
Use Cases

for

<CarParkGoWhere>

Version 1.4 approved

Prepared by Ong Jing Jie, Seow Jing Woon, Cheng Yi Feng,
Jumana Haseen, Damien Lee Han Wei, Haoren

<Group 46>

<01-Sep-2024>

Revision History

Name	Date	Reason For Changes	Version
Yi Feng	4/9/2024	Initial Version	1.0
Jing Jie	10/9/2024	Added Use Case Description	1.1
Yi Feng & Jing Jie	27/10/2024	Added new case , update use case	1.2
Yi Feng & Jing Jie	1/11/2024	Update use case	1.3
Yi Feng & Jing Jie	8/11/2024	Finalize use case	1.4

1. Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

2. User's identity has been authenticated.
3. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of the item in the database has been updated with a new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use

case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.







2.12. Assumptions

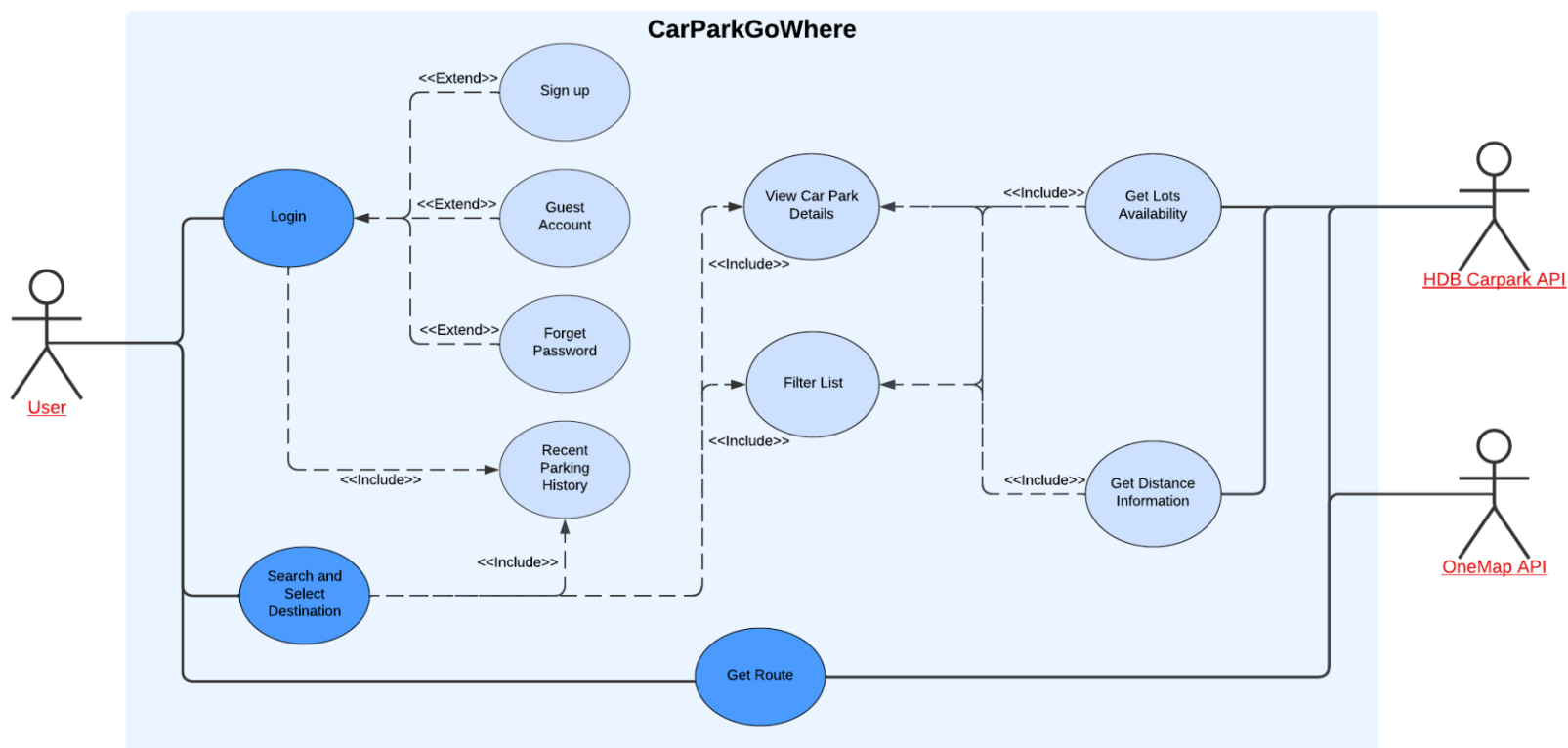
List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case Diagram Elements

Component	Symbol	Descriptions
Actor		An external entity that interacts with system
Use Case		Unit of functionality performed by system, which yields result / value for Actor
System		It contains all the use cases that describe the system's functionalities and interactions.
Association	<ol style="list-style-type: none">    	<ol style="list-style-type: none"> Connects Actors to Use Cases(s) in which they participate. <<Include>> denotes a dependency on another Use Case. <<Extend>> denotes an optional dependency on another Use Case

Use Case Diagram

Use Case Template

Use Case ID:	UID-001		
Use Case Name:	General Overview		
Created By:	Ong Jing Jie	Last Updated By:	Ong Jing Jie
Date Created:	05-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System HDB Carpark API OneMap API
Description:	General overview of how the user can use the system to access the functions of the application to find the most suitable parking lots in car parks based on the user's final destination and other factors such as pricing and availability.
Preconditions:	<ol style="list-style-type: none"> 1. The user has access to the website and is connected to the internet. 2. The system has access to real-time data from authoritative sources such as HDB Carpark and OneMap data providers. 3. The user's location services (GPS) are enabled and accessible to the system.
Postconditions:	<ol style="list-style-type: none"> 1. The user is provided with a list of parking spaces near the selected destination, which users can sort by criteria like proximity, price and lots availability . 2. The user can view real-time availability and pricing of the selected parking spaces. 3. The user can choose a parking space and get the fastest / shortest route.
Priority:	High
Frequency of Use:	High
Flow of Events:	
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	The system must ensure data privacy and comply with local regulations regarding location-based services.
Assumptions:	<p>The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date.</p> <p>The system assumes that users have basic navigation and search skills on the website.</p>
Notes and Issues:	

Flow Of Events

Actor Step	System step
1. User opens up the website	
	2. The system displays a login page, prompting the User to input his username and password to sign in, if not to sign up or continue as a guest account.
3. The user inputs his username and password correctly.	
	4. The system verifies the username and password and grants access.
	5. The system will transition to a map interface.
	6. The system will display a list of recent parking history, enabling users to select their recent car parks.
7. User inputs desired destination	
	8. The system queries real-time parking data from HDB Carpark API for parking spaces near the specified destination.
	9. The system retrieves real-time traffic and lots availability data.show
	10. The system displays car park markers around the specified destination and shows a list of car park with details like Carpark Name , price rate , lots availability, distance to carpark.
11. The user selects a preferred parking space based on the provided option.	
	12. The system displays the fastest route from the user's location to the selected parking space.
13. The user can view the route on a map interface.	
	14. The system refreshes the page to update real-time availability and pricing information

	for the selected parking space.
15. The user navigates to the selected parking space using the route provided by the system.	

Flow of events: Alternative Flow

ID	Description
AF-S3	The user's username or password is incorrect. The system would not grant access and prompt the user to try again until the user chooses guest login, sign up for a new account or reset the password, and input the correct username and password. The system will return to step 2.
AF-S6	If the guest account is chosen, this step will be skipped and the system to proceed to step 7.
AF-S10	Users can customise the selection based on their needs instead of the result of the algorithm. The system will proceed to step 13 upon User selecting car park.

Flow of Events: Exception

EX1-Destination does not exist	System will display an error message "No Results Found" .
EX1: System unavailable	The user is informed of system maintenance or downtime.

Use Case ID:	UID-002		
Use Case Name:	Login		
Created By:	Ong Jing Jie	Last Updated By:	Cheng Yi Feng
Date Created:	05-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System Database
Description:	User login using username and password to access system services.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The system does not currently have a User logged in. 3. The System must be connected to the database. 4. The database must be online.
Postconditions:	<ol style="list-style-type: none"> 1. The user can access the map interface to access system services.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The system will display the login interface. 2. The user inputs a username and password and clicks the "sign in" button. 3. The system verifies the login credentials with the database. 4. Upon successful verification, the system will transition to the map interface. 5. Users then have access to system services.
Alternative Flows:	<p>AF-S2:New User signup</p> <ol style="list-style-type: none"> 1. The system will have a signup interface and allow users to create their account. The system will store the details in the database for further use. 2. Upon successful creation, the system will revert to the login interface, step 1. <p>AF-S2:User chooses to use a guest account</p> <ol style="list-style-type: none"> 1. The System will bypass step 3 and go to step 4. <p>AF-S3:Incorrect credentials</p> <ol style="list-style-type: none"> 1. The system will inform the user that the username or password is wrong. 2. The system will return to Step 1.
Exceptions:	<p>EX1:System unavailable</p> <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime.
Includes:	UID-005
Special Requirements:	
Assumptions:	<p>The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date.</p> <p>The system assumes that users have basic navigation and search skills on the website.</p>
Notes and Issues:	

Use Case Name:	Sign Up		
Created By:	Ong Jing Jie	Last Updated By:	Ong Jing Jie
Date Created:	05-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System Database
Description:	Creating a new account to access system services.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The system does not currently have a User logged in. 3. The System must be connected to the database. 4. The database must be online. 5. TheUser has no existing account
Postconditions:	<ol style="list-style-type: none"> 1. The user can log in using his newly created account
Priority:	High
Frequency of Use:	Once for new users
Flow of Events:	<ol style="list-style-type: none"> 1. The system requests the User to input a username, email and password and re-entered password. 2. The system verifies that the email used is not repeated with other accounts. 3. Upon successful verification, the system updates its database with the credentials 4. The system will transition back to the login interface.
Alternative Flows:	AF-S1:Text fields are not filled appropriately <ol style="list-style-type: none"> 1. The system will prompt the user to fill in the required field AF-S2:Username is already in use. <ol style="list-style-type: none"> 1. The system will prompt an error message "Username is taken" and go back to step 1.
Exceptions:	EX1:System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime.
Includes:	
Special Requirements:	
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-004
--------------	---------

Use Case Name:	Guest Account		
Created By:	Ong Jing Jie	Last Updated By:	Cheng Yi Feng
Date Created:	05-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	Guest User
Description:	An alternative way to access system services without having to have an account.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The system does not currently have a User logged in. 3. The System must be connected to the database. 4. The database must be online. 5. TheUser has no existing account
Postconditions:	<ol style="list-style-type: none"> 1. The user can access the map interface to access system services.
Priority:	High
Frequency of Use:	High for Users that do not wish to create an account, no use for users with an account.
Flow of Events:	<ol style="list-style-type: none"> 1. The system will display the login interface. 2. The user will select "Continue as Guest". 3. The system will transition to the map interface.
Alternative Flows:	
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime.
Includes:	
Special Requirements:	
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-005		
Use Case Name:	Recent Parking History		
Created By:	Ong Jing Jie	Last Updated By:	Cheng Yi Feng
Date Created:	05-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System Database
Description:	To allow users to quickly access a history of car parks they have recently visited, facilitating easy reselection of preferred locations.
Preconditions:	<ol style="list-style-type: none"> 1. The user has access to the website and is connected to the internet. 2. The system has access to real-time data from authoritative sources such as HDB Carpark API and OneMap API data providers. 3. The user's location services (GPS) are enabled and accessible to the system. 4. System must have an account logged in. 5. The system has previously stored data on the user's parking activities.
Postconditions:	<ol style="list-style-type: none"> 1. The user is provided with a list of parking spaces that the user has recently visited.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user accesses the recently visited car parks feature from the application menu. 2. The system retrieves the user's parking history from the database. 3. The system presents a list of car parks the user has visited, sorted by date with the most recent visits at the top. 4. The user browses through the list and can select a car park to view more details or to reinitiate a route search to that location.
Alternative Flows:	AF-S2: If no history is available <ol style="list-style-type: none"> 1. The system displays a message indicating that there are no recently visited car parks.
Exceptions:	
Includes:	
Special Requirements:	The system must ensure data privacy and comply with local regulations regarding location-based services.
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-006		
Use Case Name:	Search And Select Destination		
Created By:	Cheng Yi Feng	Last Updated By:	Cheng Yi Feng
Date Created:	06-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User HDB Carpark API OneMap API
Description:	Allow users to input and select a destination for finding nearby parking lots.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database must be online. 4. The user must have logged in or using a guest account.
Postconditions:	<ol style="list-style-type: none"> 1. A destination is set for the parking lot search.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User inputs the address line or postal code of his destination. 2. The system verifies the destination and prepares to display nearby parking options. 3. System will display the route from user location to destination. 4. The system will query for data such as lot availability, price and car park coordinates from HDB Carpark API. 5. The system will calculate the distance between the current location and the car park. 6. The system will display the first car park detail with respective prices, lot availability, distance between the current location and car park. 7. The user will then select his preferred car park. 8. System will display the route from the user location to the selected car park. 9. System will log the selected car park in the parking history in the database.
Alternative Flows:	<p>AF-S1: Users can view car park details by clicking/tapping on the car park from the list in the sidebar.</p> <p>AF-S5: Users can utilize the filters to indicate his preference of car parks in terms of:</p> <ol style="list-style-type: none"> 1. Shortest distance between current location to car park 2. Cheapest car park 3. Highest parking lots available
Exceptions:	<p>EX1: System unavailable</p> <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime.
Includes:	<p>UID-011</p> <p>UID-012</p> <p>UID-013</p>
Special Requirements:	

Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-007		
Use Case Name:	Get Route		
Created By:	Cheng Yi Feng	Last Updated By:	Ong Jing Jie
Date Created:	06-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User OneMap API		
Description:	To provide the user with the fastest route to the selected car park.		
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database and API must be online. 4. Users must have logged in or using a guest account. 5. System and map API must have access to the user's current location. 		
Postconditions:	<ol style="list-style-type: none"> 1. The user views the driving route to the selected car park and the walking route from the car park to the destination. 		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> 1. Users select the desired car park from the list of choices. 2. The system captures the user's current location and the location of the selected car park. 3. The system sends a request to the OneMap API to generate a route from the user's current location to the selected car park. 4. The OneMap API provides the route data, which the system then displays on the map, highlighting the driving route in green. 5. The recommended route to the selected car park will be displayed for the user's aid. 		
Alternative Flows:			
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime. 		
Includes:			
Special Requirements:			
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.		
Notes and Issues:			

Use Case ID:	UID-008		
Use Case Name:	Get Lot Availability		
Created By:	Ong Jing Jie	Last Updated By:	Ong Jing Jie
Date Created:	06-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System HDB Carpark API OneMap API
Description:	To provide the system with real-time information on the availability of parking spaces near their destination.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database and API must be online. 4. Users must have logged in or using a guest account. 5. System and map API must have access to the user's current location. 6. The user has input a destination.
Postconditions:	<ol style="list-style-type: none"> 1. The system is provided with up-to-date information about parking availability near their destination.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The system sends a request to the HDB Carpark API to retrieve the real-time availability of nearby parking spaces. 2. The system receives and stores parking availability data, indicating which car parks have available spots.
Alternative Flows:	AF-S1: No Lot Availability Data <ol style="list-style-type: none"> 1. If the API fails to return availability data, the system informs the user.
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime. EX2: API Failure <ol style="list-style-type: none"> 1. If the HDB Car Park API is down, the system alerts the user and provides cached or default data where possible.
Includes:	
Special Requirements:	
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-009		
Use Case Name:	Get Distance Information		
Created By:	Ong Jing Jie	Last Updated By:	Ong Jing Jie
Date Created:	06-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System HDB Carpark API OneMap API
Description:	To provide the system with the distance from their current location to nearby parking lots.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database and API must be online. 4. Users must have logged in or using a guest account. 5. System and map API must have access to the user's current location. 6. The user has input a destination.
Postconditions:	<ol style="list-style-type: none"> 1. The system receives accurate distance information for nearby parking lots.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The system calculates the distance from the user's current location to parking lots near the destination. 2. The system displays the distance in kilometers.
Alternative Flows:	AF-S1: No GPS Data Available <ol style="list-style-type: none"> 1. If the system cannot access the user's current location, it prompts the user to manually enter a starting point for distance calculation.
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime. EX2: API Failure <ol style="list-style-type: none"> 1. If the HDB Car Park API is down, the system alerts the user and provides cached or default data where possible.
Includes:	
Special Requirements:	
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-010		
Use Case Name:	View Car Park Details		
Created By:	Cheng Yi Feng	Last Updated By:	Ong Jing Jie
Date Created:	10-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System HDB Carpark API OneMap API
Description:	Allow users to view detailed information about the car park such as pricing rates of car park, distance from current location to car park, lots availability.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database and API must be online. 4. Users must have logged in or using a guest account. 5. System and map API must have access to the user's current location. 6. The user has input a destination.
Postconditions:	<ol style="list-style-type: none"> 1. System displays detailed car park information.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The system shows a list of car parks available within a specified radius of the user's destination. 2. The user can scroll through the list of available car parks and select a specific car park to view more details. 3. When the user chooses to view more details of the car park, the car park tab will enlarge and display the lot availability, distance between current location to car park and price.
Alternative Flows:	AF-S2: User does not select a specific car park from list of available car parks <ol style="list-style-type: none"> 1. System by default shows the first car park detailed information
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime. EX2: API Failure <ol style="list-style-type: none"> 1. If the HDB Car Park API is down, the system alerts the user and provides cached or default data where possible.
Includes:	
Special Requirements:	

Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-011		
Use Case Name:	Filter List		
Created By:	Cheng Yi Feng	Last Updated By:	Ong Jing Jie
Date Created:	10-Sep-2024	Date Last Updated:	08-Nov-2024

Actor:	User System HDB Carpark API OneMap API
Description:	Allows users to customize their car park search results by toggling filters on or off. Users can apply filters like the cheapest car park rates, shortest distance from their current location to the selected car park, and highest parking lot availability to refine their search based on their preferences.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database and API must be online. 4. Users must have logged in or using a guest account. 5. System and map API must have access to the user's current location. 6. The user has input a destination.
Postconditions:	<ol style="list-style-type: none"> 1. Users successfully apply filters to view car park options that match their preferences.
Priority:	High
Frequency of Use:	High for users that want to filter their search results, no use for users that do not require to filter their search results.

Flow of Events:	<ol style="list-style-type: none"> 1. The user navigates to the filter list section. 2. The system displays available filter options: <ul style="list-style-type: none"> ○ Cheapest car park rates ○ Shortest distance between current location to the car park ○ Highest parking lot availability 3. The user toggles the desired filters on or off. 4. The system dynamically updates the search results based on the selected filters. 5. The user views the refined list of car parks that meet their selected criteria.
Alternative Flows:	AF-S3: User did not turn on any filters <ol style="list-style-type: none"> 1. By default, the system shows the first car park detail information in the list.
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime. EX2: API Failure <ol style="list-style-type: none"> 1. If the HDB Car Park API is down, the system alerts the user and provides cached or default data where possible.
Includes:	
Special Requirements:	
Assumptions:	The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date. The system assumes that users have basic navigation and search skills on the website.
Notes and Issues:	

Use Case ID:	UID-012		
Use Case Name:	Forget Password		
Created By:	Cheng Yi Feng	Last Updated By:	Cheng Yi Feng
Date Created:	14-Oct-2024	Date Last Updated:	08-Nov-2024

Actor:	User System
Description:	For users with an existing account that want to change their password.
Preconditions:	<ol style="list-style-type: none"> 1. The User's device must have access to the internet. 2. The System must be connected to the database. 3. The database must be online. 4. The User must have existing account
Postconditions:	<ol style="list-style-type: none"> 1. The user successfully changed to a new password.
Priority:	High
Frequency of Use:	High for Users that want to change their password and no use for users that don't.

Flow of Events:	<ol style="list-style-type: none"> 1. User Requests to Reset Password. 2. System transition to forget password page and request for user's registered email. 3. User input registered email and system transition to reset password page. 4. OTP (One-Time Password) is generated and sent to the user's registered email address and expires in 5 mins 5. User input correct OTP and new password. 6. The user's new password is updated in the database. 7. System redirects to the success reset password page where the user can choose login/register/guest.
Alternative Flows:	AF-S4: <ol style="list-style-type: none"> 1. If the OTP is incorrect, an error message is displayed. 2. If the OTP expires after 5 mins, an error message is displayed.
Exceptions:	EX1: System unavailable <ol style="list-style-type: none"> 1. The user is informed of system maintenance or downtime.
Includes:	
Special Requirements:	
Assumptions:	<p>The system assumes that the data provided by HDB Carpark API, and other sources is accurate and up-to-date.</p> <p>The system assumes that users have basic navigation and search skills on the website.</p>
Notes and Issues:	