Chan

Use Cases

for

SC2006 LAB1

Version 1.0 approved

Prepared by <author>

Milky Shake 04

30th August 2022

Revision History

Name	Date	Reason For Changes	Version

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 Determineds) 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 account by checking the user's credentials stored in a cloud database. 4. The system authenticates the user to login successfully.
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 2. 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 2.
Exceptions:	-
Includes:	1. Verify Credentials
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:	30th August 2022

Actor:	User, system
Description:	Registration for a new user account.
Preconditions:	1. The system is on the Login Page.
Postconditions:	1. A new account is created and stored on a cloud database.
Priority:	Medium
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. User taps the “sign up” button. 2. User inputs first name, last name, email address, and password. 3. User taps the “register” button. 4. The system sends a verification email to user’s email. 5. After user clicks on the verification link, the system confirms the new account. The new account is stored on a cloud database.
Alternative Flows:	AF-S2: If 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-S2: If 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.
Exceptions:	-
Includes:	1. Change Password 2. Validate Account Availability
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:	30th August 2022

Actor:	User, system
Description:	Alteration of a user account password.
Preconditions:	1. A user account must exist in the database. 2. The system is on the Login Page.
Postconditions:	1. A new password replaces the old one in 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 sends an email to user’s email to reset the password. 4. User inputs the old password, a new password and taps the “update” button. 5. The system checks the validity of old password. 6. The system updates the password.
Alternative Flows:	AF-S2: If input address is not found on 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-S4: If user’s new password is not valid 1. The system lets the user input another password to be valid. 2. Return to Step 2.
Exceptions:	-
Includes:	1. Validate Account Availability
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:	Verify Credentials		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	Verification of whether the credentials are valid.
Preconditions:	1. A user fills credentials and taps the “login” block.
Postconditions:	1. The account system passes information on whether the credentials are valid to the login system.
Priority:	Medium
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. User taps the “login” block after filling an email address and password on the login page. 2. The system fetches an account data selected by email address from a cloud database. 3. The system verifies whether the password is matched to the one on the database. 4. The system sends the result of the verification to the login system.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	5		
Use Case Name:	Validate Account Availability		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	Validation of a user account availability.
Preconditions:	1. A user tries to update account information.
Postconditions:	1. The account system passes information on whether the account is available to the login system.
Priority:	Medium
Frequency of Use:	1 – 3 times per lifetime
Flow of Events:	1. When a user taps the “register” button on the account registration page or the “update” block on the change of password page, email address, user name, and password are checked. 2. The system checks whether email address, username, and password satisfy the condition. 3. The system checks with the database if the email address, username and password are unique 4. The system sends the result of verification to the login system.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	6		
Use Case Name:	Add Car Park to Favourite List		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	Addition of Car Park to Favourite List.
Preconditions:	1. The system is on either the user profile or the end of the car parking Page.
Postconditions:	1. A car park is added to a favourite list.
Priority:	High
Frequency of Use:	0 - 5 times per day
Flow of Events:	1. A user taps an unfilled heart diagram shown beside the Car Park description. 2. The car park is added to the user's favourite list. A heart-shaped diagram is filled with red.
Alternative Flows:	AF-S1: If the heart has already been filled with red 1. The system displays a pop-up notification "This car park has already been in your favourite list. Would you like to delete it?" with "Yes" and "No" tabs. 2. If the user taps "Yes", the car park is deleted from the user's favourite list.
Exceptions:	-
Includes:	1. Update User Profile
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	7		
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:	30th August 2022

Actor:	User
Description:	Uploading a photo used for an icon of a user profile.
Preconditions:	1. The system 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:	Medium
Frequency of Use:	1 - 3 times per lifetime
Flow of Events:	1. When a user taps the profile icon, a notification with messages “change photo” and “not now” shows up. 2. If a user taps the “change photo” block, the user chooses a photo from the user’s photo album. 3. User confirms a photo. 4. The system returns to the user profile page and the icon photo is updated.
Alternative Flows:	AF-S2: If a user chooses cancellation 1. Return to the user profile page.
Exceptions:	-
Includes:	1. Update User Profile
Special Requirements:	-
Assumptions:	- Either jpg or png file is supported.
Notes and Issues:	-

Use Case Template

Use Case ID:	8		
Use Case Name:	View Car Park History		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	An interface for displaying car park history.
Preconditions:	1. The system is on the Home Page.
Postconditions:	1. The system displays the “My History” Page.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. A user taps the sidebar on the Home Page. 2. The system opens a menu. 3. A user taps the “My History” tab on the menu. 4. The system fetches the history data from a cloud database. 5. The system displays the history in descending order from the latest parking.
Alternative Flows:	AF-S4: If no history data is found 1. The system shows a message “there is no car park history!” 2. Return to the Home Page.
Exceptions:	-
Includes:	-
Special Requirements:	1. Up to 10 results are stored and visible.
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	9		
Use Case Name:	Update User Profile		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	An interface for updating the user profile.
Preconditions:	1. The request of adding a new car park to the favourite list or updating an icon photo is sent from a user.
Postconditions:	1. The information related to the user profile on the database is updated.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. When the user clicks on ‘my account’, then the system displays the ‘first name’, ‘last name’ and ‘email’ of the client. 2. When the user tries to edit their ‘first name’, ‘last name’, ‘email’ or profile photo, they just need to click the ‘edit’ button 3. User enters changes the info 4. User presses the update button 5. If user updates email, system checks if the updated email is unique and if it fulfils the appropriate requirements 6. If the user updates the username, the system checks if the username is unique. 7. Information tagged to the account is updated. 8. The database is also updated
Alternative Flows:	-AF-S6 If the email does not fulfil the requirements, the user is prompted to re enter their email. -AF-S5 If the email does not fulfil
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	10		
Use Case Name:	Search for Car Park Locations		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	31th August 2022

Actor:	User
Description:	Search for Car Park Location as a user's destination.
Preconditions:	1. The system 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 car parks near to the selected destination.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. A user taps a searching blank displaying "Find Carpark". 2. The system displays the "Search Start Location and Destination". 3. The system pops up "choose current location" as an option bar. 4. A user inputs a departure location in the blank, which is set as a departure location by the system. 5. A user taps a destination searching blank. 6. A user inputs a destination, which is set as a destination by the system. 7. A user taps the "Search" button. 8. The system displays car parks nearby to the destination.
Alternative Flows:	AF-S4: If a user inputs nothing on the departure location blank 1. The system sets the departure location as the user's current location. 2. Return to Step 5.
Exceptions:	Ex1: If a user does not give a permission of using the current location and the system tries to set the departure location as the current location 1. The system displays an error message: "Please give permission to use the current location or input the departure location."
Includes:	1. Check for Car Park Availability
Special Requirements:	-
Assumptions:	1. At least one option can be found.
Notes and Issues:	-

Use Case Template

Use Case ID:	11		
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:	31th August 2022

Actor:	User
Description:	A user interface to switch pages on the application.
Preconditions:	-
Postconditions:	1. The selected Page is displayed.
Priority:	High
Frequency of Use:	0 - 30 times per day
Flow of Events:	1. A user taps a sidebar. 2. The list of options displaying ‘My account’, ‘My favourites’, ‘My history’. 3. The user can choose to click any of the following of the above options and access the tabs.
Alternative Flows:	AF-S2: If a user taps the side bar again 1. The system closes the list of options.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	12		
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:	30th August 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. A user taps a blank for the departure location input. 2. The system displays a pop-up window. The uppermost bar of the window displays the message "use the current location." 3. A user taps the "use the current location" bar. 4. The current location is retrieved from the Google Map 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:	13		
Use Case Name:	Check for Car Park Availability		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

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

Use Case Template

Use Case ID:	14		
Use Case Name:	Choose Car Park		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	31th August 2022

Actor:	User
Description:	To pick a car park as a user's destination.
Preconditions:	1. The system is one the Home Page or the Search Page. 2. A user taps the "Search" button on the Home Page.
Postconditions:	1. The selected car park 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 mimic information including the distance and the time to be taken. 3. The system displays candidacy car parks near to the destination in Google Maps as pointers. 4. A user taps the right-handed arrow symbol with respect to one of the candidacy car parks. 5. The system displays detailed information about the car park. 6. A user taps the "Start Parking" button. 7. The system displays the "Parking Settings" Page.
Alternative Flows:	AF-S4: If user wishes to choose the other car park 1. A user taps a left hand-sided arrow symbol displayed on the "Carpark Selected" Page. 2. Return to Step 2.
Exceptions:	-
Includes:	1. Display Drive Route Trail
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	15		
Use Case Name:	Retrieve Car Park Information		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	Retrieval of car park information from the public API.
Preconditions:	1. A user taps the “Start Parking” button..
Postconditions:	-
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. The system fetches data related to the selected car park from a database. 2. The system displays such data in “Carpark Selected” Page.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	16		
Use Case Name:	Set Timer for Parking		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	To set a timer as a duration of car parking.
Preconditions:	1. The system is on the Car Park Selected Page.
Postconditions:	1. The duration of the timer is stored on a database.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. A user taps the “Start Parking” button. 2. The system displays the “Parking Setting” Page. 3. The system shows the user interface of setting the timer in the middle of the page. 4. After setting the timer by using the interface and parking, a user taps the “Continue” button. 5. The system moves to the “Parking Ongoing” Page.
Alternative Flows:	AF-S3: If a user sets the timer as 0 minutes 1. The system sets no timer. 2. The system flags that no timer is needed. 3. Return to Step 4.
Exceptions:	-
Includes:	1. Start Timer
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	17		
Use Case Name:	Start Timer		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 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. A user taps the “Continue” button on the “Parking Settings” Page. 2. The system displays the “Parking Ongoing” 3. The system starts the timer set by the user.A user taps the “Continue” button on the “Parking Settings” Page. 4. The system displays the “Parking Ongoing”. 5. In the middle of the window, the system displays “alert { set time } mins before end times.” by calculating from a set timer. 6. The system starts the timer set by the 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:	18		
Use Case Name:	Return to Car Park		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	An interface to facilitate returning to the selected car park.
Preconditions:	1. The system is on the Parking Ongoing Page. 2. The user wishes to go back to the selected car park.
Postconditions:	1. The system displays the navigation for destination or the selected car park.
Priority:	High
Frequency of Use:	0 - 10 times per day
Flow of Events:	1. A user taps the “Navigate” button displayed on the right side. 2. The system sends Google Maps about departure and destination information. 3. The set destination is sent as the departure point and the selected car park is sent as the destination. 4. The system directs a user to Google Maps by opening a browser.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Display Walk Route Trail
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	19		
Use Case Name:	Determine Trails to Car Park Location		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	Determination of trail route for car park.
Preconditions:	1. The system is on the Parking Ongoing Page. 2. A 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. A user taps the “Navigating” button on the left side. 2. The system sends a user to Google Maps by opening the browser. 3. Concurrently the system sends information of the location of the car park as a departure and the destination to the Google Map.
Alternative Flows:	AF-S1: A user taps the “Navigating” button on the right side. 1. The system sends a user to Google Maps by opening the browser. 2. Concurrently the system sends information of the location of the car park as a destination and the destination that a user has already input as a departure to the Google Map.
Exceptions:	-
Includes:	1. Display Walk Route Trail 2. Display Drive Route Trail
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	20		
Use Case Name:	Rate and Comment		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	User
Description:	A space for rating and commenting for a car park.
Preconditions:	1. The system is on the Comment and Ratings Page.
Postconditions:	1. A user sends rate and comment and those are stored on a cloud database.
Priority:	Medium
Frequency of Use:	0 - 5 times per day
Flow of Events:	1. A user fills comments or taps white stars on the Feedback blank. 2. The user taps the “Done” button. 3. The rating and commenting information is sent to the cloud database.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Update Rate and Comment
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case Template

Use Case ID:	21		
Use Case Name:	Update Rate and Comment		
Created By:	Hirashima Shunya	Last Updated By:	Hirashima Shunya
Date Created:	30th August 2022	Date Last Updated:	30th August 2022

Actor:	System
Description:	An interface for updating the rate and comment.
Preconditions:	1. The request of giving a rate and comment is sent from a user.
Postconditions:	1. The rate and comment on the database is updated.
Priority:	Medium
Frequency of Use:	0 - 5 times per day
Flow of Events:	1. When a user taps the “Done” block on the Rate and Comment Page, the request for updating a database is sent. 2. The system adds the comment to the database. 3. The system adds ratings to the database and updates the average ratings.
Alternative Flows:	AF-S1: If there is no filled information 1. Return to the Home Page.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-