

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

SC2006 Software Engineering

Project Report

Team Members:

Joshua Adrian Cahyono U2220391C

Kenny Seah Yong Jie U2121539K

Lim Wei Zi U2122572B

Tan Chuan Liang U2121095E

Lab Group A36

Team 3 (Team Cirno)

Contents

Project Mission Statement.....	3
Functional Requirements.....	3
Non-Functional Requirements.....	5
Data Dictionary	5
Tech Stack	6
Design Patterns	6
Model-View-Controller (MVC) Design Architecture	6
Strategy Design Pattern	6
Diagrams	7
Use Case Diagram	7
Class Diagram.....	8
Architecture Diagram.....	9
Dialog Map / UI State Machine Diagram	9
Use Case Description	10
UI Mockups	37
Testing.....	40
Black Box Testing (Search for Carparks).....	40
Black box testing (Favourites & Blacklist)	41
White box testing.....	41
Demo Script.....	42

Project Mission Statement

Cirno Team will develop a parking app, ParkWhere, to allow motorists to look up carpark information. The project is complete when the app is tested and approved by the ZEA (CZ2006/CE2006 Authority). This project supports the Data-Driven Smart Nation Competition to exploit publicly available government data to help motorists find suitable carparks easily.

Functional Requirements

1. Main Menu
 - 1.1. User must be able to input a search query into the main search bar.
 - 1.2. When the user taps the “Search” button, the system must parse the user’s search query and conduct a search based on that query.
 - 1.2.1. The system must be able to parse the user’s search query into a search type. Search types include:
 - 1.2.1.1. Full address
 - 1.2.1.2. Partial address (only road name or general location)
 - 1.2.1.3. Postal Code
 - 1.2.1.4. Carpark number
 - 1.2.2. The system must be able to conduct a search based on a search query.
 - 1.2.2.1. The search query will be listed on the Search Listing page.
2. Search Listing
 - 2.1. The system must display the resolved search query on the Search Listing page.
 - 2.2. The system must display, in pages, the nearest carparks to the searched location in the order of closest to furthest.
 - 2.3. The system must display at most 10 carpark listings per page in a table format.
 - 2.3.1. In each row of the table, the system must display each carpark listing’s Basic Information.
 - 2.3.1.1. The system must show the carpark listing as greyed out if the carpark’s available lots is zero.
 - 2.3.2. When the user taps a carpark listing, the system must show that carpark’s Advanced Information.
 - 2.3.3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Favorites list.
 - 2.4. When the user taps the “Next Page” button, the system shows the next 10 carpark listings in order.
 - 2.5. The system must be able to filter search results returned from a search query.
 - 2.6. When the user taps the “Filter” button, the system must show the filtering options. The filtering options are:
 - 2.6.1. Maximum distance to carpark
 - 2.6.2. Minimum number of available lots
 - 2.6.3. Parking System (Electronic/Coupon)
 - 2.6.4. Short Term Parking availability
 - 2.6.5. Free Parking availability
 - 2.6.6. Night Parking availability

- 2.7. When any of the search filters are applied, the system must repopulate the search listing with carpark that meet the search filters' criteria.
3. Favorites Listing
 - 3.1. The user must be able to add and remove carpark to a Favorites list. (see 2.3.3)
 - 3.2. The system must display, in pages, the Favorite carpark in alphabetical order of carpark number.
 - 3.3. The system must display at most 10 carpark listings per page in a table format.
 - 3.3.1. In each row of the table, the system must display each carpark listing's Basic Information.
 - 3.3.1.1. The system must show the carpark listing as greyed out if the carpark's available lots is zero.
 - 3.3.2. When the user taps a carpark listing, the system must show that carpark's Advanced Information.
 - 3.3.3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Favorites list.
 - 3.4. When the user taps the "Next Page" button, the system shows the next 10 carpark listings in order.
4. Blacklist
 - 4.1. The user must be able to add and remove carpark to a Blacklist. (see 2.3.3)
 - 4.2. The system must display, in pages, the Blacklisted carpark in alphabetical order of carpark number.
 - 4.3. The system must display at most 10 carpark listings per page in a table format.
 - 4.3.1. In each row of the table, the system must display each carpark listing's Basic Information.
 - 4.3.1.1. The system must show the carpark listing as greyed out if the carpark's available lots is zero.
 - 4.3.2. When the user taps a carpark listing, the system must show that carpark's Advanced Information.
 - 4.3.3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Blacklist.
 - 4.4. When the user taps the "Next Page" button, the system shows the next 10 carpark listings in order.
5. Map
 - 5.1. When the user requests a map of nearby carpark, the system must show a map with markers indicating the 10 nearest carpark to the user's current location.
 - 5.1.1. The map's zoom level must be at the finest level appropriate that shows all 10 nearest carpark.
 - 5.1.1.1. If the user zooms out from default zoom level shown, carpark other than the specified 10 carpark must be shown as markers on the map.
 - 5.1.2. When the user taps on a marker for a carpark, an info box must be shown by the system.

This info box should contain:

 - 5.1.2.1. Carpark number
 - 5.1.2.2. Available lots
 - 5.1.2.3. Total lots
 - 5.1.2.4. A button to request navigation to this carpark

- 5.2. From a specific carpark's Advanced Information screen, or a specific carpark selected in the map described in 5.1, the user must be able to request navigation to this carpark. Upon requesting navigation, the system must find the fastest route from the user's current location to the location of the specified carpark.
 - 5.2.1. The system must show on a map a highlighted fastest route between the user's current location and the location of the requested carpark.
6. Graph
 - 6.1. The user must be able to query the graph of a certain carpark.
 - 6.2. The system must show the graph of the availability history of that carpark.
 - 6.2.1. The system must retrieve the data from the history database.
 - 6.2.2. The history database must give the last 24 hours of history of the requested carpark
 - 6.3. The system must show the trending hour graph of that carpark
 - 6.3.1. The system must show the graph of the trend of the carpark

Non-Functional Requirements

1. System must not crash.
2. System must not hang.
3. Search must return result within 5 seconds.
4. 90% of users must be able to get the search results within 2 minutes upon accessing the system.

Data Dictionary

Carpark – A building or infrastructure that allows cars to be parked legally.

Carpark Number – The unique alpha-numeric identifier for a singular carpark.

Carpark Location – The details of where the carpark is located. This may include street name, block number, postal code.

Carpark Lots – Individual spaces for a single car to be parked.

Available Lots in Carpark – The current number of carpark lots that are not occupied by a parked car.

Total Lots in Carpark – The maximum number of carpark lots that a carpark accommodates.

Carpark Basic Information – A collection of basic information about a carpark, that is displayed in the Search Listing. This includes: Carpark Number, Carpark Distance from queried location, Carpark Location, Available Lots in Carpark, Total Lots in Carpark

Carpark Advanced Information – A collection of advanced information about a carpark, that is displayed when the user selects a carpark. This includes: Carpark Rates, Parking System (Electronic/Coupon), Short Term Parking Times, Free Parking Availability, Night Parking Availability, HDB/URA/LTA affiliation

Tech Stack

UI – Figma, Bootstrap

Frontend – React.js

Backend – Django REST Framework

Database – SQLite3

Design Patterns

Model-View-Controller (MVC) Design Architecture

For this project, MVC design architecture is chosen as its primary benefit to our team is concurrent development. Different people can work on both front-end system and back-end system without risk of much conflict.

Also, MVC has high modularity, so Views can be hot swapped to change the UI at any time. This is useful as UI requirements can change over time over the whole Software Development Lifecycle.

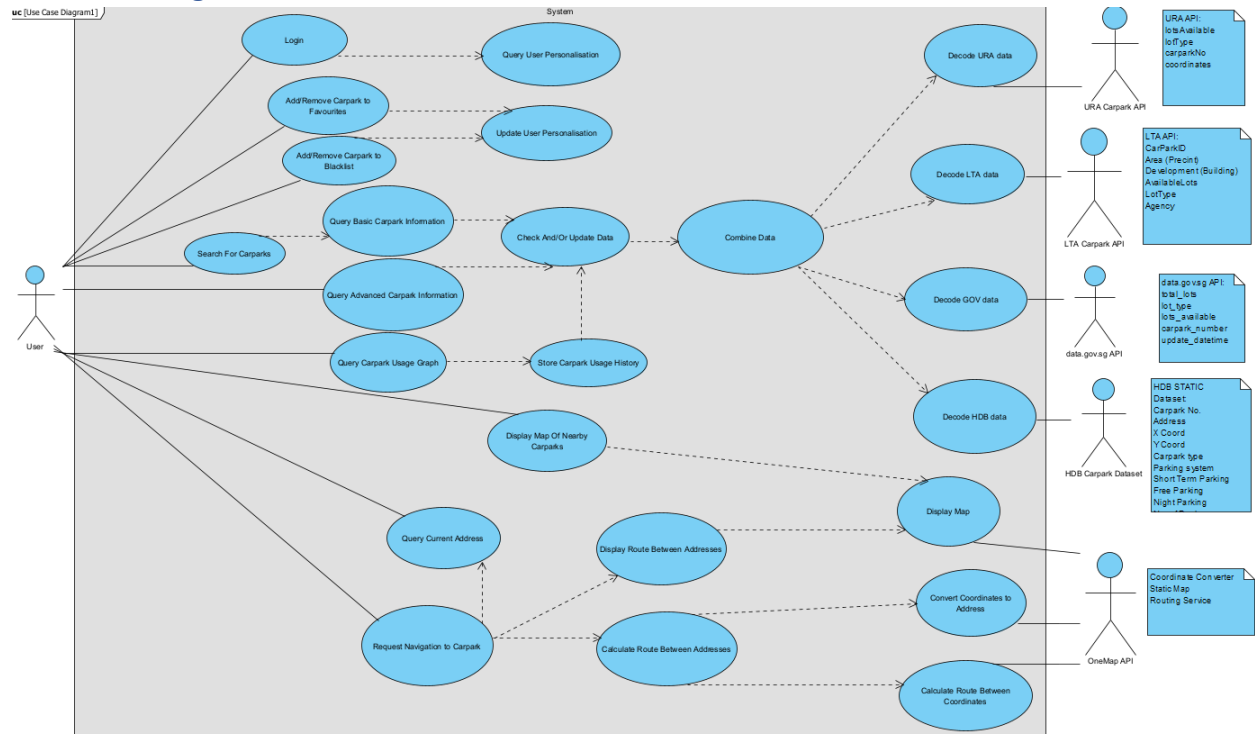
Lastly, MVC promotes easy extensibility. Our system passes data using JSON and REST, so API extensions can easily be added to the system.

Strategy Design Pattern

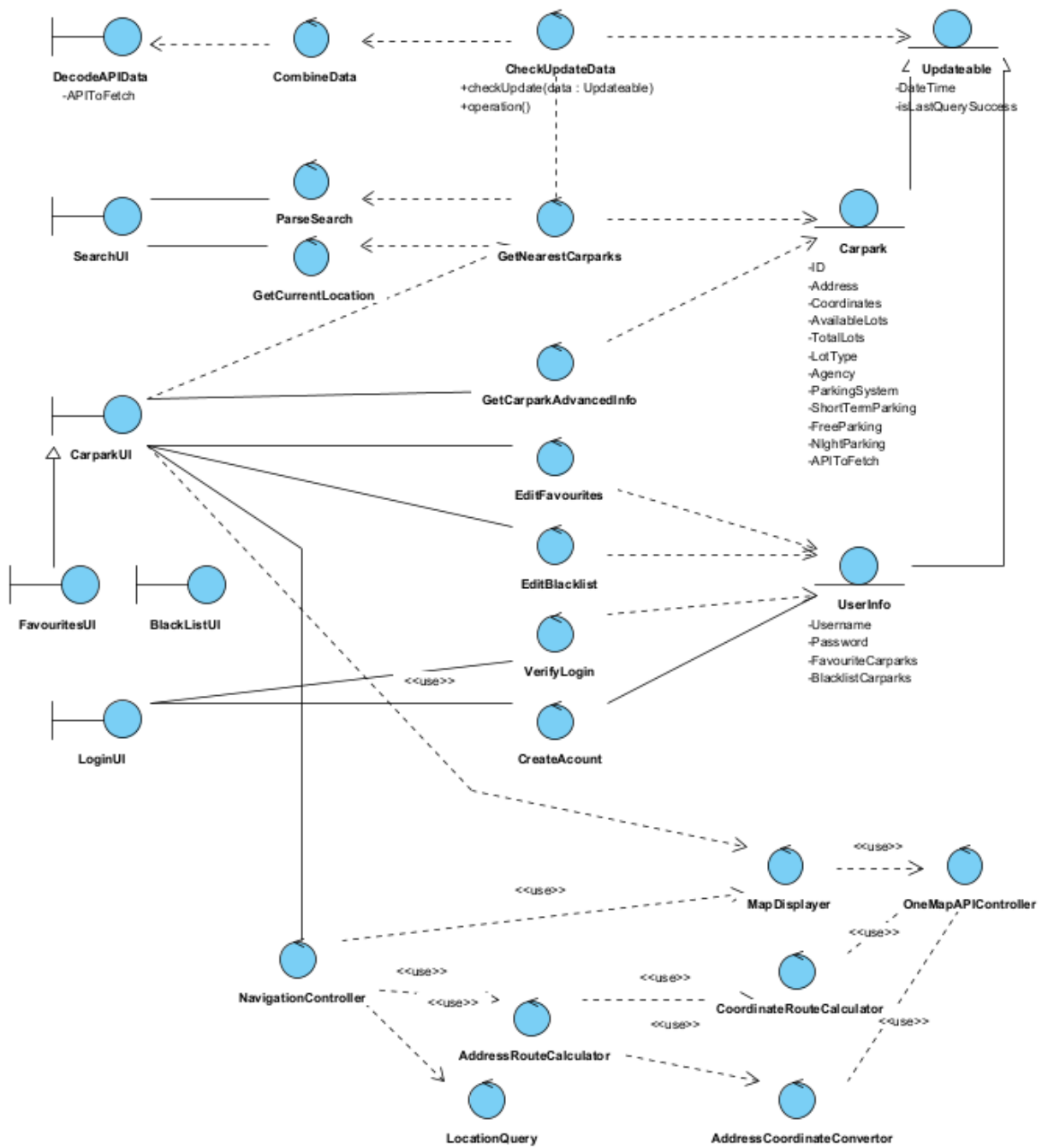
In the implementation of Favourites and Blacklist feature, there is high similarity between the two. As a result, both features inherit from the same root. In the front-end controller, the two features extend a parent function and class. In the back-end database, both favourites and blacklist are represented in the same way as foreign keys to User data. All these are applications of Strategy Design Pattern as they are similar and interchangeable.

Diagrams

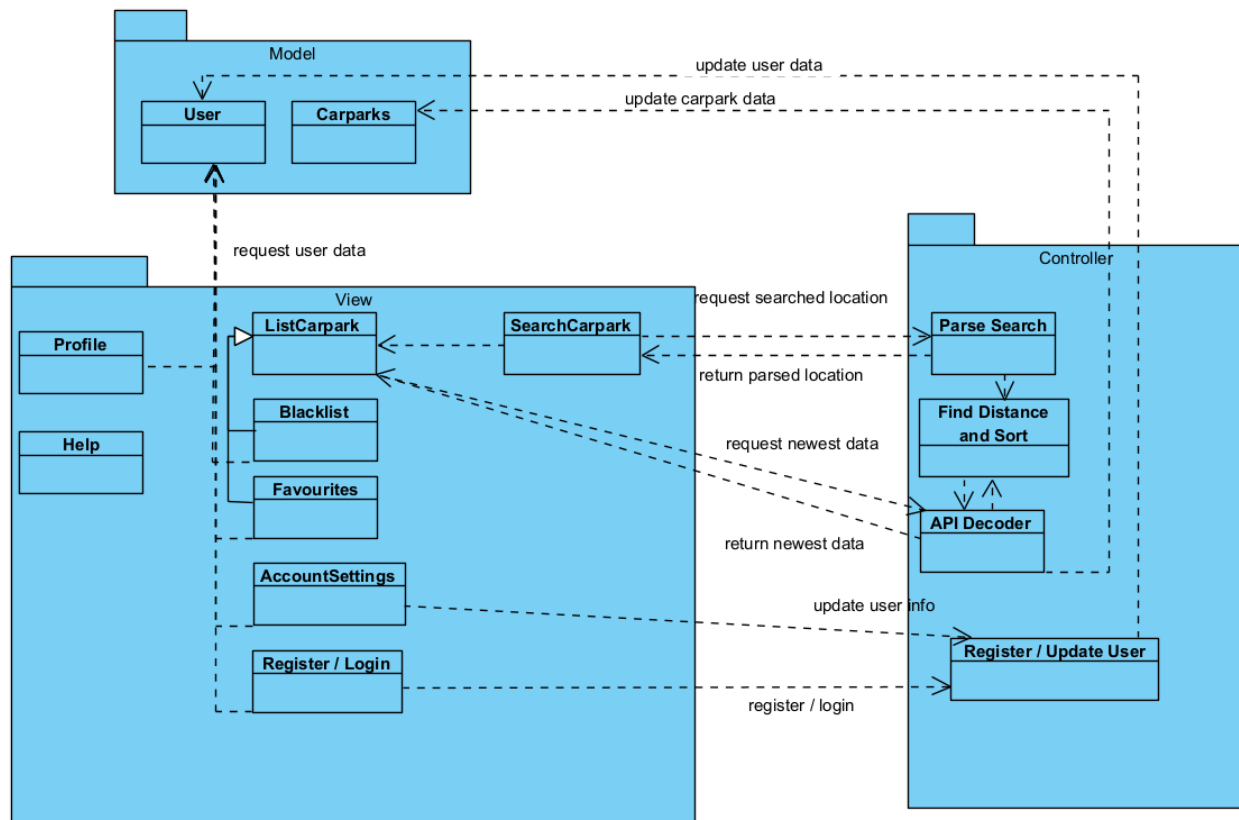
Use Case Diagram



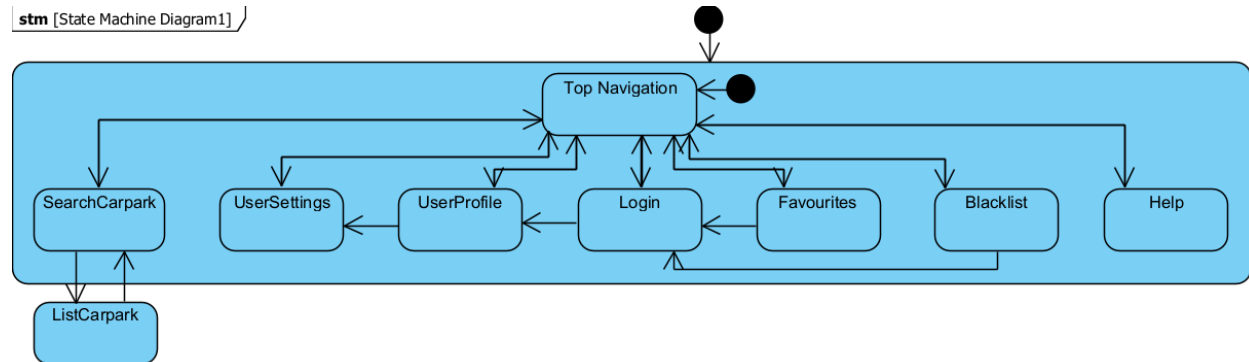
Class Diagram



Architecture Diagram



Dialog Map / UI State Machine Diagram



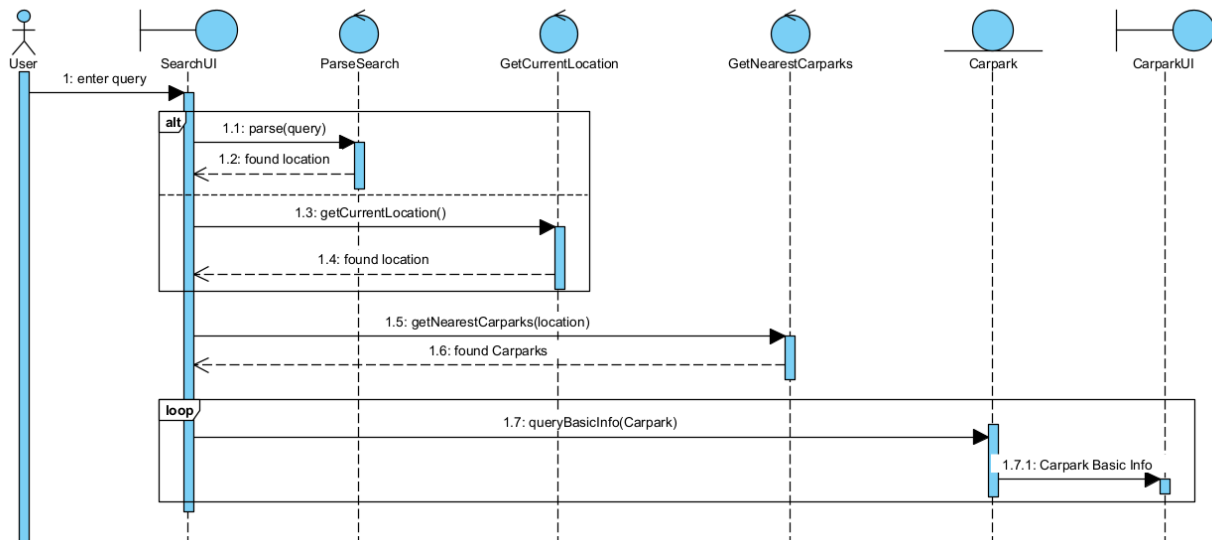
Use Case Description

Use Case ID:	1		
Use Case Name:	Search For Carparks		
Created By:	Tan Chuan Liang	Last Updated By:	Tan Chuan Liang
Date Created:	1/28/2023	Date Last Updated:	8/2/2023
Actor:	User (Initiating Actor)		
Description:	Search for nearby carparks based on search query.		
Preconditions:	1. App must be connected to WiFi/Mobile Data		
Postconditions:	1. User is able to view a list of nearby carparks		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User enters search query into search bar 2. System parses search query into a resolvable location 3. System finds at most 10 (or other preset value) nearest carparks to the queried location 4. System displays at most 10 (or other preset value) nearest carparks to the queried location, with each carpark's Basic Information, using the included use case Query Basic Carpark Information 		
Alternative Flows:	<p>AF-S1: User taps "Back" button while system is searching.</p> <ol style="list-style-type: none"> 1. Return to step 1. <p>AF-S2a: User does not enter search query into search bar and commences search.</p> <ol style="list-style-type: none"> 1. System assumes queried location to be current location. <p>AF-S2b: Search query cannot be resolved to a location.</p> <ol style="list-style-type: none"> 1. System assumes queried location to be current location. <p>AF-S4: User taps "Next Page" button</p> <ol style="list-style-type: none"> 1. If there are more carpark listings, system repopulates search listing with next 10 (or other preset value) carpark listings, continue from step 3. <p>AF-S4: User taps "Previous Page" button</p> <ol style="list-style-type: none"> 1. System repopulates search listing with previous 10 (or other preset value) carpark listings, continue from step 3. 		
Exceptions:	<p>EX-AF-S2a/b: Current location cannot be determined</p> <ol style="list-style-type: none"> 1. System displays error message "Cannot find current location, please allow and turn on GPS" <p>EX-AF-S2: Internet connection gets cut off</p> <ol style="list-style-type: none"> 1. System displays error message "No internet connection, please check if your mobile data or Wifi is turned on." 2. Return to step 1. 		
Includes:	<p>#2 Query Basic Carpark Information</p> <p>#3 Query Advanced Carpark Information</p>		
Special Requirements:	-		
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Basic Information for found carparks is retrieved successfully. 		

Notes and Issues:	-
-------------------	---

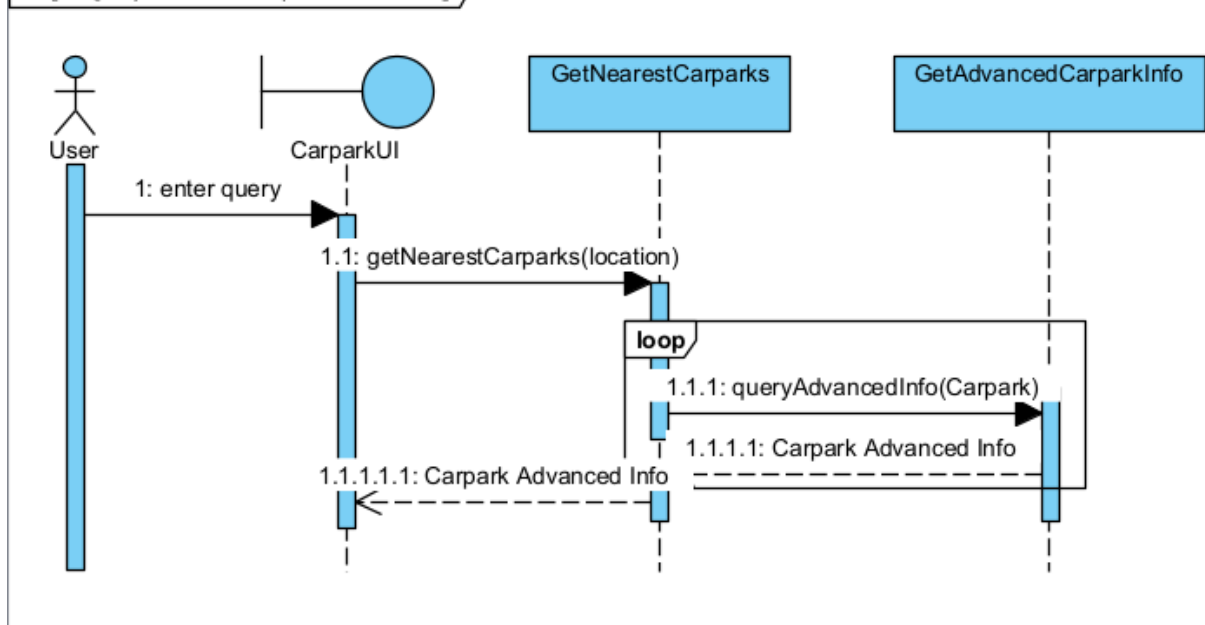
Use Case ID:	2		
Use Case Name:	Query Basic Carpark Information		
Created By:	Tan Chuan Liang	Last Updated By:	Tan Chuan Liang
Date Created:	1/28/2023	Date Last Updated:	8/2/2023
Actor:	User (Initiating Actor)		
Description:	Query basic carpark information from stored carpark data		
Preconditions:	1. Request is made for basic information of a list of car parks		
Postconditions:	1. Basic information of a list of car parks is returned		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. Use Case #1 "Search for Carparks" requests the basic information of a list of car parks. 2. System must find the data of the list of car parks given using Use Case #4 "Check And/Or Update Data". 3. System must extract the basic information of each carpark given. 4. System returns the basic carpark information of the list of car parks given. 		
Alternative Flows:	-		
Exceptions:	EX-2: Basic data of at least 1 carpark not returned by Use Case #4 "Check And/Or Update Data" <ol style="list-style-type: none"> 1. System displays error message "Carpark data not complete, some fields may be blank". 2. Return null values for basic data that is not found 		
Includes:	#4 Check And/Or Update Data		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	-		

sd [1. Search for Carparks; 2. Query Carparks Basic Information]



Use Case ID:	3		
Use Case Name:	Query Advanced Carpark Information		
Created By:	Tan Chuan Liang	Created By:	Tan Chuan Liang
Date Created:	8/2/2023	Date Created:	8/2/2023
Actor:	User (Initiating Actor)		
Description:	Query basic carpark information from stored carpark data		
Preconditions:	1. Request is made for advanced information of a list of carpark		
Postconditions:	1. Advanced information of a list of carpark is returned		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. Use Case #1 "Search for Carparks" requests the advanced information of a list of carpark. 2. System must find the data of the list of carpark given using Use Case #4 "Check And/Or Update Data". 3. System must extract the advanced information of each carpark given. 4. System returns the advanced carpark information of the list of carpark given. 		
Alternative Flows:	1. -		
Exceptions:	EX-2: Advanced data of at least 1 carpark not returned by Use Case #4 "Check And/Or Update Data" <ol style="list-style-type: none"> 1. System displays error message "Carpark data not complete, some fields may be blank". 2. Return null values for advanced data that is not found 		
Includes:	#4 Check And/Or Update Data		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	-		

sd [3. Query Advanced Carpark Information]



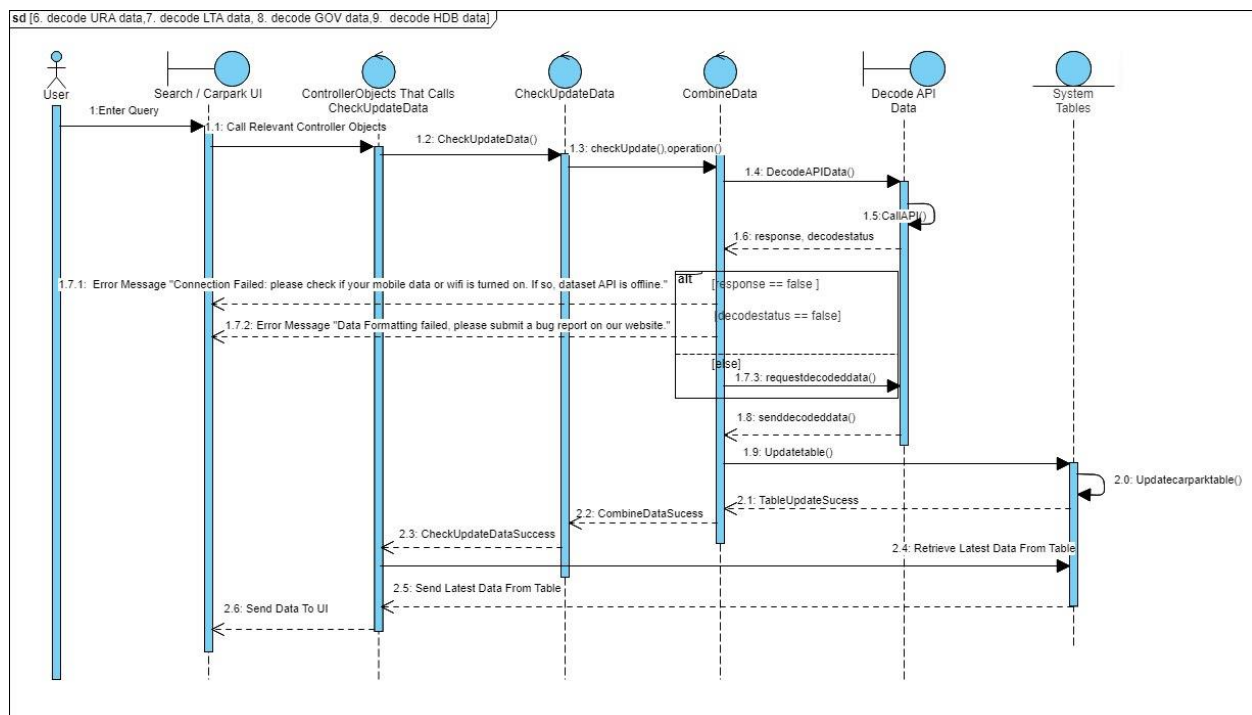
Use Case ID:	4		
Use Case Name:	Check And/Or Update Data		
Created By:	Tan Chuan Liang	Created By:	Tan Chuan Liang
Date Created:	8/2/2023	Date Created:	8/2/2023
Actor:	User (Initiating Actor)		
Description:	Check the cached data on carpark information and update the information if it is more than 1 minute old		
Preconditions:	1. Request is made for newest carpark data		
Postconditions:	1. Newest carpark data is returned		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. Use Case #2 "Query Basic Carpark Information" or Use Case #3 "Query Advanced Carpark Information" requests newest carpark data. 2. System must check the age of carpark data currently stored in the system. 3. If the carpark data currently stored is older than 1 minute, the system must use Use Case #5 "Combine Data" to request and store an updated copy of carpark data. 4. System returns the newest carpark data. 		
Alternative Flows:	-		
Exceptions:	<p>EX-3: Latest data is not returned by Use Case #5 "Combine Data"</p> <ol style="list-style-type: none"> 1. System displays error message "Caution, carpark data is not up-to-date". 2. Return null values for carpark data that is not returned by Use Case #5 "Combine Data" 		
Includes:	#5 Combine Data		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	-		

Use Case ID:	5		
Use Case Name:	Combine Data		
Created By:	Tan Chuan Liang	Created By:	Tan Chuan Liang
Date Created:	8/2/2023	Date Created:	8/2/2023
Actor:	User (Initiating Actor)		
Description:	Combine data provided by APIs and static dataset.		
Preconditions:	1. Request is made to create a new compilation of carpark data		
Postconditions:	1. New compilation of carpark data is returned		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	1. Use Case #4 "Check And/Or Update Data" requests updated copy of carpark data.		

	<ol style="list-style-type: none"> 2. System must request for decoded carpark data from Use Cases #6 Decode URA, #7 Decode LTA, #8 Decode GOV, #9 Decode HDB 3. System must compile all received data into a single data structure that can be queried according to carpark, and contains all data fields described in Carpark Basic Data and Carpark Advanced Data 4. System must return the compilation of carpark data.
Alternative Flows:	-
Exceptions:	<p>EX-2: Latest decoded carpark data is not returned from Use Cases #6 Decode URA, #7 Decode LTA, #8 Decode GOV, and/or #9 Decode HDB System</p> <ol style="list-style-type: none"> 1. Return null values for carpark data that is not returned by Use Case #6, #7, #8, and/or #9
Includes:	<p>#6 Decode URA #7 Decode LTA #8 Decode GOV #9 Decode HDB System</p>
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	decode URA data		
Created By:	Lim Wei Zi	Last Updated By:	Lim Wei Zi
Date Created:	7/2/2023	Date Last Updated:	8/2/2023
Actor:	System, User, URA Carpark API		
Description:	Processing URA Carpark API response from 'Combine Data' use case's requests when User searches for carparks.		
Preconditions:	<ol style="list-style-type: none">1. App must have an Internet Connection.2. API must be online.3. User must have logged in.4. User has called one of the following use cases:<ol style="list-style-type: none">1. 'Search For Carpark'2. 'Query Advanced Carpark Information'3. 'Query Carpark Usage Graph'		
Postconditions:	<ol style="list-style-type: none">1. The URA Carpark API response is successfully retrieved, formatted as required and passed on to the 'Combine Data' use case.		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none">1. User calls one of the use cases under 'Preconditions'.		

	<ol style="list-style-type: none"> The 'Combine Data' use case is called at the end. 'Combine Data' calls 'decode URA' for the data it needs. 'decode URA data' sends request to URA Carpark API. URA Carpark API responds with requested data. 'decode URA data' reformats data from URA Carpark API into a format acceptable to 'Combine Data'. 'decode URA data' then passes formatted data to 'Combine Data'.
Alternative Flows:	
Exceptions:	<p>EX-S5: Dataset does not respond to request.</p> <ol style="list-style-type: none"> System displays error message "Connection failed, please check if your mobile data or Wifi is turned on. If so, Dataset API is offline." <p>EX-S6: Use case is unable to correctly reformat data.</p> <ol style="list-style-type: none"> System displays error message "data formatting failed, please submit a bug report to our website." Return to step 1.
Includes:	1. Combine Data
Special Requirements:	-
Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-



Use Case ID:	7
Use Case Name:	decode LTA data

Created By:	Lim Wei Zi	Last Updated By:	Lim Wei Zi
Date Created:	7/2/2023	Date Last Updated:	8/2/2023
Actor:	System, User, LTA Carpark API		
Description:	Processing LTA Carpark API response from 'Combine Data' use case's requests when User searches for carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. API must be online. 3. User must have logged in. 4. User has called one of the following use cases: <ol style="list-style-type: none"> 1. 'Search For Carpark' 2. 'Query Advanced Carpark Information' 3. 'Query Carpark Usage Graph' 		
Postconditions:	<ol style="list-style-type: none"> 1. The LTA Carpark API response is successfully retrieved, formatted as required and passed on to the 'Combine Data' use case. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls one of the use cases under 'Preconditions'. 2. The 'Combine Data' use case is called at the end. 3. 'Combine Data' calls 'decode LTA' for the data it needs. 4. 'decode LTA data' sends request to LTA Carpark API. 5. LTA Carpark API responds with requested data. 6. 'decode LTA data' reformats data from LTA Carpark API into a format acceptable to 'Combine Data'. 7. 'decode LTA data' then passes formatted data to 'Combine Data'. 		
Alternative Flows:			
Exceptions:	<p>EX-S5: Dataset does not respond to request.</p> <ol style="list-style-type: none"> 1. System displays error message "Connection failed, please check if your mobile data or Wifi is turned on. If so, Dataset API is offline." <p>EX-S6: Use case is unable to correctly reformat data.</p> <ol style="list-style-type: none"> 1. System displays error message "data formatting failed, please submit a bug report to our website." 2. Return to step 1. 		
Includes:	<ol style="list-style-type: none"> 1. Combine Data 		
Special Requirements:	-		
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Data from API accurately reflects real-time info. 		
Notes and Issues:	-		

Use Case ID:	8
--------------	---

Use Case Name:	decode GOV data		
Created By:	Lim Wei Zi	Last Updated By:	Lim Wei Zi
Date Created:	7/2/2023	Date Last Updated:	8/2/2023
Actor:	System, User, data.gov.sg API		
Description:	Processing data.gov.sg API response from 'Combine Data' use case's requests when User searches for carparks.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. API must be online. 3. User must have logged in. 4. User has called one of the following use cases: <ol style="list-style-type: none"> 1. 'Search For Carpark' 2. 'Query Advanced Carpark Information' 3. 'Query Carpark Usage Graph' 		
Postconditions:	<ol style="list-style-type: none"> 1. The data.gov.sg API response is successfully retrieved, formatted as required and passed on to the 'Combine Data' use case. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls one of the use cases under 'Preconditions'. 2. The 'Combine Data' use case is called at the end. 3. 'Combine Data' calls 'decode GOV' for the data it needs. 4. 'decode GOV data' sends request to data.gov.sg API. 5. data.gov.sg API responds with requested data. 6. 'decode GOV data' reformats data from data.gov.sg API into a format acceptable to 'Combine Data'. 7. 'decode GOV data' then passes formatted data to 'Combine Data'. 		
Alternative Flows:			
Exceptions:	<p>EX-S5: Dataset does not respond to request.</p> <ol style="list-style-type: none"> 1. System displays error message "Connection failed, please check if your mobile data or Wifi is turned on. If so, Dataset API is offline." <p>EX-S6: Use case is unable to correctly reformat data.</p> <ol style="list-style-type: none"> 1. System displays error message "data formatting failed, please submit a bug report to our website." 2. Return to step 1. 		
Includes:	<ol style="list-style-type: none"> 1. Combine Data 		
Special Requirements:	-		
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Data from API accurately reflects real-time info. 		
Notes and Issues:	-		

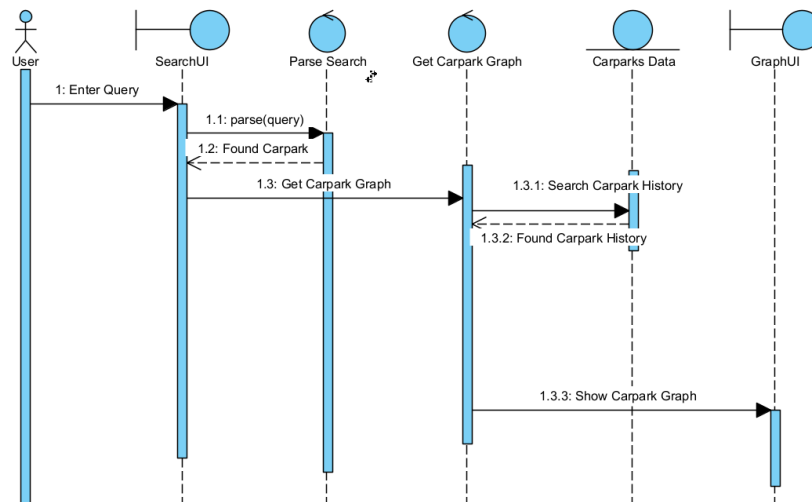
Use Case ID:	9
--------------	---

Use Case Name:	decode HDB data		
Created By:	Lim Wei Zi	Last Updated By:	Lim Wei Zi
Date Created:	7/2/2023	Date Last Updated:	8/2/2023
Actor:	System, User, HDB Carpark Dataset		
Description:	Processing HDB Carpark Dataset response from 'Combine Data' use case's requests when User searches for carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. API must be online. 3. User must have logged in. 4. User has called one of the following use cases: <ol style="list-style-type: none"> 1. 'Search For Carpark' 2. 'Query Advanced Carpark Information' 3. 'Query Carpark Usage Graph' 		
Postconditions:	<ol style="list-style-type: none"> 1. The HDB Carpark Dataset is successfully retrieved, formatted as required and passed on to the 'Combine Data' use case. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls one of the use cases under 'Preconditions'. 2. The 'Combine Data' use case is called at the end. 3. 'Combine Data' calls 'decode HDB' for the data it needs. 4. 'decode HDB data' sends request to HDB Carpark Dataset. 5. HDB Carpark Dataset responds with requested data. 6. 'decode HDB data' reformats data from HDB Carpark Dataset into a format acceptable to 'Combine Data'. 7. 'decode HDB data' then passes formatted data to 'Combine Data'. 		
Alternative Flows:			
Exceptions:	<p>EX-S5: Dataset does not respond to request.</p> <ol style="list-style-type: none"> 1. System displays error message "Connection failed, please check if your mobile data or Wifi is turned on. If so, Dataset API is offline." <p>EX-S6: Use case is unable to correctly reformat data.</p> <ol style="list-style-type: none"> 1. System displays error message "data formatting failed, please submit a bug report to our website." 2. Return to step 1. 		
Includes:	<ol style="list-style-type: none"> 1. Combine Data 		
Special Requirements:	-		
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Data from dataset accurately reflects real-time info. 		
Notes and Issues:	-		

Use Case ID:	10		
Use Case Name:	Query Graph Information		
Created By:	Joshua Adrian Cahyono	Last Updated By:	Joshua Adrian Cahyono
Date Created:	7/2/2023	Date Last Updated:	7/2/2023

Actor:	User, System, HDB Carpark Dataset
Description:	Query graph information of the user chosen carpark
Preconditions:	1. App must be connected to WiFi/Mobile Data
Postconditions:	1. Graph is created from the current data
Priority:	Medium
Frequency of Use:	0-4 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on a specific carpark from the main menu 2. System retrieves the necessary data from the database 3. System creates and displays the graph of the availability of the carpark from the recent 24 hours data to the user. 4. System also shows the trend information with a graph.
Alternative Flows:	-
Exceptions:	<p>EX-1: API does not return graph of the queried carpark</p> <ol style="list-style-type: none"> 1. System displays error message "Server error, try again later". 2. Return to main menu page. <p>EX-AF-S2: Internet connection gets cut off</p> <ol style="list-style-type: none"> 2. System displays error message "No internet connection, please check if your mobile data or Wifi is turned on." 3. Return to step 1.
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. The information is retrieved successfully from the database 2. There is at least 24 hours of information collected by the system
Notes and Issues:	-

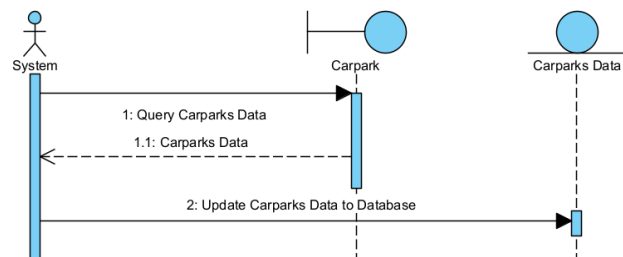
sd [Query Graph Information]



Use Case ID:	11		
Use Case Name:	Store History		
Created By:	Joshua Adrian Cahyono	Last Updated By:	Joshua Adrian Cahyono
Date Created:	7/2/2023	Date Last Updated:	7/2/2023

Actor:	System, URA Carpark API, LTA Carpark API, data.gov.sg API, HDB Carpark Dataset
Description:	System stores the current availability of all the carparks
Preconditions:	<ol style="list-style-type: none"> 1. Server must be running 2. API must be working
Postconditions:	<ol style="list-style-type: none"> 1. Data is stored to the database
Priority:	Medium
Frequency of Use:	10-12 times per hour
Flow of Events:	<ol style="list-style-type: none"> 1. System automatically requests API data of every carpark every 5 minutes. 2. API returns Basic availability of all the carparks 3. System stores the data to the server database
Alternative Flows:	-
Exceptions:	EX-1: Server is not running <ol style="list-style-type: none"> 1. No API will be called
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. The APIs are working properly 2. The information is retrieved successfully from the database
Notes and Issues:	-

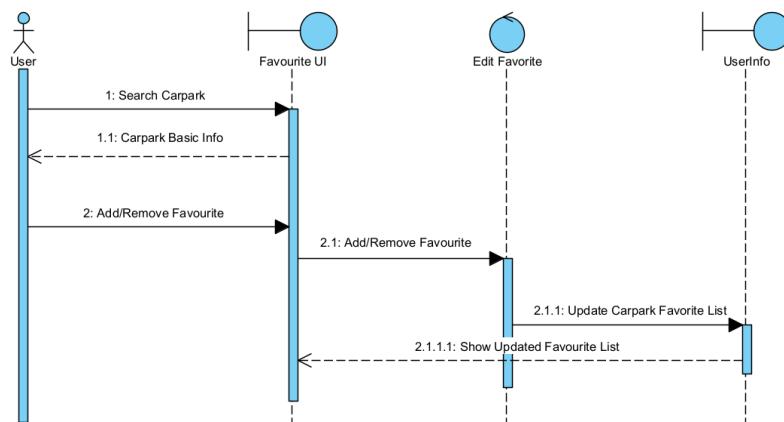
sd [Store History]



Use Case ID:	12		
Use Case Name:	Add/Remove Favorite List		
Created By:	Joshua Adrian Cahyono	Last Updated By:	Joshua Adrian Cahyono
Date Created:	7/2/2023	Date Last Updated:	7/2/2023
Actor:	User, System		
Description:	User can add or remove carparks to their favorite list. The chosen carparks will be prioritized when the user queries.		
Preconditions:	<ol style="list-style-type: none"> 1. App must be connected to WiFi/Mobile Data 2. User must be logged in and authorized 		
Postconditions:	<ol style="list-style-type: none"> 1. Favorite list will be updated 		
Priority:	Low		
Frequency of Use:	0-5 times per hour		

Flow of Events:	<ol style="list-style-type: none"> 1. User goes to the favorite tab 2. User search the carpark to remove it from the list 3. System will update the favorite list for that user
Alternative Flows:	<p>AF-S1: User taps “Back” button while system is searching.</p> <ol style="list-style-type: none"> 2. Return to step 1. <p>AF-S2b: Search query cannot be resolved to a carpark.</p> <ol style="list-style-type: none"> 2. System returns error message. <p>AF-S4: User taps “Next Page” button</p> <ol style="list-style-type: none"> 2. If there are more carpark listings, system repopulates search listing with next 10 (or other preset value) carpark listings, continue from step 3. <p>AF-S4: User taps “Previous Page” button</p> <ol style="list-style-type: none"> 2. System repopulates search listing with previous 10 (or other preset value) carpark listings, continue from step 3.
Exceptions:	<p>EX-AF-S1: Internet connection gets cut off</p> <ol style="list-style-type: none"> 1. System displays error message “No internet connection, please check if your mobile data or Wifi is turned on.” 2. Return to step 1. <p>EX-AF-S2: User is not logged in</p> <ol style="list-style-type: none"> 1. System displays message “You are not logged in yet” 2. System redirects the user to login page
Includes:	1. Update User Personalization
Special Requirements:	-
Assumptions:	1. Carpark list is retrieved successfully
Notes and Issues:	-

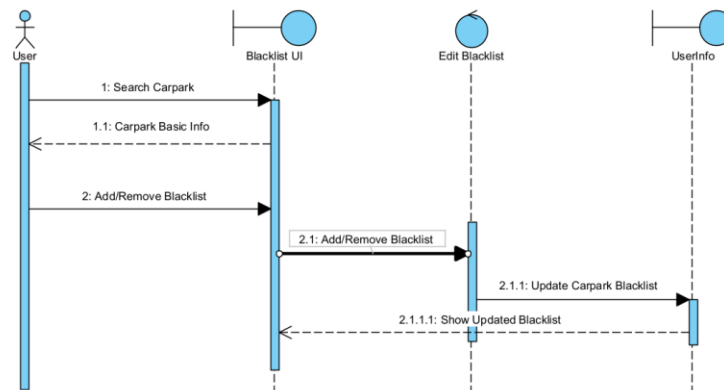
sd [Add/Remove Favourite List]



Use Case ID:	13		
Use Case Name:	Add/Remove Blacklist		
Created By:	Joshua Adrian Cahyono	Last Updated By:	Joshua Adrian Cahyono
Date Created:	7/2/2023	Date Last Updated:	7/2/2023
Actor:	User, System		

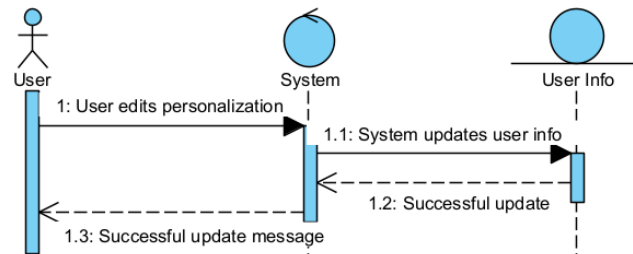
Description:	User can add or remove carpark to their blacklist. The chosen carpark will be hidden when the user queries.
Preconditions:	<ol style="list-style-type: none"> 1. App must be connected to WiFi/Mobile Data 2. User must be logged in and authorized
Postconditions:	<ol style="list-style-type: none"> 1. Blacklist will be updated
Priority:	Low
Frequency of Use:	0-5 times per hour
Flow of Events:	<ol style="list-style-type: none"> 1. User goes to the blacklist tab 2. User search the carpark to add/remove it from the list 3. System will update the blacklist for that user
Alternative Flows:	<p>AF-S1: User taps "Back" button while system is searching.</p> <ol style="list-style-type: none"> 3. Return to step 1. <p>AF-S2a: User does not enter search query into search bar and commences search.</p> <ol style="list-style-type: none"> 2. System assumes queried location to be current location. <p>AF-S2b: Search query cannot be resolved to a location.</p> <ol style="list-style-type: none"> 3. System assumes queried location to be current location. <p>AF-S4: User taps "Next Page" button</p> <ol style="list-style-type: none"> 3. If there are more carpark listings, system repopulates search listing with next 10 (or other preset value) carpark listings, continue from step 3. <p>AF-S4: User taps "Previous Page" button</p> <ol style="list-style-type: none"> 3. System repopulates search listing with previous 10 (or other preset value) carpark listings, continue from step 3.
Exceptions:	<p>EX-AF-S1: Internet connection gets cut off</p> <ol style="list-style-type: none"> 1. System displays error message "No internet connection, please check if your mobile data or Wifi is turned on." 2. Return to step 1. <p>EX-AF-S2: User is not logged in</p> <ol style="list-style-type: none"> 1. System displays message "You are not logged in yet" 2. System redirects the user to login page
Includes:	<ol style="list-style-type: none"> 1. Update User Personalization
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. Carpark list is retrieved successfully
Notes and Issues:	-

sd [Add/Remove Blacklist]



Use Case ID:	14		
Use Case Name:	Update User Personalization		
Created By:	Joshua Adrian Cahyono	Last Updated By:	Joshua Adrian Cahyono
Date Created:	7/2/2023	Date Last Updated:	7/2/2023
Actor:	System		
Description:	System updates user preferences		
Preconditions:	<ol style="list-style-type: none"> 1. App must be connected to WiFi/Mobile Data 2. User must be logged in and authorized 		
Postconditions:	<ol style="list-style-type: none"> 1. Data will be updated 		
Priority:	Low		
Frequency of Use:	0-5 times per hour		
Flow of Events:	<ol style="list-style-type: none"> 1. System accepts edit request from the user 2. System updates the data in the database 		
Alternative Flows:			
Exceptions:	<p>EX-AF-S1: Internet connection gets cut off</p> <ol style="list-style-type: none"> 1. System displays error message "No internet connection, please check if your mobile data or Wifi is turned on." 2. Return to step 1. <p>EX-AF-S2: User is not logged in</p> <ol style="list-style-type: none"> 1. System displays message "You are not logged in yet" 2. System redirects the user to login page 		
Includes:	-		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	-		

sd [Update User Personalization]



Use Case ID:	15		
Use Case Name:	Display Map		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Displays a map when the users request to display map of showing nearby carparks or when the 'Display Route between Addresses' use case is called when users request for navigation to a carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. OneMap API must be online. 3. User must have logged in. 4. User has called one of the following use cases: <ol style="list-style-type: none"> 1. 'Display Map of Nearby Carparks' 2. 'Request Navigation to Carpark' 		
Postconditions:	<ol style="list-style-type: none"> 1. Map is successfully display on the viewport. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls 'Display Map of Nearby Carparks' or 'Request Navigation to Carpark' 2. 'Display Map' sends request to OneMap API. 3. OneMap API responds with requested data. 4. Map is displayed on the viewport. 		

Alternative Flows:	<p>AF-S1-S4: User taps “Back” button while ‘Display Map’ use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S4: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S4: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	<p>EX-S1: OneMap API does not respond to request.</p> <ol style="list-style-type: none"> 1. Determine if device has an internet connection. 2. If it does not, system displays error message “No internet connection, please check if your mobile data or Wifi is turned on.” 3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

Use Case ID:	16		
Use Case Name:	Convert Coordinates to Address		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Converts coordinates data to addresses when the 'Calculate Route between Addresses' use case is called upon when the user request for navigation to carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 		

	<ol style="list-style-type: none"> 2. OneMap API must be online. 3. User must have logged in. 4. User has called the 'Request Navigation to Carpark' use case.
Postconditions:	<ol style="list-style-type: none"> 1. System has received the address data provided by OneMap API response 2. Address data is being passed to the 'Calculate Route between Addresses' use case.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User calls the 'Request Navigation to Carpark' use case. 2. 'Request Navigation of Carparks' calls upon the 'Calculate Route between Addresses' use case. 3. 'Calculate Route between Addresses' calls upon the 'Convert Coordinates to Address' use case. 4. 'Convert Route between Addresses' sends request to OneMap API. 5. OneMap API responds with requested data. 6. Data is transferred back to 'Calculate Route between Addresses' use case.
Alternative Flows:	<p>AF-S1-S6: User taps "Back" button while 'Convert Coordinates to Address' use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S6: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S6: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	<p>EX-S1: OneMap API does not respond to request.</p> <ol style="list-style-type: none"> 1. Determine if device has an internet connection. 2. If it does not, system displays error message "No internet connection, please check if your mobile data or Wi-Fi is turned on."

	3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	-
Special Requirements:	-
Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

Use Case ID:	17		
Use Case Name:	Calculate Route between Coordinates		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Calculates the routes between coordinates when the 'Calculate Route between Addresses' use case is called upon when the user request for navigation to carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. OneMap API must be online. 3. User must have logged in. 4. User has called the 'Request Navigation to Carpark' use case. 		
Postconditions:	<ol style="list-style-type: none"> 1. System has received the address data provided by OneMap API response 2. Address data is being passed to the 'Calculate Route between Addresses' use case. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls the 'Request Navigation to Carpark' use case. 2. 'Request Navigation of Carparks' calls upon the 'Calculate Route between Addresses' use case. 3. 'Calculate Route between Addresses' calls upon the 'Calculate Route between Coordinates' use case. 		

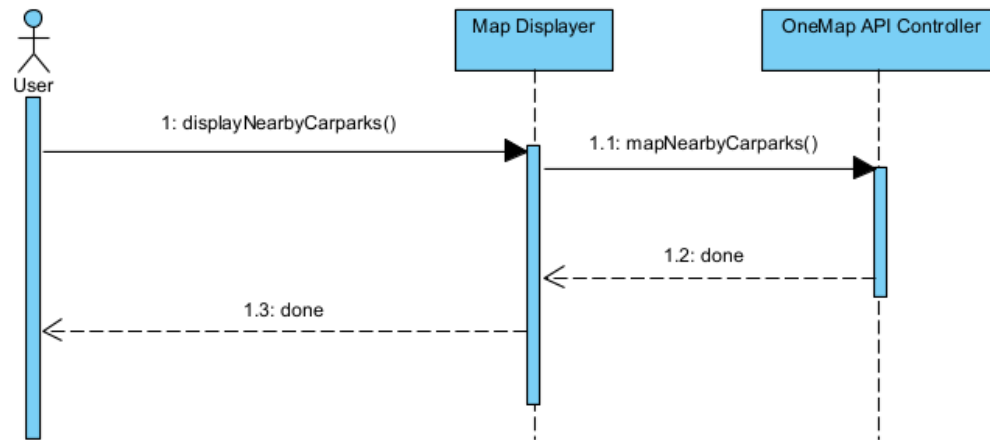
	<ol style="list-style-type: none"> 'Convert Route between Coordinates' sends request to OneMap API. OneMap API responds with requested data. Data is transferred back to 'Calculate Route between Addresses' use case.
Alternative Flows:	<p>AF-S1-S6: User taps "Back" button while 'Calculate Route between Addresses' use-case is not done.</p> <ol style="list-style-type: none"> Cancel any request or wait for response. Return to main menu. <p>AF-S1-S6: User exits App.</p> <ol style="list-style-type: none"> Cancel any request or wait for response. <p>AF-S1-S6: User switches to another App.</p> <ol style="list-style-type: none"> Cancel any request or wait for response. Restart from previous step when user returns.
Exceptions:	<p>EX-S1: OneMap API does not respond to request.</p> <ol style="list-style-type: none"> Determine if device has an internet connection. If it does not, system displays error message "No internet connection, please check if your mobile data or Wi-Fi is turned on." If it does, system displays error message "An error has occurred. Please try again later"
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

Use Case ID:	18		
Use Case Name:	Display Map of Nearby Carparks		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023

Actor:	System, User, OneMap API
Description:	Displays map of carparks nearby target carpark on viewport.
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. OneMap API must be online. 3. User must have logged in. 4. User has clicked on the 'Show nearby Carparks' button.
Postconditions:	<ol style="list-style-type: none"> 1. Map of nearby carparks is displayed on viewport.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the 'Show nearby Carparks' button. 2. 'Display Map of Nearby Carparks' calls upon the 'Display Map' use case. 3. 'Display Map' executes. 4. Map of nearby carparks is displayed on the viewport.
Alternative Flows:	<p>AF-S1-S4: User taps "Back" button while 'Display Map of Nearby Carparks' use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S4: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S4: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	<p>EX-S1: OneMap API does not respond to request.</p> <ol style="list-style-type: none"> 1. Determine if device has an internet connection. 2. If it does not, system displays error message "No internet connection, please check if your mobile data or Wi-Fi is turned on." 3. If it does, system displays error message "An error has occurred. Please try again later"

Includes:	Display Map
Special Requirements:	-
Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

sd [Display Map of Nearby Carparks]



Use Case ID:	19		
Use Case Name:	Display Route between Addresses		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Displays route between user location and target carpark on map on viewport.		
Preconditions:	1. App must have an Internet Connection. 2. OneMap API must be online.		

	<ol style="list-style-type: none"> 3. User must have logged in. 4. User has called upon the 'Request Navigation to Carpark' use case.
Postconditions:	Route between user's current location to target carpark is displayed on map on viewport.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User calls upon the 'Request Navigation to Carpark' use case. 2. 'Request Navigation to Carpark' calls upon the 'Display Route between Addresses' use case. 3. 'Display Route between Addresses' calls upon the 'Display Map' use case. 4. 'Display Map' executes. 5. Map of nearby carpark is displayed on the viewport.
Alternative Flows:	<p>AF-S1-S5: User taps "Back" button while 'Display Route between Addresses' use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S5: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S5: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	<p>EX-S4: OneMap API does not respond to request.</p> <ol style="list-style-type: none"> 1. Determine if device has an internet connection. 2. If it does not, system displays error message "No internet connection, please check if your mobile data or Wi-Fi is turned on." 3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	Display Map
Special Requirements:	-

Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

Use Case ID:	20		
Use Case Name:	Calculate Route between Addresses		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Calculates the optimal route between user's current location and the target carpark.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. OneMap API must be online. 3. User must have logged in. 4. User has called upon the 'Request Navigation to Carpark' use case. 		
Postconditions:	<ol style="list-style-type: none"> 1. Optimal route between user and target carpark, estimate travel time, and addresses' name is obtained. 2. Data is passed to 'Request Navigation to Carpark' use case. 		
Priority:	High		
Frequency of Use:	0-10 times per day		
Flow of Events:	<ol style="list-style-type: none"> 1. User calls upon the 'Request Navigation to Carpark' use case. 2. 'Request Navigation to Carpark' calls upon the 'Calculate Route between Addresses' use case. 3. 'Calculate Route between Addresses' calls upon the 'Convert Coordinates to Address' use case. 4. 'Convert Coordinates to Address' executes and return address data. 5. 'Calculate Route between Addresses' calls upon 'Calculate Route between Coordinates' use case. 		

	6. Calculate Route between Coordinates executes and return optimal route and estimated travel time. 7. Data is passed back to 'Request Navigation to Carpark' use case.
Alternative Flows:	AF-S1-S7: User taps “Back” button while ‘Calculate Route between Addresses’ use-case is not done. <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. AF-S1-S7: User exits App. <ol style="list-style-type: none"> 1. Cancel any request or wait for response. AF-S1-S7: User switches to another App. <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	EX-S4/S6: OneMap API does not respond to request. <ol style="list-style-type: none"> 1. Determine if device has an internet connection. 2. If it does not, system displays error message “No internet connection, please check if your mobile data or Wifi is turned on.” 3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	Convert Coordinates to Address, Calculate Route between Coordinates
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-

Use Case ID:	21		
Use Case Name:	Query Current Location		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		

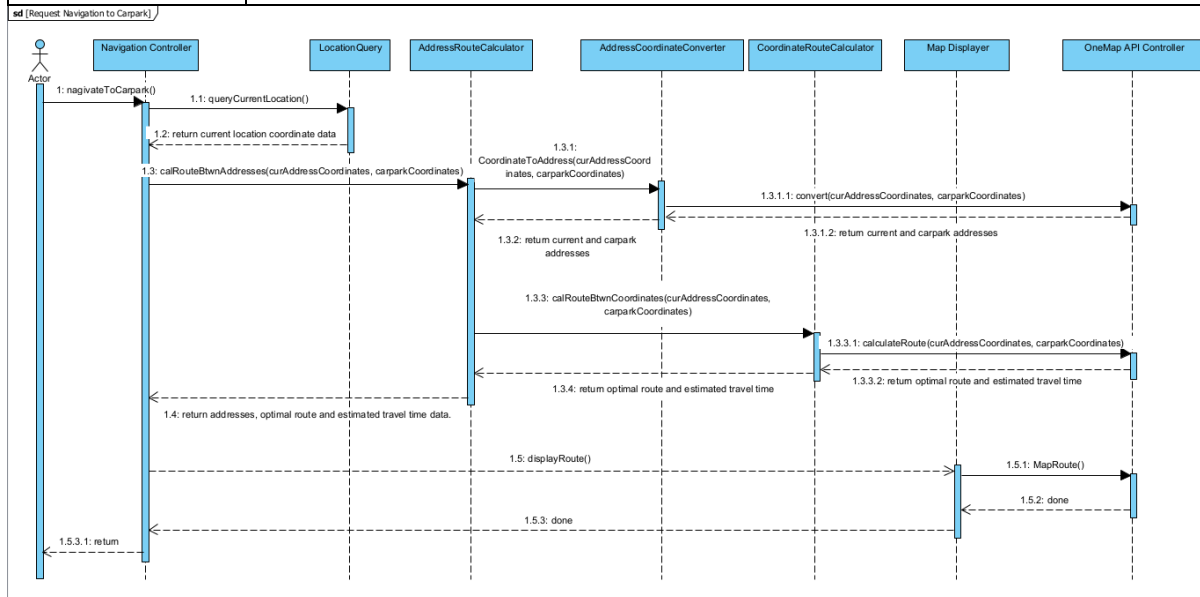
Description:	Gets user's coordinates data from user system.
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. User must have logged in. 3. User has called upon the 'Request Navigation to Carpark' use case.
Postconditions:	<ol style="list-style-type: none"> 1. User's coordinates data is obtained. 2. User's coordinates data is passed to 'Request Navigation to Carpark' use case.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	<ol style="list-style-type: none"> 1. User calls upon the 'Request Navigation to Carpark' use case. 2. 'Request Navigation to Carpark' calls upon the 'Query Current Location' use case. 3. Check if GPS is turned on at user's system. 4. Get user's current coordinate data from system. 5. Return user's current coordinates data back to 'Request Navigation to Carpark' use case.
Alternative Flows:	<p>AF-S1-S5: User taps "Back" button while 'Query Current Location' use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S5: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S5: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns. <p>AF-S3: GPS is not turned on:</p> <ol style="list-style-type: none"> 1. Display message to prompt user to turn on GPS 2. Wait for user to click on 'Yes' or 'No' 3. If user clicked on 'No', return to main menu.

	4. If user clicked on 'Yes, turn on GPS automatically and continue step 4.
Exceptions:	EX-S4: Unable to get user's current coordinates data: <ol style="list-style-type: none"> 1. Determine if device has an internet connection and GPS is turned on. 2. If it does not, system displays error message "No internet connection, please check if your mobile data or Wi-Fi is turned on." 3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	-
Special Requirements:	-
Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-


Use Case ID:	22		
Use Case Name:	Request Navigation to Carpark		
Created By:	Kenny Seah Yong Jie	Last Updated By:	Kenny Seah Yong Jie
Date Created:	9/2/2023	Date Last Updated:	9/2/2023
Actor:	System, User, OneMap API		
Description:	Displays optimal route between user's current location and target carpark, and estimated travel time.		
Preconditions:	<ol style="list-style-type: none"> 1. App must have an Internet Connection. 2. OneMap API must be online. 3. User must have logged in. 4. User has clicked on the 'Navigate to Carpark' button. 		
Postconditions:	Optimal route between user's current location and target carpark, and estimated travel time is displayed on viewport.		
Priority:	High		
Frequency of Use:	0-10 times per day		

Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on 'Navigate to Carpark' button. 2. 'Request Navigation to Carpark' calls upon 'Query Current Location' use case. 3. 'Query Current Location' executes and returns user's coordinates data. 4. 'Request Navigation to Carpark' calls upon 'Calculate Route between Addresses' use case. 5. 'Calculate Route between Addresses' executes and returns optimal route and estimated travel time data. 6. 'Request Navigation to Carpark' calls upon 'Display Route between Addresses' use case. 7. 'Display Route between Addresses' executes and displays map. 8. Estimated travel time is printed on the screen.
Alternative Flows:	<p>AF-S1-S8: User taps "Back" button while 'Request Navigation to Carpark' use-case is not done.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Return to main menu. <p>AF-S1-S8: User exits App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. <p>AF-S1-S8: User switches to another App.</p> <ol style="list-style-type: none"> 1. Cancel any request or wait for response. 2. Restart from previous step when user returns.
Exceptions:	<p>EX-S3/S5/S7: Included use case fails to execute to completion:</p> <ol style="list-style-type: none"> 1. Determine if device has an internet connection and GPS is turned on. 2. If it does not, system displays error message "No internet connection, please check if your mobile data or Wifi is turned on." 3. If it does, system displays error message "An error has occurred. Please try again later"
Includes:	Query Current Address, Display Route between Addresses, Calculate Route between Addresses
Special Requirements:	-

Assumptions:	1. Real-time Data from API accurately reflects real-time info.
Notes and Issues:	-



UI Mockups



ParkWhere?

Search Box:

Start Search!

Favorites

Blacklist

?

User Guide:

- Hitting the 'Start Search!' Button without info in the search box will run a search with your present GPS-based location.
- Click on a row in the results list for advanced info on the carpark.
- For address info in search box, it can be in the format of a:
 - A postal code.
 - Full/Partial address.
 - Carpark number

<- Return

Advanced Info

Info:	Results:
Carpark Rate:	
Parking System (Electronic/Co upon)	
Short Term Parking Times	
Free Parking Availability	
Night Parking Availability	
HDB/URA/LTA affiliation	

<- Return

Add to favorites

Add to blacklist

Advanced Info

Info:	Results:
Carpark Rate:	
Parking System (Electronic/Co upon)	
Short Term Parking Times	
Free Parking Availability	
Night Parking Availability	
HDB/URA/LTA affiliation	

<- Return

Remove from favorites

Advanced Info

Info:	Results:
Carpark Rate:	
Parking System (Electronic/Co upon)	
Short Term Parking Times	
Free Parking Availability	
Night Parking Availability	
HDB/URA/LTA affiliation	

<- Return

Remove from blacklist

Testing

Black Box Testing (Search for Carparks)

Manual testing is done due to a large number of distinct inputs. Test cases are not numerical and cannot be tested sequentially using automated testing.

Test Input	Input description	Expected Output	Actual Output
Singapore 099253	Test for Postal Code checking	Carparks near Harbourfront e.g. VivoCity, Seah Im	<ul style="list-style-type: none">• Harbfront Harbourfront Centre• Harbfront VivoCity P2• MARITIME SQUARE D OFF STREET• Harbfront VivoCity P3• SEAH IM ROAD OFF STREET• BLK 42/43 TELOK BLANGAH RISE
Singapore 258507	Test for Postal Code checking	Carparks near Ministry of Foreign Affairs e.g. Dawson Road, west end of Orchard	<ul style="list-style-type: none">• BLK 95A DAWSON ROAD• BLK 89A DAWSON ROAD• ANGULLIA PARK OFF STREET• Orchard Wheelock Place• BLK 129A CLARENCE LANE
Jurong Point	Test for Partial address	Carparks near Jurong Point i.e. Jurong West	<ul style="list-style-type: none">• BLK 664 JURONG WEST ST 64• BLK 686 JURONG WEST CENTRAL 1• BLK 692A JURONG WEST CENTRAL 1• BLK 691A JURONG WEST CENTRAL 1• BLK 663 JURONG WEST ST 65
313 Somerset	Test for Partial address	Carparks near 313@Somerset i.e. Orchard	<ul style="list-style-type: none">• Orchard 313@Somerset• Orchard Orchard Gateway• Orchard Orchard Central• Orchard Centrepont• Orchard The Heeren
Hougang Street 21	Test for Full address	Carparks along Hougang Street 21 and nearby roads	<ul style="list-style-type: none">• BLK 208-210 HOUGANG STREET 21• BLK 229 HOUGANG STREET 21

			<ul style="list-style-type: none"> • BLK 231/233/235/237 HOUGANG STREET 21 • BLK 201-204 HOUGANG STREET 21 • BLK 230/232/234/236/238 HOUGANG AVENUE 1
Yishun Ave 9	Test for Full address	Carparks along Yishun Ave 9 and nearby roads	<ul style="list-style-type: none"> • BLK 317 YISHUN AVE 9 • BLK 243/257 YISHUN RING ROAD/ AVENUE 9 • BLK 315 YISHUN AVE 9 • BLK 334 YISHUN ST 31 • BLK 258/264 YISHUN STREET 22

Black box testing (Favourites & Blacklist)

Test Input	Input description	Expected Output	Actual Output
<ul style="list-style-type: none"> - Log in to system - Add "ALJUNIED ROAD OFF STREET" to Favourites list - Search for "ALJUNIED ROAD OFF STREET" 	Test the highlighting of favoured car parks	Star emoji next to favoured car park appears	Star emoji next to "ALJUNIED ROAD OFF STREET" car park entry
<ul style="list-style-type: none"> - Log in to system - Add "DICKSON ROAD OFF STREET" to Favourites list - Search for "DICKSON ROAD OFF STREET" 	Test the hiding of blacklisted car parks	Blacklisted car park is not shown	"DICKSON ROAD OFF STREET" car park entry is not shown

White box testing

White box testing is done on user login and register functions. The basis paths for these functions are well-defined so they lend well to white box testing.

Test Input	Input description	Expected Output	Actual Output
Basis path: <ul style="list-style-type: none"> - Register new user "user0" with password "password" - Login user "user0" with password "password" 	A new user is registered and immediately logged in with correct username and password. Log in status is checked.	Logged in successfully. User profile page shows details.	Logged in successfully. User profile page shows details.

- Navigate to user profile page to ensure login			
Alternative path 1: - Register new user "user1" with password "password" - Login user "wrong" with password "password" - Navigate to user profile page to ensure login	A new user is registered but login uses wrong username. Log in status is checked	Log in failed due to wrong username, user profile page does not show, prompting login.	User profile page does not show logged in user, prompting login.
Alternative path 2: - Register new user "user2" with password "password" - Login user "user2" with password "wrong" - Navigate to user profile page to ensure login	A new user is registered but login uses wrong password. Log in status is checked	Log in failed due to wrong password, user profile page does not show, prompting login.	User profile page does not show logged in user, prompting login.
Alternative path 3: - Without logging in, navigate to user profile page	No user is registered or logged in	User profile page does not show logged in user, prompting login.	User profile page does not show logged in user, prompting login.
Alternative path 4: - Log in a pre-set user "admin" with password "admin" - Navigate to user profile page to ensure login	Without using the registration feature, log in to the pre-set admin account with correct password. Log in status is checked.	Logged in successfully. User profile page shows details.	Logged in successfully. User profile page shows details.
Alternative path 5: - Register new user "user3" with password "password"	A new user is registered but log in feature is not used. Server console is used to check If user is successfully registered.	User successfully registered as indicated by server console.	User successfully registered as indicated by server console.

Demo Script

Presentation Flow:

- Expected Users
- Main Functionalities
- Use Case Diagram

- Highlighted Use Case: Search for car parks (class & sequence diagrams)
- Testing
- Good Software Engineering Practices
- Live Demo