It can be used on mobile or desktop web browsers.

Purpose

This document details the steps taken throughout the Software Development Lifecycle of Carpark NF. The software requirements of Carpark NF covered in this document is of the final version of the whole application.

Product scope

Carpark NF allows users to search for a location and in real time displays a list of nearby carparks within a 500-metre radius. What sets us apart is that we also provide information about carpark availability and parking rates, helping users plan their journeys. Users can also favourite their frequently visited carparks for easy access, and access navigation directly from our app.

With Carpark NF, users can plan their journeys ahead and save time. As our application allows users to view a carpark's availability in real time, he can plan his journey accordingly and look for a carpark with sufficient parking lots so that he will not be disappointed when he reaches his destination. This helps him save time looking for a carpark that has available parking lots as he can find out directly by using Carpark NF, instead of driving around looking for carparks.

UI Mockups

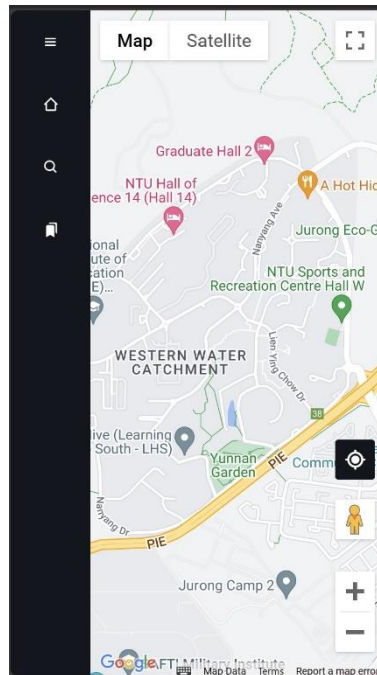


Figure 1: Home Page

Figure 1 shows the Home Page. This is the page the user sees when he opens the application on his browser. He can click on the sidebar to go to the Search Page, to search for a location, or Favourites Page, to view his list of favoured carpark.

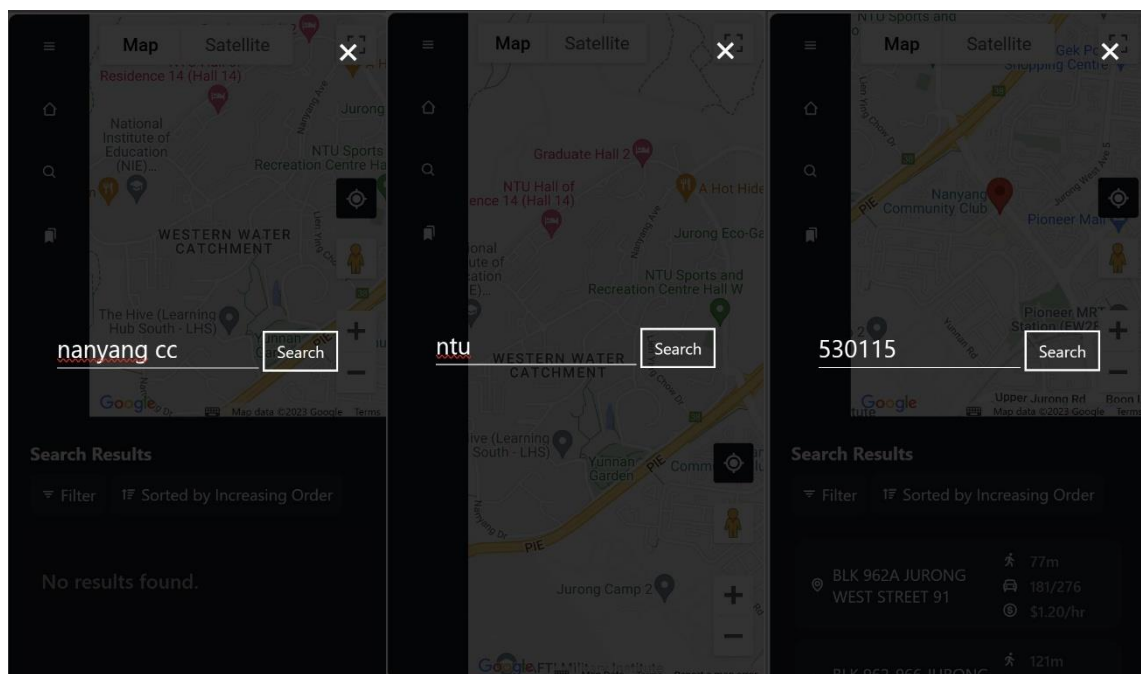


Figure 2: Search Page

Figure 2 shows the Search Page. The user is allowed to search for a location by postal code or street name.

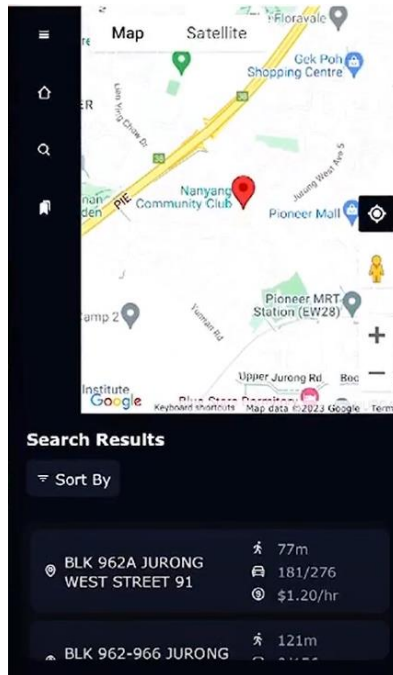


Figure 3: Results Page

Figure 3 shows the Results Page. After the user searches for a location, a list of carpark, and its details, within 500 metres of the searched location is shown to the user. The location which he searched for is also pinpointed on the map.

By default, the list of carpark is shown to the user sorted according to ascending order of the distance from the searched location. The user can choose to sort the carpark according to descending order of the carpark availability by click on the “Sort By” button.

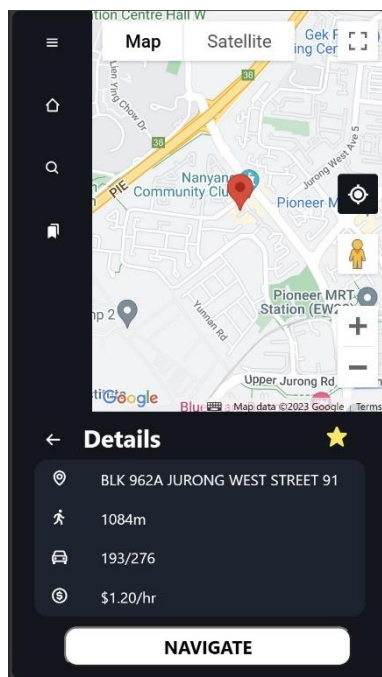


Figure 4: Details Page

Figure 4 shows the Details Page when the user selects a carpark either from the Favourites Page or search results. It shows the details of the carpark selected.

From here, he can choose to navigate to the carpark from his current location by clicking on the “Navigate” button. He can also add or remove the carpark to his favourites list by click on the star icon (solid star indicates that the carpark is in his favourites; hollow star indicates that the carpark is not in his favourites).

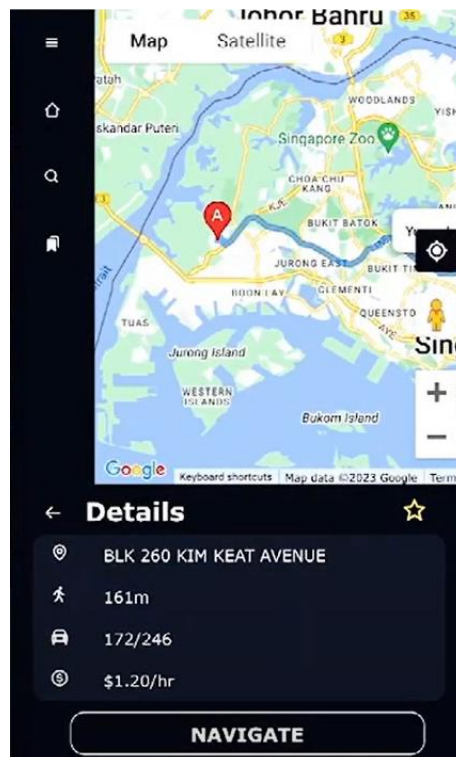


Figure 5: Navigation

When the user chooses to navigate to the selected carpark, a route will be displayed on the map, as shown in Figure 5. There is also an option for the user to open the navigation in Google Maps.

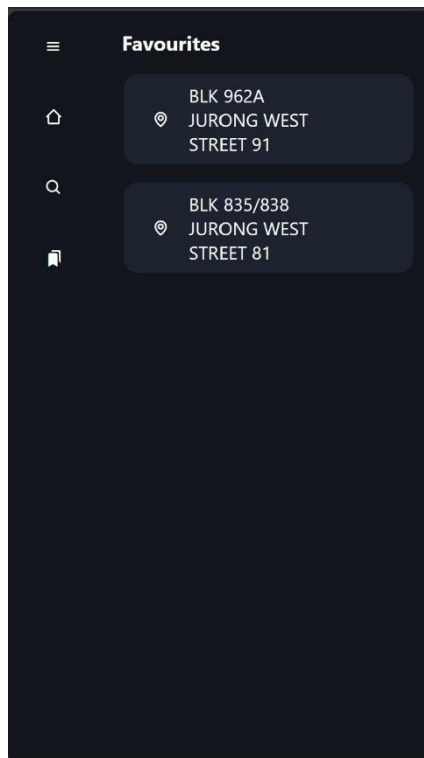


Figure 6: Favourites Page

Figure 6 shows the Favourites Page where a list of the user's favourited carpark is shown. He can click on any of the carparks to select a carpark, which will redirect him to view more details of the carpark on the Details Page.

Requirements Elicitation

Functional Requirements

The functional requirements defined for Carpark NF are as follows.

1. General
 - a. The system must be able to display a map.
2. Search
 - a. The system must allow user to search for a location.
 - i. The user must be able to search for a location by entering a 6-digit postal code or a street name.
 - ii. The system must be able to recommend a list of carpark within a radius of 500 metres of the searched location.
 1. Each carpark in the list of carpark displayed must contain information related to that carpark.
 - a. Information includes:
 - i. Carpark distance from searched location
 - ii. Carpark address
 - iii. Carpark availability
 - iv. Carpark rate
 - iii. The user must be able to sort the list of carpark by ascending order of distance from searched location or descending order of availability.
 1. By default, the list of carpark is sorted by ascending order of distance from searched location.
3. Favourite
 - a. The user must be able to favourite carpark.
 - b. The user must be able to view a list of their favourite carpark.
 - i. Each carpark in the list of favourite carpark must contain information related to that carpark.
 1. Information includes:
 - a. Carpark distance from user's location
 - b. Carpark address
 - c. Carpark availability
 - d. Carpark rate
 - c. The user must be able to remove a favourited carpark.
4. Select carpark to visit
 - a. The user must be able to select a carpark to visit from his/her list of favourite carpark or list of carpark from searching for a destination.
 - i. The system shall warn the user when the carpark being selected's availability is below 5% of its maximum capacity.

- ii. The system must be able to get the carpark rates of the selected carpark.
 - b. The user must be able to unselect a carpark that he/she has selected to visit.
5. Retrieve carpark availability information
- a. The system must be able to retrieve carpark availability information via Data.gov.sg's Carpark Availability API.
 - i. Information includes:
 - 1. Carpark number
 - 2. Carpark availability
6. Retrieve carpark information
- a. The system must be able to retrieve a carpark's information using the carpark number by using the list of HDB Carpark Information by Data.gov.sg.
 - i. Carpark's information includes:
 - 1. Carpark address
 - 2. Carpark's X Coord
 - 3. Carpark's Y Coord
7. Locations
- a. The system must be able to retrieve locations via Google Maps API.
 - i. Locations include:
 - 1. Location of carparks
 - 2. With respect to functional requirement 2(a), a searched location
 - 3. With respect to functional requirement 2(a)(ii), 500 metres radius of a searched location
 - 4. Location of user
 - b. With respect to functional requirement 7(a)(i)(4), the location of user shall be retrieved from the user's device GPS module.
8. Navigation
- a. The user must be able to search for a route to a selected carpark from his/her location.
 - b. The system must be able to display the route searched for by the user on the map.
9. Carpark rates
- a. The system must be able to retrieve carpark rates for user's selected carpark from the URA API.

Non-functional Requirements

The non-functional requirements defined for Carpark NF are as follows.

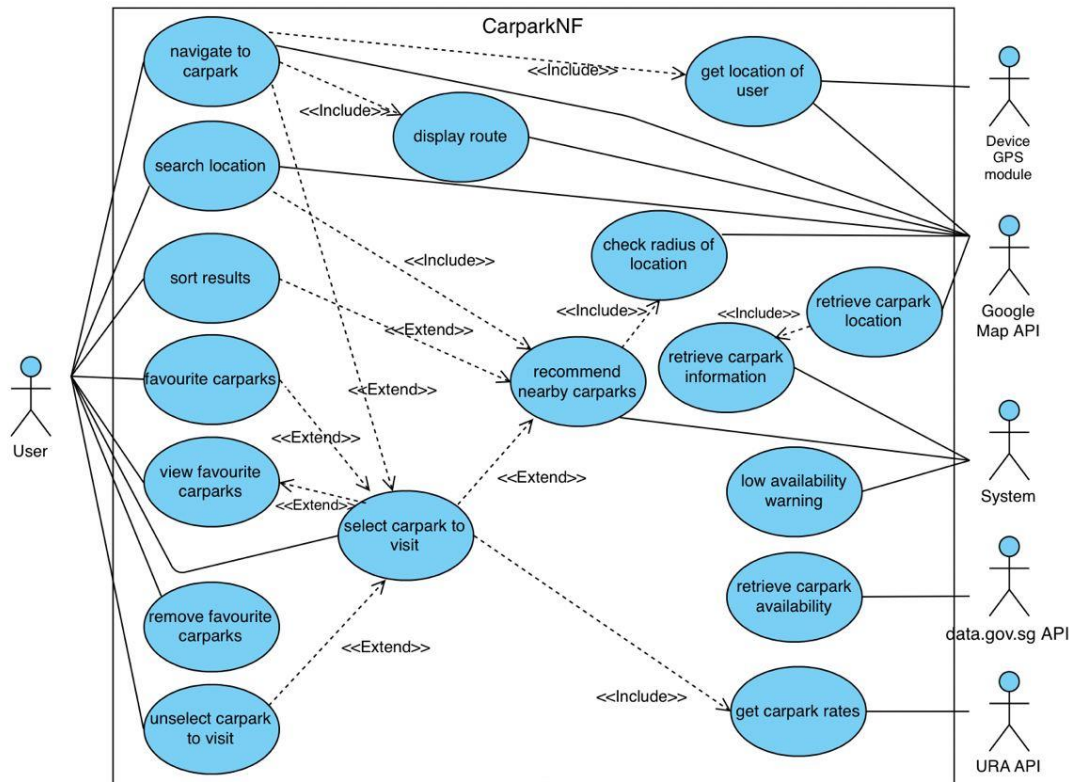
1. Usability requirements
 - a. The system and user must have internet connection.
 - b. With respect to functional requirement 3(a), the user must be allowed to favourite a carpark by clicking on a star icon.
 - i. To keep the user interface simple and easy to understand.
2. Reliability requirements
 - a. The system must update the carpark availability information every 1 minute.
 - b. After a system reboot, the full system functionality must be restored within 5 minutes.
3. Performance requirements
 - a. The system must not crash when the user opens the application.
 - b. The user must be able to use the application within 20 seconds of opening the application.
 - c. With respect to functional requirement 2(a), the system must be able to return the search results to the user within 10 seconds.
4. Supportability requirements
 - a. The user must be able to access the application from web browsers on their mobile device and desktop.

Data Dictionary

Term	Definition
System / application	The Car Park NF web application.
User	A person using the application to find carpark locations and carpark availability.
Map	An interactive map of Singapore.
Location	A point on the map of a particular place or building.
Search	A feature of the application that allows users to find carparks within a 500 metres radius of his/her searched location.
Favourite	A feature of the application that allows users to favourite, view favourited, or remove carparks for future ease of reference.
6-digit postal code	A 6-digit postal code representing an address of a location.
Street name	Name of a location.
Carpark number	A 5-digit carpark code.
Carpark availability	Number of remaining free lots in a carpark.
Carpark availability information	Information retrieved from Carpark Availability API which consists of, among other things, carpark number and carpark availability.
Carpark Availability API	API provided by Data.gov.sg that gives information on the latest carpark availability in Singapore.
Carpark('s) information	A carpark's information retrieved from HDB Carpark Information based on the carpark's carpark number, which consists of, among other things, carpark address, Carpark's X Coord and Y Coord.
HDB Carpark Information	Data provided by Data.gov.sg that gives information about HDB carparks.
Carpark's X Coord and Y Coord	Geo coordinate X and geo coordinate Y respectively of the carpark, indicating the carpark's location on the map.
Google Maps API	An API that allows developers to access Google Maps data and functionality
Route	A path from the location of user to a selected carpark.

Use Case Model

Use Case Diagram



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Search Location		
Created By:	Haozheng	Last Updated By:	Jin Min
Date Created:	30/08/2023	Date Last Updated:	13/09/2023

Actor:	User, Google Maps API
Description:	To search for car parks near the user's searched location
Preconditions:	1. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	1. Car parks that fall within the radius of the searched location will be found and saved.
Priority:	High
Frequency of Use:	0-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User navigate to the Search page from the sidebar 2. User will enter the 6-digit postal code or street name of his desired location 3. System will search for car parks that are near the location 4. Nearby car parks will be saved as a search result.
Alternative Flows:	-
Exceptions:	-
Includes:	Recommend nearby car park
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Favourite Carpark		
Created By:	Haozheng	Last Updated By:	Jia Ying
Date Created:	30/08/2023	Date Last Updated:	19/10/2023

Actor:	User
Description:	To add the carpark to user's favourite list of car parks
Preconditions:	<ol style="list-style-type: none"> 1. Device must be connected to Wi-Fi/Mobile Data 2. User has already searched a location
Postconditions:	<ol style="list-style-type: none"> 1. User will be able to save their favourite carpark 2. Carpark information will be added to 'View Favourite Carpark' list
Priority:	Medium
Frequency of Use:	1-10 times per lifetime
Flow of Events:	<ol style="list-style-type: none"> 1. User have selects a carpark 2. DetailsPage will get the carpark information and display page 3. User clicks on the hollow star icon on DetailsPage 4. FavouritesHandler will add carpark into UserInfo 5. UserInfo favourites list now contains the selected carpark
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	View Favourite Carpark		
Created By:	Haozheng	Last Updated By:	Jin Min
Date Created:	30/08/2023	Date Last Updated:	09/09/2023

Actor:	User
Description:	To view user's favourite car parks
Preconditions:	1. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	1. User will be able to see their favourite car park
Priority:	Medium
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on View Favourite Page from the Sidebar 2. FavouritePage gets the user's favourite car parks from the FavouritesHandler 3. FavouritesHandler retrieves the user's list of favourite car parks 4. FavouritesHandler returns the user's list of favourite car parks to FavouritePage 5. FavouritePage will display user's favourite car parks to the User
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	<p>User can have no favourite car park</p> <p>If user has no favourite car park, system will display empty page</p>

Use Case ID:	4		
Use Case Name:	Remove Favourite Carpark		
Created By:	Haozheng	Last Updated By:	Jin Min
Date Created:	30/08/2023	Date Last Updated:	11/10/2023

Actor:	User
Description:	To remove the user's favourite carpark
Preconditions:	<ol style="list-style-type: none"> 1. Device must be connected to Wi-Fi/Mobile Data 2. User must have at least one favourite carpark.
Postconditions:	<ol style="list-style-type: none"> 1. The favourite carpark selected by the user will be removed from the system
Priority:	Medium
Frequency of Use:	0-10 times per lifetime
Flow of Events:	<ol style="list-style-type: none"> 1. User select the carpark that he wish to remove from favourite list 2. User will be redirected to the details page of the selected carpark 3. User click on the solid star icon 4. DetailsPage invoke removeFavouriteCarpark() of the FavouritesHandler 5. FavouritesHandler removes the carpark from UserInfo's favourites list
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Retrieve Carpark Locations		
Created By:	Wei Hong	Last Updated By:	Jin Min
Date Created:	01/09/2023	Date Last Updated:	22/10/2023

Actor:	Google Maps API
Description:	To retrieve and display the locations of carpark in Singapore on a digital map.
Preconditions:	<ol style="list-style-type: none"> 1. The list of carpark in Singapore must be obtained 2. The Google Maps API must be reachable.
Postconditions:	<ol style="list-style-type: none"> 1. The locations of the carpark in Singapore are displayed on the digital map
Priority:	High
Frequency of Use:	1 time per backend startup
Flow of Events:	<ol style="list-style-type: none"> 1. The locations of the carpark are queried from the Google Maps API using the coordinates. 2. The Google Maps API marks the locations 3. The markers are displayed on the map
Alternative Flows:	-
Exceptions:	EX.5: If the Google Maps API does not respond to the query <ol style="list-style-type: none"> 1. A "Google Maps API is unreachable" error is shown on the screen.
Includes:	Retrieve Carpark Information
Special Requirements:	-
Assumptions:	The Google Maps API is in an operational state.
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	Retrieve Carpark Information		
Created By:	Wei Hong	Last Updated By:	Jia Ying
Date Created:	01/09/2023	Date Last Updated:	19/10/2023

Actor:	System
Description:	To get the coordinates of all the carpark in Singapore, along with their ID for cross-referencing with its availability.
Preconditions:	1. The dataset containing Singapore's carpark coordinates and IDs must be available.
Postconditions:	1. The system obtains information on Singapore's carpark coordinates and IDs.
Priority:	High
Frequency of Use:	1 time per backend startup
Flow of Events:	<ol style="list-style-type: none"> 1. The system retrieves the carpark dataset from data.gov.sg API 2. The system unpacks the dataset. 3. System retrieves list of carpark alongside with their IDs from Carpark List
Alternative Flows:	-
Exceptions:	EX.6: The data.gov.sg dataset cannot be retrieved. <ol style="list-style-type: none"> 1. A "The carpark dataset could not be retrieved." error is shown.
Includes:	-
Special Requirements:	-
Assumptions:	The dataset from data.gov.sg is available.
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Unselect Carpark to Visit		
Created By:	Wei Hong	Last Updated By:	Jin Min
Date Created:	01/09/2023	Date Last Updated:	11/10/2023

Actor:	User
Description:	To unselect a previously selected carpark
Preconditions:	<ol style="list-style-type: none"> 1. The user must have previously selected a carpark to visit. 2. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	<ol style="list-style-type: none"> 1. The carpark previously selected by the user to park at will be unselected in the system.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User presses unselect carpark button 2. SelectionHandler's unselectCarpark() is invoked 3. The green tick indicating carpark is selected on the ResultsPage is removed
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Check radius		
Created By:	Wei Hong	Last Updated By:	Jia Ying
Date Created:	01/09/2023	Date Last Updated:	19/10/2023

Actor:	Google Maps API
Description:	To search for carparks within 500 metres of user's searched location
Preconditions:	1. A location has been entered by the user
Postconditions:	1. A list of carparks within 500 metres is saved in Carpark List
Priority:	High
Frequency of Use:	0-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. The user chooses a location from the Search Page. 2. Search Page sends the location to Google Maps API 3. Google Maps API query the Data.gov.sg API for the list of carparks within 500 metre radius 4. Data.gov.sg API adds the list of carparks to Carpark List
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	The Google Maps API is in an operational state.
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Recommend Nearby Carparks		
Created By:	Jia Ying	Last Updated By:	Wei Hong
Date Created:	30/08/2023	Date Last Updated:	13/09/2023

Actor:	System
Description:	To sort the nearby carparks from the location by distance in ascending order and display the list of sorted carparks.
Preconditions:	<ol style="list-style-type: none"> 1. User must key in a search location 2. Google Maps API must be operational 3. User has already searched for a location 4. Nearby carparks of the location has been found
Postconditions:	<ol style="list-style-type: none"> 1. System will display a list of carparks within 500 metre radius of search location 2. System will display the carpark availability for the corresponding carparks
Priority:	High
Frequency of Use:	0-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. SearchHandler has nearby carparks found from searching location 2. SearchHandler loop through each carpark 3. For each carpark, use GoogleMapAPI to compute distance between carpark and location 4. Sort the carparks by distance in ascending order. 5. Display the sorted list of carparks on SearchPage.
Alternative Flows:	-
Exceptions:	-
Includes:	Check radius of location
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	Select Carpark To Visit		
Created By:	Jia Ying	Last Updated By:	Jia Ying
Date Created:	30/08/2023	Date Last Updated:	19/10/2023

Actor:	User
Description:	To select which carpark they would like to visit, and view the information regarding that carpark.
Preconditions:	1. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	1. The information of selected carpark will be displayed on the web
Priority:	High
Frequency of Use:	0-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User will search for a location 2. Search Handler will return a list of nearby carpark 3. User selects the carpark and the Selection Handler will return the information of selected carpark 4. Search Page will display the information
Alternative Flows:	AF-S1: User enters selects from list of favourited carpark <ol style="list-style-type: none"> 1. User enters Favourite Page 2. FavouritesHandler returns a list of user's favourite carpark 3. User selects the carpark and the Selection Handler will return the information of selected carpark 4. Search Page will display the information
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	Low availability warning		
Created By:	Jia Ying	Last Updated By:	Jia Ying
Date Created:	30/08/2023	Date Last Updated:	01/09/2023

Actor:	System
Description:	To display a low carpark availability warning to user
Preconditions:	<ol style="list-style-type: none"> 1. User has selected a carpark 2. Carpark availability for the chosen carpark is less than 5% of its maximum capacity
Postconditions:	<ol style="list-style-type: none"> 1. System will display a notification to users, informing them that the carpark availability for chosen carpark is running low
Priority:	Low
Frequency of Use:	0-5 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a carpark 2. If the selected carpark availability is less than 5% of its maximum capacity, system will display a low availability notification
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	Retrieve Carpark Availability		
Created By:	Jia Ying	Last Updated By:	Jin Min
Date Created:	30/08/2023	Date Last Updated:	11/10/2023

Actor:	Data.gov.sg API
Description:	To retrieve the number of available parking lots across all carpark in Singapore
Preconditions:	1. The Carpark Availability API must be reachable
Postconditions:	1. The carpark availability for all carpark will be updated in the list of recommended carpark
Priority:	High
Frequency of Use:	1 time per minute
Flow of Events:	<ol style="list-style-type: none"> 1. The data is queried from data.gov.sg API 2. Data.gov.sg API returns the carpark availability data 3. The corresponding carpark availability along with the carpark number will be updated in CarparkList
Alternative Flows:	-
Exceptions:	EX.1: The Carpark Availability API is inaccessible <ol style="list-style-type: none"> 1. System will display the last updated carpark availability 2. System will show time for the last updated information
Includes:	-
Special Requirements:	-
Assumptions:	The Carpark Availability API is in an operational state.
Notes and Issues:	-

Use Case ID:	13		
Use Case Name:	Get Location of User		
Created By:	Wei Hong	Last Updated By:	Jin Min
Date Created:	05/09/2023	Date Last Updated:	11/10/2023

Actor:	Google Maps API, Device GPS Module
Description:	To get the current location of the user through the user device's GPS module and pinpoint it on the map using the Google Maps API.
Preconditions:	<ol style="list-style-type: none"> 1. User must have granted the application permission to precisely access the device GPS module. 2. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	<ol style="list-style-type: none"> 1. The location of the user is determined and shown on the map.
Priority:	High
Frequency of Use:	1-30 times per minute
Flow of Events:	<ol style="list-style-type: none"> 1. Google Maps API gets location of the user from GPS Module 2. GPS Module returns user's coordinates to the Google Maps API 3. Google Maps API searches the location based on user's coordinates 4. The location is marked on the map.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	The device GPS module is in operational condition and able to retrieve the exact location of the device.
Notes and Issues:	User may only grant approximate location permission instead of precise location which can lead to an inaccurate pinpoint of the user's location.

Use Case ID:	14		
Use Case Name:	Navigate to Carpark		
Created By:	Wei Hong	Last Updated By:	Haozheng
Date Created:	05/09/2023	Date Last Updated:	12/09/2023

Actor:	Google Maps API, User
Description:	To search for a route to the carpark and display the resulting route using on the map.
Preconditions:	<ol style="list-style-type: none"> 1. A destination carpark has been selected by the user. 2. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	<ol style="list-style-type: none"> 1. A route to the destination carpark is computed by the Google Maps API.
Priority:	High
Frequency of Use:	1-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a destination carpark on the NavigatePage Interface 2. Google Maps API retrieves the location of the user from Use Case ID 13: "Get Location of User" 3. Google Maps API calculates the route to the destination carpark from the current location of user. 4. Use Case ID 15: "Display Route" displays the route to the destination on the map.
Alternative Flows:	-
Exceptions:	-
Includes:	Get Location of User, Display Route
Special Requirements:	-
Assumptions:	The device GPS module is in operational condition and able to retrieve the exact location of the device.
Notes and Issues:	User may only grant approximate location permission instead of precise location which can lead to an inaccurate pinpoint of the user's location.

Use Case ID:	15		
Use Case Name:	Display Route		
Created By:	Wei Hong	Last Updated By:	Jia Ying
Date Created:	05/09/2023	Date Last Updated:	19/10/2023

Actor:	Google Maps API
Description:	To display the navigation route to a chosen destination.
Preconditions:	1. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	1. A route with directions to the destination carpark is shown visually to the user on the map.
Priority:	High
Frequency of Use:	1-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. Google Maps API retrieves the route from RouteHandler 2. RouteHandler will compute the best route and return it to Google Maps API 3. Google Maps API will display the route on Maps.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	16		
Use Case Name:	Sort Results		
Created By:	Haozheng	Last Updated By:	Jia Ying
Date Created:	05/09/2023	Date Last Updated:	19/10/2023

Actor:	User
Description:	To sort the result of carparks recommended by either distance or availability
Preconditions:	1. User has keyed in a search location
Postconditions:	1. System will search display the carpark results based on user's selection
Priority:	High
Frequency of Use:	1-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User selects to sort the results by distance or availability via the dropdown on ResultsPage 2. SortHandler will apply the filter 3. SortHandler will sort the carparks according to the filter applied
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	The default selection will be sort by distance.

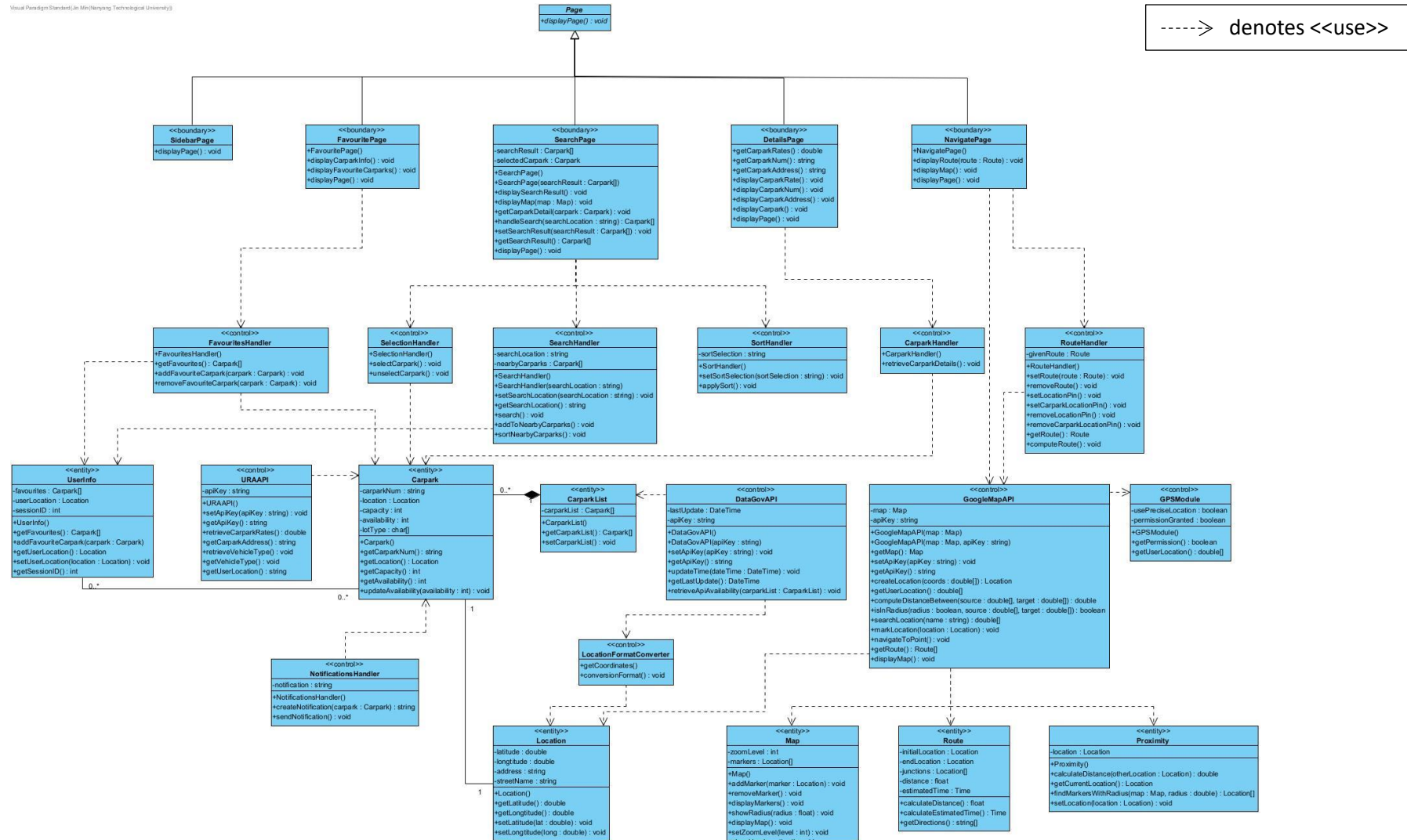
Use Case ID:	17		
Use Case Name:	Get carpark rates		
Created By:	Haozheng	Last Updated By:	Jia Ying
Date Created:	05/09/2023	Date Last Updated:	19/10/2023

Actor:	URA API
Description:	To display the carpark rates for the carpark that was selected by the user
Preconditions:	<ol style="list-style-type: none"> 1. Device must be connected to Wi-Fi/Mobile Data 2. The URA API must be reachable. 3. User must have selected a carpark
Postconditions:	<ol style="list-style-type: none"> 1. System will display the parking rates for the carpark selected by the user.
Priority:	High
Frequency of Use:	1-20 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. URA API retrieves the carpark rates for selected carpark 2. System will display the carpark rate onto the Details Page
Alternative Flows:	-
Exceptions:	Ex 17: The URA'S API is not operational. <ol style="list-style-type: none"> 1. A "URA's API could not be reached." error is shown.
Includes:	-
Special Requirements:	-
Assumptions:	The URA's API is in an operational state.
Notes and Issues:	-

Requirements Analysis

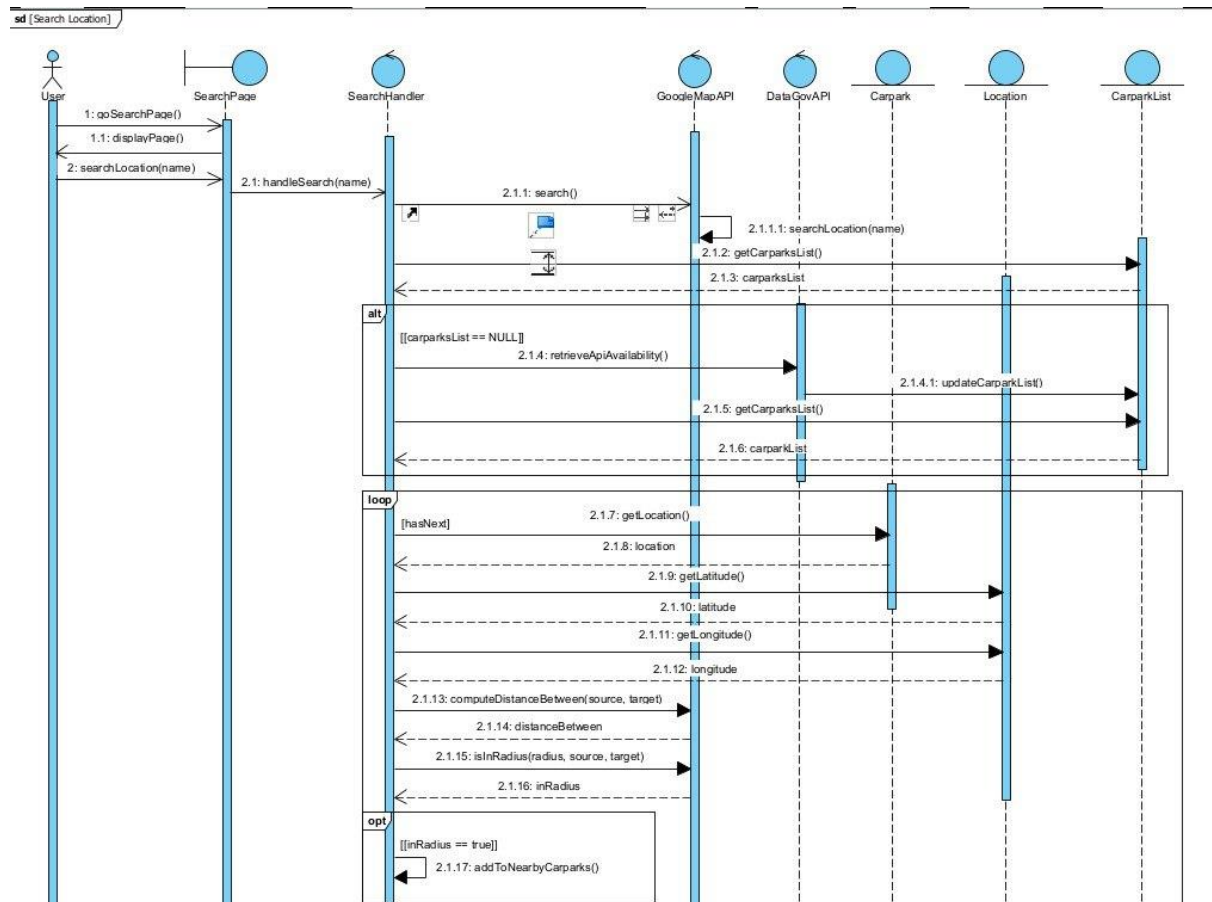
Class Diagram

Visual Paradigm Standard (in Melbourne Technology University)

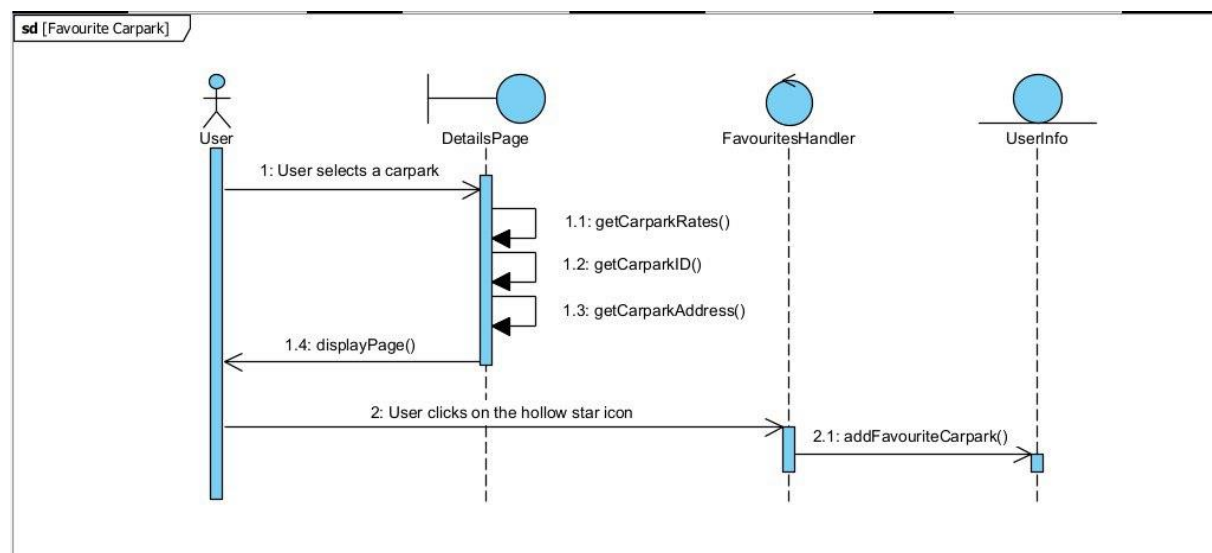


Sequence Diagrams

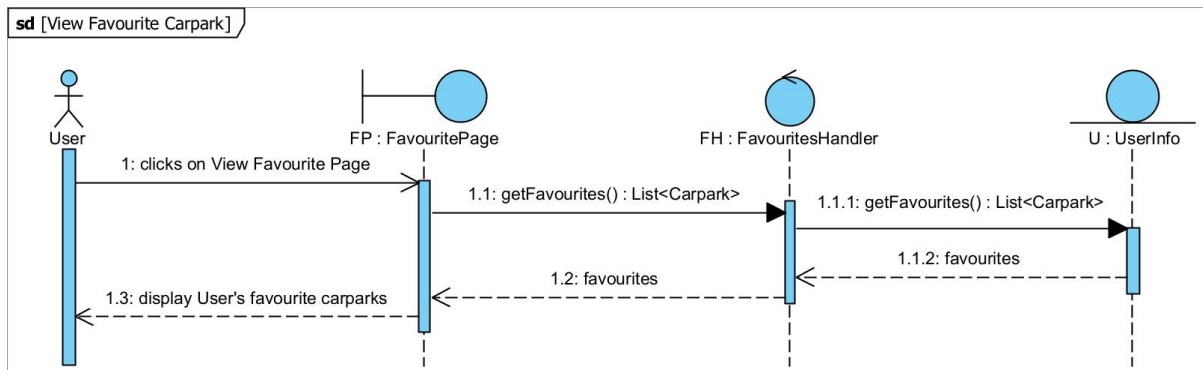
1. Search location



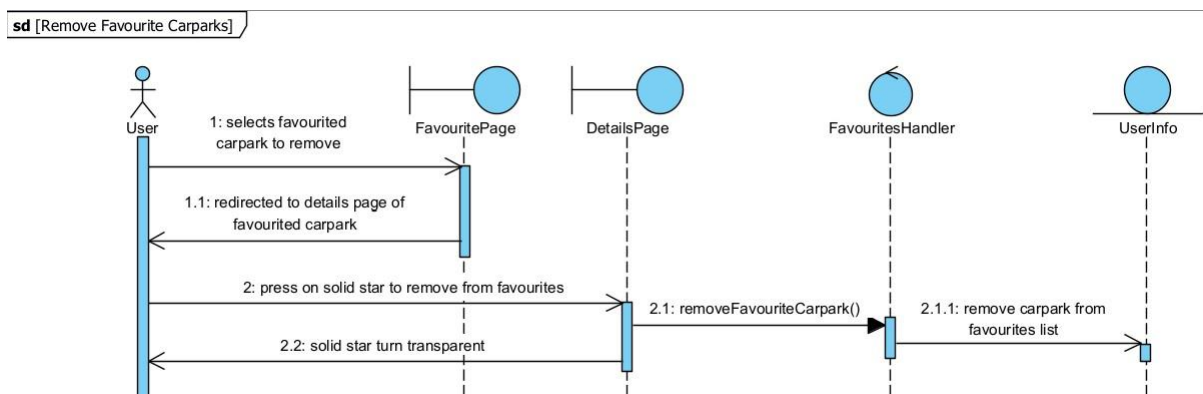
2. Favourite Carpark



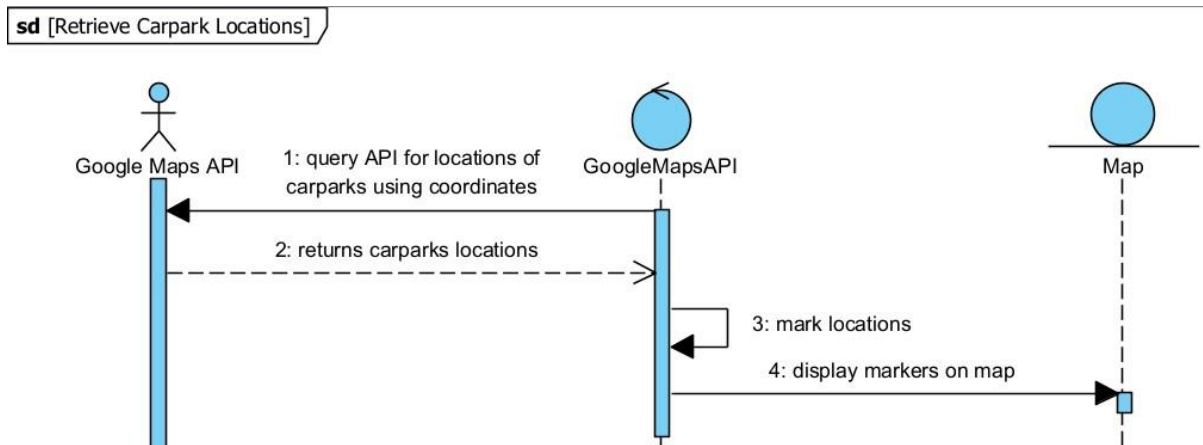
3. View Favourite Carpark



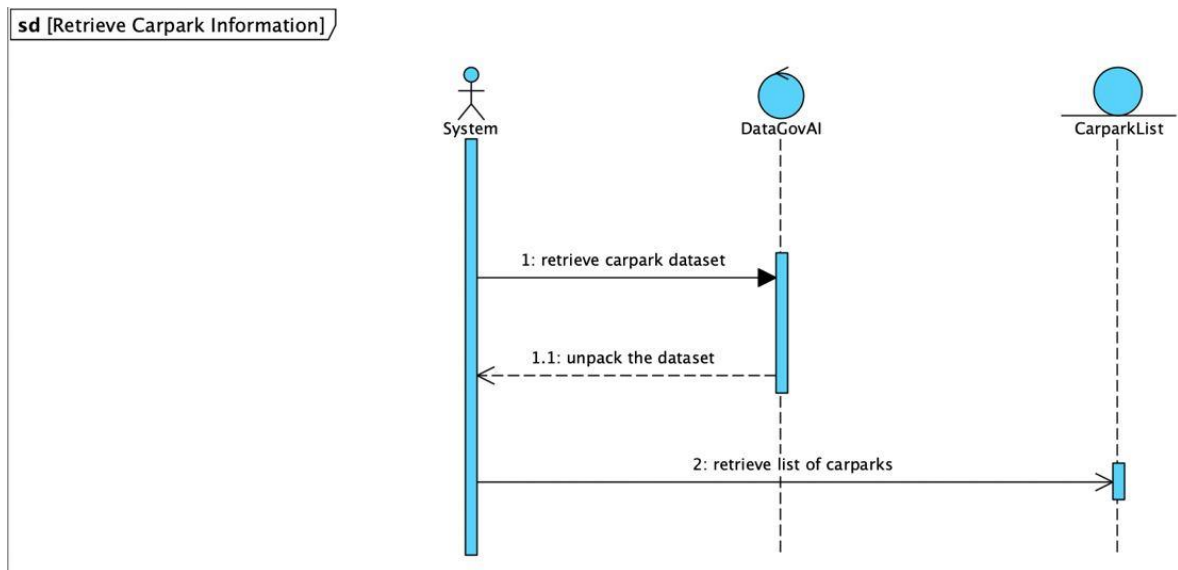
4. Remove Favourite Carpark



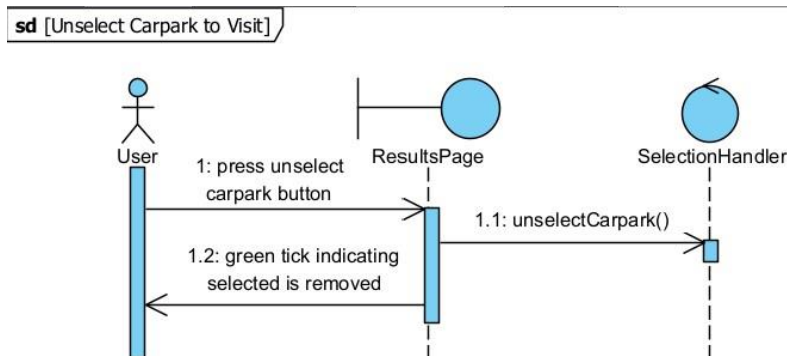
5. Retrieve Carpark Locations



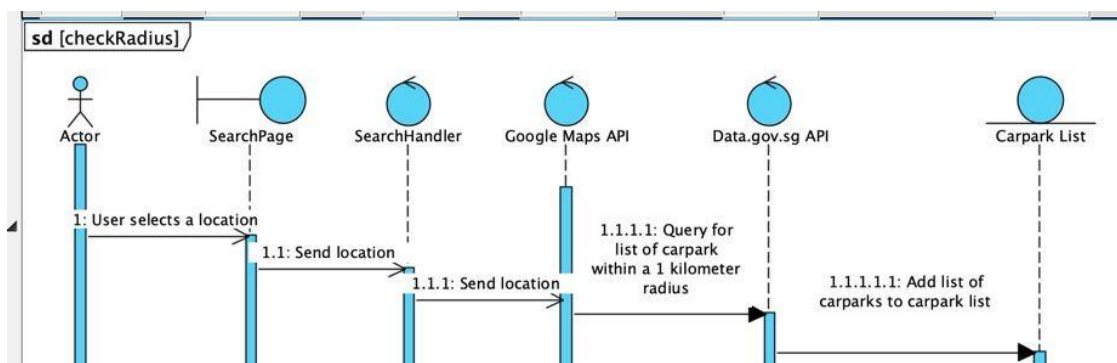
6. Retrieve Carpark Information



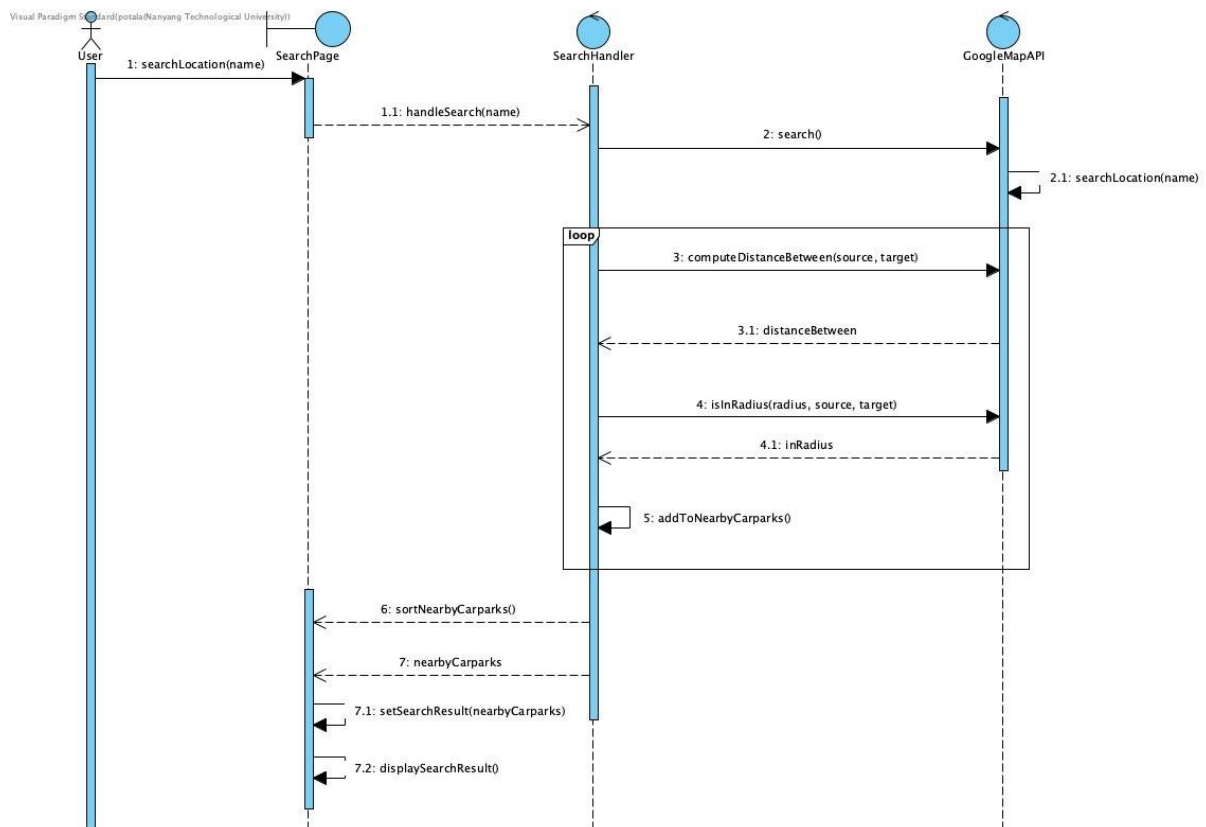
7. Unselect Carpark to Visit



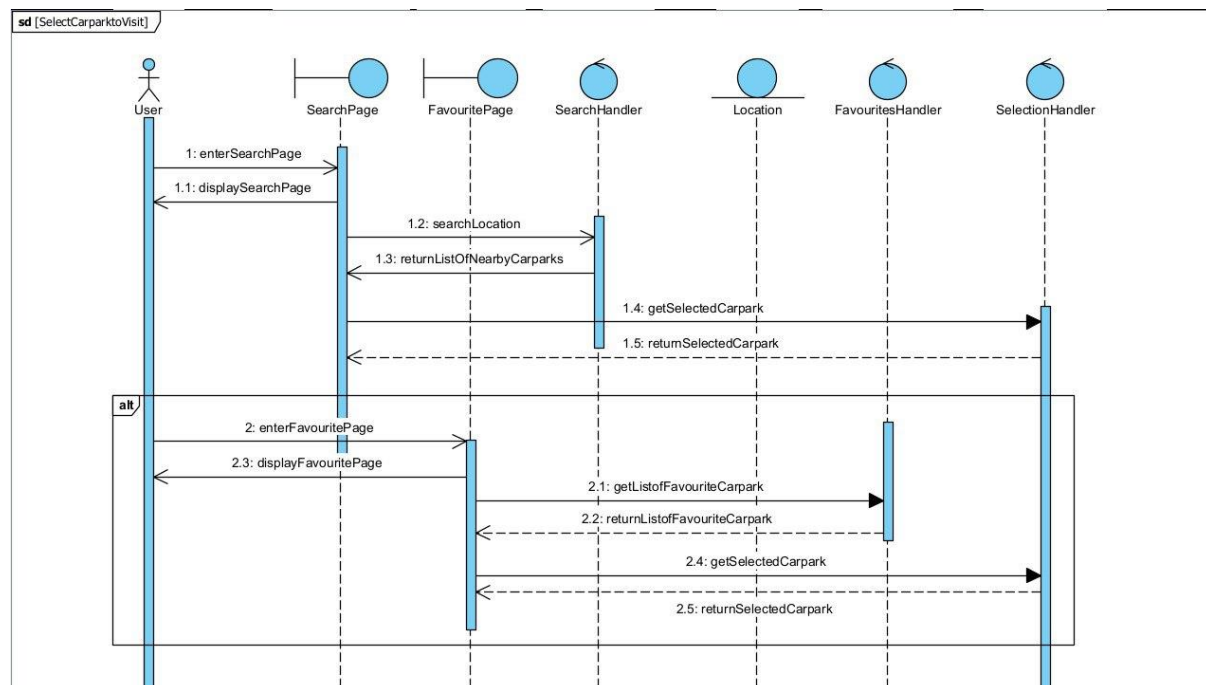
8. Check Radius



9. Recommend Nearby Carpark

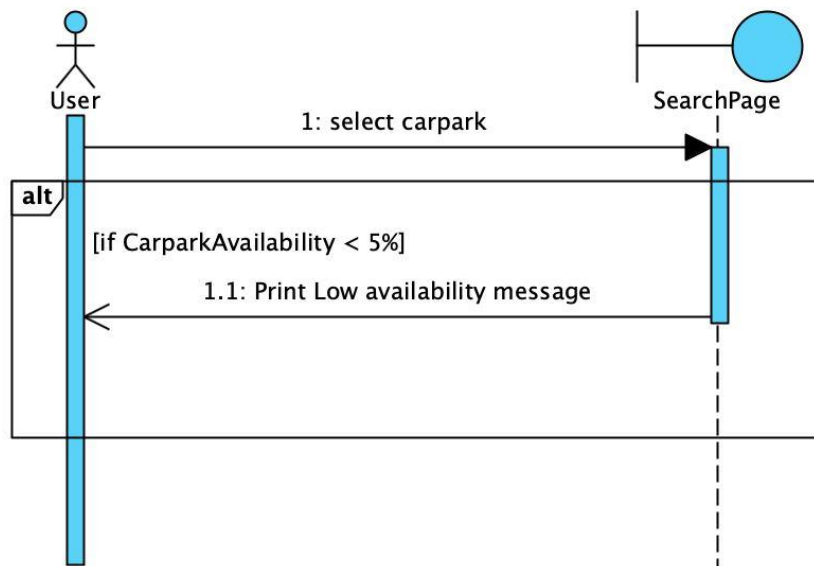


10. Select Carpark to Visit



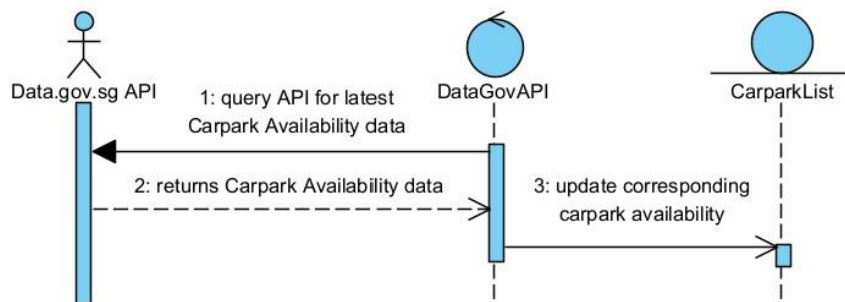
11. Low Availability Warning

sd [Low availability warning]



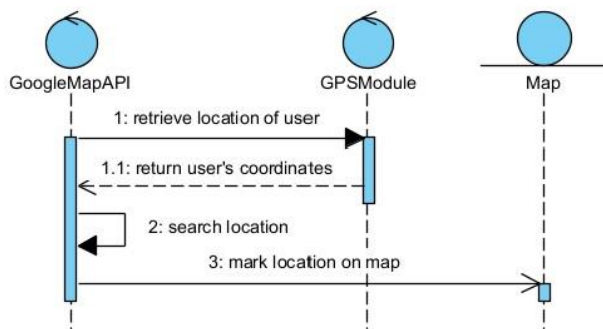
12. Retrieve Carpark Availability

sd [Retrieve Carpark Availability]

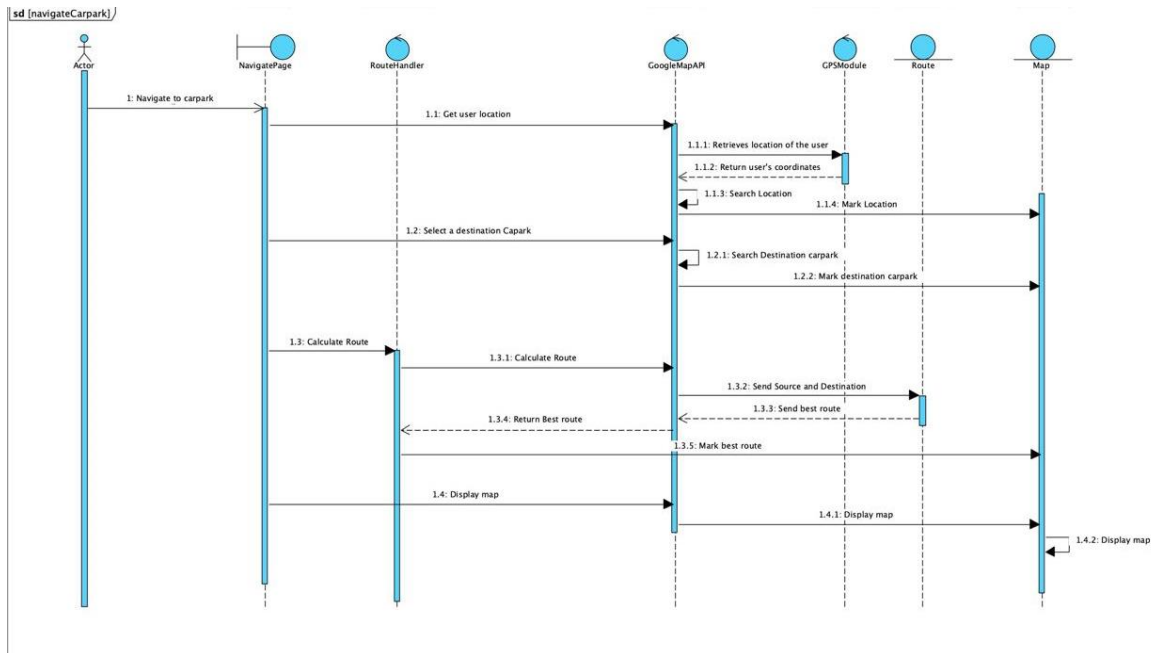


13. Get Location of User

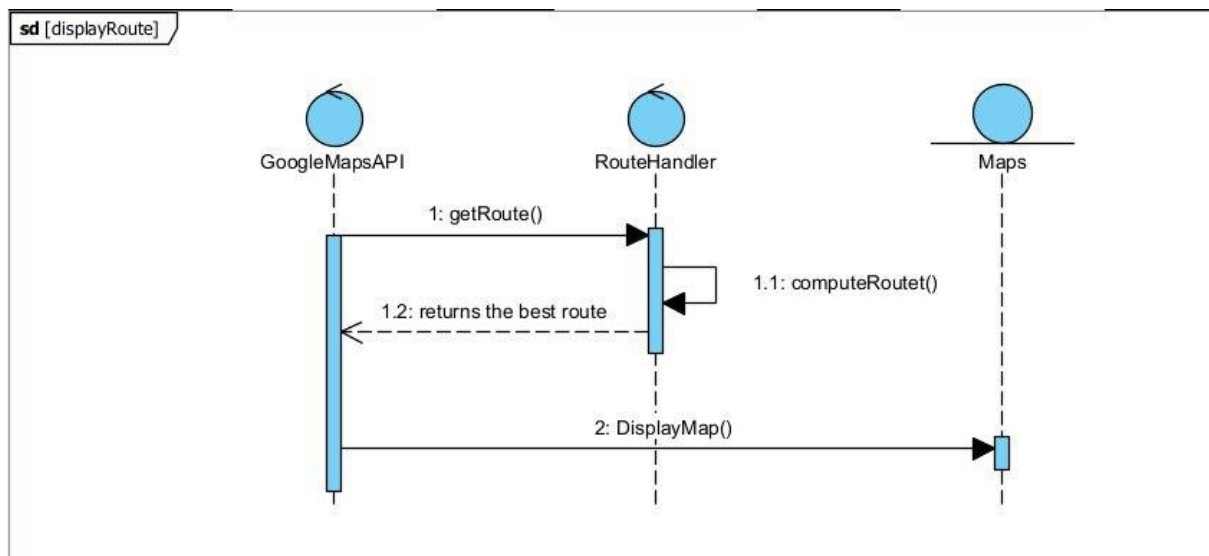
sd [Get Location of User]



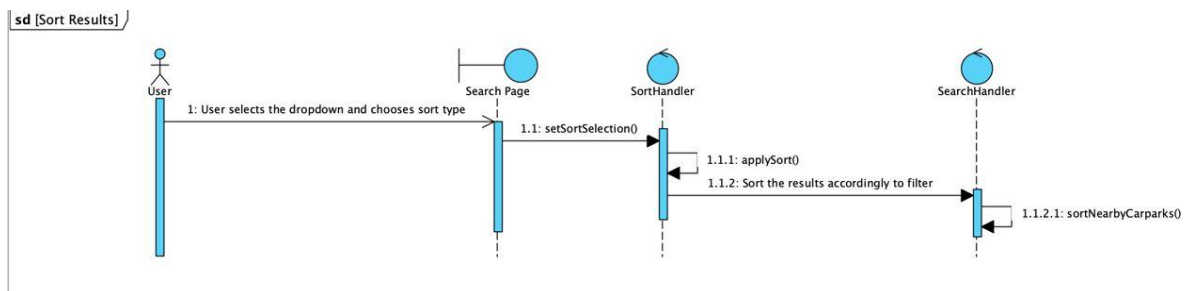
14. Navigate to Carpark



15. Display Route

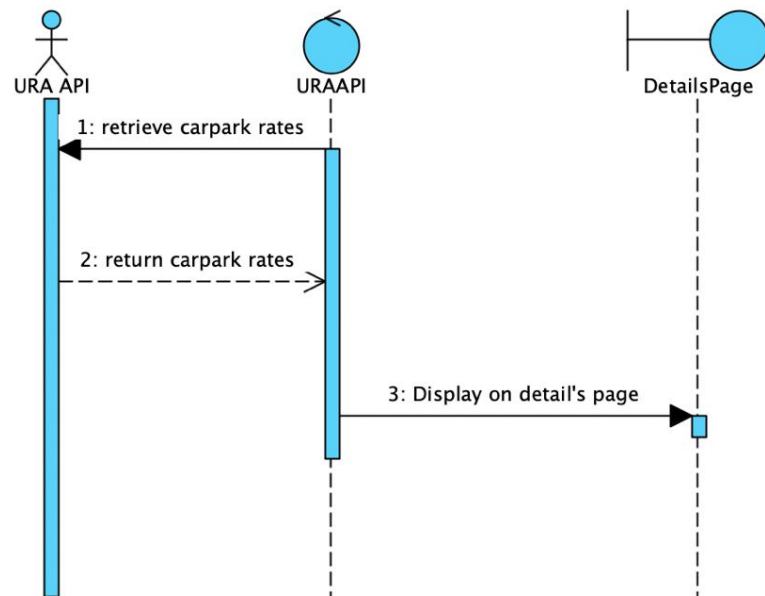


16. Sort Results

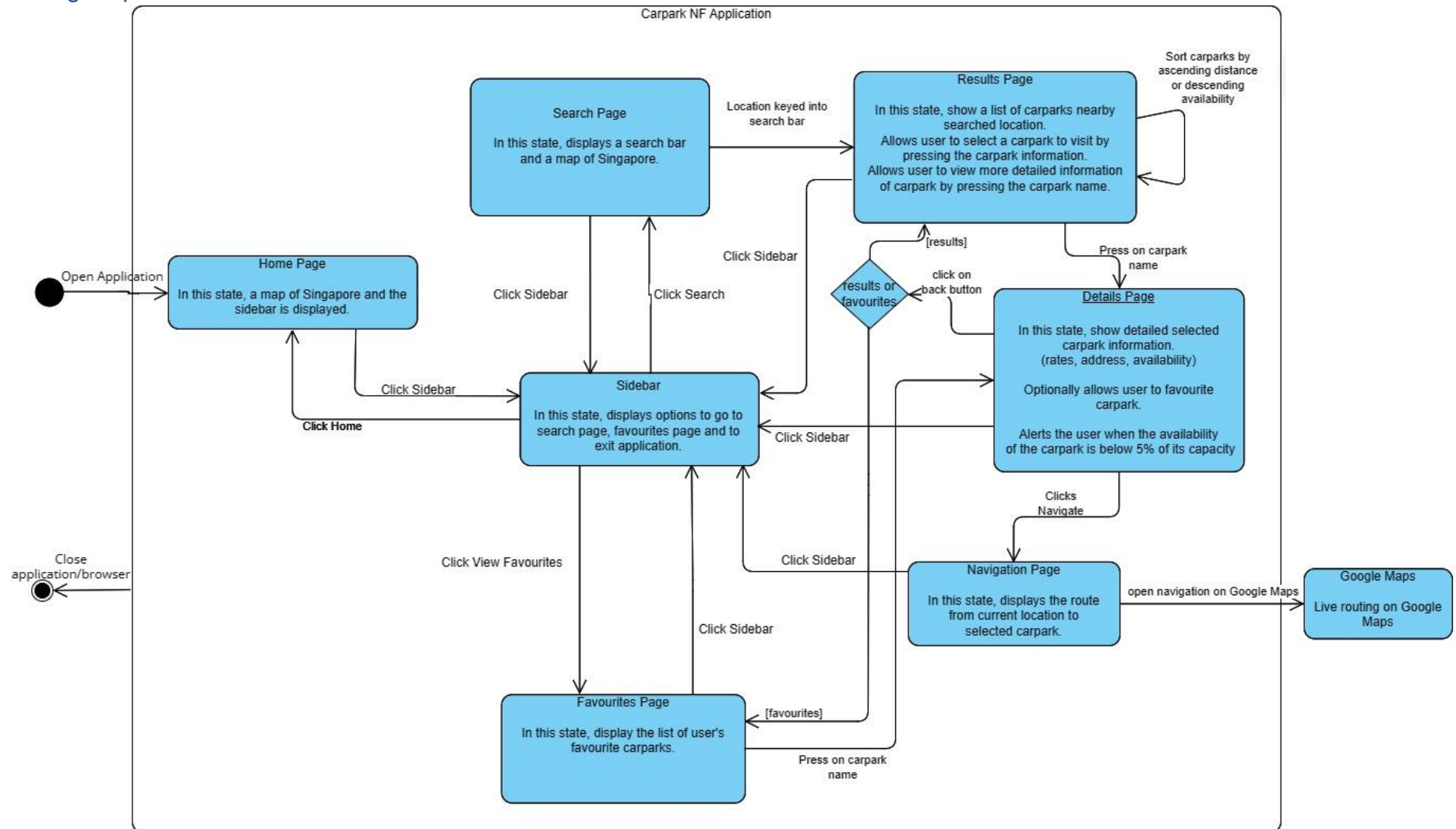


17. Get Carpark Rates

sd [Get Carpark Rates]

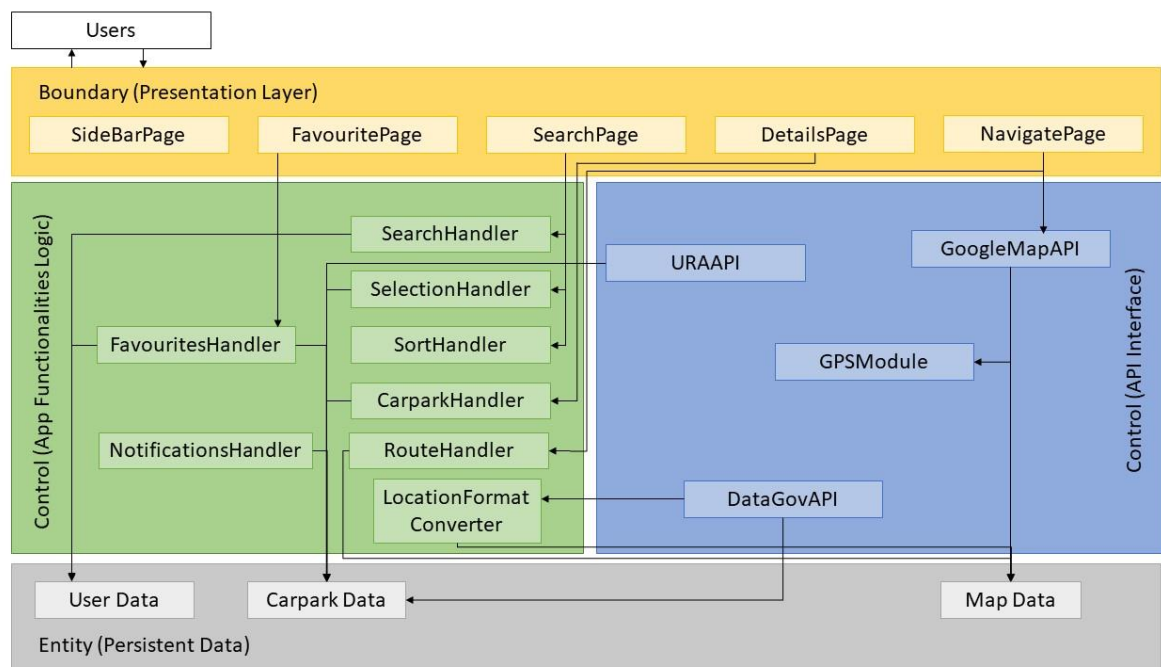


Dialog Map



System Design

System Architecture



In accordance with our system requirements and architectural considerations, we have implemented a layered architecture system design, partitioning it into three distinct layers.

Boundary Layer: Consists of all our interfaces and serves as the presentation layer to the users.

Control Layer: Consists of all our handler classes and API implementations, facilitating the management and execution of system functionality.

Entity Layer: Encapsulate all relevant data within entity classes, ensuring a well-structured and organized representation of our core data components.

This layered approach not only enhances the modularity and extensibility of our system but also greatly contributes to the overall ease of maintenance, allowing for the seamless addition of new classes tailored to specific functionalities.

Software Engineering Practices

Single Responsibility Principle (SRP)

We adhered to the SRP, ensuring that each class in our system embodies a singular, well-defined responsibility.

Dependency Inversion Principle (DIP)

We adhered to the DIP, encouraging a flexible and adaptable design by making sure that the different parts of our system are less tightly connected, achieving reduced coupling and flexibility.

Facade Pattern

We leveraged the Facade pattern to create a simplified interface to the complexity of our subsystems. This approach minimizes interdependencies between classes, ensuring that changes in one part of the system do not ripple through unrelated components.

Observer Pattern

We leveraged on the Observer pattern to establish a clean and decoupled mechanism for notifying classes of changes, enhancing modularity and reducing dependencies. This not only facilitates maintainability but also promotes the isolation of concerns within our codebase.

Overall, our design patterns have had a positive impact on the maintainability as well as extensibility of our system.

Improved Maintainability

Our code is easier to understand and modify due to the clear responsibilities of each class and the minimised interdependencies. This makes debugging and maintenance tasks more straightforward and less error-prone, greatly enhancing our efficiency.

Enhanced Extensibility

We reduced the dependencies between our classes, and only linked them when necessary. Hence when a class needs to be changed, only a few will be affected, Enhancing our reusability as well as extensibility.

Test Cases & Testing Results

Black Box Testing

1. Search for Carparks
 - a. Generic cases

Test ID	Scenario	Expected Result	Actual Result
1	Search by 6-digit postal code (searched location)	List of carparks within 500 metres radius of searched location is displayed to the user	List of carparks within 500 metres radius of searched location is displayed to the user
2	Search by street name (searched location)	List of carparks within 500 metres radius of searched location is displayed to the user	List of carparks within 500 metres radius of searched location is displayed to the user
3	Search empty string	Popup stating "No results found for search."	Popup stating "No results found for search."

- b. Specific cases

Search string	Expected Result	Actual Result
"530115" (Kovan)	List of carparks within 500 metres radius of an HDB block in Kovan is displayed to the user	List of carparks within 500 metres radius of an HDB block in Kovan is displayed to the user
"Nanyang CC"	List of carparks within 500 metres radius of Nanyang CC is displayed to the user	List of carparks within 500 metres radius of Nanyang CC is displayed to the user
"NTU"	"No results found" message displayed on results page. (No carparks nearby)	"No results found" message displayed on results page. (No carparks nearby)
"abcdefg"	Popup stating "No results found for search."	Popup stating "No results found for search."
"" (empty string)	Popup stating "No results found for search."	Popup stating "No results found for search."

2. Favourites

Test ID	Scenario	Expected Result	Actual Result
1	Favourite a carpark	The system saves the carpark into the favourite list	The system saves the carpark into the favourite list
2	Favouriting a carpark that is already favourited	The system removes the carpark from the favourite list	The system removes the carpark from the favourite list

3. Select Carpark to Visit

a. Generic cases

Test ID	Scenario	Expected Result	Actual Result
1	Select carpark from search results	Detailed information including distance from the searched location, capacity of the carpark and carpark rate is displayed to the user.	Detailed information including distance from the searched location, capacity of the carpark and carpark rate is displayed to the user.
2	Select carpark from favourites page	Detailed information including distance from the user, capacity of the carpark and carpark rate is displayed to the user.	Detailed information including distance from the user, capacity of the carpark and carpark rate is displayed to the user.
3	Select a carpark with carpark availability below 5% of its capacity	Alert warning stating that the carpark availability is low.	Alert warning stating that the carpark availability is low.

b. Specific cases

Selected Carpark	Expected Result	Actual Result
BLK 109-127 Hougang Avenue 1 (from search results)	Detailed information including distance from the searched location,	Detailed information including distance from the searched location,

	capacity of the carpark and carpark rate is displayed to the user.	capacity of the carpark and carpark rate is displayed to the user.
BLK 962A Jurong West Street 91 (from favourites page)	Detailed information including distance from the user, capacity of the carpark and carpark rate is displayed to the user.	Detailed information including distance from the user, capacity of the carpark and carpark rate is displayed to the user.
Blk 19 to 23 Lorong 7 Toa Payoh (when availability is 0/114)	Alert warning stating that the carpark availability is low.	Alert warning stating that the carpark availability is low.

4. Unselect Carpark to Visit
a. Generic cases

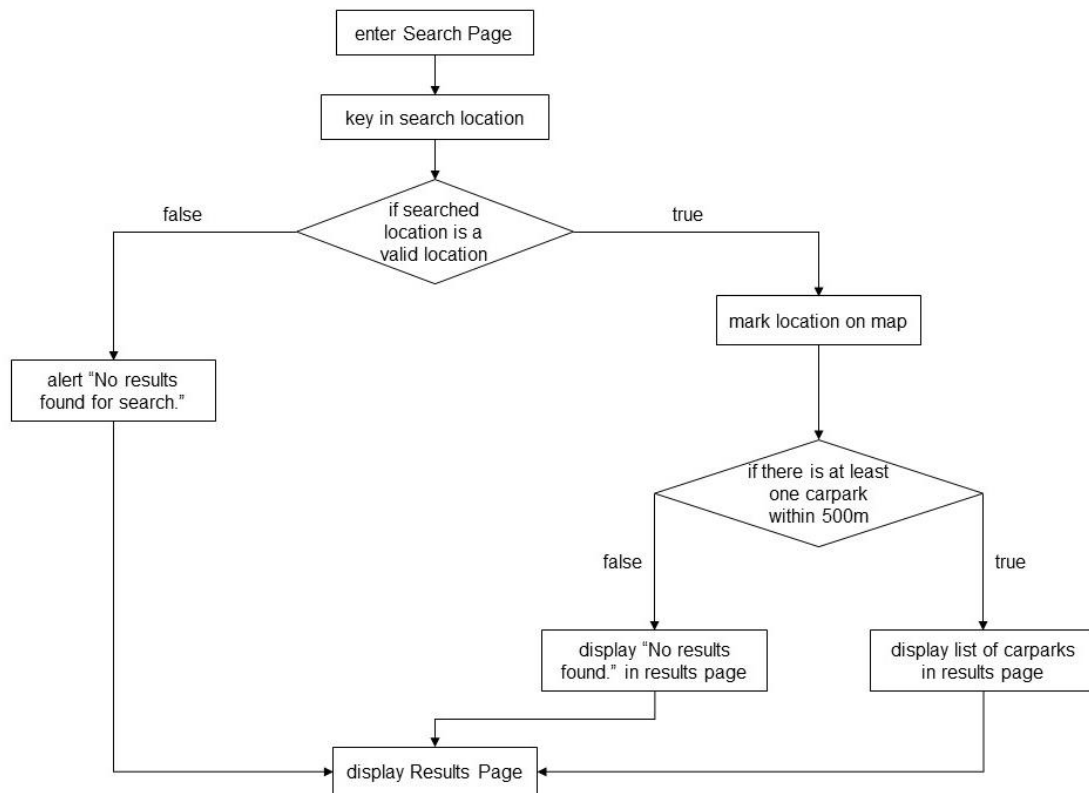
Test ID	Scenario	Expected Result	Actual Result
1	Unselect carpark to visit (carpark is selected from search results)	Return to the search result page.	Return to the search result page.
2	Unselect carpark to visit (carpark is selected from favourites page)	Return to the favourites page.	Return to the favourites page.

b. Specific cases

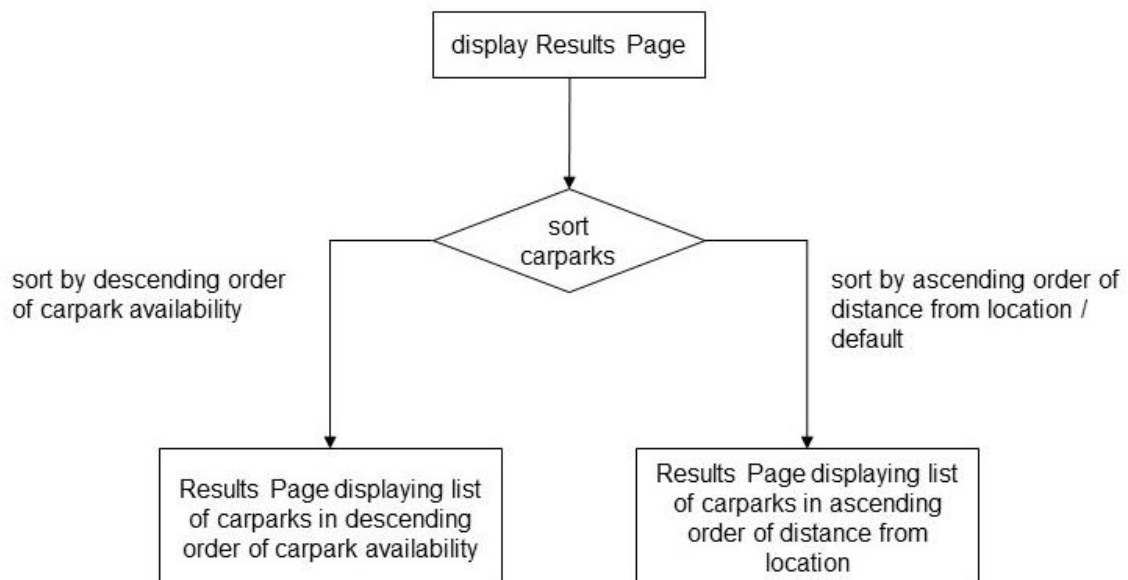
Search string	Expected Result	Actual Result
BLK 109-127 Hougang Avenue 1 (from search results)	Return to the search result page.	Return to the search result page.
BLK 962A Jurong West Street 91 (from favourites page)	Return to the favourites page.	Return to the favourites page.

White Box Testing (Control Flow Testing)

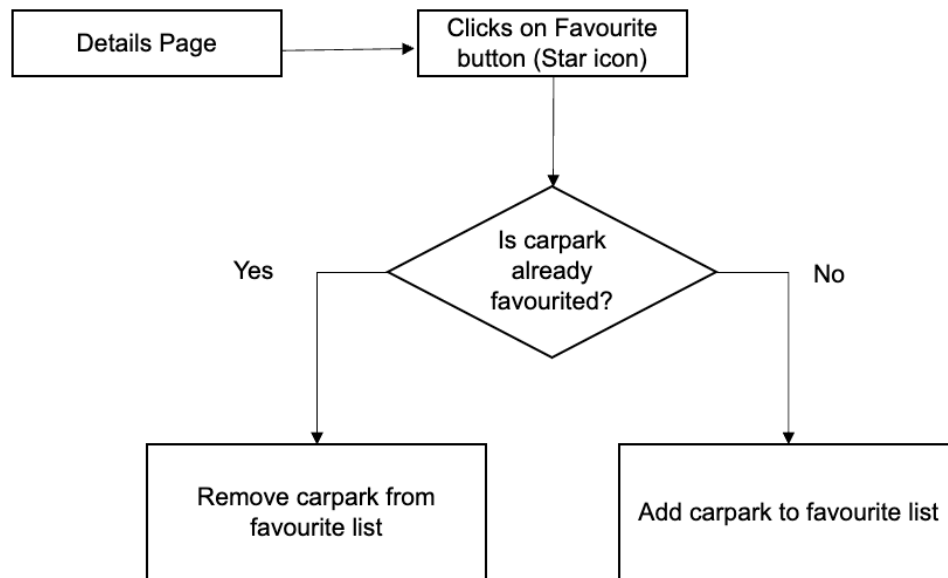
1. Search for Carparks



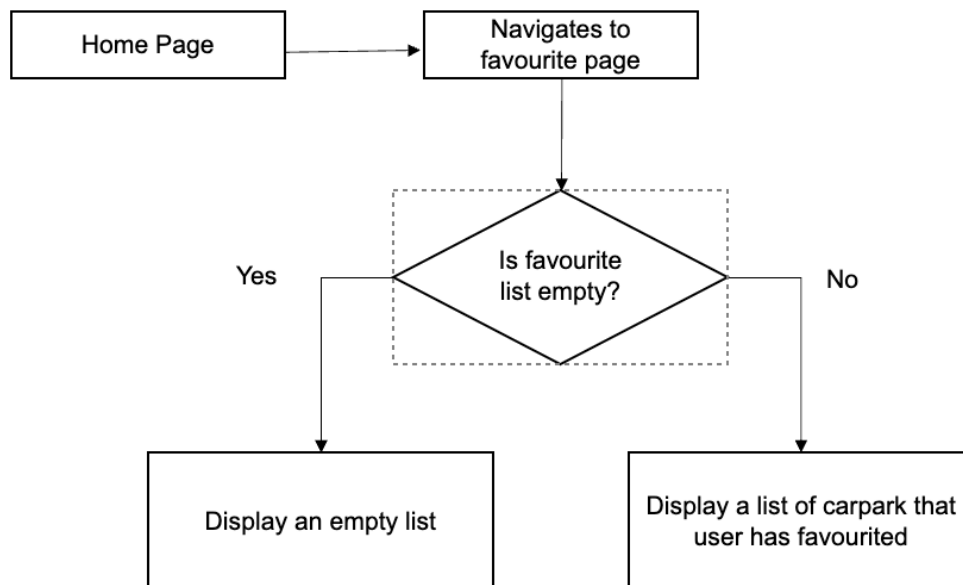
2. Sort Results



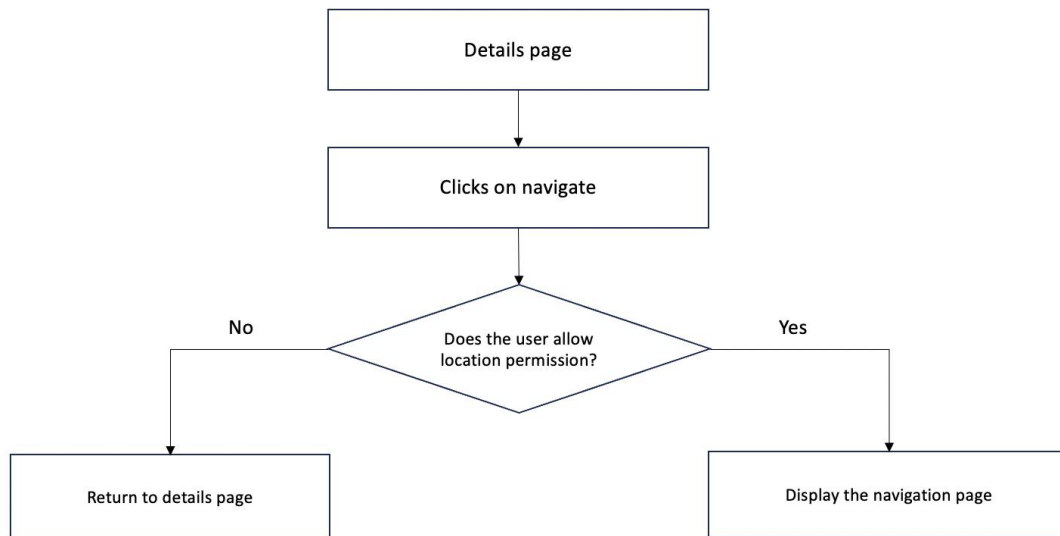
3. Favourite Carpark



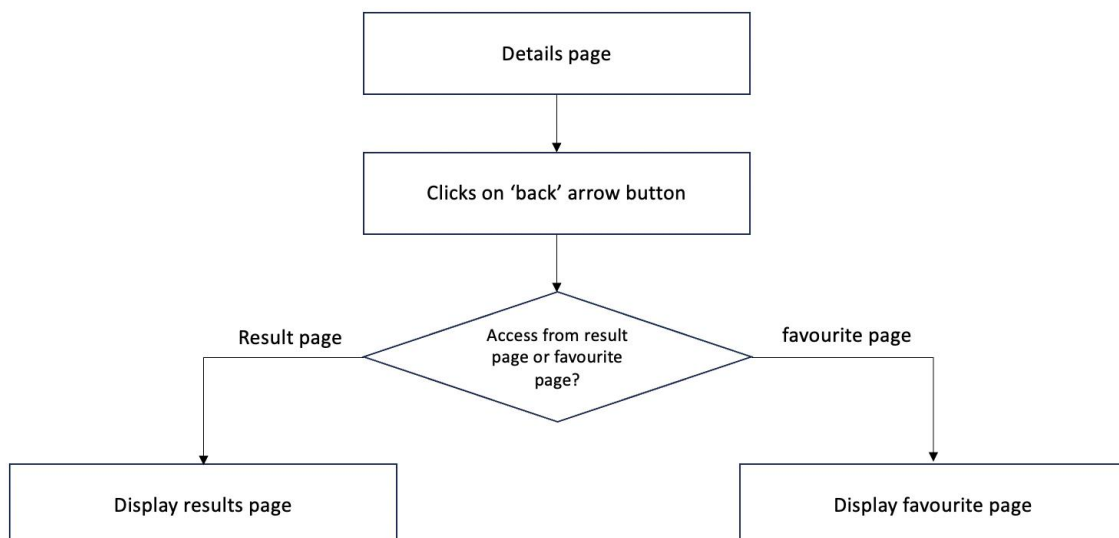
4. View Favourite Carparks



5. Select Carpark to Visit
a. Navigate to Carpark



b. Unselect Carpark to Visit



Video

A demo video demonstrating the features of Carpark NF can be found at <https://www.youtube.com/watch?v=iTqXrlhFv7U>

The video is best viewed in 1080p.

Source Code

The source code for Carpark NF can be found at <https://github.com/jinmin314/SC2006-CarParkNF/tree/main/Final/Code>