Use Cases

for

EasyPark Application

Version 1.0 approved

Prepared by Anh Tu, Madhav

Team NoobSW

8th September 2019

Revision History

Name	Date	Reason For Changes	Version
Anh Tu	14/9/2019	Refining use case descriptions based on feedback from lab 1	1.1
Madhav	22/9/2019	Minor fixes on presentation of the documents	1.2

1. Enter Destination

Use Case ID:	1.12		
Use Case Name:	Enter Destination		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary), Google Map (supporting)
Description:	The use case collects destination keywords from the user, validates it and shows a list of potential matching locations for user to choose. This allows user to pinpoint exact destination they are heading for. (E.g. going to a mall in Orchard Area)
Preconditions:	1.The application must be opened by the user.
Postconditions:	1. The application must receive the exact destination the user wanted to search for car parks. Or 2. The application must notice that the destination entered by the user could not be found.
Priority:	1
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application requests the User for their interested destination on the first page of application. 2. The User enters their interested destination into the text box. 3. The application queries the Google Map application by including its List Destination use cases, while the User is entering their interested destination. 4. The Google Map application verifies if the destination being entered by the User is within Singapore. 5. The Google Map application provides a list of possible destinations in Singapore based on the keywords entered by the User, who is searching for his interested destination. 6. The application displays the list of possible destinations as a drop-down list on the same page. 7. The User selects a destination from the drop-down list which is closest to the interested destination they have entered.
Alternative Flows:	*1.12.AC.1: At step 3, if the application is unable to query the Google Map Interface* 1. The application displays the message "Unable to connect to Google Map. Please check your connection!". 2. The application returns to step 3 of Normal Flows to query the Google Map application interface. *1.12.AC.2: At step 4, if the destination is not within Singapore or the Google Map cannot find the relevant destinations* 1. The application displays the error "Sorry, your destination is not

	found in Singapore. Please try again! ". 2. The application returns to step 2 to wait for User's new input
Exceptions:	N/A
Includes:	Provide List Destination
Special Requirements:	N/A
Assumptions:	1. User must have Google Map. 2. User must have active internet connection. 3. User is aware of their destination.
Notes and Issues:	

2. Search Car Parks

Use Case ID:	1.15		
Use Case Name:	Search Car Parks		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary), Government Car Park Database (supporting)
Description:	The use case helps user to search for carparks within 5km area away from the destination they are heading for.
Preconditions:	1. User must already have selected their destination from the destination list provided by Google Map And 2. Connection to government database API must already have been established
Postconditions:	1. A list of nearby car parks to the interested destination must be shown.
Priority:	1
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The User chooses to search for the car parks near their selected destination by clicking on the "Next" button, besides the destination's search box. 2. The application sends query with the User's destination to the Government Car Park Database application interface (API). 3. The Government Car Park Database returns a list of car parks available in Singapore to the application via the API. 4. The application filters out car parks that are more than 5km away from the destination. 5. The application changes to a new page to display the car parks. 6. By default, a list view of nearby car parks is displayed using the included Display Car Parks use case.
Alternative Flows:	N/A
Exceptions:	*1.15.EX.1: User can go back to the first page and reselect their destination*

	1. The User clicks on Back button in application. 2. The application pops the first page out from the navigator stack. 3. The application displays the first page and requests for User amendment in destination textbox.
Includes:	Display Car Parks
Special Requirements:	N/A
Assumptions:	1. User must have Google Map. 2. User must have active internet connection. 3. User is aware of their destination.
Notes and Issues:	

3. Select Car Park

Use Case ID:	1.21		
Use Case Name:	Select Car Park		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary), Google Map (supporting)
Description:	After the car parks are displayed to the user, this use case describe how user can select a car park and navigate to it.
Preconditions:	1. The application must currently be displaying a list of car parks in List View for selection. Or 2. The application must currently be displaying pin points of car parks in Map View for selection.
Postconditions:	1. Google Map must be invoked to guide the user to their destination
Priority:	2
Frequency of Use:	1 to 5 times per use of the application.
Flow of Events:	1. The User clicks on a car park card in the list view (or a pinpoint on the map view) in Display Car Parks use case. 2. The application displays a dialog box with selected car park full information: car park name, car park address, car park number, number of spaces available and distance from destination. 3. The application offers to navigate the User to the selected car park by button "Navigate". 4. The User clicks on the button. 5. The Google Map application opens with the selected car-park’s address as the location. 6. The User follows the instructions given by Google Map in the Navigate included use case to reach the desired car park.

Alternative Flows:	N/A
Exceptions:	*1.21.EX.1: User can go back to the display view of Car Parks:* *1. User clicks on Back button on the application.* 2. The application closes the dialog box and goes back to the previous selected display view (either list view or map view) using the Display Car Parks use case
Includes:	Display Car Parks, Navigate
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

4. Change Display Type

Use Case ID:	1.17		
Use Case Name:	Change Display Type		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary)
Description:	The user can select to view the car parks as pinpoints on a map or as items in a list. This use case describes how to change in between the two displays.
Preconditions:	1.The application must be in either List View or Map View
Postconditions:	1.Either list view or map view must be invoked, depending on the user's choice.
Priority:	3
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application presents choice for User to change the layout to display the car parks by buttons. 2. The User makes decision to change the display view for car parks as a list or on a map by clicking either of the buttons. 3. The application takes note of the User decision and check it with the current display view of car parks. 4. The application makes changes to the current view UI using the included Display Car Parks as needed. 5. The application presents the car parks as either on a Map (Map View) or in a list (List view).
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Display Car Parks
Special Requirements:	N/A
Assumptions:	N/A

Notes and Issues:	

5. Change Search Distance

Use Case ID:	1.19		
Use Case Name:	Change Search Distance		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary)
Description:	The user can optionally narrow down the search distance for their car parks. The maximum and default searching distance is 5km but the user can narrow this down through a slider should there be too many nearby car parks around in the area.
Preconditions:	1. The application must be in either List View or Map View
Postconditions:	1. The application hides the destination outside the search range
Priority:	3
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application displays a sliding bar with distance range to search for the car parks. By default, 5km is selected. 2. The User slides the point on the sliding bar to a desirable range. 3. The application reads the adjusted distance range as stipulated by User. 4. The application iterates through information of car parks provided by the Government Car Park Database in Provides Car Parks Information and hide all car parks further than the distance chosen to the destination. 5. The application displays the data provided by the Government Car Park Database using the included Display Car Parks use case
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Display Car Parks
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

6. Change Sort

Use Case ID:	1.20		
Use Case Name:	Change Sort		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary)
Description:	The user can choose to view car parks displayed in the list view with two orders: by vacancy left and by distance away from them. This can be done by clicking on the sorting buttons displaying on UI.
Preconditions:	The application must only be in List View.
Postconditions:	The order of the car parks cards on the List View must be changed based on the criteria selected
Priority:	3
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application provides the User with 2 choices of sorting the car parks within the range to destination: by distance and by empty slots available. 2. The User selects 1 of the 2 choices to sort the list of car parks. 3. The application arranges the available car parks according to the information received from Provides Car Parks Information. 4. The application uses the included Display Car Parks use case to show a sorted list by either the shortest distance or by the emptiest space. 5. The User further investigates the displayed car parks and selects.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Display Car Parks
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

7. Get Help

Use Case ID:	1.24		
Use Case Name:	Get Help		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary)
Description:	First-time users and users that require help on using the application can click on the Help button on the first page to receive helping information.

Preconditions:	N/A
Postconditions:	1. User must be informed on how to use the application
Priority:	2
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The User clicks on "Help" button on first page of the application to require help. 2. The application displays a dialog box to display the helpful information. 3. The User reads the information. 4. The User clicks on the "X" sign to close the help dialog box. 5. The application goes back to the first page and wait for User's next interaction
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

8. Go Back

Use Case ID:	1.23		
Use Case Name:	Go Back		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary)
Description:	The user can traverse to previous pages or closing the app (on first page) using the Back button on the application UI. This allow them to amend information if needed.
Preconditions:	
Postconditions:	1. The application must show the previous page on the navigation stack to the user. Or 2. The application must close itself and return to the phone’s home screen
Priority:	3
Frequency of Use:	1 to 5 times per use of the application
Flow of Events:	1. The User clicks on Back button on their phone. 2. The application terminates any ongoing tasks on the current page view. 3. The application restores the state of the previous page view by popping them out from the navigator stack. 4. The application UI displays the previous page view. 5. The User interacts with the previous page view.

Alternative Flows:	*1.23.AC.1: If the back button is clicked on the application's home page:* 1. A dialog box pops up and asks the user to confirm about exiting the application. 2. User chooses "Yes". 3. The application terminates, freeing up resources on user's phone.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

9. Display Car Parks

Use Case ID:	1.18		
Use Case Name:	Display Car Parks		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	User (primary), Google Map (supporting)
Description:	Once information of car parks are provided by the Government API, the application needs to display them in a suitable UI for user. This UI need to be interactive to users and need to be refreshable whenever user make a change on range, order of sorting or type of display.
Preconditions:	1. The application must receive information of car parks from the API of Government Car Parks Database.
Postconditions:	1. The car parks information must be displayed in List View or Map View with different customisation depending on user's selection in other use cases.
Priority:	1
Frequency of Use:	1 to 5 times per use of the application
Flow of Events:	1. The application receives car parks information from the Government Car Park Database in the included Provides Car Parks Information use case. 2. The application checks for requested display view sent by User in Change Display Type. By default, the List View is selected. 3. The application filters out car parks that is out of the range chosen by the User in Change Search Distance. By default, it will filter out any car parks more than 5km away from the User's destination 4. The application orders the remaining car parks based on distance or empty space available depending on sorting criteria in Change Sort. By default, sorting order in distance is selected. 5. The application displays a new page with a list of car parks in the

	sorting order. 6. The application shows information of the car parks: car park name, car park address, car park number, number of spaces available and distance from destination; each in the form of a card with color-coded boundary to indicate vacancy. 7. The User views the information and select car park for further investigation as needed.
Alternative Flows:	*1.18.AC.1: If user selects Map View Display at step 2:* 1. The application filters out car parks that are out of the range chosen by the User in Change Search Distance. By default, it will filter out any car parks more than 5km away from the User's destination 2. The application displays a new page with a map view of car parks. Car parks are shown as pins on the map view. 3. The application hides car parks out of 5km 4. The application shows the car parks within 5km to destination on the map with a pin point color-coded symbol, indicating the empty space of the car parks. *1.18.AC.2: If the application is in List View and user changes the range of searching for car parks in step 3:* 1. The application orders the remaining car parks within selected range based on distance or empty space available depending on sorting criteria in Change Sort. By default, sorting order in distance is selected. 2. The application refreshes the list view of car parks. 3. The application hides car parks that are out of the search range as determined by User in Change Search Distance. 4. The application shows information of the car parks: their number, distance to destination and empty slots, each in a card with color-coded boundary to User. 5. The user view the information and select car park for further investigation as needed. *1.18.AC3: If the application is in Map View and user changes the range of searching for car parks in step 3:* 1. The application filters out car parks that is out of the range chosen by the User in Change Search Distance. 2. The application refreshes the map view of car parks. 3. The application hides car parks out of range of search. 4. The application shows the car parks within range of search to destination on the map with a pin point color-coded symbol, indicating the empty space of the car parks. 5. The User can see the distance between the destination and the car parks by interacting with the map view. *1.18.AC.4: If the user selects sorts by empty space in List View in step 4:* 1. The application orders the remaining car parks based on empty space available.

	2. The application refreshes with a list of car parks in order of the emptiest space available first. 3. The application shows information of the car parks: their number, distance to destination and empty slots, each in a card with color-coded boundary. 4. The User views the information and select car park for further investigation as needed.
Exceptions:	N/A
Includes:	Provides Car Parks Information
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

10. Provides Car Parks Information

Use Case ID:	1.16		
Use Case Name:	Provides Car Parks Information		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	Government Car Park Database (primary)
Description:	After the user selects their destination and decide to search for nearby car parks, the application and Government API need to communicate so that information of nearby car parks are given to the application for display and further interaction with users.
Preconditions:	1. The application must supply the destination and its local range to search for car parks. And 2. The application must include the information it needs from the Government Database
Postconditions:	1. A List of carparks, their vacancies and their distances to destination must be returned to the application for display. Or 2. A carpark with detailed information must be returned to the destination (when the user selects carpark). Or 3. A message must be sent back to the application if it is unable to connect to the database's API
Priority:	2
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application sends a query to the Government Car Park Database using its application interface (API) for relevant

	information and attributes of car parks. 2. The Government Car Park Database finds all schema with car parks in it. 3. The Government Car Park Database select those attributes of car parks being requested. 4. The Government Car Park Database API sends the car parks information back to the application. 5. The application receives the data. 6. The application proceeds with displaying the data in its Display Car Parks use case .
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

11. Navigate

Use Case ID:	1.22		
Use Case Name:	Navigate		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	Google Map (primary)
Description:	After user selects a car park and choose to navigate to it on the application, the Google Map application will help the user to reach the selected car parks
Preconditions:	1. A car park must be selected by user. And 2. The application must send data of current location and the carpark destination to Google Map
Postconditions:	1. The user must reach his destination by following Google Map's instruction
Priority:	3
Frequency of Use:	1 time per use of the application
Flow of Events:	1. The application sends the selected car park's address to the Google Map application. 2. The Google Map application opens with the above information. 3. The User selects “Navigate” to start the Navigation mode 4. The Google Map starts navigation mode and gives directions to the user to reach the selected car parks. 5. The User follows the direction to reach the destination. 6. The Google Map tells the User once he has reached the

	destination.
Alternative Flows:	N/A
Exceptions:	*1.22.EX.1: User can go back to EasyPark application from the Google Map application*: 1. The User clicks on Back button in Google Map application. 2. The Google Map pauses current activity. 3. The Google Map switches to EasyPark application.
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

12. Provide List Destination

Use Case ID:	1.13		
Use Case Name:	Provide List Destination		
Created By:	Anh Tu	Last Updated By:	Madhav
Date Created:	8/9/2019	Date Last Updated:	22/9/2019

Actor:	Google Map (primary)
Description:	As user enters their intended destination or keywords relating to their destination, the Google Map API needs to return a list of possible matching destination, which will be display as a drop down on the application for user to choose.
Preconditions:	1. The Application must receive and forward user's input of their interested destination to the Google Map API
Postconditions:	1. Google Map API must return a list of possible destination that the user is searching for. Or 2. Google Map API must return a message to the application informing about the connection problem to the API. Or 3. Google Map API must return a message to the application informing if no possible destination is found.
Priority:	
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application provides the User's destination input to the Google Map application interface (API). 2. The Google Map API searches for matching destinations to the user input within Singapore area in its database while the user is typing. 3. The Google Map API sends an approval message to the application if there is at least 1 location in Singapore containing the user's keywords 4. The Google Map API queries its database and collect all

	possible destinations in Singapore. 5. The Google Map API sends the possible destinations to the application. 6. The application displays these suggested destinations in drop down list, under User input textbox.
Alternative Flows:	*1.13.AC.1: If there is no internet connection:* 1. The Google Map database cannot be queried. 2. The Google Map API sends message to inform the application about the connection problem. 3. The Google Map API returns to step 2 and continuously query the database until timeout again. *1.13.AC.2: If there is no matching destination in Singapore* 1. The Google Map API sends a message to inform about the search results. 2. The application display a message "Your destination is not found! Please try again!"re
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	