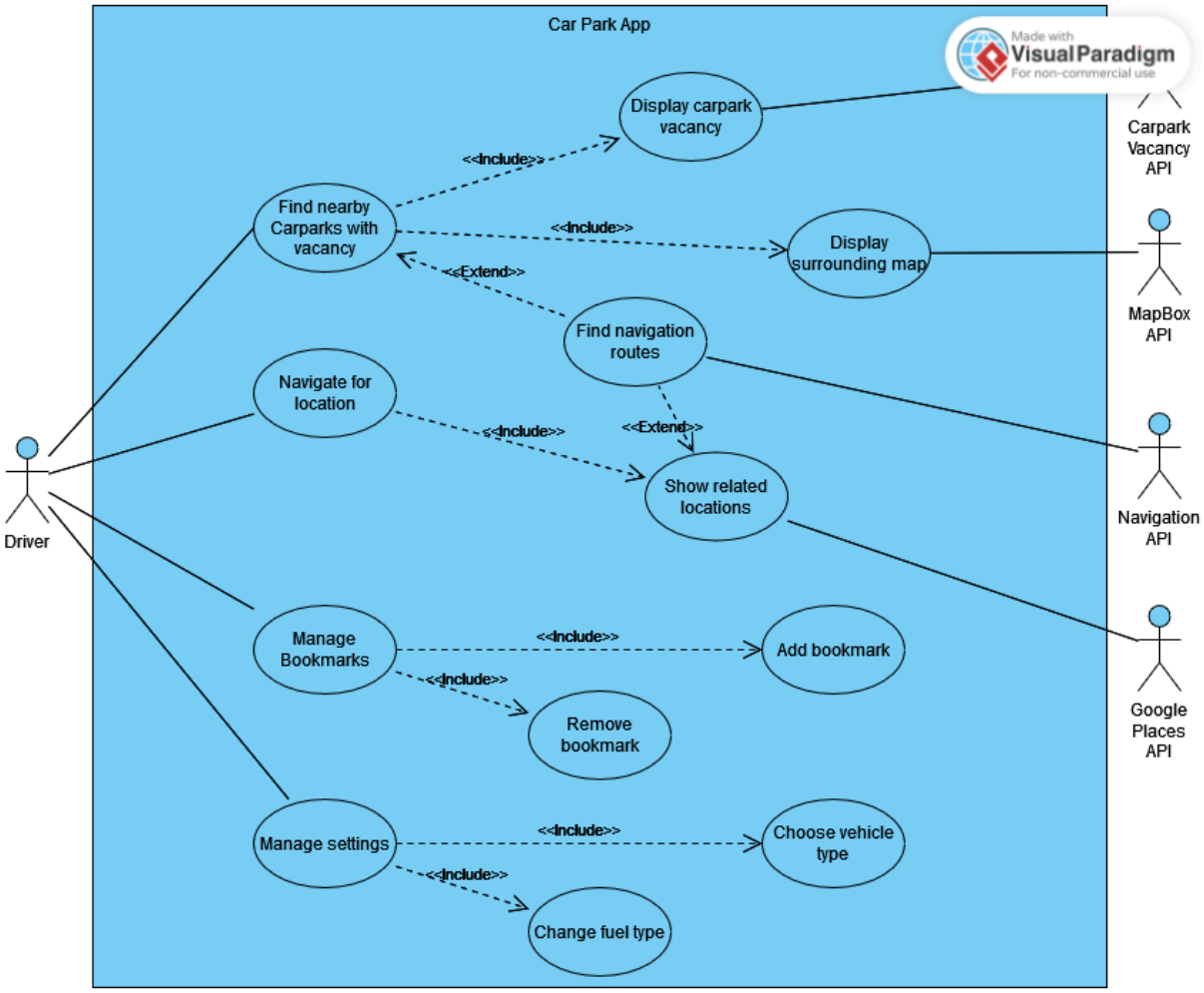
Use Case Diagrams and Descriptions

Table of Contents

1. Use Case Diagram



2. Use case descriptions

2.1 View Map

2.1.1 Description and Priority

Users should be able to view the map. If location access is enabled, the map will hone in on the user's current location and show the surrounding area. If location access is not enabled, the map will show the general area of the country. This map will be used extensively, across other features such as querying for locations and showing navigation routes.

Overall Priority	**Description**
High	Functionality of location search and navigation routing relies heavily on the map as a display.

2.1.2 Stimulus/Response Sequences

Use Case ID:	#1		
Use Case Name:	View Map		
Created By	Jiang Zong Zhe	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (Initiating), Mapbox Library/API
Description	Users can **view the geospatial 2D map**
Preconditions	1. User must be connected to the Internet 2. The Mapbox API or library must be accessible and functioning without service interruptions.
Postconditions	Upon map load, user's is prompted for their current location to be revealed. Allowing current location would display the user's location on the map OR The user's browser blocks location access
Priority	High

Frequency of use	High Users will view the map every time the app is initialised, and other functionalities rely on the map display.
Flow of events	1. User opens app and the map initialises with the general map of Singapore 2. User enables location access 3. Map hones in on the user's current location and surrounding area.
Alternative flows	1. User disables location access a. Map remains at the general map of Singapore
Exceptions	1. User has no internet access a. An empty screen will be rendered and only the top-navbar will be shown.
Includes	None
Special requirements	Stable Internet Connectivity
Assumptions	None
Notes and Issues	None

2.1.3 Functional Requirements

REQ-1: System must initialise and display a map interface upon the launch of the application using Mapbox API.

REQ-1.1: System must be able to display a loading indicator while map is being initialised.

REQ-1.2: System must be able to handle map initialisation failures

REQ -2: System must request location access permission from users upon first initialisation.

REQ-2.1: When location access is granted, the system must centre the map on the user's current location.

REQ-2.2: When location access is denied, the system must display the default Singapore map view.

REQ-3: The system must support user interaction with the map

REQ-3.1: The system must provide standard map controls:

REQ-3.1.1: Enable users to zoom in/out

REQ-3.1.2: Allow map to be rotatable

REQ-3.1.3: Allow smooth transition while panning across the map

2.2 Search for a Location

2.2.1 Description and Priority

Users must be able to query for locations in Singapore

Overall Priority	**Description**
High	Users must be able to find nearby carparks for other locations other than their current location so as to plan their navigation routes.

2.2.2 Stimulus/Response Sequences

Use Case ID:	#2-1		
Use Case Name:			
Created By	Jiang Zong Zhe	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (Initiating), Mapbox API
Description	Drivers can **search** for a **location of interest**
Preconditions	4. The app user has launched the app 5. The app user has access to the internet
Postconditions	1. Successful exit condition: The app user receives information about location of interest and presses the ‘leave’ tab 2. Unsuccessful exit condition: The system informs the user that no relevant locations were found with the search query or there was an invalid location input
Priority	High
Frequency of use	High
Flow of events	1. User enters a valid location into the search box. 2. System sends a request to the MapBox API and fetches relevant location results. 3. System displays consolidated details of most relevant location results in a dropdown list - LocationName, Address, DistanceAway. 4. User selects the chosen location. 5. System shows the selected location on the map.

	6. System shows nearby carparks to that location using the extended use case '*View Nearby Carparks*'. 7. System shows valid navigation routes using the extended use case '*Find Navigation Routes*'.
Alternative flows	1. User enters an invalid location into the search box. a. System fails to find any relevant locations based on the search query b. System informs the app user that no results were found and suggests refining the search c. The app user refines the query and the system resumes from the normal flow at step #1
Exceptions	1. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown.
Includes	*View Nearby Carparks* *Get Navigation Route*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.2.3 Functional Requirements

REQ-1: The system must allow users to manually enter a location.

REQ-2: The system must process the user's search input.

REQ-2.1: The system must send the search request to Mapbox API.

REQ-2.2: The system must validate the user's input.

REQ-2.2.1: If input is valid, the system must display the nearby carparks' information on the user interface.

REQ-2.2.2: If input is not valid, the system must provide feedback.

2.3 Get Navigation Routes

2.3.1 Description and Priority

Users will be shown the navigation route (via car/motorcycle) to a chosen location.

Overall Priority	**Description**
High	This works in tandem with the functionality to view the vacancy of

	nearby carparks, as drivers will likely want to see the route to the carpark after checking its vacancies.

2.3.2 Stimulus/Response Sequences

Use Case ID:	#2-2		
Use Case Name:	Get Navigation Routes		
Created By	Tan Chong Yao	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (initial actor), Mapbox API
Description	Users can **view navigation routes** to chosen location
Preconditions	1. User enables location access 2. User has internet access
Postconditions	Successful exit condition: User can view the navigation route to their desired location
Priority	High
Frequency of use	High
Flow of events	1. User searches for a location on the map using extended use case '*Search for a Location*', and selects the desired location. 2. System sends a request to the MapBox API to find the fastest route. 3. Map displays the route to the location.
Alternative flows	1. User chooses a bookmarked location to navigate to. a. User selects "Bookmark" tab on the UI. b. User selects "Navigate To" button on a saved bookmark. c. User is routed to the Main Page. d. System displays navigation route to the saved location.
Exceptions	1. User does not allow location access a. No navigation route is shown to the selected location

Includes	*Search for a Location*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.3.3 Functional Requirements

REQ-1: The system must check for prerequisites for navigation

REQ-1.1: System must check for location access status.

REQ-1.1.1: If location access is not granted, the system must display an error message.

REQ-1.2: System must check for internet connectivity.

REQ-1.2.1: If the device is not connected to the internet, the system must notify the user.

REQ-2: The system must display navigation routes when all prerequisites for navigation are met.

REQ-3: The system must be able to navigate to bookmarked locations without searching for the exact location in the search function.

REQ-3.1: System must display navigation button in bookmarks page.

REQ-3.2: The system must redirect to the main page with the route displayed.

2.4 View Nearby Carparks from a Location

4.4.1 Description and Priority

Users should be able to view a list of nearby carparks, along with information such as available parking spaces and price, within a 500m radius of their current location or any specified location on the map.

Overall Priority	**Description**
High	Viewing nearby carparks is a central role in the app's functionality.

4.4.2 Stimulus/Response Sequences

Use Case ID:	#2-3
Use Case Name:	View Nearby Carparks from a Location

Created By	Solis Aaron Mari Santos	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (Initiating), Mapbox API, LTA DataMall and HDB API
Description	Users can **find carparks with vacancies within 500m near a of a specific location**
Preconditions	1. User has GPS and internet access turned on 2. User is located in Singapore
Postconditions	1. User may click on a certain carpark to trigger the Search For a Location functionality (4.3) with that carpark's address 2. Each carpark selected generates a route to that location and leaves a persistent marker on the map to allow users to compare relative carpark locations
Priority	High
Frequency of use	High
Flow of events	1. User opens app and application immediately triggers the search for nearby carparks within 500m. 2. System will fetch vacancy and pricing data from LTA DataMall and HDB API, and routing from Mapbox API. 3. System will obtain 10 nearby carparks after fetching the data. 4. Pop-up box on the map will display 3 of the 10 nearby carparks a. Users may choose to expand the pop-up to see all 10 nearby carparks. b. Each carpark will display its price and lot information
Alternative flows	1. User enters a location and triggers the search for nearby carparks within 100m. 2. System will fetch vacancy and pricing data from LTA DataMall and HDB API, and routing from Mapbox API. 3. System will obtain 10 nearby carparks after fetching the data. 4. Pop-up box on the map will display 3 of the 10

	nearby carparks a. Users may choose to expand the pop-up to see all 10 nearby carparks. b. Each carpark will display its price and lot information
Exceptions	1. There are no nearby carparks within 500m 2. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown. 3. There is no lot information for a carpark a. Carparks with no lot information will show a No information available line 4. There is no price information for a carpark a. Carparks with no price information will show a No information available line
Includes	*Search for a Location* *Get Navigation Route*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.4.3 Functional Requirements

REQ-1: The system must automatically search for nearby carparks upon application initialisation.

REQ-2: The system must be able to fetch data(availability of lots, price, navigation route) from backend APIs.

REQ-2.1: The system must retrieve real-time vacancy and price data from LTA DataMall and HDB API.

REQ-2.2: The system must fetch routing information from Mapbox API.

REQ-2.3: The system must display appropriate error messages when it fails to retrieve data from API calls.

REQ-3: The system must be able to find nearby carparks based on user location.

REQ-3.1: The system must search within 100m radius of user's location

REQ-3.1.1: If there are no carparks within 100m radius, the system will search for nearest carparks beyond 100m radius.

REQ-3.1.2: If there are carparks within 100m radius, the system must identify and sort the 10 nearest carparks based on distance, from nearest to furthest.

REQ-4: The system must display the nearby carparks' information on user interface.

REQ-4.1: The system must display the initial-pop-up of 3 nearest carparks.

REQ-4.1.1: The system must display the vacancy information for each carpark.

REQ-4.1.2: The system must display the price per hour for each carpark.

REQ-4.1.3: The system must display the name of each carpark.

REQ-4.1.4: The system must allow users to navigate to each carpark.

REQ-4.1.4.1: The system must display the navigation route to the chosen carpark.

2.5 Save and Load Bookmarks

2.5.1 Description and Priority

Users are able to save locations or carparks they visit regularly to a bookmarks page.

Overall Priority	**Description**
Medium to High	Users will find it more convenient to be able to see frequented locations and carparks.

2.5.2 Stimulus/Response Sequences

Use Case ID:	#3-1		
Use Case Name:	Save and Load Bookmarks		
Created By	Sun Si Tong	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (initial actor)
Description	Users can **bookmark locations** on the app for search convenience. Users can also view a list of bookmarked locations that they have previously added.
Preconditions	1. User must have internet access on their phone 2. User allows cookie access

Postconditions	1. A new bookmark is added OR 2. A bookmark is deleted OR
Priority	Medium
Frequency of use	Medium
Flow of events	1. User selects “Bookmark” tab on the UI. 2. System retrieves any bookmarks saved by the user from server-side database and displays them on the UI. 3. User searches for a location on the map using extended use case ‘*Search for a Location*’. 4. User selects “Add Bookmark” button on the UI. 5. System saves the bookmarks added by the user to server-side database. 6. System displays the new bookmark on the UI.
Alternative flows	1. User saves bookmarks from the Main Page e. User searches for a location on the map using extended use case ‘*Search for a Location*’. f. System displays nearby carparks to the location g. User selects “Add Bookmark” button for a specific carpark h. System saves the bookmarks added by the user to server-side database. 2. User removes an existing bookmark a. User selects “Remove Bookmark” button on the UI. b. System removes the bookmark from the server-side database, and updates the UI correspondingly.
Exceptions	1. User does not allow cookie access a. If the user saves a bookmark and refreshes the page, the saved bookmark will not be reflected on the app. 2. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown.
Includes	*Search for a Location*
Special requirements	- Stable Internet Connectivity
Assumptions	- None

Notes and Issues	- None

2.5.3 Functional Requirements

REQ-1: System must verify prerequisite conditions.

REQ-1.1: System must check internet connectivity.

REQ-1.1.1: System must notify user if internet is unavailable.

REQ-1.1.2: System must provide retry option when connection is restored.

REQ-1.2: System must verify cookie access permission.

REQ-1.2.1: System must prompt user to enable cookie access if disabled.

REQ-1.2.2: System must explain importance of cookie access for bookmark functionality.

REQ-2: The system must display a bookmark tab in the main user interface.

REQ-3: The system must handle bookmark addition.

REQ-3.1: The system must allow users to save locations to bookmarks directly from the location search function at the navigation interface.

REQ-3.2: The system must handle bookmark storage

REQ-3.2.1: The system must save bookmarks to the database.

REQ-3.2.2: The system must update UI to show new bookmarks saved.

REQ-3.3: The system must not save duplicate bookmarks.

REQ-3.3.1: When the user tries to save a duplicate bookmark from bookmark interface, the bookmark will not be added

REQ-3.3.2: When the user tries to save a duplicate bookmark from the navigation interface, it will be indicated that the location has already been added.

REQ-4: The system must handle bookmark removal.

REQ-4.1: The system must allow user to remove bookmark from the bookmark interface.

REQ-4.2: The system must allow user to remove bookmark from the navigation interface.

REQ-4.3: The system must update the UI to reflect the bookmark has been removed.

REQ-4.4: The system must remove bookmark from the database.

REQ-5: System must handle data persistence.

REQ-5.1: System must maintain bookmark synchronisation.

REQ-5.1.1: System must sync bookmarks across sessions.

REQ-5.2: System must manage cookie-related scenarios.

REQ-5.2.1: System must store bookmark data in cookies.

EQ-5.2.2: System must handle cookie access denial.

2.6 Toggle Bookmark to Navigation

2.6.1 Description and Priority

Users are able to click on bookmarks to toggle location navigation route and find nearby carparks to that bookmarked location

Overall Priority	**Description**
Medium to High	Users will find it more convenient to be able to get the navigation routes to their bookmarked locations

2.6.2 Stimulus/Response Sequences

Use Case ID:	#3-2		
Use Case Name:	Toggle Bookmark to Navigation		
Created By	Tan Yu Xiu	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (initial actor)
Description	Users can toggle a bookmark by clicking on the navigation icon next to the display bookmark to enable navigation route calculation to that location
Preconditions	3. User must have internet access on their phone 4. User allows cookie access 5. User may have saved bookmarks before
Postconditions	3. User is routed to the navigation page, route is shown
Priority	Medium
Frequency of use	Medium
Flow of events	4. User selects “Bookmark” tab on the UI. 5. System retrieves any bookmarks saved by the user from server-side database and displays them on the UI. 7. User clicks on the “navigate” button next to each Bookmark card. 8. User is routed to the main navigation page 9. Nearby carparks are shown

Alternative flows	
Exceptions	1. User does not allow cookie access a. If the user saves a bookmark and refreshes the page, the saved bookmark will not be reflected on the app. 2. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown.
Includes	*Save and Load Bookmarks* *Get Navigation Routes* *Show Nearby Carparks*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.6.3 Functional Requirements

REQ-1: System must verify prerequisite conditions.

REQ-1.1: System must check internet connectivity.

REQ-1.1.1: System must notify user if internet is unavailable.

REQ-1.1.2: System must provide retry option when connection is restored.

REQ-1.2: System must verify cookie access permission.

REQ-1.2.1: System must prompt user to enable cookie access if disabled.

REQ-1.2.2: System must explain importance of cookie access for bookmark functionality.

REQ-2: The system must display a bookmark tab in the main user interface.

REQ-3: The system must handle bookmark selection.

REQ-3.1: The system must showcase a navigation to button next to each bookmarked location

REQ-3.1.1: The system must allow users to click on bookmark's navigate to button

REQ-3.2: The system must handle the routing to the main map page

REQ-3.3: The system must load in the bookmark's coordinates into the search function of the map page

REQ-3.3.1: The map must be recentered and zoomed into that location

REQ 3-4: The system must generate the route from the user's current location to said bookmarked location.

REQ-3.5: The system must display the list of nearby carparks to the bookmarked location upon loading the route

2.7 Toggle Carpark Sorting Filter

2.7.1 Description and Priority

Users are able to toggle the filter by which nearby carparks are sorted by. These filters are distance, number of vacancies, and pricing rates respectively.

Overall Priority	**Description**
Medium	Users will not be likely to toggle the sorting filter much.

4.7.2 Stimulus/Response Sequences

Use Case ID:	#4-1		
Use Case Name:	Toggle Carpark Sorting Filter		
Created By	Quek Jun Siong	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (initial actor)
Description	Users can **toggle sorting filter** for carparks.
Preconditions	1. User must have internet access on their phone
Postconditions	1. User's search options are updated to reflect changes in the search results
Priority	Medium
Frequency of use	Low
Flow of events	1. System displays nearby carparks to user's current location, or a searched location, using extended use case '*View Nearby Carparks*'. 2. User selects "Settings" tab on the UI. 3. User selects sorting filter out of 3 options - Sort by

	Distance, Sort by Vacancies, and Sort by Pricing Rates. 4. Nearby carparks will be sorted depending on the filter.
Alternative flows	-
Exceptions	1. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown. 2. There are no lot information for a carpark a. Carparks with no lot information will be pushed to the back in the sort by lot filter b. If all carparks have no price, the sort by distance will be used 3. There are no price information for a carpark a. Carparks with no price information will be pushed to the back in the sort by price filter b. If all carparks have no price, the sort by distance will be used
Includes	*View Nearby Carparks*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.7.3 Functional Requirements

REQ-1: System must integrate with carpark display functionality.

REQ-1.1: System must link with "View Nearby Carparks from a location" feature.

REQ-1.1.1: System must access current carpark list.

REQ-1.1.2: System must maintain real-time data synchronisation.

REQ-2: System must provide filter interface.

REQ-2.1: System must display "Settings" tab in UI.

REQ-2.2: System must show sorting options.

REQ-2.2.1: System must provide "Sort by Distance" option.

REQ-2.2.2: System must provide "Sort by Vacancies" option.

REQ-2.2.3: System must provide "Sort by Pricing Rates" option.

REQ-2.3: System must indicate currently active filter.

REQ-2.3.1: System must highlight selected filter option.

REQ-3: System must implement sorting functionality.
REQ-3.1: System must sort by distance.
REQ-3.1.1: System must calculate distance from current/searched location.
REQ-3.1.2: System must arrange carparks in ascending order of distance.
REQ-3.2: System must sort by vacancies.
REQ-3.2.1: System must retrieve real-time vacancy data.
REQ-3.2.2: System must arrange carparks in descending order of vacancies.
REQ-3.3: System must sort by pricing rates.
REQ-3.3.1: System must compare carpark pricing data.
REQ-3.3.2: System must arrange carparks in ascending order of rates.

REQ-4: System must handle display updates.
REQ-4.1: System must refresh carpark list display.
REQ-4.1.1: System must update list immediately after filter change.
REQ-4.1.2: System must maintain smooth transition between sorts.
REQ-4.2: System must preserve essential information.
REQ-4.2.1: System must maintain all carpark details during sort.
REQ-4.2.2: System must retain user's view position after sorting.

REQ-5: System must maintain filter state.
REQ-5.1: System must save user's filter preference.
REQ-5.1.1: System must store last used filter.
REQ-5.1.2: System must apply saved filter on app restart.

2.8 Toggle Vehicle Type

2.8.1 Description and Priority

Users are able to toggle their preferred vehicle type. Available options are fuel cars, motorcycles, and EV cars.

Overall Priority	**Description**
Medium	Most users are likely to be fuel car drivers, and likely only drive one vehicle type, thus being unlikely to have to toggle vehicle types.

2.8.2 Stimulus/Response Sequences

Use Case ID:	#4-2		
Use Case Name:	Toggle Vehicle Type		
Created By	Quek Jun Siong	Last Updated By:	Tan Yu Xiu
Date Created	27/8/2024	Date Last Updated:	15/11/2024

Actor	User (initial actor), LTA DataMall and HDB API
Description	Users can **toggle their preferred vehicle type.**
Preconditions	1. User must have internet access on their phone
Postconditions	1. User's vehicle options are updated to reflect changes in the search results
Priority	Medium
Frequency of use	Low
Flow of events	1. System displays nearby carparks to user's current location, or a searched location, using extended use case '*View Nearby Carparks*'. 2. User selects "Settings" tab on the UI. 3. User selects preferred vehicle typeout of 3 options - Fuel Car, Motorcycle, or EV car. 4. System sends a request to LTA DataMall and HDB API to fetch carpark results that match the chosen vehicle type. 5. System returns number of vacancies depending on the chosen vehicle type.
Alternative flows	-
Exceptions	4. User does not have internet access a. An empty screen will be rendered and only the top-navbar will be shown. 5. There are no lot information for motorcycle parking a. Carparks with no lot information will be pushed to the back in the sort by lot filter b. If all carparks have no price, the sort by distance will be used

	6. There are no price information for motorcycle parking a. Carparks with no price information will be pushed to the back in the sort by price filter b. If all carparks have no price, the sort by distance will be used
Includes	*View Nearby Carparks*
Special requirements	- Stable Internet Connectivity
Assumptions	- None
Notes and Issues	- None

2.8.3 Functional Requirements

REQ-1: System must integrate with carpark display system.
- REQ-1.1: System must link with "View Nearby Carparks" feature.
  - REQ-1.1.1: System must access current carpark list.
  - REQ-1.1.2: System must maintain vehicle type filters.

REQ-2: System must provide vehicle type interface.
- REQ-2.1: System must display "Settings" tab in UI.
- REQ-2.2: System must show vehicle type options.
  - REQ-2.2.1: System must provide "Fuel Car" option.
  - REQ-2.2.2: System must provide "Motorcycle" option.
  - REQ-2.2.3: System must provide "EV Car" option.
- REQ-2.3: System must indicate active vehicle type.
  - REQ-2.3.1: System must highlight selected vehicle option.
  - REQ-2.3.2: System must persist vehicle selection.

REQ-3: System must handle API interactions.
- REQ-3.1: System must communicate with LTA DataMall API.
  - REQ-3.1.1: System must send vehicle type parameter.
  - REQ-3.1.2: System must handle API response data.
- REQ-3.2: System must communicate with HDB API.
  - REQ-3.2.1: System must send vehicle type parameter.
  - REQ-3.2.2: System must handle API response data.
- REQ-3.3: System must manage API errors.
  - REQ-3.3.1: System must implement error handling for API failures.
  - REQ-3.3.2: System must provide retry mechanisms.

REQ-4: System must process vacancy data.

- REQ-4.1: System must filter vacancies by vehicle type.
  - REQ-4.1.1: System must show car lots for fuel car and EV selection.
  - REQ-4.1.2: System must show motorcycle lots for motorcycle selection.
- REQ-4.2: System must update vacancy display.
  - REQ-4.2.1: System must refresh vacancy numbers after type change.
  - REQ-4.2.2: System must indicate lot type on display.