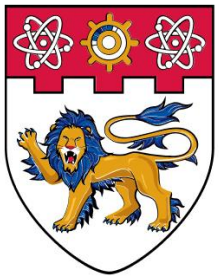


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

School of Computer Science & Engineering
SC2006 Software Engineering

Project Name: **ParkAnywhere**

Team Ryan

Lab 1

Team Members:

Ryan Teo Cher Kean (Leader)	U2122540D
Bryan Lee Wei Han	U2121789E
Chow Wei Jie	U2121056D
Chua Boon Chong, Jaedon	U2122153J
Colin Jun Meehan	U2120168L
Donna Chua	U2121254E

1. **Software description**

The car park availability app is a program for tracking the parking availability and rates for car parks across Singapore.

A. Data is taken from the real time car park database provided by the Housing and Development Board (HDB). The database provides real time updates on carpark availability every minute.
B. User can create a new account with unique user ID and password.
C. User profile will contain information such as parking history.
D. The software will take in user's inputs such as destination, parking rates, indoor/outdoor or charging points for electric vehicles,time(if applicable).
E. If car parks within the specified area are full, they will not be suggested to the user.
F. The system will return a list of suitable car parks, based on the user's inputs. Each carpark will provide an estimated cost based on time input (If no time input then show cost/hour). (if more people use the app, we're able to collect data and provide estimated waiting time)
G. The user may choose the desired car park based on the filters selected. The system will then provide a travel route to reach the destination from the user's current location.
H. There will also be a feedback system where users leave reviews and comments about car parks or the app in general.

2. Functional Requirements

1. Account

1.1. User must be able to create an account
- 1.1.1. Account must have unique ID
- 1.1.2. Password must be strong
  - 1.1.2.1. Password must
    - 1.1.2.1.1. Be at least 12 characters long
    - 1.1.2.1.2. Contain uppercase and lowercase letters
    - 1.1.2.1.3. Contain numbers
    - 1.1.2.1.4. Contain symbols

1.2. User must be able to log in with created account
- 1.2.1. User can request for help if password is lost

1.3. Users must be able to view their account information when logged in
- 1.3.1. Account information must contain parking history
  - 1.3.1.1. Parking history must contain location
  - 1.3.1.2. Parking history may contain spending on car park

2. Search

2.1. User must be able to request for a map/list of all available car parks
- 2.1.1. Results must contain
  - 2.1.1.1. Car park description
  - 2.1.1.2. Rates
  - 2.1.1.3. Availability
  - 2.1.1.4. Shelter availability
  - 2.1.1.5. Electrical vehicle charging point availability
  - 2.1.1.6. Wheelchair accessibility

2.2. User must be able to make a search
- 2.2.1. Search must contain destination
  - 2.2.1.1. Search may contain filters for
    - 2.2.1.1.1. Price range
    - 2.2.1.1.2. Shelter availability
    - 2.2.1.1.3. Electrical vehicle charging point availability
    - 2.2.1.1.4. Wheelchair accessibility
  - 2.2.1.2. User may input their stipulated time of use

2.3. A list of car parks that are near and available would be displayed after search
- 2.3.1. List must contain at least one car park
  - 2.3.1.1. Results must contain car park description
  - 2.3.1.2. Results must contain rates
    - 2.3.1.2.1. Total price will be displayed if user provided time input
  - 2.3.1.3. Results must contain
    - 2.3.1.3.1. Availability
    - 2.3.1.3.2. Shelter availability

2.3.1.3.3. Electrical vehicle charging point availability
2.3.1.3.4. Wheelchair accessibility
2.3.2. Car parks that are full will not be suggested
2.4. User can either choose car park from search results or from list/map of car parks
2.4.1. Users can input stipulated parking time
2.4.2. Route to chosen car park will be displayed
2.4.2.1. Real time navigation if GPS is enabled

3. Feedback

3.1. User must be able to leave feedback for car parks
3.1.1. User must select car park to rate
3.1.2. Feedback must contain a rating
3.1.3. Feedback must contain a comment
3.2. User must be able to leave feedback for the app
3.2.1. Feedback must contain a rating
3.2.2. Feedback must contain a comment

3. Non-Functional Requirements

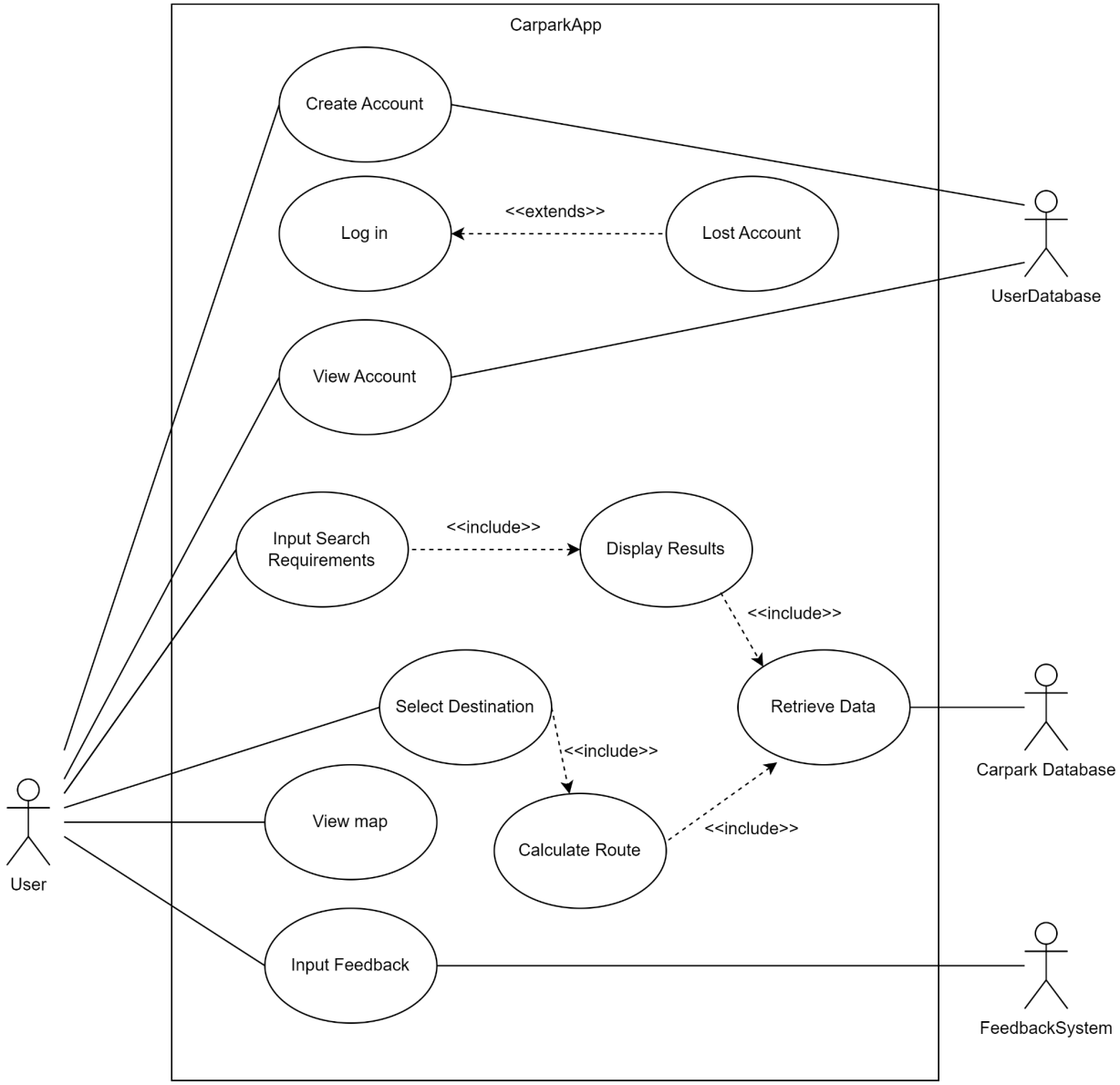
1. The app must hash the user's parking information before storing the data.
2. The app must produce consistent and accurate results.
3. The app must produce search results in less than 10 seconds.
4. The app must be able to restart in less than 5 minutes in event of failure.
5. The app must be able to support at least 2 different languages other than English.
6. The app must not have more than 4 hours of downtime per year.

4. **Data Dictionary**

Parking availability	The parking availability of a car park is the number of vacant parking lots in it. A car park with no vacant lots is labelled as unavailable.
Parking rate	The parking rate of a car park is the rate at which drivers will be charged for parking there over a given period of time(from). Usually it will be expressed in S$ per hour.
Sheltered	A boolean parameter for every car park, which is true if it is indoors and false if it is outdoors.
Wheelchair accessible	An integer parameter that indicates the presence and number of vacant wheelchair accessible parking lots in the car park. This actively updates based on if a wheelchair accessible parking lot has been occupied or is vacant.
Charging points	An integer parameter that indicates the presence and number of charging points at the car park for recharging electric cars. This actively updates based on if a charging point parking lot has been occupied or is vacant.
User preference	A set of parameters outlined by the user of the app for it to filter out and find the ideal car park for the user. These are based on the parking rate, sheltered, wheelchair accessible, charging points parameters of car parks.
Destination	This is the location the user wishes to travel to, the app searches for suitable nearby car parks from around this location.
Time input	This is the amount of time that the user expects to park at the car park, it can be used in conjunction with the parking rate to calculate the expected parking fee.
User feedback	User feedback allows users to leave a comment and rating for any car park, as well as for the app itself.
Rating	A rating from between 0 to 5 can be given by users for car parks during a review. An average rating can be displayed for each car park for other users to assess the quality of car parks and their amenities.
Account	Every user must create an account, which allows for tracking of data and details that makes it easier for the user to use the app.
Username	The username is for account login and is a publicly available form of identification to distinguish themselves from other accounts.

Password	The password is used to log into an account and should only be kept private for security reasons.

5. Use Case Diagram



6. **Use Case Descriptions**

Use Case ID:	1		
Use Case Name:	Create Account		
Created By:	Ryan Teo	**Last Updated By:**	Ryan Teo
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User (Initiating Actor), UserDatabase
Description:	User can create an account to use, which is stored in the UserDatabase
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. UserDatabase must be up and running.
Postconditions:	3. User creates an account with a unique username and password 4. User notified why they were unable to successfully create an account
Priority:	High
Frequency of Use:	-
Flow of Events:	1. User opens app. 2. User selects the option “Create Account”. 3. User sets a unique username. 4. User sets a unique password with at least 8 characters, 1 capital letter, 1 number and 1 symbol. 5. New User account is stored in UserDatabase. 6. User account is linked to a mobile number/email. 7. User is logged in and moved to the main screen.
Alternative Flows:	AF-S3: Username already in use. 1. User inputs a username that is already in use. 2. System displays error message “Username already in use. Please choose another username.” 3. System returns to step 3 AF-S4: Password did not meet requirements. 1. User inputted a password that did not meet the requirements. 2. System displays error message “Password did not meet the requirements. Please input a password with at least 8

	characters, 1 capital letter, 1 number and 1 symbol." 3. System returns to step 4.
Exception:	EX-1: User repeatedly attempts to register for an account more than ten times despite errors in inputs. 1. On the 11th attempt, when the User chooses "Create Account", the system displays the message "Too many attempts. Please try again in 10 minutes." 2. The "Create Account" button is unavailable for ten minutes. 3. The system only accepts registration from the App User's IP address after ten minutes.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Log in		
Created By:	Ryan Teo	**Last Updated By:**	Ryan Teo
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User (Initiating Actor), UserDatabase
Description:	User wants to log in to the app.
Preconditions:	1. User account must already exist in the database. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. User logs in to app. 2. User's account is locked due to too many login attempts
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. User opens app. 2. User chooses the log in option. 3. User inputs a username and password and selects "Login" 4. System checks username and password against UserDatabase. 5. User successfully logs in.
Alternative Flows:	AF-S4: System detects an invalid username or password. 1. When the User selects "Login", the system displays the message "Invalid username and/or password." 2. The system returns to Step 4.
Exception:	EX-1: The User inputs incorrect username or password more than five times. 1. On the sixth attempt at logging in, when User clicks on "Login", the system displays the message "Account suspended. Please try again after 10 minutes." 2. The "Login" button is unavailable for ten minutes. 3. The system only accepts registration from the User's IP address after ten minutes. EX-2: The App User forgot his/her login credentials . 1. The App User clicks on "Forgot Username/Password" on the login page. 2. User can recover their account using extended use case Lost Account

Includes:	-
Extends:	1. Lost Account
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Lost Account		
Created By:	Ryan Teo	**Last Updated By:**	Ryan Teo
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User (Initiating Actor), UserDatabase
Description:	User forgot their username/password and wants to recover their account.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. User must already have an existing account. 3. UserDatabase must be up and running.
Postconditions:	1. User successfully recovers their account details. 2. User is locked out due to too many recovery attempts.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. User opens app. 2. User selects “Forgot Username/Password” option. 3. User is brought to account recovery page. 4. User is prompted to input their registered mobile number/email. 5. UserDatabse is checked for matching mobile number/email. 6. Account recovery instructions are sent to User’s mobile number/email. 7. User resets username/password. 8. User successfully recovers account details. 9. New details are saved in UserDatabase. 10. User is returned to login page.
Alternative Flows:	AF-S5: Invalid mobile number/email. 1. Inputted mobile number/email is unable to be retrieved from UserDatabase. 2. System displays error message “Invalid mobile number/email. Please try again.” 3. User is returned to step 4.
Exception:	EX-1: User attempts to recover account details more than 3 times. 1. On the fourth attempt, the system displays “Account suspended. Please try again after 10 minutes.” 2. The “Forgot Username/Password” button is unavailable for

	ten minutes. 3. The system only accepts registration from the User's IP address after ten minutes.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	View Account		
Created By:	Ryan Teo	**Last Updated By:**	Ryan Teo
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User (Initiating Actor), UserDatabase
Description:	User wants to view account details and history (previous journeys etc.)
Preconditions:	1. User account must already exist in the database. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. User is able to retrieve account information.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. User taps on account icon. 2. System displays user information.
Alternative Flows:	-
Exception:	EX-1: User has not logged any information. 1. When user tries to look for information in the user account page, no information is displayed. 2. The message “No information.” is displayed.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Input Search Requirements		
Created By:	Ryan Teo	**Last Updated By:**	Colin
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User
Description:	User wants to find a suitable car park, inputs search requirements to System.
Preconditions:	1. User account must already exist in the database. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. User is able to retrieve a list of carparks. 2. User is able to select a carpark to set as a location.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. User taps on the search bar. 2. User inputs his desired location into the search bar. 3. User adjusts the filter based on his priority for the best carpark. 4. System returns a list of carparks.
Alternative Flows:	AF-S2: Desired location not found. 1. System displays the message “No matching locations.” 2. User is returned to step 2.
Exception:	-
Includes:	1. Display Results
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	Display Results		
Created By:	Ryan Teo	**Last Updated By:**	Colin
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	System
Description:	System filters results based on the specified search requirements by the User and displays them.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. User's search requirements are valid. 3. Carpark Database is up and running.
Postconditions:	1. Relevant results are displayed to the User by the System. 2. No relevant results are found.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. System receives input from User. 2. System sends input to Carpark Database. 3. System retrieves relevant results from Carpark Database. 4. System displays results to User.
Alternative Flows:	AF-S3: Input does not return any results from Carpark Database. 1. Carpark Database returns "null" to System. 2. System displays error message "No carparks found with those search requirements. Please refine your requirements." 3. Return to Step 1.
Exception:	-
Includes:	1. Retrieve Data
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Select Destination		
Created By:	Ryan Teo	**Last Updated By:**	Colin
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User
Description:	After System returns a list of suitable car parks, the User selects a destination based on that list.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. System must have returned a list with carparks for the User to choose from.
Postconditions:	1. User selects a destination from the list. 2. User does not like the options suggested and chooses to edit his search requirements.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. System returns a list of carparks for User to choose from. 2. User selects a carpark as their destination. 3. System assigns that carpark location as the destination.
Alternative Flows:	AF-S2: User wants to edit his search after seeing the results 1. User clicks on the search bar to re-enter search requirements. 2. User inputs his search requirements. 3. System retrieves search results from Carpark Database. 4. System displays search results. 5. Return to Step 2.
Exception:	-
Includes:	1. Calculate Route
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Calculate Route		
Created By:	Ryan Teo	**Last Updated By:**	Colin
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	System
Description:	After User has selected a destination, System will determine the best route to take to get there.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. User has inputted their intended destination. 3. System is able to detect User's starting location.
Postconditions:	1. System is able to calculate a relevant route to the intended destination from the starting location.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. System registers the destination chosen by the User. 2. System retrieves the starting location of the User based on mobile phone GPS. 3. System generates route using Google Maps API.
Alternative Flows:	-
Exception:	-
Includes:	1. Retrieve Data
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Retrieve Data		
Created By:	Ryan Teo	**Last Updated By:**	Colin
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	System (Initiating Actor), Carpark Database
Description:	System retrieves necessary info from the Carpark Database so it can generate results for suggested carpark.
Preconditions:	1. System receives a request for data from User.
Postconditions:	1. System receives relevant data from Carpark Database. 2. System is unable to receive relevant data from Carpark Database.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. System sends input requirements to Carpark Database. 2. Carpark Database is searched based on the requirements and returns all relevant results to the System. 3. System consolidates all results and displays them for the User.
Alternative Flows:	AF-S2: Carpark Database is unable to find any suitable results. 1. Carpark Database returns “null” to System. 2. System displays the error message “No carparks found with those search requirements. Please refine your requirements.” 3. Return to Step 1.
Exception:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

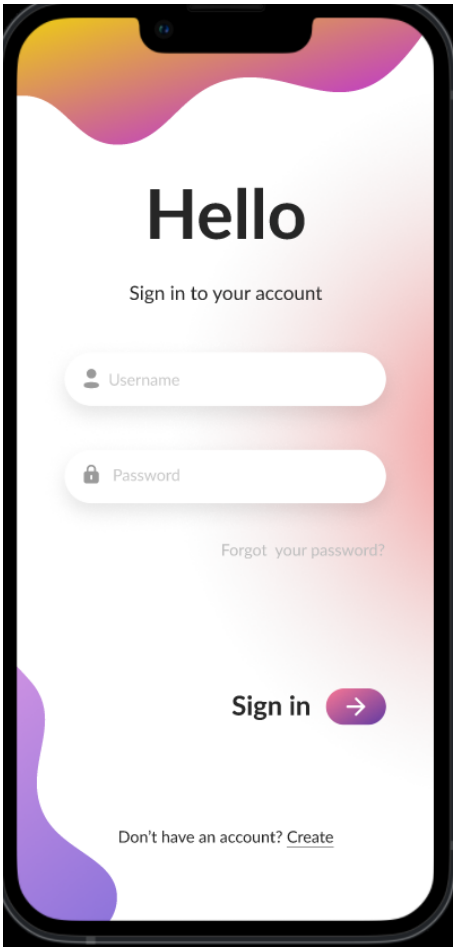
Use Case ID:	10		
Use Case Name:	View Map		
Created By:	Ryan Teo	**Last Updated By:**	Ryan Teo
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

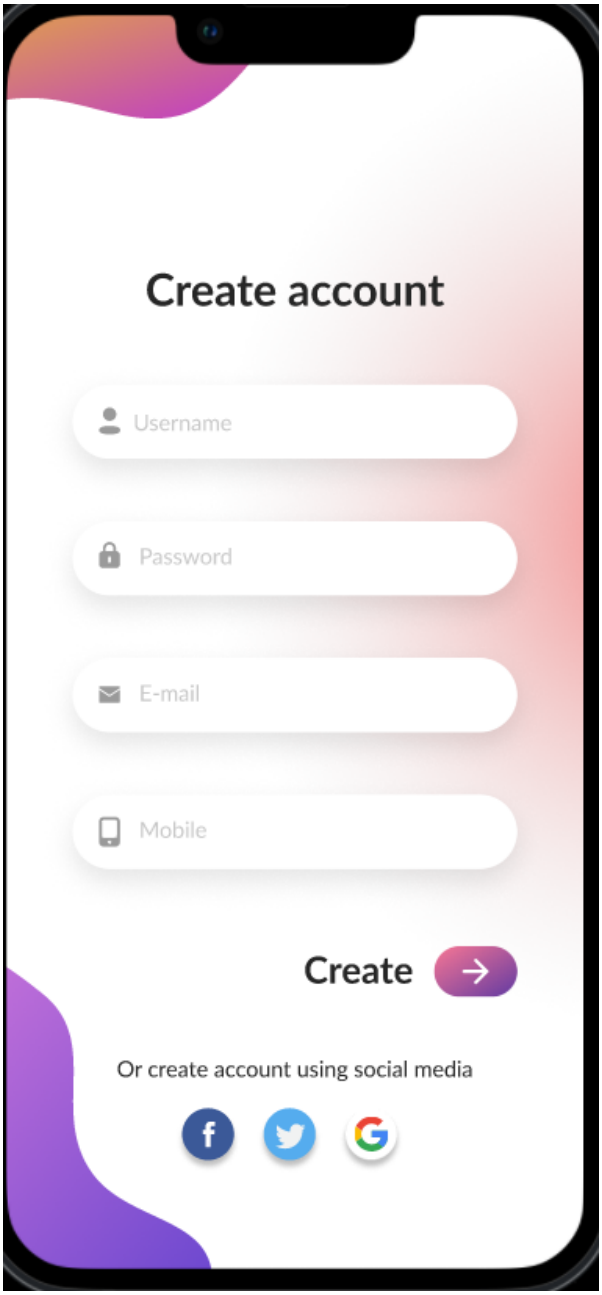
Actor:	User (Initiating Actor), System
Description:	System displays map information on the screen.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. System calculates routes and displays them with the map of Singapore. 2. System is unable to calculate a route.
Priority:	Medium
Frequency of Use:	-
Flow of Events:	1. User opens app and logs in. 2. System retrieves User's current location using mobile phone GPS. 3. System displays the map of Singapore centred around User's location.
Alternative Flows:	AF-S2: System cannot retrieve location information. 1. System is unable to retrieve current location. 2. System displays map centred around last known location.
Exception:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	Input Feedback		
Created By:	Ryan Teo	**Last Updated By:**	Wei Jie
Date Created:	6th February 2023	**Date Last Updated:**	9th February 2023

Actor:	User
Description:	User wants to input feedback on a specific car park or on the CarparkApp.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data. 2. User must be logged in.
Postconditions:	1. User feedback is uploaded.
Priority:	Medium
Frequency of Use:	0-5 times a day
Flow of Events:	1. User chooses feedback section. 2. System will display option for app or carpark feedback. 3. User chooses app option. 4. User inputs rating out of 5. 5. User inputs comments for the rating. 6. System will display feedback for user to confirm. 7. User chooses to confirm. 8. System will upload feedback to Feedback System.
Alternative Flows:	AF-S3: User chooses car park option. 1. System displays list of previously used car parks. 2. User chooses car park to rate. 3. Return to Step 4. AF-S6: User does not confirm the feedback. 1. Return to Step 2.
Exception:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

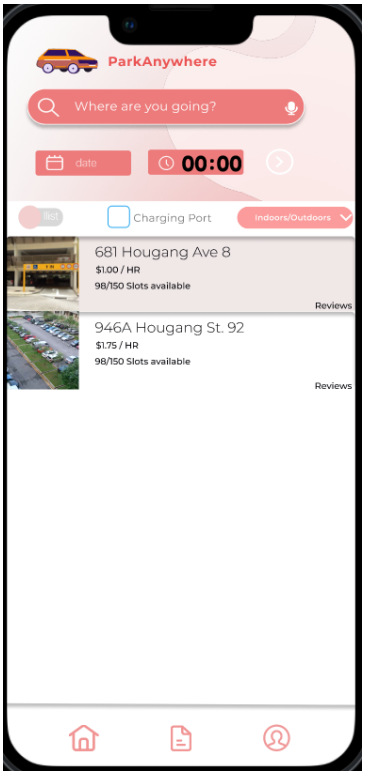
7. UI Mockups

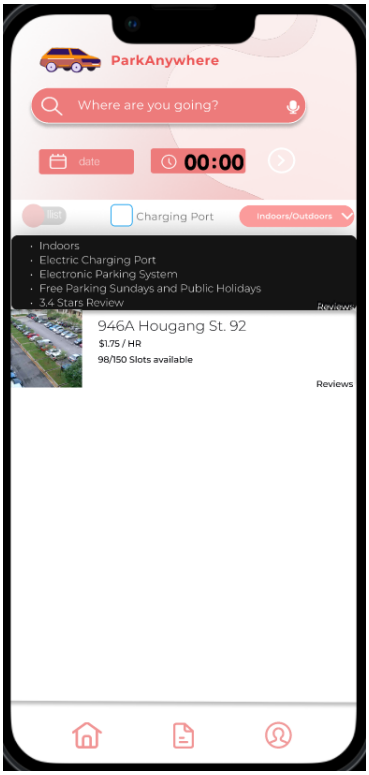
Log in Page	
	This is the login page, where users can sign in.

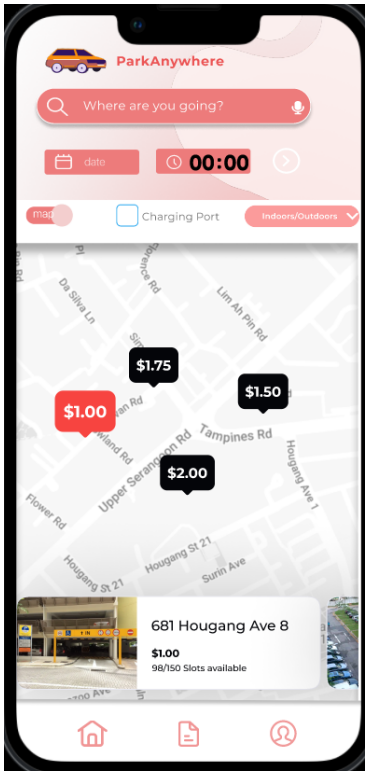
Sign up Page	
	This is the sign-up page where users can sign up for an account.

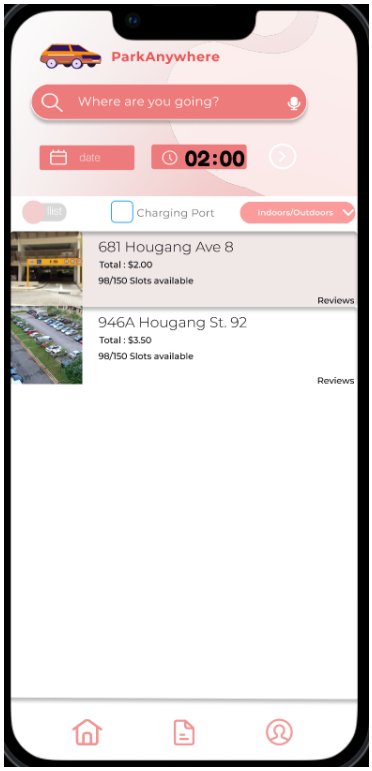
Loading Page	
	This is the loading page, the car will move to the left.

Main page

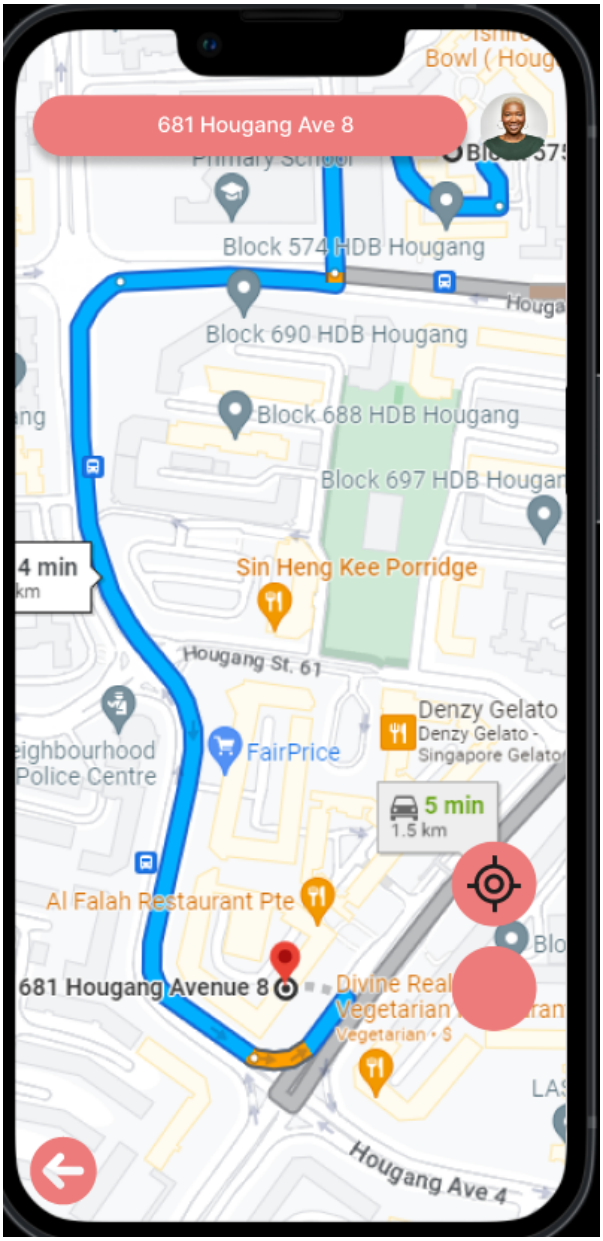


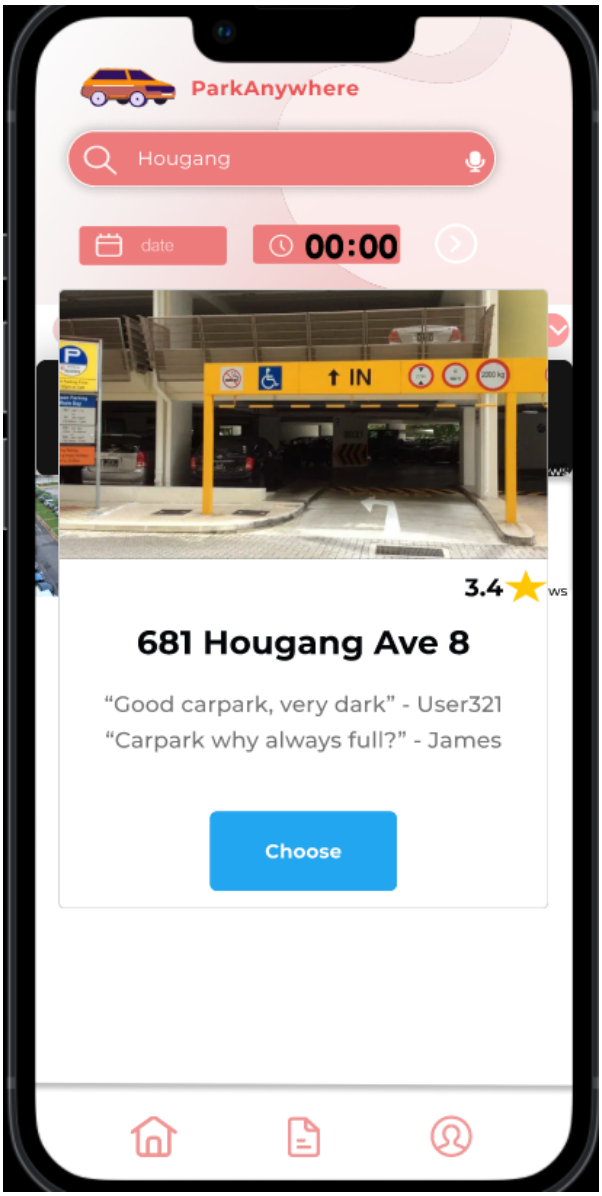




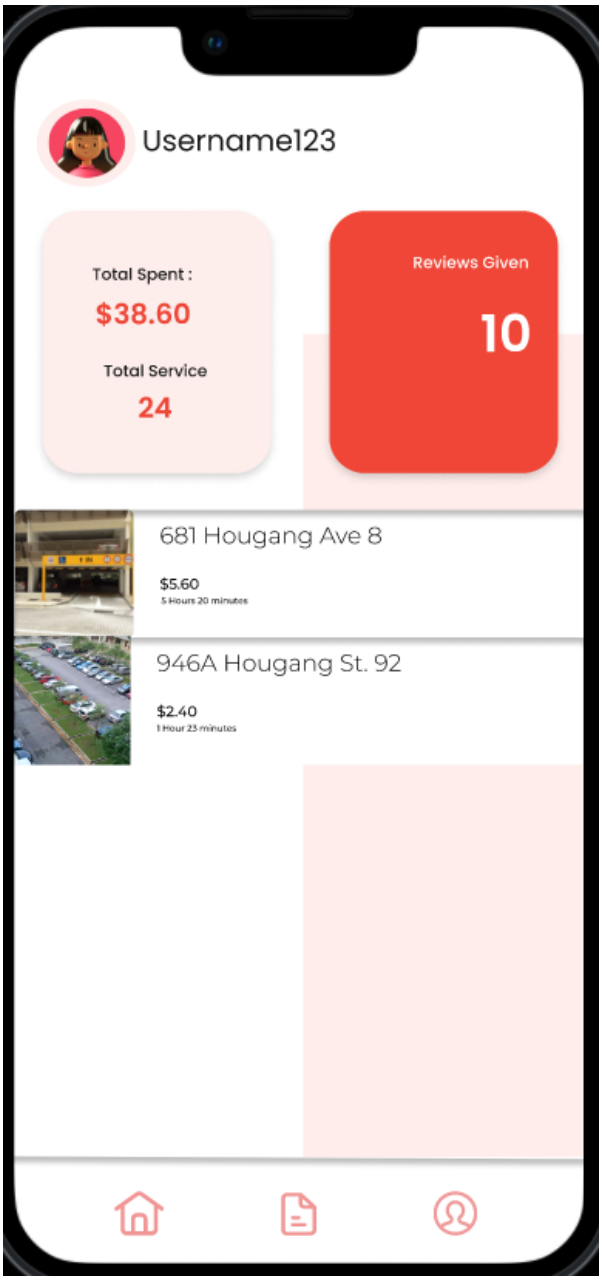


This is the main page where users can find carparks near the area they want to go to. They can apply filters to choose a carpark of their liking. There is also a map view to view based on map. In the last phone, we have chosen Hougang as our destination and the duration of parking will be 2 hours, as such, the total price will be calculated.

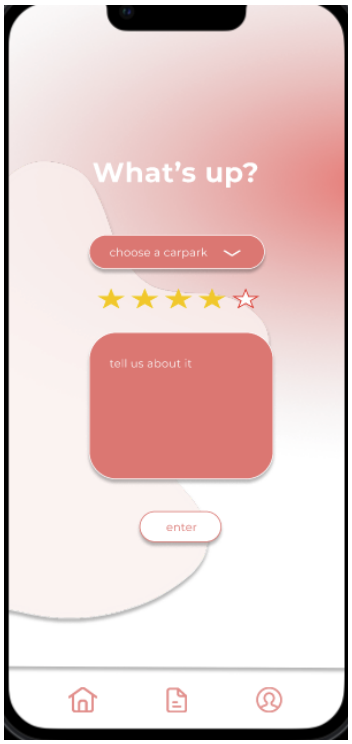
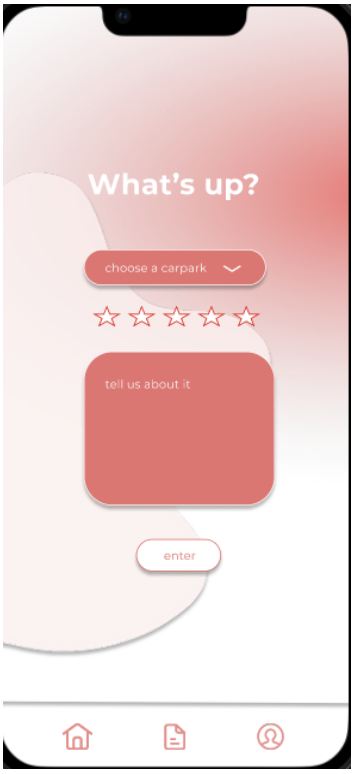
Choose of Carpark	
	Upon choosing of carpark, a GPS map will pop up and give you the fastest route to your location.

Reviews Page	
	Users can see reviews of other users by clicking the review button.

Profile Page



This is the user's profile page, the user can see how much they spent, how many carparks they went to, how many reviews and history of the carpark they went to.

Feedback page	
 	Users can feedback a carpark and other users can see it.