



Software Engineering

Lab 2 (ParkingSlot)

Submitted by:

Tan Jia Wei U1821234H

Wong Chun Foong U1822939L

Goh Jun Le U1820246G

An Zhi Andrew Cai N1902005C

Emmi Mannonen N1902011B

Nurul Sabrina Binte Mohammed Riduwan U1822849A

Functional Requirements

1. The motorist shall be able to sign up for an account
 - 1.1. The motorist must input their username, password, email, contact number and address
 - 1.2. The motorist shall be able to login to his account
 - 1.2.1. If the username and password is valid, the system shall connect the motorist to the database
 - 1.2.2. If the username and password is invalid, the motorist will be denied access to the system
2. Once logged in, the system shall retrieve the motorist user profile.
 - 2.1. The motorist data display must consist of motorist name
 - 2.2. The motorist data display must consist of motorist password
 - 2.3. The motorist data display must consist of motorist email
 - 2.4. The motorist data display must consist of motorist contact number
 - 2.5. The motorist data display must consist of motorist address
3. The motorist shall be able to change his/her account password if needed.
 - 3.1. The motorist must input either his/her email or username
 - 3.1.1. The system shall send a reset password link to the motorist's email address
4. The motorist shall be able to delete his account
5. The system must allow motorist to query the system
 - 5.1.1. The motorist must be able to filter parking cost
 - 5.1.2. The motorist must be able to filter height
 - 5.1.3. The motorist must be able to filter carpark type
 - 5.1.4. The motorist must be able to search for the nearest car parks
 - 5.1.5. The motorist must be able to calculate his parking fees
 - 5.1.6. The motorist shall be able to enter his parking duration
6. The motorist must be able to view information of a carpark.
 - 6.1. The system must display carpark height
 - 6.2. The system must display duration of parking
 - 6.3. The system must display carpark type
 - 6.4. The motorist must be able to save their choice of car park if needed
 - 6.5. System must be able to calculate the estimated cost based on the number of hours
 - 6.6. System must be able to calculate the estimated cost based on the carpark type
 - 6.7. System must be able to calculate the estimated cost based on the date/time
7. The motorist must be able to change his/her settings on when/how he/she would like to be notified
 - 7.1. The motorist must be able to change the frequency of the notification based on hours/days/weeks
8. The system must display information in the correct format.
 - 8.1. The system must display the current time in 24 hours format
 - 8.2. The system must display the motorist's real time location in address
 - 8.3. The system must display real time number of car park lots in lists and map

- 8.4. The system must be able to provide recommendations of other nearby lots availability if the chosen car park is full

Non-functional Requirements

1. Usability
 - 1.1. The system functions must be organized such that the motorist will not have any problem navigating through the application
 - 1.2. Errors or exception messages must be handled to eliminate any code warning messages to be shown to the motorist.
 - 1.3. Error messages must be user friendly and understood by the normal motorist.
 - 1.4. There must be a help button at all pages which is easily identifiable for motorist that are unfamiliar with the application
2. Performance
 - 2.1. System must be able to handle a sudden surge of new users no more than 10,000
 - 2.2. System must be able to provide timely feedback of no more than 2 seconds to guide the motorist through errors
 - 2.3. System must return a response within 2 seconds upon motorist interaction
3. Reliability
 - 3.1. System must be available to motorist at least 99.9% of the time from 7am to 12am
 - 3.2. System must be able to handle a sudden surge of new users no more than 10,000
 - 3.3. After restarting, full system functionality must be restored in less than 5 minutes
 - 3.4. Upon entering the application again after exiting, the motorist previous carpark data must still be intact
 - 3.5. There must be monthly backups for disaster recovery in case the database gets wiped out due to corruption
 - 3.6. 99.9% of the systems failure must be repairable in an hour
4. Security
 - 4.1. System shall encrypt the motorist password upon registration and no plaintext are to be stored in the database
 - 4.2. System must be able to track every motorist login date and time and in addition display the last access date upon sign on
 - 4.3. System shall be able to set the number of login attempts before locking the motorist account
 - 4.4. System shall be able to enforce a certain password requirement
 - 4.5. The website/application and database connections must be secured with encryption so as to prevent either system from getting compromised
 - 4.6. The system must be able to detect denial-of-service (DoS) attacks
5. Supportability
 - 5.1. The website/application shall be able to be deployed on any cloud server such as Amazon Web Services, Azure, HostGator, Dream host and etc.
 - 5.2. The database must be able to migrate to any other commercial products that support standard SQL queries

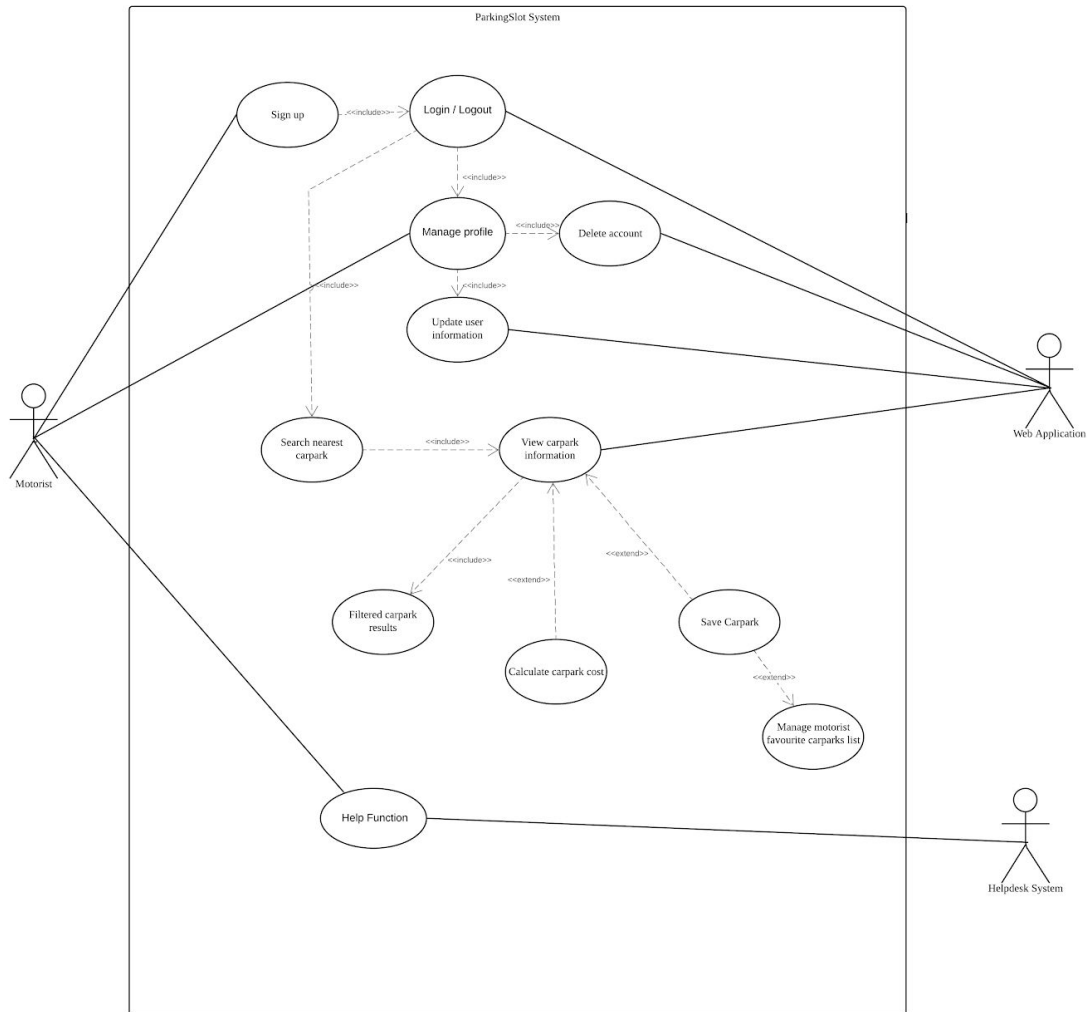
Data Dictionary

Term	Definition
Motorist	Someone who drives a car or other road vehicle
Vehicle	A mode of transporting people or goods, especially on land, such as a motorcycle ,car, lorry or van
Multi Story Carpark	A large building with several floors for parking vehicles in. Commonly found nearby HDB flats
Parking Lot	A dedicated area that is intended for parking vehicles
Car park Availability	The real-time number of available lots in the car park
Underground Carpark	Parking facilities that are built beneath the ground, either in the basement of a building or beneath a street
HDB Carpark	Car parks for Housing Development Board flats which are subsidised public housing for Singaporeans
URA Carpark	Car parks managed by Urban Redevelopment Authority which are outside of HDB estates
LTA Carpark Data	The LTA carpark data consist of major shopping malls and developments within Orchard, Marina, HarbourFront, Jurong Lake District.
Map	A representation of Singapore land showing buildings, cities, roads, car parks and etc

Use Case Diagram

ParkingSlot Use Case Diagram

#CAI AN ZHI ANDREW W | September 17, 2019



Use Case Description

Use Case ID:	001		
Use Case Name:	Sign Up		
Created By:	Chun Foong	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	Motorist registers account
Preconditions:	Motorist does not have the account
Postconditions:	Motorist successfully creates account
Priority:	High
Frequency of Use:	Motorist will only do this once
Flow of Events:	<ol style="list-style-type: none">1. Motorist key in Username, Email, Password, Phone Number2. System verifies Motorist details3. If valid, Motorist able to proceed to use the application
Alternative Flows:	Motorist application invalid if invalid email address: The application will be rejected
Exceptions:	N. A
Includes:	N. A
Special Requirements:	N. A
Assumptions:	N. A
Notes and Issues:	N. A

Use Case ID:	002		
Use Case Name:	Manage Profile		
Created By:	Jia Wei	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	Motorist is able to update their information or delete their account
Preconditions:	Motorist have logged in the application
Postconditions:	Motorist have updated their information or deleted their account
Priority:	Medium
Frequency of Use:	Motorist will use this when they wish to change their profile settings
Flow of Events:	<ol style="list-style-type: none"> 1. Motorist logged into the application 2. Click on Manage profile 3. Once Motorist is in the Manage profile: <ol style="list-style-type: none"> 3.1. Motorist able to update their information 3.2. Motorist able to delete their account
Alternative Flows:	N.A.
Exceptions:	N.A
Includes:	Logged into the application
Special Requirements:	N. A
Assumptions:	N. A
Notes and Issues:	N. A

Use Case ID:	003		
Use Case Name:	Update Favorites		
Created By:	Andrew	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	User can add or delete a carpark on his favorites' list
Preconditions:	1. The motorist must be logged in to the application
Postconditions:	1. The motorist must be able to save or delete his desired carpark into his favorites
Priority:	Medium
Frequency of Use:	Often
Flow of Events:	<ol style="list-style-type: none"> 1. The motorist logs into the application 2. The motorist clicks on 'Update Favorites' 3. The motorist chooses to add carpark or delete carpark by entering Carpark ID or Carpark Address
Alternative Flows:	If the carpark ID or carpark address is not found, display a message "Please find another carpark. ID or Address is unable to be saved into favorites."
Exceptions:	N.A.
Includes:	Login
Special Requirements:	N.A.
Assumptions:	N.A.
Notes and Issues:	N.A.

Use Case ID:	004		
Use Case Name:	Login		
Created By:	Sabrina	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	Motorist logs into the application
Preconditions:	Motorist must have signed up for an account
Postconditions:	Motorist stays logged in the application
Priority:	High
Frequency of Use:	Whenever motorist wants to use the application, he has to login
Flow of Events:	<ol style="list-style-type: none"> 1. Motorist signs up for an account 2. Motorist enters username 3. Motorist enters password 4. Motorist successfully logs in
Alternative Flows:	If the motorist enters wrong username or password, display message "Wrong username/password entered."
Exceptions:	If the motorist enters wrong username or password 5 or more times in a row, display message "Account is locked out for an hour. Please try again later."
Includes:	Sign Up
Special Requirements:	N. A
Assumptions:	N.A.
Notes and Issues:	N. A

Use Case ID:	005		
Use Case Name:	Search For Nearest Carparks		
Created By:	Jun Le	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	The motorist searches for the nearest carparks, and a list of up to 10 nearest carparks is returned
Preconditions:	<ol style="list-style-type: none"> 1. The motorist have logged into the application 2. The motorist have switched on GPS on his phone 3. The motorist must enter his vehicle type 4. The motorist must enter his intended parking duration
Postconditions:	<ol style="list-style-type: none"> 1. The motorist receives a list of up to 10 nearest carparks on his phone or 2. The motorist receives a message that no carparks are found within the parameter
Priority:	High
Frequency of Use:	Often
Flow of Events:	<ol style="list-style-type: none"> 1. The motorist login into the application 2. The motorist enters his vehicle type and parking duration 3. The motorist can choose his filter options by Carpark Rates, Carpark Type, Carpark Height 4. The motorist can choose to use current location or manually select location 5. The motorist presses "Search For Nearest Carparks" button 6. The motorist views a list of up to 10 nearest carparks within motorist's parameters
Alternative Flows:	If the motorist's filter options are invalid The system displays the message "No carparks found. Please refine your filter options."
Exceptions:	N. A
Includes:	Login
Special Requirements:	N. A
Assumptions:	N. A
Notes and Issues:	N. A

Use Case ID:	006		
Use Case Name:	Filter Search		
Created By:	Emmi	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	The motorist filters his search by parameters like Carpark Rates, Carpark Type and Carpark Height
Preconditions:	The motorist needs to be logged in
Postconditions:	The motorist must be able to make a search with the carpark height parameters specified
Priority:	High
Frequency of Use:	Often
Flow of Events:	<ol style="list-style-type: none"> 1. The motorist logs into the application 2. The motorist enters his vehicle type and parking duration 3. The motorist can choose his filter options by Carpark Rates, Carpark Type and Carpark Height 4. The motorist makes a search 5. The motorist must be able to view the results with the specified parking parameters applied
Alternative Flows:	<p>If there are no results found: The system displays the message "No carpark found. Please refine your filter options."</p>
Exceptions:	N. A
Includes:	N. A
Special Requirements:	N. A
Assumptions:	N. A
Notes and Issues:	N. A

Use Case ID:	007		
Use Case Name:	Save Parked Car's Location		
Created By:	Jun Le	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist
Description:	The motorist saves his parked car's location as a reminder for himself
Preconditions:	The motorist must be logged in to the application
Postconditions:	The motorist must be able to successfully save his parked car's location
Priority:	Medium
Frequency of Use:	The motorist could choose to use this after he parks his car
Flow of Events:	<ol style="list-style-type: none"> 1. The motorist logs into the application 2. The motorist saves his parked car's location
Alternative Flows:	N. A
Exceptions:	N. A
Includes:	Login
Special Requirements:	N. A
Assumptions:	N. A
Notes and Issues:	

Use Case ID:	008		
Use Case Name:	Help Function		
Created By:	Chun Foong	Last Updated By:	Jun Le
Date Created:	21/8/2019	Date Last Updated:	12/9/2019

Actor:	Motorist, Help Desk
Description:	Motorist can seek help when they face any problems.
Preconditions:	NA
Postconditions:	Motorist must be able to reach the help desk for help
Priority:	Low
Frequency of Use:	NA
Flow of Events:	<ol style="list-style-type: none"> 1. Motorist face problems 2. Motorist use the help function 3. System received help request and provide necessary help to the Motorist
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	N.A.
Notes and Issues:	NA

Use Case ID:	009		
Use Case Name:	Recover Account		
Created By:	Jia Wei	Last Updated By:	Jun Le
Date Created:	1/9/2019	Date Last Updated:	12/9/2019

Actor:	Web App Administrator
Description:	The administrator can recover deleted user accounts
Preconditions:	The web app administrator must be logged in
Postconditions:	Administrator must be able to successfully recover deleted user accounts
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The web app administrator logs into the application 2. The web app administrator sees a list of requests for recovered accounts 3. The web app administrator approves the requests for recovered accounts
Alternative Flows:	NA
Exceptions:	NA
Includes:	Login
Special Requirements:	NA
Assumptions:	N.A.
Notes and Issues:	NA

Use Case ID:	010		
Use Case Name:	Manage Error Log		
Created By:	Andrew	Last Updated By:	Jun Le
Date Created:	1/9/2019	Date Last Updated:	12/9/2019

Actor:	Web App Administrator
Description:	The administrator can view or delete the errors in the log
Preconditions:	The web app administrator must be logged in
Postconditions:	Administrator must be able to successfully view or delete the errors in the log
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The web app administrator logs into the application 2. The web app administrator views the error log 3. The web app administrator works on debugging the errors 4. The web app administrator deletes the errors after he has solved the errors
Alternative Flows:	NA
Exceptions:	NA
Includes:	Login
Special Requirements:	NA
Assumptions:	N.A.
Notes and Issues:	NA

Use Case ID:	011		
Use Case Name:	Update Carpark Information		
Created By:	Sabrina	Last Updated By:	Jun Le
Date Created:	1/9/2019	Date Last Updated:	12/9/2019

Actor:	Web App Administrator
Description:	The administrator can create or update carpark information in the database
Preconditions:	The web app administrator must be logged in
Postconditions:	The information created or updated must be successfully saved in the database
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The web app administrator logs into the application 2. The web app administrator can choose to create or update carpark information if information has changed in real-time 3. Data changed or created must be saved in the database
Alternative Flows:	NA
Exceptions:	NA
Includes:	Login
Special Requirements:	NA
Assumptions:	N.A.
Notes and Issues:	NA

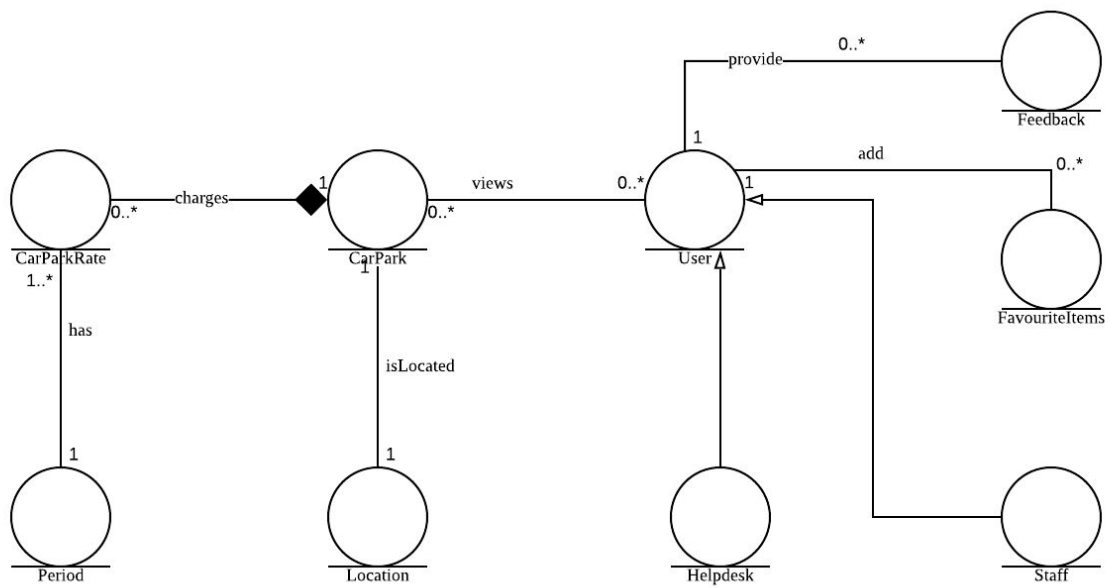
UI Mockups



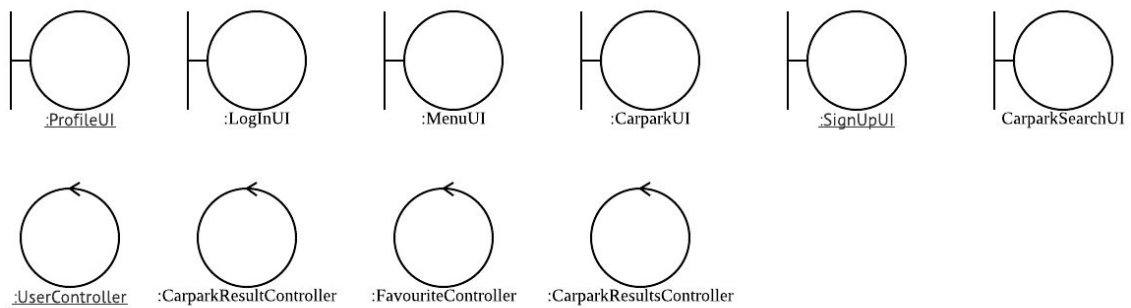
Live prototype:

<https://xd.adobe.com/view/bc8658b1-2ea2-4042-4f2f-23be6080371d-cab7/screen/546b75e4-aea3-4d5e-bc4a-f8c3de59fca1/Web-1920-5?fullscreen>

Class Diagram of entity classes

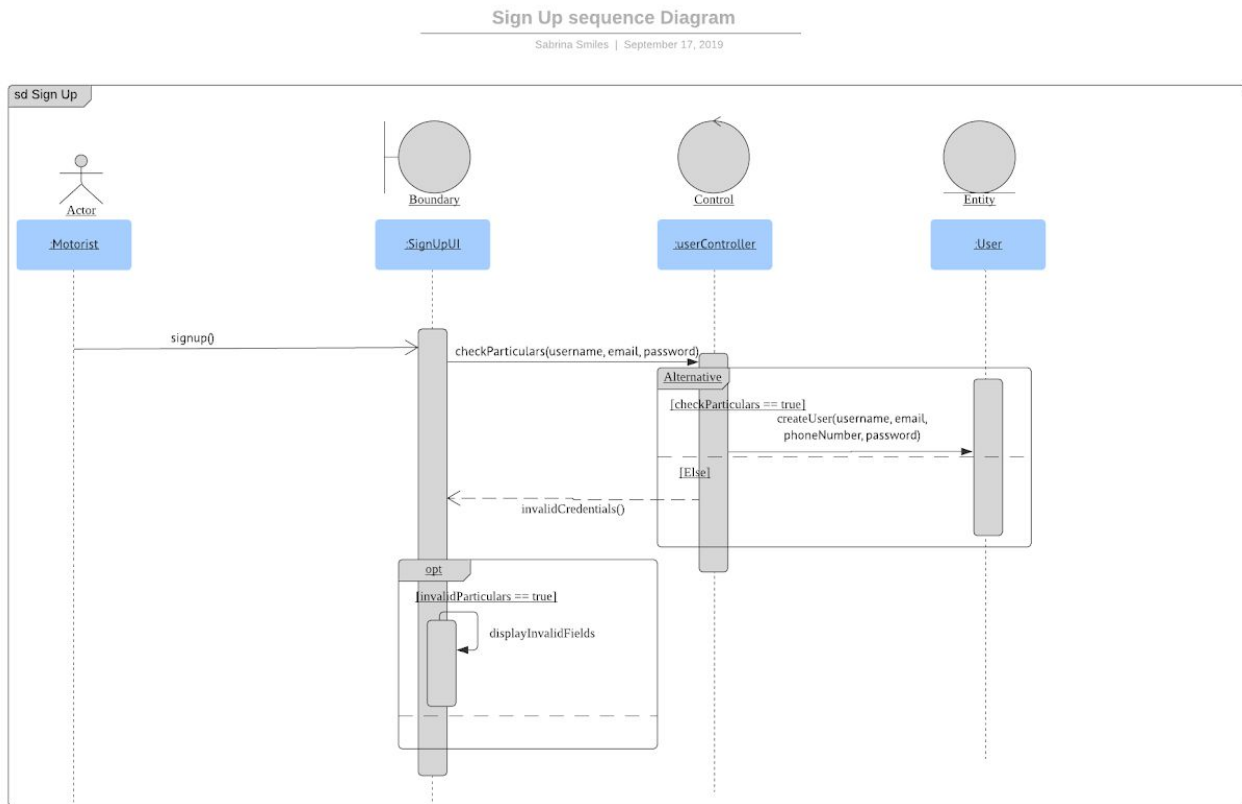


Key boundary classes and control classes

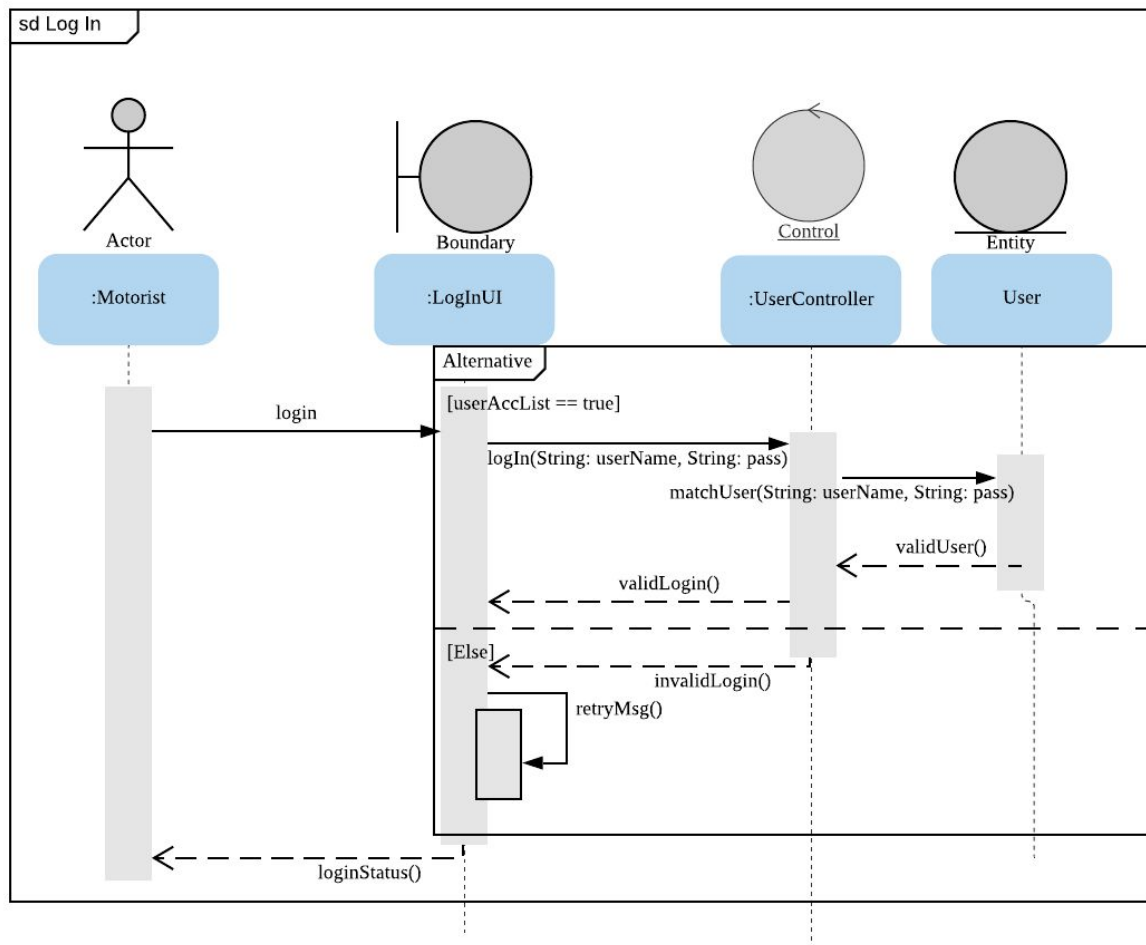


Sequence Diagram

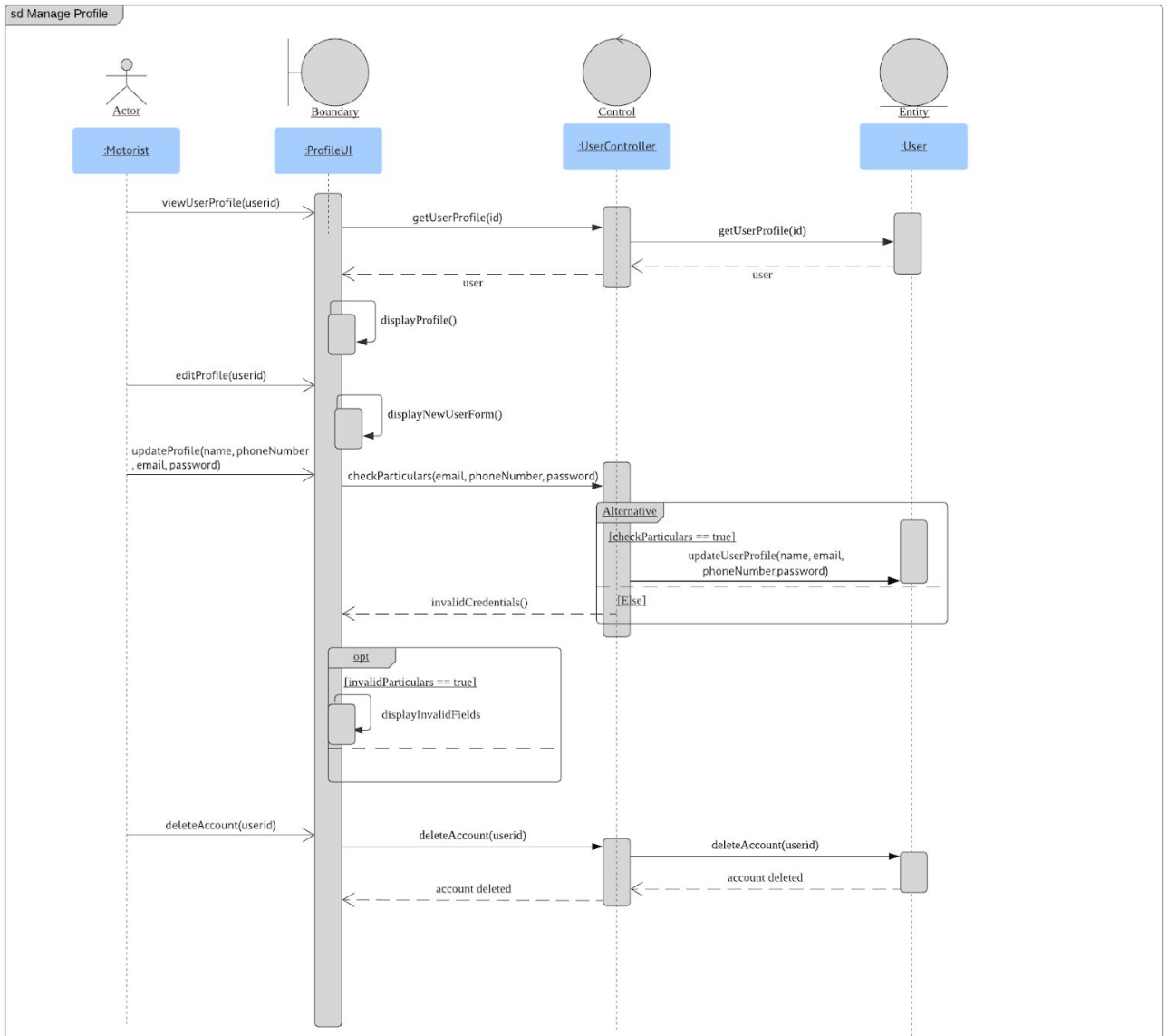
1. Sign up



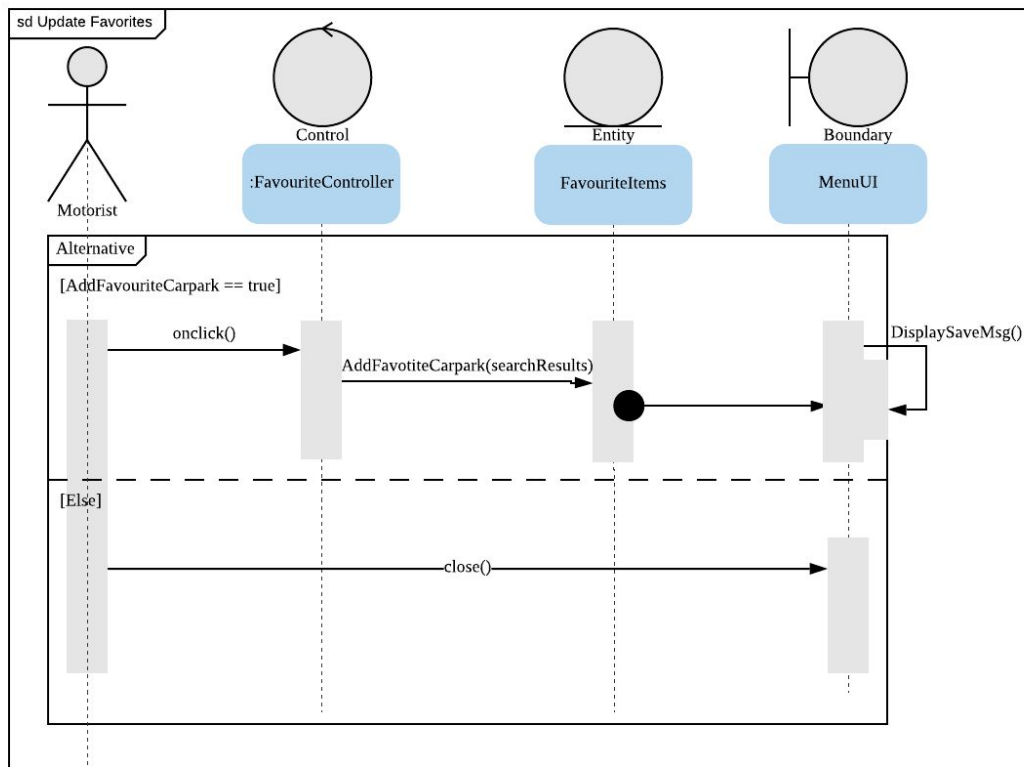
2. Login



3. Manage Profile

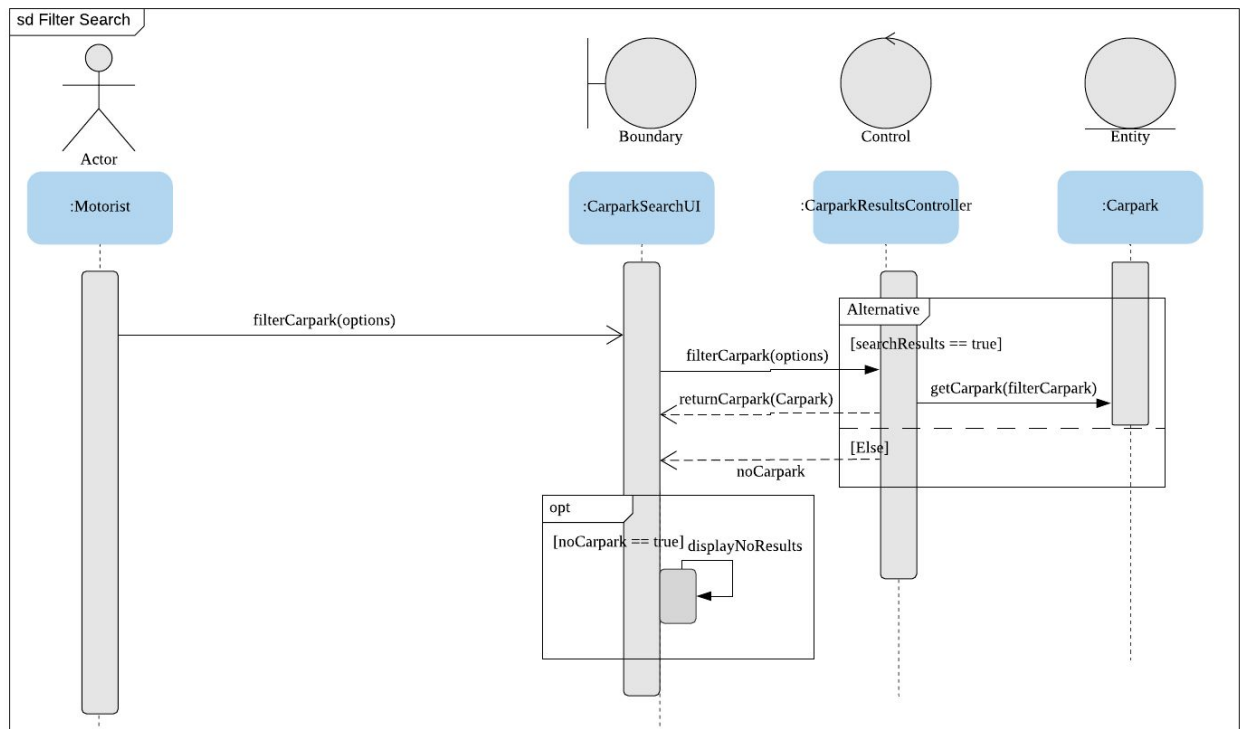


4. Update Favourites

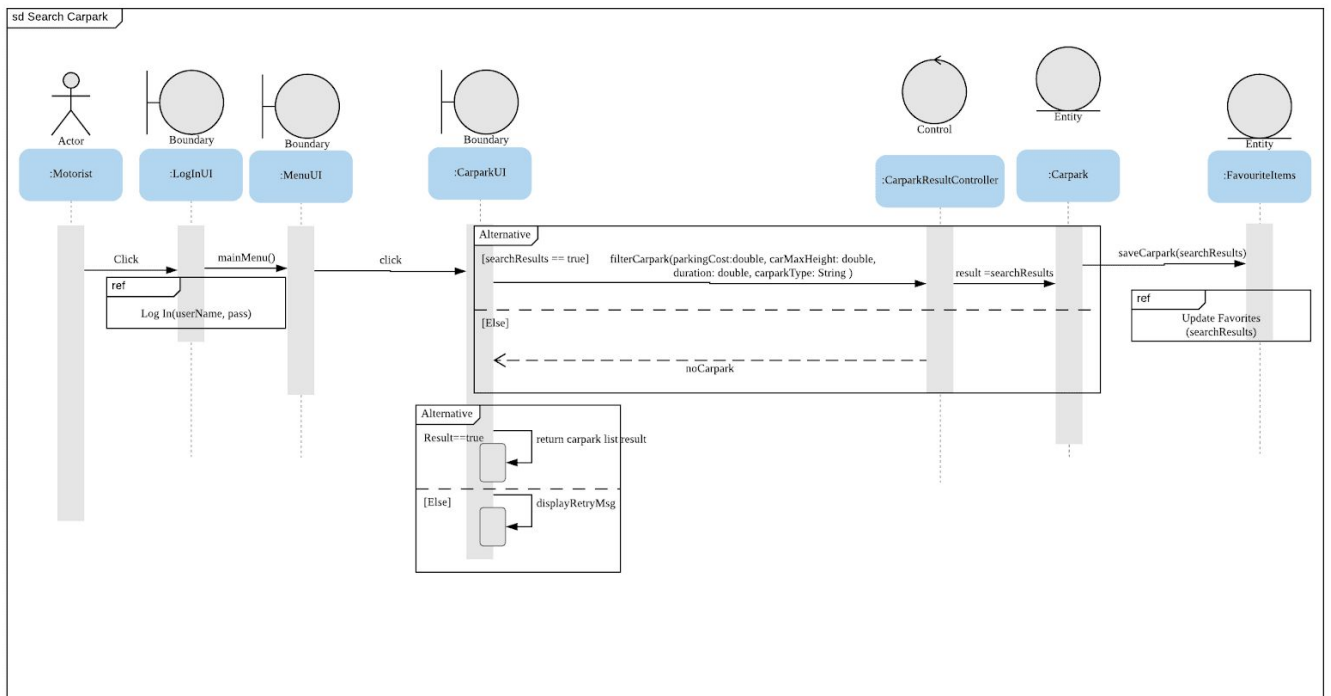


Reference from
SearchCarpark
Sequence
Diagram

5. Filter Search



6. Search Carpark



Initial Dialog map

