Functional requirements

1. The user must login to the respective user account by entering email and password to access the system.
    1.1. The user must press the ‘login’ button to login.
    1.2. The system must display error messages if the user fails to login to the system.
    1.3. The system will search the database to see if the email and password is correct.
    1.4. If the data is correct, then the user is redirected to the Home Screen.
    1.5. The user must stay in even after the application is closed.

2. If the user does not have an account, the user must be directed to the sign up screen. The user must input their email, first name, last name and their password.
    2.1. Email must be of string data type.
        2.1.1. The user must input a valid email address.
            2.1.1.1. A valid email address must include the symbol ‘@’
        2.1.2. If the email is empty, the user is prompted to fill up the field.
        2.1.3. If the email has already been used before and is stored in the database, then the user must be prompted that the email has already been used to create an account.
    2.2. First name must be of string data type.
        2.2.1. User must input a valid first name.
            2.2.1.1. A valid first name cannot contain characters other than text and integer datatype and the two special characters(.,@).
        2.2.2. If the first name is left blank, the user is prompted to fill up this field.
        2.2.3. If the input value of the first name does not meet the requirements of section 2.2.1, the user is prompted to refill the field with the correct requirements.
    2.3. Last name must be of string data type.
        2.3.1. User must input a valid last name.
            2.3.1.1. A valid last name cannot contain characters other than text and integer datatype and the two special characters(.,@).
        2.3.2. If the last name is left blank, the user is prompted to fill up this field.
        2.3.3. If the input value of the last name does not meet the requirements of section 2.2.1, the user is prompted to refill the field with the correct requirements.
    2.4. Password must be of string data type
        2.4.1. User must input a secure, valid password.
            2.4.1.1. A valid password must have a minimum of 8 characters.
            2.4.1.2. A valid password must contain at least 1 capital letter(A-Z).
            2.4.1.3. A valid password must contain at least 1 number(0-9).
            2.4.1.4. A valid password must contain at least 1 of the following characters- (‘.’,’_’,’-’).
        2.4.2. If the password is empty, the user is prompted to fill up the field.

2.4.3. If the password requirements of sections 2.2.1.1, 2.2.1.2, 2.2.1.3, 2.2.1.4 are not met, the user is prompted to refill the field with the correct requirements.

2.5. The user must press the “register” button to proceed with the registration.
- 2.5.1. If the “register” button does not receive input from the user, the state of the application does not change.
- 2.5.2. If the requirements in 2.2.1 and 2.1.1 are not met, the user will not be able to register.
- 2.5.3. If all the requirements in 2.2.1 and 2.1.1 are met, then the user will be redirected to the Login page.

2.6. If the user has forgotten their password, the system must allow the user to reset their password.
- 2.6.1. The user will be led to the forgot password page.
- 2.6.2. The user will be required to enter their email.
  - 2.6.2.1. If the email requirements in 2.1.1 are not fulfilled, then the user will be prompted to enter a valid email.
- 2.6.3. An email must be sent out to the user, using the email address that they has inputted.
- 2.6.4. The email must be sent out a link which allows the user to reset their password.
- 2.6.5. The new password must fulfil the requirements in 2.2.1
  - 2.6.5.1. If the password requirements in 2.2.1 are not fulfilled, then the user will be prompted to enter a valid password.

3. The Home Screen has the following components:

3.1. The system must display the map of Singapore.
- 3.1.1. The map must cover the screen.
- 3.1.2. The system must show the current location of the user by using a circle as a marker.

3.2. The system must display the Search function as described in section 4.

3.3. The system must display the sidebar.
- 3.3.1. The following tabs must be included as part of the sidebar of the home page
  - 3.3.1.1. My account
    - 3.3.1.1.1. Shows user account profile which includes the following:
      - 3.3.1.1.1.1. First Name
      - 3.3.1.1.1.2. Last Name
      - 3.3.1.1.1.3. Email
      - 3.3.1.1.1.4. Profile photo
        - 3.3.1.1.1.4.1. The default photo is that of a car
    - 3.3.1.1.2. User must be able to edit the information in 3.3.1.1.1.
    - 3.3.1.1.3. User must be able to change their password by clicking the “Change password”
    - 3.3.1.1.4. There must be a graph showing the total parking fare spent per month.

3.3.1.1.4.1. User must be allowed to customise the year.
3.3.1.1.4.2. User must be allowed to customise the month.
3.3.1.1.4.2.1. The months must have the following categories:
3.3.1.1.4.2.1.1. January - April
3.3.1.1.4.2.1.2. May - August
3.3.1.1.4.2.1.3. September - December
3.3.1.1.5. There must be a tab leading to <u>My favourites</u>
3.3.1.1.6. There must be a tab leading to <u>My history</u>
3.3.1.2. <u>My favourites</u>
3.3.1.2.1. Shows a list of carparks saved by user by clicking the heart shape.
3.3.1.3. <u>My history</u>
3.3.1.3.1. Shows the last few carparks the user has gone to

4. The system must allow the user to use the search function to search a location.
4.1. The search function must allow the user to input the following locations:
4.1.1. Destination
4.1.2. Departure
4.1.2.1. By default, the departure location must be the current location.
4.2. Users must be able to search the location using the following criteria:
4.2.1. The system must allow the user to search for a location using the name of the place.
4.2.2. The system must allow the user to search for a location using the address of the place.
4.2.3. The system must allow the user to search for a location using a postal code.
4.3. When using the search function, the suggested search results must pop up.
4.3.1. A message will be displayed if no results are found.
4.4. The system must display available carparks on the map near the carpark.
4.4.1. The carparks must be in a 1km radius from the destination.
4.5. The system must list out the available carparks in the slide-up box pop out from the bottom half of the screen.
4.5.1. The caparks must be grouped under the header ‘category’ as a dropdown box with the following options:
4.5.1.1. <u>Distance</u>
4.5.1.2. <u>Fare</u>
4.5.1.3. <u>Availability</u>
4.5.1.4. <u>Recommendation</u>
4.5.1.4.1. The recommendation will be based on an algorithm. The algorithm will be based on the following variables
4.5.1.4.1.1. Car park fee rate
4.5.1.4.1.2. Opening time
4.5.1.4.1.3. Availability
4.5.1.4.1.4. Distance from location
4.5.2. The carparks must be able to be sorted by under the header ‘sort by’ based on the category selected:

4.5.2.1. If the category is “distance”, the available sort options are the followings:
4.5.2.1.1. Nearest to Furthest(Departure to Carpark)
4.5.2.1.2. Furthest to Nearest(Departure to Carpark)
4.5.2.1.3. Nearest to Farthest(Carpark to Departure)
4.5.2.1.4. Farthest to Nearest(Carpark to Departure)
4.5.2.2. If the category is “fare”, the available sort options are the followings:
4.5.2.2.1. Cheapest to Most Expensive
4.5.2.2.2. Most Expensive to Cheapest
4.5.2.3. If the category is “availability”, the available sort options are the followings:
4.5.2.3.1. Least to Most
4.5.2.3.2. Most to Least
4.5.2.4. If the category is “recommendation”, the available sort options are the followings:
4.5.2.4.1. Best to worst
4.5.2.4.2. Worst to Best

4.6. The system must calculate the distance between the carpark and the destination.
4.6.1. The mode of calculation must be based on the actual walking distance between the carpark and the destination

5. The system must allow the user to select the desired carpark.
5.1. The system must display information related to the selected carpark
5.1.1. The following information must be included
5.1.1.1. Carpark name
5.1.1.2. Carpark location address
5.1.1.3. Driving time/distance (from departure to carpark)
5.1.1.4. Walking time/distance (from carpark to destination)
5.1.1.5. Carpark parking rate
5.1.1.5.1. There must be carpark calculator
5.1.1.5.1.1. User will input the number of hours they intend to stay for
5.1.1.5.1.2. Calculator will calculate the parking fare in dollars according to what the user has inputted.
5.1.1.6. Carpark availability
5.1.1.7. Carpark operating hours
5.1.1.8. Comments and ratings by other users
5.2. There must be a “Navigate” button that directs the user to Google Maps.
5.2.1. The departure location in Google Maps must be the departure location selected by user.
5.2.2. The destination location in Google Maps must be the Carpark selected by user.
5.3. There must be a “Share” button that allow user to share the selected Carpark to others via the following:
5.3.1. Whatsapp
5.3.2. Telegram

5.3.3. Email.

5.4. There must be a “Start Parking” button for the user to press once the user reaches the carpark.

5.5. The system must be able to include information on the car park availability for each and every parking lot

5.5.1. The user can choose to book the parking lot when an available one is found.

5.5.1.1. The system will inform the user once the parking lot has been booked

5.5.2. If the car does not park at the parking lot within 15 minutes, the car park lot is released

5.5.2.1. the system will inform the user once the parking lot has been released

6. While the system has started the journey to the selected carpark, in which before the user click on the “Start Parking” button, the system must check the availability of carparks near to the intended destination.

6.1. The system must conduct the checking process every 1 minute.

6.2. The checking process will consist of the following:

6.2.1. Has the number of carpark lots dropped by more than 20% in the past 10 minutes?

6.2.2. Are there 3 or less parking lots left in the carpark?

6.3. If the answer to section 6.3.1 or 6.3.2 is yes, then a notification will pop out informing the user of the respective conditions

6.3.1. If the user ignores the notification, the state of the system does not change

6.3.2. If the user clicks yes on the notification, the user can choose to reroute to another carpark

6.3.2.1. It can be another carpark recommended by the algorithm in section 4.5.1.4.

6.3.2.2. User can also select another carpark based on the other categories in section 4.5.

7. After the user press “start parking” button, the system must show the following:

7.1. The carpark address

7.2. The system must allow the user to enter user’s parking time estimation

7.2.1. The user must be able to customise the hour.

7.2.1.1. The hour must be in integer data type

7.2.1.2. The hour must have the range from 0 to 23.

7.2.1.3. The default number is 0.

7.2.2. The user must be able to customise the minute.

7.2.2.1. The minute must be in integer data type

7.2.2.2. The minute must have a range from 0 to 59.

7.2.2.3. The default number is 0.

7.2.3. The system must calculate and display the estimated carparking fare.

7.2.3.1. The format of calculation is the following:

7.2.3.1.1. The minute set by user must convert into an hour.

7.2.3.1.2. The converted minute must be added to the hour.

7.2.3.1.3. The formula is [total hour] x [carparking rate] = [total amount in Singapore dollar]

7.2.4. The system must allow the user to set timer.

7.2.4.1. The system must allow the user to customise the minute.

7.2.4.1.1. The minute must be integer data type.

7.2.4.1.2. The minute must be in the range from 0 to 59.

7.2.4.1.3. The default number is 0.

7.2.4.2. The system must send notification to user based on the selected minute before the end time estimated by the user.

7.2.4.3. The system must calculate the end time by adding the hours and minutes set by user to the start time, which is the current time once user press “continue” button.

7.2.5. The system must allow the user to input note as following:

7.2.5.1. The system must allow user to upload a photo.

7.2.5.1.1. The size of the photo must be less than 500MB.

7.2.5.2. The system must allow the user to input word notes.

7.2.5.2.1. The word notes must be in string data type.

7.2.6. There must be a “continue” button.

8. When the user has parked the car, in which after “continue” button is pressed

8.1. The system must display the followings:

8.1.1. Carpark location address

8.1.2. Start time

8.1.2.1. The start time is taken at the moment when the user press the “continue” button.

8.1.3. Estimated end time

8.1.3.1. The estimated end time is calculated by adding the estimated hour and minute set by user to the start time.

8.1.4. Timer

8.1.5. Photo uploaded and note written.

8.2. There must be a “Navigate” button at the bottom left corner of the screen with “from carpark to destination” label that directs the user to Google Map.

8.2.1. The departure location in Google Maps must be the carpark location selected by user.

8.2.2. The destination location in Google Maps must be the destination selected by user.

8.3. There must be a “Navigate” button at the bottom right corner of the screen with “from destination to carpark” label that direct the user to Google Map.

8.3.1. The departure location in Google Maps must be the destination selected by user.

8.3.2. The destination location in Google Map must be the carpark selected by user.

8.4. There must be an “end parking” button at the bottom of the screen.

9. When the user has done the parking and wishes to leave the carpark, in which after “end parking” button is pressed.

9.1. The following details will be displayed.

9.1.1. The carpark name

9.1.2. The carpark location address
9.1.3. Date
9.1.4. Start time
9.1.5. End time
9.1.5.1. End time is obtained once the user presses the “end parking” button.
9.1.6. Duration
9.1.6.1. Duration is calculated by [end time] - [start time]
9.1.7. Total fare
9.1.7.1. Total fare is calculated based on the actual duration.\
9.2. The system must save the information in 9.1 into the database.
9.2.1. The saved information must be displayed in My history.
9.3. The system must allow the user to provide feedback
9.3.1. The system must allow the user to type out the comment in the box.
9.3.1.1. The comment must be in string data type.
9.3.2. The system must allow the user to rate the carpark in the box.
9.3.2.1. The rating has a range from 0 star to 5 stars.
9.4. There must be a “done” button.
9.4.1. If the user presses this button, it represents the whole parking assistance service is ended and must direct the user back to Homepage.

9.5. The system must be able to record the destination of the parked car
9.5.1. The system must display an button to click that reads ‘reached destination’
9.5.2. If the user does not press the ‘reached destination’ button
9.5.2.1. the system must automatically assume that the location is the carpark that the car was routed to go to.
9.5.2.2. After 5 min, if the car is in the vicinity of the routed carpark, the system will automatically start the timer.
9.5.3. After the system has detected that the user has parked the car as in sections 7.1.2 or 7.1.3, the system will input the following into the database:
9.5.3.1. Start Time
9.5.3.2. Location of carpark
9.5.4. When the user inquires to get back to their car
9.5.4.1. The system must allow the user to click the ‘reach carpark’ button.
9.5.4.2. The system must lead the user back to the carpark from their current location using the in-built map.
9.6. The system must start a timer
9.6.1. The system must prompt the user to select how many hours they requires to stay in the carpark
9.6.2. The user will be able to change the timer
9.6.2.1.1. The user must manually input the number of hours
9.6.2.1.2. If the user inputs 0 minutes, then no timer will be set

9.6.3. Once the imputed time in section 7.2.2.1.1 is left on the timer, the system will notify the user that their parking time has almost run out.
9.6.4. The User must be able to choose to extend the time left.

9.7. The system must allow the user to input notes on the carpark location
9.7.1. The user must be able to input the following
9.7.1.1. Text-based notes
9.7.1.2. Pictures

9.8. The system must be able to lead the user to the destination that they searched with in section 4.

10. When the user wants to get back to the carpark
10.1. The app must use the recorded location in section 7.1.2 to lead the user back to the carpark

11. When the user finishes parking their car and wants to leave the carpark:
11.1. The system must track when the car leaves the carpark
11.1.1. The system UI must include a button which reads “end parking”
11.1.1.1. Clicking the “end parking” button will allow the system to know that the user has finished parking
11.1.1.2. If the user does not click on the “end parking” button, the system will track when the car leaves the destination by using the user’s “current location”, and determining when it leaves the radius of the carpark.
11.1.1.3. The following info will be inputted into the database:
11.1.1.3.1. End time
11.1.1.3.2. Time spent
11.1.1.4. The data will then be used in ‘my history’ tab

11.2. The system must allow the user to comment and rate the carpark
11.2.1. The rating must be a 1-5 star rating system