**NANYANG TECHNOLOGICAL UNIVERSITY**

**CZ2006 SOFTWARE ENGINEERING**

**Car Park App**

**SSP1 Team NTU SE**

**Team Members:**

Devansh Koppar (Leader) (U1823660K)

Ng Wee Hsuan (U1920658D)

Glenda Hong Zixuan (U1921784A)

Supraja Naganathan (U2021507E)

Chantharojwong Kasidis (U2020731L)

Lim Jiayi Joey (U2021903G)

Contents

[1. Product Description 3](#_Toc81941797)

[1.1. UI Mockups 4](#_Toc81941798)

[2. Functional Requirements 11](#_Toc81941801)

[2.1. Use Case Diagrams 15](#_Toc81941802)

[2.2. Use Case Descriptions 16](#_Toc81941803)

[3. Non-Functional Requirements 39](#_Toc81941804)

[4. Data Dictionary 41](#_Toc81941805)

# 1. Product Description

**Availability:** iOS App Store

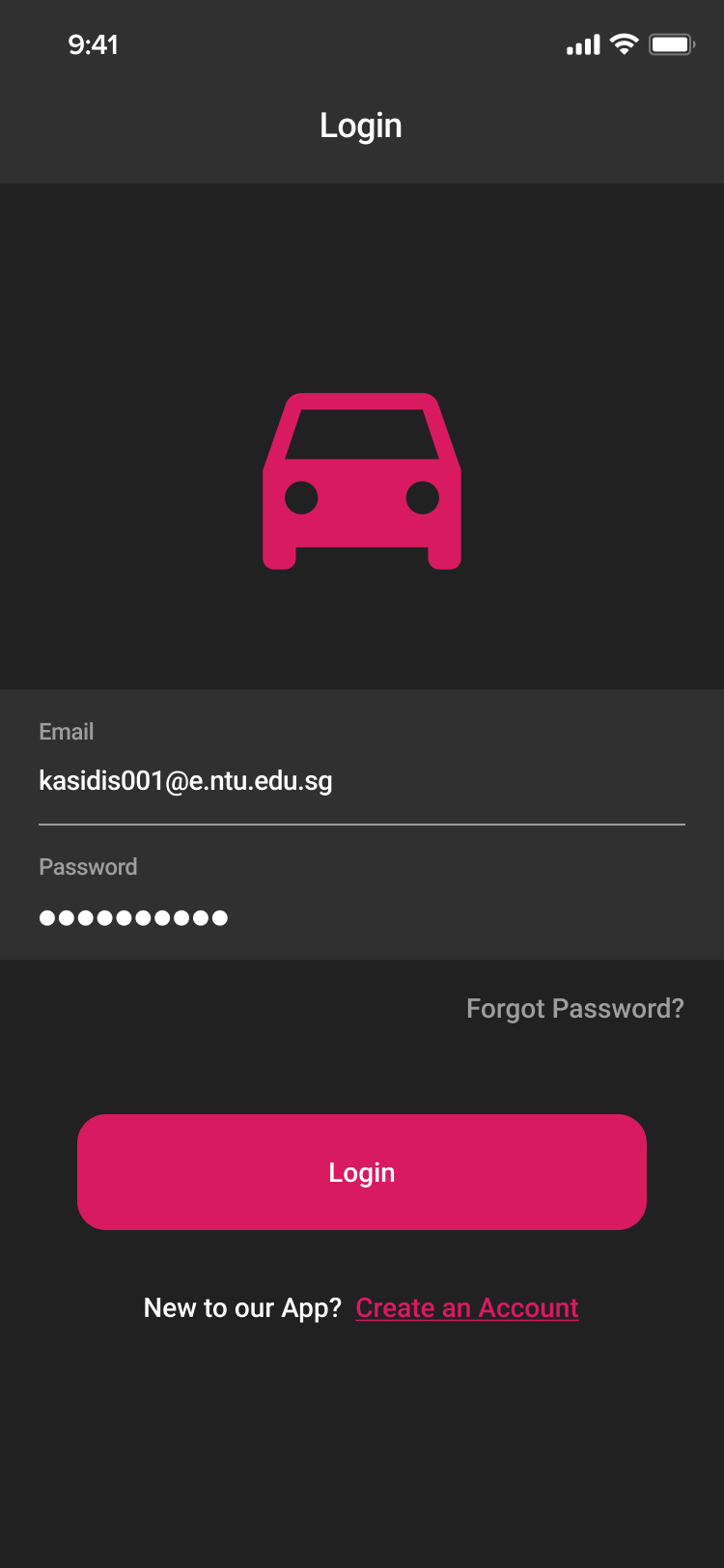
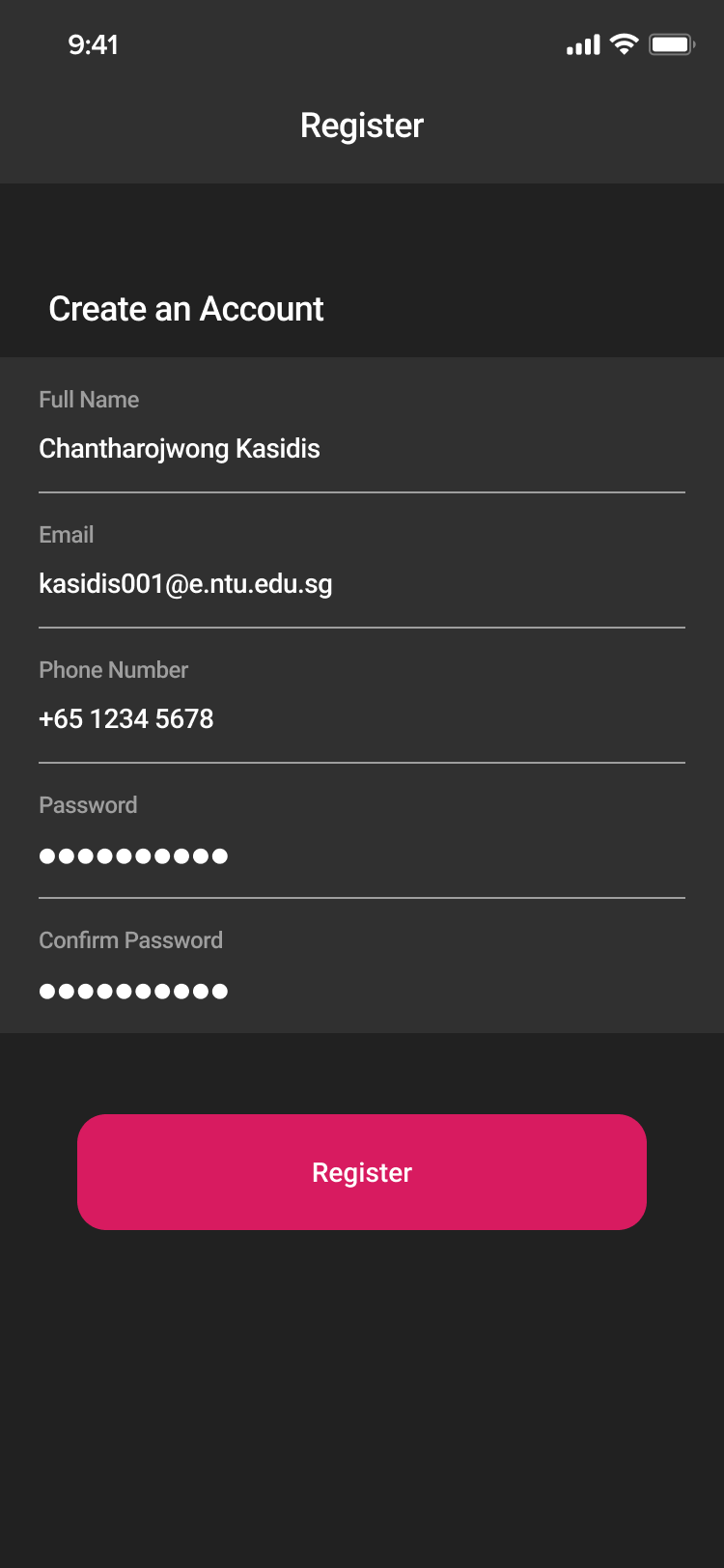
**API:** https://data.gov.sg/dataset/carpark-availability

**Target Users:** Drivers

**Description:**

Our application will allow drivers to check for availability lots in HDB car parks across Singapore. The application will also allow drivers to reserve lots in said car parks.

## 1.1. UI Mockups



# 

*Figure 1: Login and Registration Screens*

Figure 1 illustrates the login and registration screens. The login screen displays the Email and Password field, a registration link, a login button, and an optional link for resetting the password. New users can create an account by filling in their information and password at the registration screen.

# 

# 

# 

# 

# 

# 

# 

# 

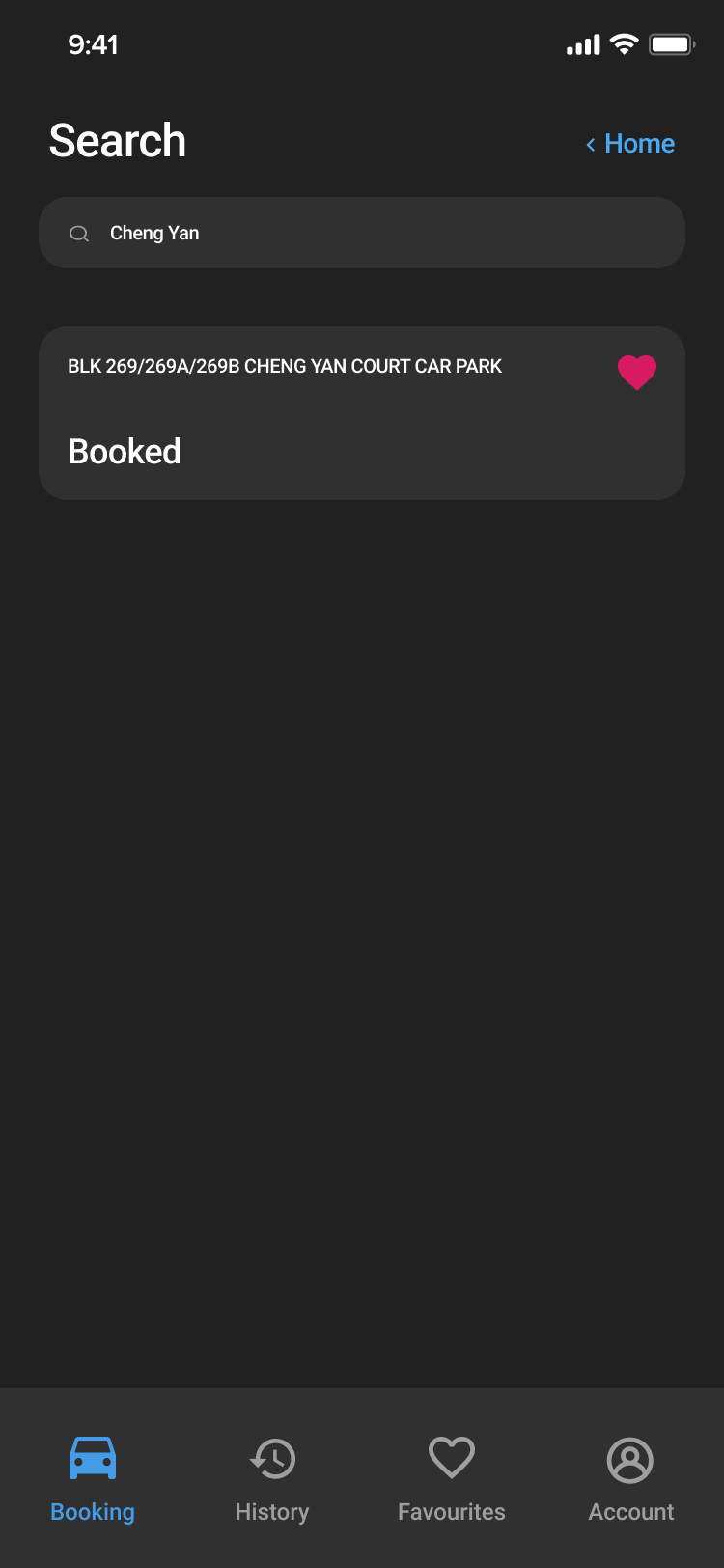
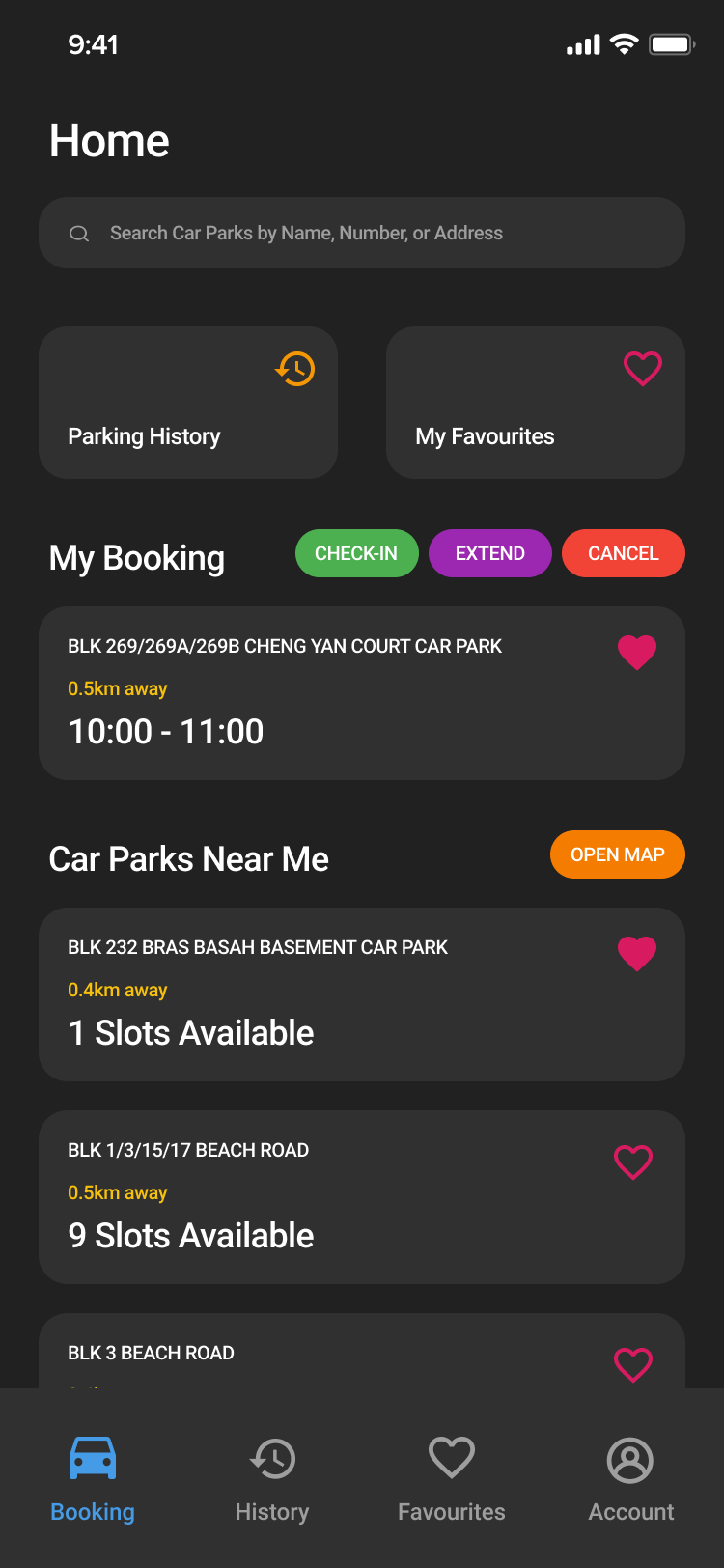
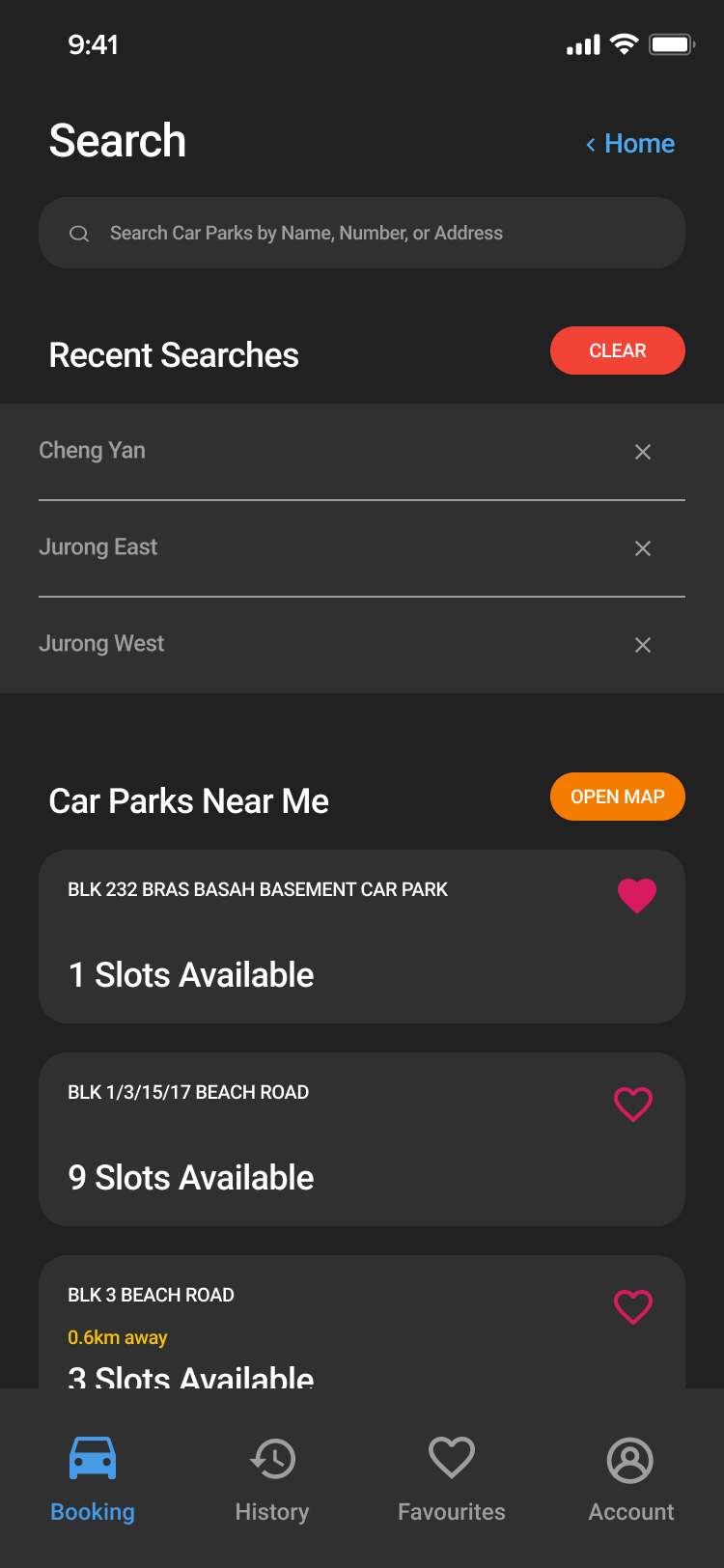
*Figure 2: Recover Password and Reset Password Screens*

Figure 2 illustrates the recover password and reset password screens. Users who have forgotten their current password can fill in their account email address. After receiving the request, the system will send a link to the email. When a user clicks on the link received, they will be navigated to reset password screen to set a new password.

# 

# 

# 



Ddd

*Figure 3: Home and Search Screens*

Figure 3 illustrates the home and search screens. After login, users will see the bottom menu bar. The home, search, map, car park information, and choose parking time screens can be accessed from the booking menu. The home screen displays a search bar and menu shortcuts at the top, followed by current booking details and a list of nearby car parks with an option to view these car parks in the map. When a user clicks on the search bar, the system will recommend recent searches and will display a list of car parks when the user starts typing.

# 

# 

# 

# 

# 

# 

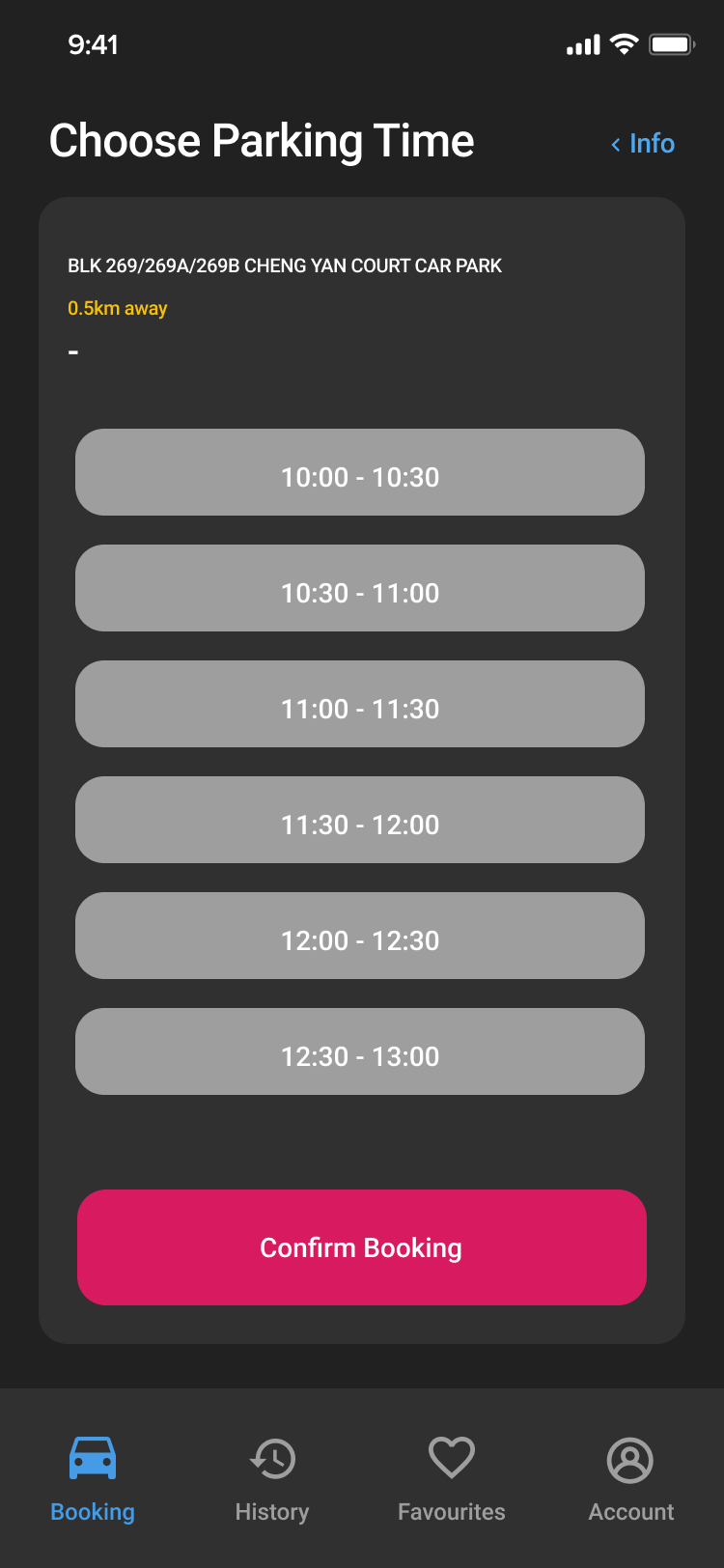
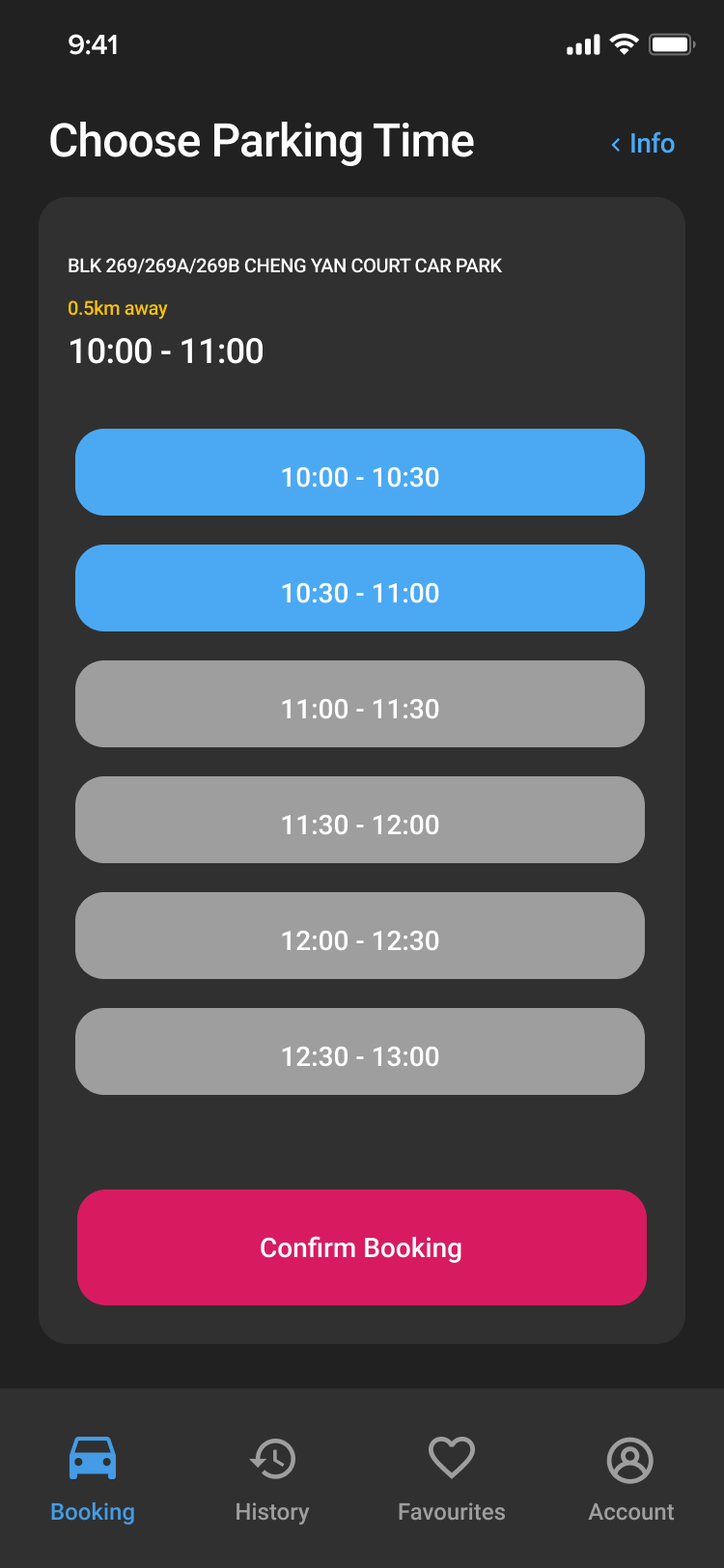
# 

# 

*Figure 4: Map and Car Park Information Screens*

Figure 4 illustrates the map and car park information screens. Whenever a user taps a car park on the map or taps an item in any car park lists from home, search, or my favourites screens, a car park information screen will appear. The car park information screen displays additional details about the car park and users can choose to view the car park on map or continue to choose a parking time for the booking.

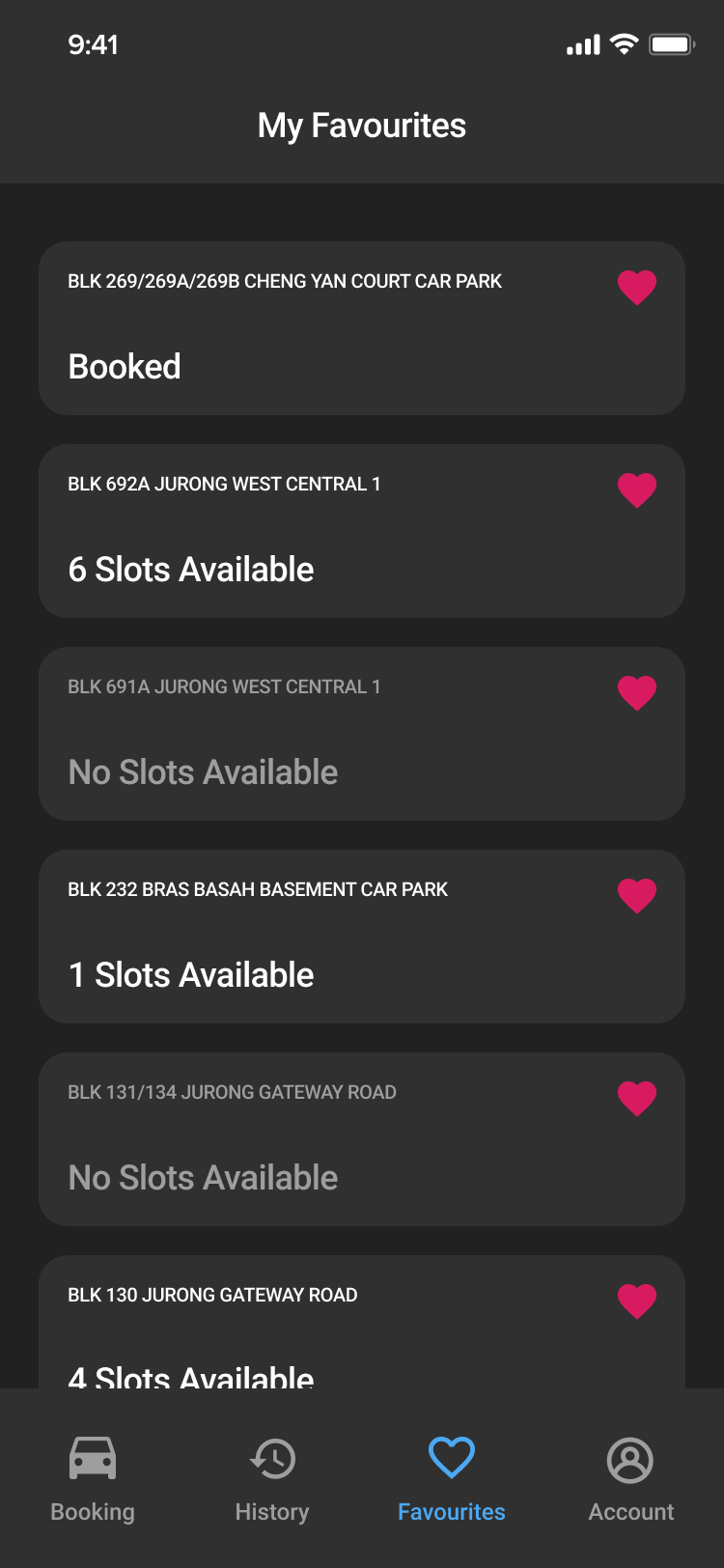
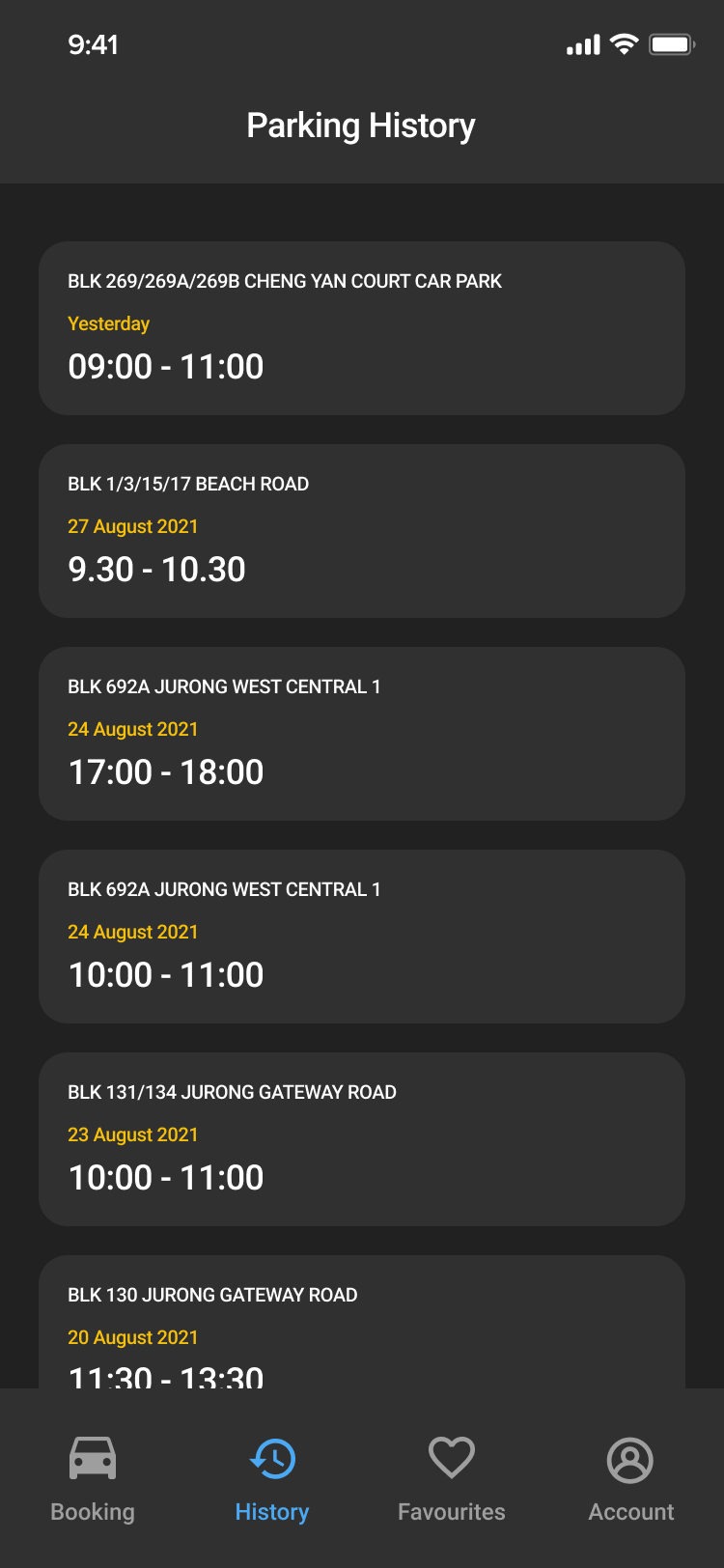
# 



*Figure 5: Choose Parking Time Screens*

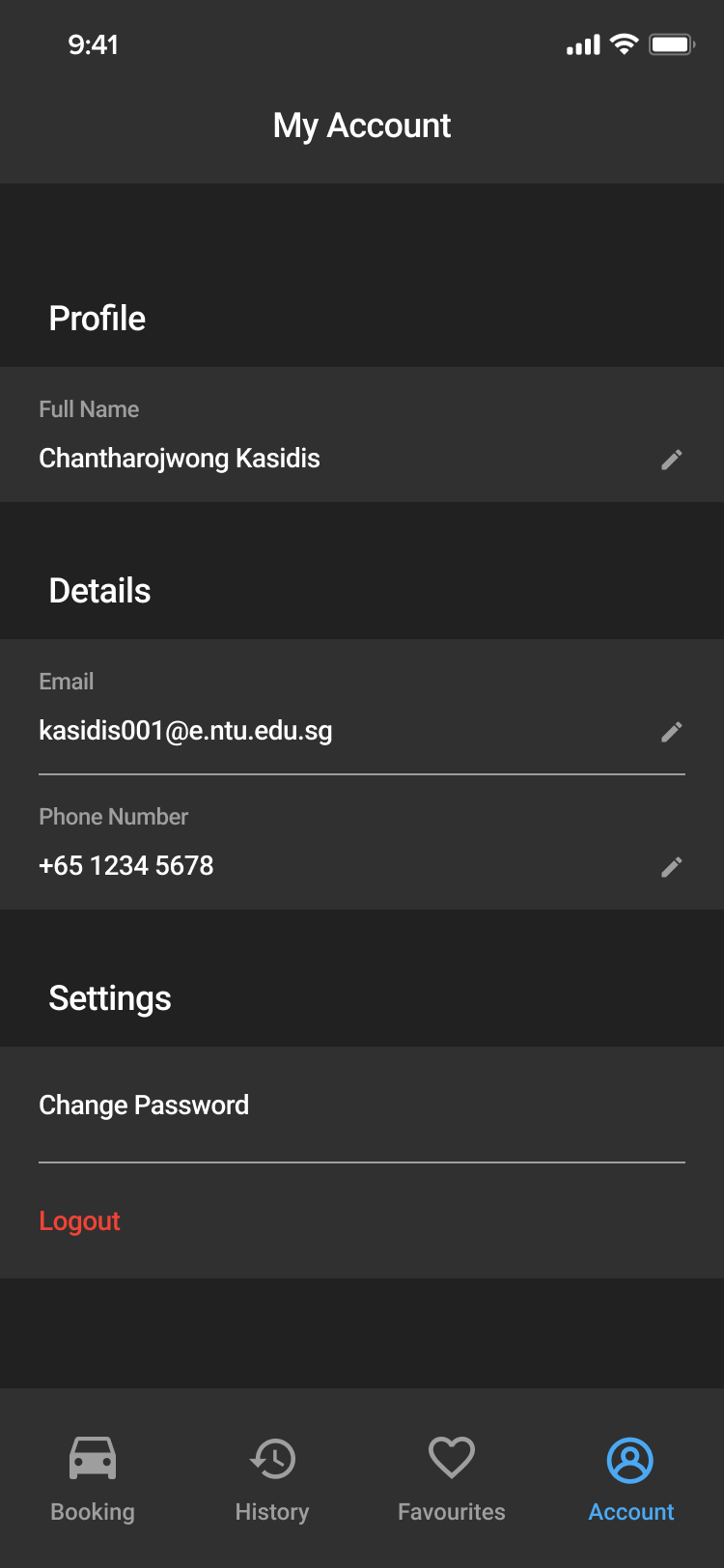
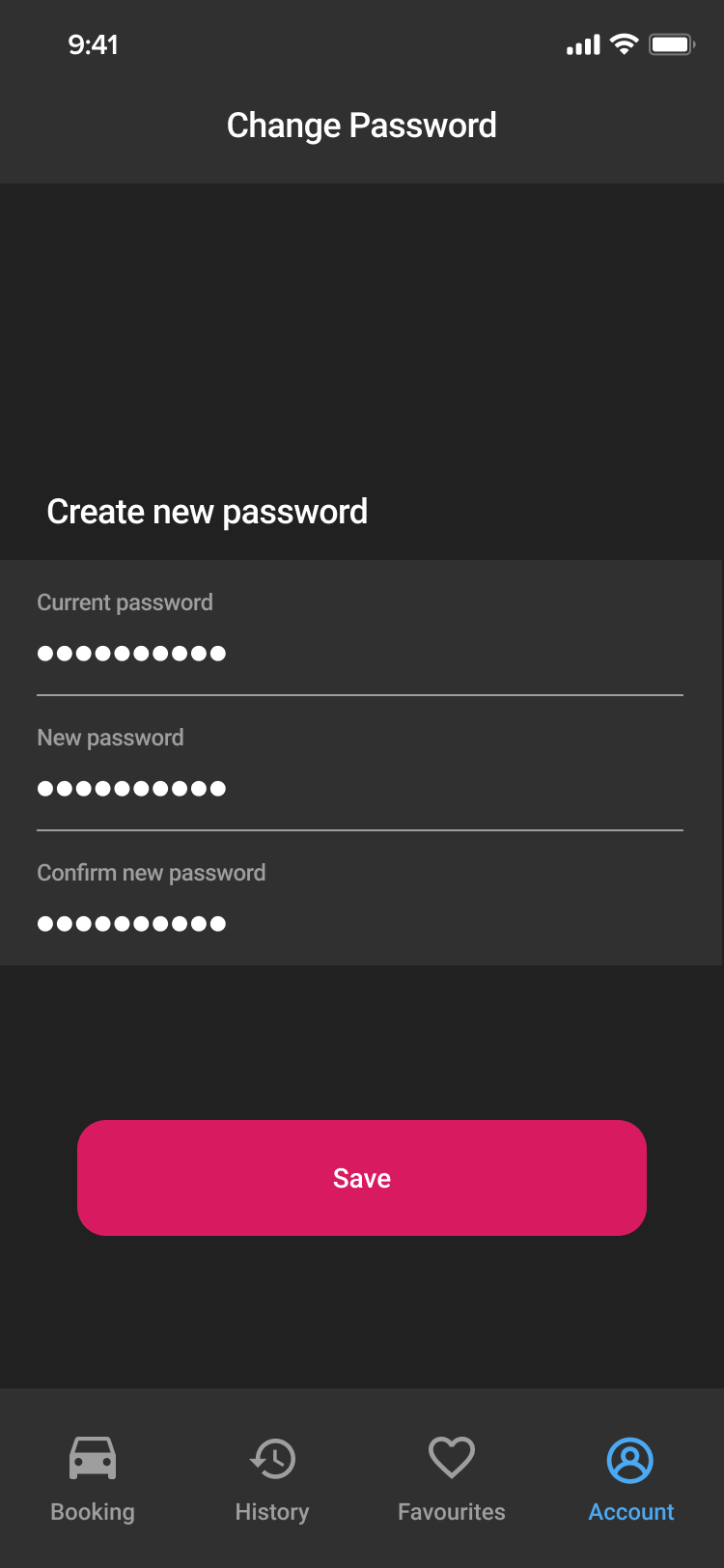
Figure 5 illustrates the choose parking time screens. The choose parking time screen displays parking time available for booking. Users can select time slots with an interval of half-hour. After selecting the preferred parking time and confirmed booking, the booking will appear in the home screen.

# 



*Figure 6: Parking History and My Favourites Screens*

Figure 6 illustrates the parking history and my favourites screens. Users can add or remove car parks to the my favourites screen easily by tapping the heart icon in any car park. From the my favourites screen, users can tap any item in the list to visit the car park information screen and book the car park.



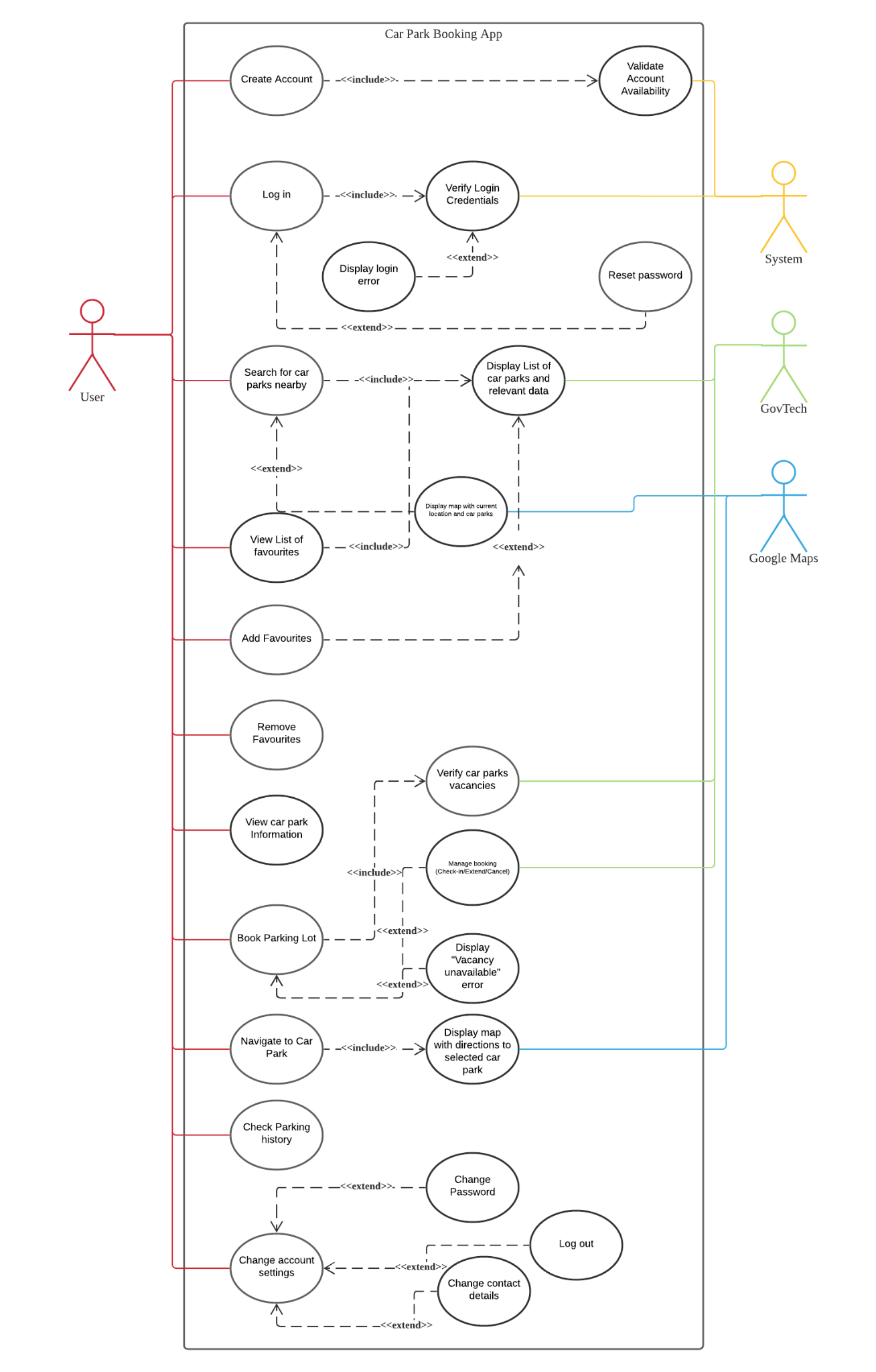
*Figure 7: My Account and Change Password Screens*

Figure 7 illustrates the account and change password screens. Users can change personal information and account password from my account screen. In the change password screen, the system requires the user to enter an old password before making any changes.

# 2. Functional Requirements

1. **Home**
   1. The application must allow the user to use the search function.
   2. The application must allow the user to select and retrieve more details about a particular car park.
      1. The details about the car pack must include the following information:
         1. The car park address
         2. The number of empty lots in the car park
         3. The car park rates
         4. The car park number
         5. The car park type
         6. The car park parking system
         7. The car park short-term parking
         8. The car park free parking
         9. The car park night parking
         10. The car park basement
   3. The application must allow the user to view current bookings
   4. The application must allow the user to view car parks near the user’s current location.
   5. The application must allow the user to make bookings.
   6. The application must allow the user to view the account booking history.
   7. The application must allow the user to view the account favourites list.
   8. The application must allow the user to manage an existing account.
2. **Search** 
   1. The application must allow the user to search for available car parks using a particular location.
      1. The location must be one of the following:
         1. The user’s current location
         2. A location specified by the user using its postal code
         3. A location specified by the user using its address
      2. The application must provide the user with the nearby available car parks from the search location.
         1. The application must at least display the 5 closest available car parks.
   2. The application must allow the user to search for available car parks using the date and time when the search was made.
      1. For a car park to be available, it must meet the following criteria:
         1. The car park must have at least one empty lot at the time of searching.
         2. The car park must be open for at least 1 hour starting from the time of searching.
3. **Registration**
   1. The application must allow the user to register for a new account.
   2. During registration, the application must validate that the user has entered valid information for the account fields.
      1. The account must have the following fields:
         1. Name
         2. Email address
         3. Password
         4. Phone number
      2. The application must validate that the password contains more than 8 characters.
      3. The application must validate that the email address is valid.
         1. The application must send out a verification email to the email address to check if the email exists.
         2. The application must receive a verification response from the verification email to check that the email address belongs to the user.
      4. The application must validate that the phone number is valid
         1. The application must send out a verification message to the phone number to check if the phone number exists.
         2. The application must receive a verification response from the verification message to check that the phone number belongs to the user.
      5. The application must only create an account after it has validated that all account field information is valid.
4. **Login and Security**
   1. The application must allow the user to login into existing accounts using the account field information.
      1. The login credentials must include the account email address and password.
   2. If the user enters invalid login credentials, the application must return an error message.
   3. The application must allow the user to recover a forgotten password
      1. The application must have a “forgot password” button that starts the password recovery process when clicked.
      2. The application must send out a password recovery email to the account email address.
5. **Account management**
   1. The application must allow the user to manage the account field information.
      1. The application must allow the user to view the account field information.
      2. The application must allow the user to edit the account field information
6. **Favourites**
   1. The application must allow the user to favourite car parks.
      1. Favourites must be tied to an account.
      2. The application must allow each account to have a favourites list of up to 10 favourites.
   2. The application must allow the user to retrieve the account favourites list.
   3. The application must allow the user to remove a favourite from the account favourite list.
7. **Booking**
   1. The application must allow the user to book a particular available lot in a particular car park.
      1. The application must only allow each account to have 1 active booking at a time.
      2. The application must only allow lots to be booked a maximum of 24 hours in advance.
      3. The application must only allow lots to be booked for a maximum time period of 24 hours.
      4. For each booking, the application must return a message stating whether the booking was successful or unsuccessful.
   2. The application must allow the user to check-in bookings.
      1. Checking-in bookings means that the user has arrived at the booked car park.
   3. The application must allow the user to cancel bookings.
   4. The application must allow the user to extend bookings in intervals of 30 minutes.
   5. The application must allow the user to view active bookings.
8. **History**
   1. The application must allow the user to retrieve details about previous bookings.
      1. The details about previous bookings must include
         1. The car park details
         2. The date and time of when the booking was made
         3. The booked time window
9. **Interface with other systems**
   1. The application must be able to retrieve the real-time data of available car parks using the GovTech Carpark Availability API
   2. The application must be able to retrieve the user’s current location using the Flutter Geolocation Plugin
   3. The application must allow the user to choose a car park and get directions to the car park using Google Maps.
10. **Formats for information to be processed.**
    1. Price rate
       1. Prices must be displayed in Singapore Dollars (i.e. S$) and have two decimal places (e.g. S$1.00)
    2. Car park rates
       1. Time units must be in hours ‘h’ and minutes ‘min’ (e.g. First 1h: S$1.50, subsequent 30 mins: S$0.50)
    3. Location
       1. Distance units must be in kilometers (i.e. km) and have two decimal places (e.g. 1.26km).
    4. Date and time
       1. Date and time must follow the Singapore Standard Time (SST) time zone.

## 2.1. Use Case Diagrams

****

## 2.2. Use Case Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Log in | | |
| Created By: | Supraja | Last Updated By: | Glenda |
| Date Created: | 6th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User enters login credentials |
| Preconditions: | 1. User account(s) exists in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User can favorite car parks 2. User can book a particular available lot in a particular car park 3. User can check-in bookings 4. User can retrieve details about previous bookings |
| Priority: | High |
| Frequency of Use: | 1-3 per lifetime |
| Flow of Events: | 1. User enters their credentials in the login interface 2. User selects the login button 3. System performs Use Case 10 Verify Login Credentials 4. User is logged into their account |
| Alternative Flows: | 1.AC.1 User resets password before logging in   1. User clicks on “Forgot Password?” 2. User enters their account’s email address 3. System will send a password recovery email to the account’s email address 4. User follows the instructions included in the password recovery email 5. System updates the account’s password in the database 6. Return to step 1 |
| Exceptions: | - |
| Includes: | 1. Verify login credentials |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Search for Nearby Parking Lots | | |
| Created By: | Supraja | Last Updated By: | Glenda |
| Date Created: | 6th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | To search for available car parks using a particular location |
| Preconditions: | 1. Phone must be connected to Wi-Fi/ mobile data 2. User needs to turn on GPS 3. User specifies location 4. User must be logged into an account |
| Postconditions: | 1. User obtains nearby available car parks from the search location. 2. User obtains car parks that are available at the date and time when the search was made |
| Priority: | High |
| Frequency of Use: | 0-20 times a day |
| Flow of Events: | 1. User inputs his/her query into the search bar in the home page 2. System retrieves a list of available car parks and their relevant data from the GovTech database 3. System displays a list of the car parks, with the car parks closer to the queried location listed first 4. User selects his/her preferred car park location |
| Alternative Flows: | 2.AC.1: User’s input does not match any keywords in search results   1. Display “No search results” and prompt user to edit his/her input 2. Return to step 1   2.AC.2.: User uses their current location to search rather than inputting one   1. User clicks on the “OPEN MAP” button on the home page 2. System retrieves a list of available car parks and their relevant data from the GovTech database based on the user’s current location 3. Google Map displays the car parks on the map 4. Return to step 4 |
| Exceptions: | - |
| Includes: | 1. Display list of parking lots and relevant data |
| Special Requirements: | - |
| Assumptions: | Google Maps uses GPS to locate car parks using data from GovTech |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | View List of Favourites | | |
| Created By: | Supraja | Last Updated By: | Glenda |
| Date Created: | 6th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | To retrieve the list of the user’s favourite car parks |
| Preconditions: | 1. Phone must be connected to Wi-Fi/ mobile data 2. User must be logged into an account |
| Postconditions: | 1. A list of up to 10 favourite car parks will be displayed on the Favourites page |
| Priority: | Low |
| Frequency of Use: | 0-5 times per day |
| Flow of Events: | 1. User navigates to the Favourites tab 2. Application will retrieve the list of user’s favourite car parks 3. Application displays a list of the user’s favourite car parks 4. User chooses car park |
| Alternative Flows: | 3.AC.1 User navigates to favourites page using “My Favourites” card on the home page   1. User clicks on the “My Favourites” card on the home page 2. Return to step 2 |
| Exceptions: | - |
| Includes: | 1. Display list of parking lots and relevant data |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Book Parking Lot | | |
| Created By: | Supraja | Last Updated By: | Glenda |
| Date Created: | 6th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User books a particular available lot in a particular car park |
| Preconditions: | 1. Mobile must be connected to Wi-Fi/ mobile data 2. User does not have any active bookings 3. User must be logged into an account |
| Postconditions: | 1. User books a lot in a car park at a particular time |
| Priority: | Medium |
| Frequency of Use: | 0-5 times per day |
| Flow of Events: | 1. User navigates to the Home/Bookings page 2. User clicks on a car park under the “Car Parks Near Me” list 3. User chooses an allocated time period for the car park 4. System returns a message stating that booking is successful |
| Alternative Flows: | 4.AC.1: Vacancy unavailable in selected car park   1. System displays “Vacancy unavailable” error 2. Return to Step 2   4.AC.2: User checks-in/ extends/ cancels booking |
| Exceptions: | - |
| Includes: | 1. Verify parking lot vacancies |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Change Account Settings | | |
| Created By: | Supraja | Last Updated By: | Glenda |
| Date Created: | 6th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User wants to change his/her account settings |
| Preconditions: | 1. User account(s) exists in the system database 2. Phone must be connected to Wi-Fi/ mobile data 3. User is logged into an account |
| Postconditions: | 1. Account details are changed according to what the user wanted |
| Priority: | Low |
| Frequency of Use: | 10 times per lifetime |
| Flow of Events: | 1. User navigates to the ‘Account’ page 2. User does one of the following to edit a specific account field:    1. User clicks on the pen icon next to the email input box to edit the email address    2. User clicks on the pen icon next to the phone no. input box to edit the phone number    3. User clicks on “change password” 3. User types in the new account field information 4. System updates the database to match the changes |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | Create Account | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Registration for a new user account |
| Preconditions: | 1. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User will have an account that can be logged into in the future |
| Priority: | High |
| Frequency of Use: | 1-3 times per lifetime |
| Flow of Events: | 1. User fills up the “name”, “email address”, “password”, “confirm password” and “phone number” fields. 2. User clicks the register button. 3. System performs Use Case 9 Validate Account Availability 4. System validates that the “password” and “confirm password” fields are identical. 5. System returns a message stating that the account has been successfully created |
| Alternative Flows: | 6.AC.1: System detects a mismatch between the password and confirm password fields   1. System displays an error message: “The 2 passwords you have keyed in are different. Please try again” 2. Return to Step 1 |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | Navigate to Car Park | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Use maps to navigate from current location to the selected car park |
| Preconditions: | 1. User account(s) exists in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User reached the selected car park |
| Priority: | Medium |
| Frequency of Use: | 0-10 times per day |
| Flow of Events: | 1. User clicks on the booking under “My Booking” 2. System will display the booking information 3. User clicks on “View on Map” 4. User clicks on the arrow button 5. System will prompt user to redirect to Maps or Google Maps 6. User clicks on either one and will be redirected to the selected application. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | 1. Display map with directions to selected car park |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 8 | | |
| Use Case Name: | Check Parking History | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User can check its parking history |
| Preconditions: | 1. User account(s) exists in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. A list of past bookings will be displayed in the system |
| Priority: | Low |
| Frequency of Use: | 0-10 times per day |
| Flow of Events: | 1. User clicks on the “History” tab at the bottom 2. A list will be displayed with the car park details, date and time of the booking made and the booked time window |
| Alternative Flows: | 8.AC.1: User navigates to history using the “Parking History” card on the home page   1. User clicks on the “Parking History” card on the home page 2. Return to step 2 |
| Exceptions: | 8.EX.2: User has not made any bookings yet   1. System will display “You have no past bookings.” Message |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 9 | | |
| Use Case Name: | Validate Account Availability | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | System |
| Description: | System will verify the account availability, send verification email and message and create the account |
| Preconditions: | 1. User account must not exist in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User account is created in the system’s database 2. User will be able to log in his/her registered account |
| Priority: | Medium |
| Frequency of Use: | 1-3 times per lifetime |
| Flow of Events: | 1. System will validate the user account availability in the system’s database. 2. System will create the user account in the system’s database. 3. System will send verification email to the registered email address. 4. System will send a verification message to the registered phone number. |
| Alternative Flows: | 9.AC.1: System detects the user account is already in the database   1. System displays an error message: “The email address you have entered is already registered.” 2. User will key in another email address. 3. User clicks on the “Register” button. 4. Return to Step 1 |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 10 | | |
| Use Case Name: | Verify Login Credentials | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | System will verify the login credentials based on the system’s database. |
| Preconditions: | 1. User account must exist in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User can access the various functions of the applications. |
| Priority: | Medium |
| Frequency of Use: | 1-3 times per lifetime |
| Flow of Events: | 1. User enters email address and password 2. User clicks on “Login” button 3. System validates the account by checking the user’s credentials against the system’s database. 4. System verifies the user login successfully. |
| Alternative Flows: | 10.AC.1 System detects the empty email address and/or password fields   1. System displays an error message: “Email address or password fields cannot be empty.” 2. User fills up the empty field. 3. User clicks the “Login” button. 4. Return to Step 3.   10.AC.2 User enter wrong credentials   1. System displays an error message: “Wrong email address and/or password. Please try again.” 2. Return to Step 1. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 11 | | |
| Use Case Name: | Add to Favourites | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User can add car park entries to the list of favourites |
| Preconditions: | 1. User account must exist in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User’s favourites will be saved to the system’s database |
| Priority: | High |
| Frequency of Use: | 0-20 times per day |
| Flow of Events: | 1. User scrolls to view the nearby car parks on the home page. 2. User clicks on the heart shape on the top right of each car park entry. 3. System will save the user’s favourite car park to the system’s database. |
| Alternative Flows: | 11.AC.1 User goes to search car park   1. User types in the search field on the home page 2. System will display a list of car parks based on the user entry. 3. Return to Step 2   11.AC.1: User goes to parking history   1. User clicks on the “History” tab on the bottom 2. System displays a list of past booking history 3. Return to Step 2 |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 12 | | |
| Use Case Name: | Remove Favourites | | |
| Created By: | Wee Hsuan | Last Updated By: | Wee Hsuan |
| Date Created: | 7th September 2021 | Date Last Updated: | 7th September 2021 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User can remove car park entries to the list of favourites |
| Preconditions: | 1. User account must exist in the system database 2. Phone must be connected to Wi-Fi/ mobile data |
| Postconditions: | 1. User’s favourites will be saved to the system’s database |
| Priority: | High |
| Frequency of Use: | 0-20 times per day |
| Flow of Events: | 1. User scrolls to view the nearby car parks on the home page. 2. User clicks on the filled heart shape on the top right of each car park entry. 3. System will remove the user’s favourite car park from the system’s database. |
| Alternative Flows: | 11.AC.1 User goes to search car park   1. User types in the search field on the home page. 2. System will display a list of car parks based on the user entry. 3. Return to Step 2   11.AC.2 User goes to parking history   1. User clicks on the “History” tab on the bottom. 2. System displays a list of past booking history. 3. Return to Step 2   11.AC.3 User goes to list of favourites   1. User clicks on the “My favourites” tab on the home page. 2. System will display the list of favourited car parks. 3. Return to step 2 |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

# 

# 3. Non-Functional Requirements

1. **Usability Requirements**
   1. When moving from page to page, the application must not require the user to remember more than 5 chunks of information. (reduce short-term memory load)
   2. The application must allow the user to reverse an action within 5 steps. (easy reversal of actions)
      1. The reversal of actions must include:
         1. Removing bookmarks from the bookmark list.
         2. Cancelling a booking.
   3. The application must return error messages when: (offer informative feedback)
      1. The user enters invalid inputs
      2. An application process was unsuccessful
   4. The application must be consistent.
      1. The application must have consistent sequences of actions for similar situations.
         1. The application must use the same transitions between pages throughout its interface.
      2. The application must have a consistent visual layout for its interface.
         1. The application must use the same font, shapes, type of labels and colour scheme throughout its interface.
         2. The application must use the same animations throughout its interface.
2. **Performance Requirements**
   1. The application must be able to maintain with little or no downtime occurring.
      1. 95% of users must be able to use the application without any system breakdown.
      2. The application must not lag when opening the application or using its features.
         1. The application must be able to return the display results to the user within 2 seconds.
      3. The application must not crash during run-time.
   2. The application must be able to support real-time retrieval of information from the GovTech Carpark Availability API.
      1. The application must be able to retrieve and return information using 1 second of the query.
3. **Security Requirements**
   1. During login, the application must mask the password field with asterisks to prevent any potential shoulder surfing.
   2. The application must apply Secure Hash Algorithm 1 (SHA-1) on all passwords before storing them into the database.
   3. During login, the application must verify whether the user has entered the correct password using SHA-1 hashing.
      1. The application must apply SHA-1 hashing on the input password.
      2. The application must compare the hashed input password with the hashed account password stored in the database.
4. **Extendibility Requirements**
   1. The application must be designed using Model-View-Controller architecture.

# 4. Data Dictionary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Car Park Type | A car park refers to an area or building where drivers can park their car temporarily. There are seven types of car park: surface car park, multi-story car park, basement car park, covered car park, mechanised car park, mechanised and surface car park and surface/multi-storey car park |
| Car Park Rates | Rates refer to the amount charged to drivers for parking their cars in the car park. Rates could be charged per 30 minutes or per hour. |
| Car Park Number | Every car park will have a unique number to identify them |
| Car Park Address | Car park address refers to the exact location the car park is located in. |
| Car Park Basement | A boolean function (Yes/No) to show if the car park has basements |
| Car Park Lots | A car park will have multiple lots for drivers to park their car. |
| Night Parking | A boolean function (Yes/No) to show if the car park allows cars to be parked overnight |
| Free Parking | Some car parks provide free parking during certain days such as Sunday and public holiday |
| Short term parking | Some car parks only allow season parking holders to park their cars there while some car parks allow non-season parking holders to park during a timeframe or the whole day |
| Type of Parking System | Each car park has a type of parking system which refers to the payment type. There are two types of parking system: Electronic parking and Coupon parking |
| X-/Y- Coordinates | The X- and Y- coordinates locate the exact location of a car park based on the scale of the map. |
| User | User refers to anyone using the application |
| Favourites | Favourites is a feature that allows users to add their favourite or frequently used car park and users can easily retrieve this list. |
| Registration | Registration is a feature that allows new user to sign up for an account and use the application features such as bookmarks and book a car park |
| GPS | GPS refers to Global Positioning System which can detect the user’s current location |
| Shoulder Surfing | Shoulder spying is the practice of spying on the user when they are typing their credentials into an electronic device in order to obtain their password. |
| Search | Search is a feature that enables users to access available car parks based on a few filters. |
| Error message | An error message is a message displayed on the screen when an incorrect input is given by the user or when an application process is unsuccessful. |
| Booking | Booking is a feature that allows users to reserve a lot at a specific car park for a specified amount of time. |
| Check-In | Check-in is a feature that allows the user to indicate that he/she has arrived at their booked car park and will be subsequently using the booking. |
| History | History is a list of previous bookings that the user had made using a specific account. Each account has a different history. |
| Account Fields | Account fields are information fields that are unique to each user account. Every account has four account fields: name, email address, password, and phone number. |
| Login Credentials | Login credentials are account fields used to login to a specific user account. There are two account fields needed to login: email address and password. |
| Verification Email | Verification email is an email sent out to the email address that the user enters during registration. The user can verify that the email address belongs to him/her by following the instructions written in the verification email. |
| Verification Message | Verification message is a message sent out to the phone number that the user enters during registration. The user can verify that the phone number belongs to him/her by following the instructions written in the verification message. |
| Password Recovery Email | Password recovery email is an email sent out to the account email address at the user’s request. The user can recover his/her forgotten account password by following the instructions written in the verification email. |