Use Cases

for

SC2006 LAB5

Version 2.0 approved

Prepared by Hirashima Shunya

Milky Shake 04

30th August 2022

Revision History

Name	Date	Reason For Changes	Version
Hirashima Shunya	3/11/2022	For final submission. Did minor changes.	2.0

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:

1. User's identity has been authenticated.
2. 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 item in database has been updated with 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 Template

Use Case ID:	1		
Use Case Name:	Login to User Account		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	Account login process through email address and password.
Preconditions:	1. User accounts must exist in the database. 2. The system is on the login page.
Postconditions:	1. A user must be able to access the user's favourite list. 2. A user must be able to access the user's searching history.
Priority:	Medium
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. User enters an email address and password in the login interface. 2. User selects the login button. 3. The system validates the email address and password. 4. The system verifies the account by checking the user's credentials stored in a cloud database. 5. The system authenticates the user to login successfully by direct user to Home screen.
Alternative Flows:	AF-S3: If the system detects empty email address or password fields 1. The system displays an error message "Please input both email address and password." 2. Return to Step 1. AF-S4: If user enters the wrong credentials 1. The system displays an error message "Either address or password is wrong. Please try again." 2. Return to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	A user account must be accessed from a cloud database.
Notes and Issues:	-

Use Case Template

Use Case ID:	2		
Use Case Name:	Register for a New Account		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Registration for a new user account.
Preconditions:	1. The window is on the Login page.
Postconditions:	1. A new account is created and stored on a cloud database.
Priority:	High
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. User taps the "sign up" button. 2. User inputs username, first name, last name, email address, and password. 3. User taps the "register" button. 4. The system validates the user inputs. 5. The system verifies the account by checking the availability of user's account in the cloud database. 6. The system creates a new account for the user in the cloud database. 7. The system sends a verification email to user's email. 8. After user clicks on the verification link, the system activate the new account.
Alternative Flows:	AF-S4: The user inputs an invalid email address 1. The system displays an error message, "Invalid address. Please input a valid address." 2. Return Step 2. AF-S4: The user inputs an invalid password 1. The system displays an error message "Invalid password. Please input a valid password by referring to the conditions below." 2. Return Step 2. AF-S5: The system detect account existed in the cloud database 1. The system displays an error message "Account already exists. Please try again." 2. Return Step 2. AF-S8: The system did not receive confirmation by the verification link. 1. The system will not activate the new account until confirmation.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	A user account must be stored in a cloud database.

Notes and Issues:	-

Use Case Template

Use Case ID:	3		
Use Case Name:	Change Password		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Alteration of a user account password.
Preconditions:	1. A user account must exist in the database. 2. The window is on the Login page.
Postconditions:	1. A new password replaces the old one in a cloud database.
Priority:	Medium
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. User taps “Forget password” strings. 2. User enters an email address for an existing account. 3. The system validates the user's input. 4. The system verifies the account by checking the user’s email stored in a cloud database. 5. The system sends an email to the user's email to reset the password. 6. User inputs a new password and taps the “update” button. 7. The system checks the validity of the new password. 8. The system updates the password in the cloud database.
Alternative Flows:	AF-S3: If the system detects empty email address field. 1. The system displays an error message “Please input an email address.” 2. Return to Step 2. AF-S4: If input address is not found on a database 1. The system lets the user input email address again, displaying an error message, “Input email address is not found. Please input an address that you register for”. 2. Return to Step 2. AF-S7: If user’s new password is not valid 1. The system displays an error message “Invalid password. Please input a valid password by referring to the conditions below.” 2. The system lets the user input another password to be valid. 3. Return to Step 6.
Exceptions:	-
Includes:	-
Special Requirements:	-

Assumptions:	- A user account can be accessed from cloud databases.
Notes and Issues:	-

Use Case Template

Use Case ID:	4		
Use Case Name:	Add Carpark to Favorite List		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Addition of Carpark to Favorite List.
Preconditions:	1. The window is on either the user profile or the end of the carparking page.
Postconditions:	1. A selected carpark is added to a favorite list.
Priority:	High
Frequency of Use:	0 - 5 times per day
Flow of Events:	1. User taps an unfilled heart-shaped diagram shown beside the Carpark description. 2. The system retrieves the selected carpark information from the CarparkAPI. 3. The system verifies the existence of the selected carpark in the user's favorite list in the cloud database. 4. The carpark is added to the user's favorite list. The heart-shaped diagram is filled with red.
Alternative Flows:	AF-S3: If the selected carpark already existed in the user's favorite list. 1. The system displays a pop-up notification "This carpark has already been in your favorite list. Would you like to delete it?" with "Yes" and "No" tabs. 2. If user taps "Yes", the system will remove the selected carpark from the user's favorite list in the cloud database.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	5		
Use Case Name:	Upload Photo for User Profile		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Uploading a photo used for an icon of a user profile.
Preconditions:	1. The window is on the user profile page.
Postconditions:	1. An icon photo of a user profile is updated. 2. The photo is stored on a cloud database.
Priority:	Low
Frequency of Use:	1 - 3 times per lifetime
Flow of Events:	1. User taps the profile icon. 2. The system displays a notification with messages "change photo" and "not now". 3. User taps the "change photo" block. 4. The system accesses and displays the user's photo album. 5. User chooses a photo from the user's photo album. 6. User confirms a photo. 7. The system returns the window to the user profile page and the icon photo is updated.
Alternative Flows:	AF-S2: If a user taps the "not now" block 1. The system returns the window to the user profile page. AF-S4: The system unable to access user's photo album 1. The system displays an error message "Unable to access photo album. Please allow permission"
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	- jpg and png files are supported.
Notes and Issues:	-

Use Case Template

Use Case ID:	6		
Use Case Name:	Search for Carpark Locations		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Search for Carpark Location as a user's destination.
Preconditions:	1. The window is on the Home page.
Postconditions:	1. The system accepts the departure location and the destination. 2. The system displays a map interface showing pointers on carparks near to the selected destination.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. User taps a searching blank displaying "Find Carpark". 2. The system moves to the "Search Start Location and Destination" page. 3. The system pops up "choose current location" as an option bar. 4. User inputs a departure location in the blank, which is set as a departure location by the system. 5. User inputs a destination, which is set as a destination by the system. 6. User taps the "Search" button. 7. The system checks for carpark availability near the destination. 8. The system displays carparks near to the destination.
Alternative Flows:	AF-S7: If a user inputs nothing on the departure location blank 1. The system sets the departure location as the user's current location 2. The system retrieves the user's current location. 3. Return to Step 7.
Exceptions:	Ex1: If a user does not give a permission of using the current location and the system attempts to set the departure location as the current location 1. The system displays an error message: "Please give permission for the usage of the current location or input the departure location."
Includes:	1. Retrieve User's current location 2. Check for Carpark Availability 3. Choose carpark
Special Requirements:	-
Assumptions:	1. At least one option can be found.
Notes and Issues:	-

Use Case Template

Use Case ID:	7		
Use Case Name:	Check for Carpark Availability		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	System
Description:	Check whether the respective carpark is available.
Preconditions:	1. At least one carpark is fetched from Google Maps API. 2. The user had chosen a destination location.
Postconditions:	1. The checked availability is sent to the user interface for searching functionality.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. The system fetches carparks near to the selected destination from Google Maps API. 2. The system fetches information about availability of each carpark. 3. The availability data is tied to the carpark location. The availability data is shown on the "Carpark Selected" page. 4. The system repeats updating the carpark availability as a background process after a CarPark is selected.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	8		
Use Case Name:	Retrieve User's current location		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	System
Description:	Retrieval of user's current location.
Preconditions:	1. User gives a permission for the application to access the current location.
Postconditions:	1. The retrieved current location can be used as a departure location in search functionality.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. The system displays a pop-up window. The uppermost bar of the window displays a bar with the message "use the current location." 2. User taps the "use the current location" bar. 3. The current location is retrieved from the Google Maps API and GPS.
Alternative Flows:	AF-S2: If a user does not give a permission to use the current location 1. The uppermost message changes to be "no permission to use the current location."
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	9		
Use Case Name:	Choose Carpark		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	To pick a carpark as a user's destination.
Preconditions:	1. The system is on the Home page or the Search page. 2. User taps the "Search" button on the Home page.
Postconditions:	1. The selected carpark is displayed on the Search page as a destination. 2. The system calculates a route for driving.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. The system displays 2 dropdown menus: the one determines the category of sorting and the other determines the order of the result display. 2. The system displays 3 of candidates on the bottom side of the user interface with trivial information including the distance and the time to be taken. 3. The system displays candidate carparks near to the destination in Google Maps in a pointer form. 4. User taps the right-handed arrow symbol with respect to one of the candidate carparks. 5. The system displays detailed information about the carpark. 6. The user pushes the navigate button. 7. The system determines trails to carpark location using Google Maps API. 8. The system displays the trail to carpark location on the map until user reach destination or user press "Close" button.
Alternative Flows:	AF-S4: If user desires to choose the other carpark 1. User taps a left hand-sided arrow symbol displayed on the "Carpark Selected" page. 2. Return to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	10		
Use Case Name:	Share Carpark Location		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Allow user to share the carpark location.
Preconditions:	1. User reach destination OR 2. User press the "Close" button after navigation.
Postconditions:	
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. User presses the "Share" button. 2. The system displays the sharing options. 3. User chooses a sharing option. 4. The system shares the carpark's information to user's sharing options.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	11		
Use Case Name:	Parking Settings		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Allow user to estimate fares, set timer, add images, and add notes
Preconditions:	1. User presses the "Start Parking" button and is in the Parking Setting page.
Postconditions:	1. The system displays the navigation for destination or the selected carpark.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. The system displays "Estimation", "Timer", "Add Image", "Add Note buttons" in the parking setting page. 2. If user selects the "Add Image" box, then the system saves the image added in the user account in the cloud database and displays the image on the "Add Image" box. 3. If user enters text in the "Add Note box", then the system saves the user's notes in the user account in the cloud database. 4. If user selects "Estimation", then the user uses the included use case "Estimation Fare". 5. If user sets a timer by selecting the "Timer" buttons, then the system saves the user timer in a cloud database. 6. User presses the "Continue" button. 7. The system displays the Parking Ongoing UI.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Estimation Fare 2. Start Timer
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	12		
Use Case Name:	Start Timer		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	System
Description:	Activation of the set timer.
Preconditions:	1. The system is on the "Parking Ongoing" page.
Postconditions:	1. The timer system activates the set timer.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. User taps the "Continue" button on the "Parking Settings" page. 2. The system makes the window display the "Parking Ongoing" page. 3. When user taps the "Continue" button on the "Parking Settings" page, the system starts the timer set by the user. 4. The system makes the window display the "Parking Ongoing" page. 5. In the middle of the window, the system makes the window display "alert { set time } mins before end times." by calculating from a set timer. 6. The system starts the timer set by user.
Alternative Flows:	AF-S5: If no timer is set 1. The system does display the set time as 0 minutes. 2. Return to Step 6.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	13		
Use Case Name:	Estimate Fare		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Estimate fare needed based on user desired duration of stay
Preconditions:	1. The user selects Estimation buttons in the Parking Setting page. 2. The system is able to retrieve the user's parking location's parking fare.
Postconditions:	-
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	1. User input his estimation duration of carparking. 2. The system calculates the estimated fare needed for the user estimated duration. 3. The system saves the estimated fare in cloud database and displays it in the Parking Setting page.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	14		
Use Case Name:	Carpark To and From Destination		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	Show trail from car park to destination and vice versa.
Preconditions:	1. The system is on the Parking Ongoing page. 2. User requests to determine the trail route by tapping the "Navigation" button.
Postconditions:	1. The system calculates and finds the trail route.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. User taps the "Navigating" button on the left side. 2. The system directs user to Google Maps by opening the browser. 3. The system concurrently sends information about the location of the car park as a departure location and the destination to the Google Map.
Alternative Flows:	AF-S1: User taps the "Navigating" button on the right side. 1. The system directs user to Google Maps by opening the browser. 2. The system concurrently sends information about the location of the car park as a destination and the destination that a user has already input as a departure location to the Google Map.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	15		
Use Case Name:	Rate and Comment carpark location		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	A space for rating and commenting for a carpark.
Preconditions:	1. The system is on the Comment and Ratings page.
Postconditions:	1. User's rate and comment are stored on a cloud database.
Priority:	Medium
Frequency of Use:	0 - 5 times per day
Flow of Events:	1. User fills comments or taps white stars on the Feedback blank. 2. User taps the "Done" button. 3. The system sends the content of the rating and commenting to the cloud database. 4. The system displays the Home page UI.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	16		
Use Case Name:	Access other screen display and functionality		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	User
Description:	A user interface to switch pages on the application.
Preconditions:	-
Postconditions:	1. The window moves to a selected page.
Priority:	High
Frequency of Use:	0 - 30 times per day
Flow of Events:	1. User taps a sidebar. 2. The system displays the list of options: ‘My account’, ‘My favorites’, and ‘My history’. 3. User can choose to click any of the following of the above options. 4. The system makes the window display the selected page.
Alternative Flows:	AF-S2: If a user taps the sidebar again 1. The system closes the list of options.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	17		
Use Case Name:	Update User Profile		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	2nd September 2022

Actor:	System
Description:	An interface for updating the user profile.
Preconditions:	1. The request of adding a new carpark to the favorite list or updating an icon photo is sent from user.
Postconditions:	1. The information related to the user profile on a database is updated.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. When user taps the 'my account' on a submenu, the system displays the user interface including first name, last name, and email of the user. 2. User clicks the 'edit' button. 3. The system displays the {EDIT_PROFILE} page. 4. User inputs particulars to each blank. 5. After filling particulars, the user taps the 'update' button. 6. If user updates the email address, the system checks whether it is unique and satisfies the requirements. 7. If user updates the username, the system checks whether the username is unique. 8. The system updates user information on a cloud database.
Alternative Flows:	AF-S6: If the email does not fulfill the requirements 1. The system prompts user to enter email again.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	18		
Use Case Name:	Alternative Carpark Location		
Created By:	Sim Oi Liang	Last Updated By:	Sim Oi Liang
Date Created:	13th September 2022	Date Last Updated:	13th September 2022

Actor:	User
Description:	Allow user to choose a better carpark.
Preconditions:	1. User had chosen a carpark location and proceeding towards the chosen carpark location.
Postconditions:	1. User proceeds to an alternative carpark location.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	1. The system checks carpark availability around the destination every one minute. 2. The system determines whether new carpark locations are better than user selected carpark location based on an algorithm. 3. If new carpark locations are better, the system sends notification about the information of new carpark locations. 4. User press the "Yes" button in the notification. 5. The system determines trails to carpark location using Google Maps API. 6. The system displays the new trail to the alternative carpark location on the map. 7. Return to Step 1 until the user reaches the carpark or presses the "Close" button.
Alternative Flows:	AF-S4: User ignores or presses "No" in the notification. 1. System displays the existing drive route trail. 2. Return to Step 1
Exceptions:	-
Includes:	1. Check carpark availability
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-