Use Cases

for

<SC2006>

Version 1.2 approved

Prepared by <Lim Weijun>

<Fiver's Makan>

<20th April 2024>

Revision History

Name	Date	Reason For Changes	Version
Lim Weijun	01/02/2024	First draft	1.0
Lim Weijun	19/02/2024	New and improved use cases	1.1
Lim Weijun	20/04/2024	Refined use cases	1.2

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.

3. Use Case Descriptions

3.1 Sign up

Use Case ID:	UC001		
Use Case Name:	Sign up		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	User
Description:	This use case allows the user to sign up an account in our system.
Preconditions:	User has navigated to the Signup Page from Login Page.
Postconditions:	A "Register Successful!" message will be returned to the user, and the user will be redirected to the Login Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The User navigates to the Signup Page from the Login Page. 2. This is done by clicking the "Register here" button at bottom corner of the Login Page, which will redirect them to the Signup Page. 3. The User enters the necessary information such as user type (patron or restaurant owner), email and password. 4. The User clicks the "Register" button to sign up.
Alternative Flows:	AF1.1: At Step 4, if the user did not provide complete information such as weak password/invalid email format. 1. The system will display an error message depending on the error type and prompts the user to provide complete information. 2. Go to Step 3. AF1.2: Cancellation of registration process by closing the tab. 1. No new user account is created in the database.
Exceptions:	EX1: User already exists in the system. 1. The System will return an error message "The email address is already in use by another account."
Includes:	UC001.1: Create New Account Instance
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.1.1 Create New Account Instance

Use Case ID:	UC001.1		
Use Case Name:	Create New Account Instance		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	15/04/2024	Date Last Updated:	20/04/2024

Actor:	Database
Description:	This use case outlines the process where the database will create a new account instance based on the information received.
Preconditions:	A request is sent to database to store the personal information.
Postconditions:	A “Register Successful!” message will be returned to the user, and the user will be redirected to the Login Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. A request is sent to the database to store the personal information. 2. The database will validate if the personal information already exists within the stored data. 3. If personal information is new, it will create a new user account instance and store it in the database and the system opens up the Login Page.
Alternative Flows:	
Exceptions:	EX1: User already exists in the database. 1. The System will return an error message “The email address is already in use by another account.”
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.2 Login

Use Case ID:	UC002		
Use Case Name:	Login		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	User
Description:	This use case allows the user to login to an existing account on the web platform.
Preconditions:	User has started the web application and navigated to the Login Page.
Postconditions:	A “Login Successful!” message will be returned to the user, and the user will be redirected to the Home Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The User started the web application and navigated to the Login Page. 2. The User enters the necessary information such as email and password. 3. The User clicks on the “Login” button.
Alternative Flows:	AF2.1: Cancellation of login process by closing the tab. 1. User will not login.
Exceptions:	EX2: Email/password is wrong. 1. The system will display an error message “Authentication failed. Please try again.”
Includes:	UC002.1: Validate user credentials
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.2.1 Validate user credentials

Use Case ID:	UC002.1		
Use Case Name:	Validate user credentials		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	15/04/2024	Date Last Updated:	20/04/2024

Actor:	Database
Description:	This use case allows the user to login to an existing account on the web platform.
Preconditions:	User has started the web application and navigated to the Login Page.
Postconditions:	A “Login Successful!” message will be returned to the user, and the user will be redirected to the Home Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. A request is sent to the database to check the credentials against the stored data. 2. The database will validate if the submitted password matches exactly with the password of the submitted email address in the credentials. 3. If the password matches, the credentials are validated, and the system opens the Home Page.
Alternative Flows:	
Exceptions:	EX2: Email/password is wrong. The system will display an error message “Authentication failed. Please try again.”
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.3 Search for restaurants by filters (Home Page)

Use Case ID:	UC003		
Use Case Name:	Search for restaurants by filters		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	Patron, Google Map API
Description:	This use case allows the patron to search for his desired restaurant by using filters such as distance proximity, dietary requirements, and minimum ratings.
Preconditions:	Patron must be logged into our application and navigated to the Home Page.
Postconditions:	The google map API must return a list of restaurants that are filtered according to the patron's needs.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the patron navigates to the Home Page on the Navbar. 2. The patron searches for his desired restaurant by using filters at the Home Page. 3. The patron clicks on the "Search" button. 4. Google Map API will return a list of restaurants on the Home Page.
Alternative Flows:	AF3.1: If no restaurants match the filters, the google map API will display a message "Unfortunately, there are no Fivers' Restaurants specific to your needs".
Exceptions:	
Includes:	UC3.1: Submit Reservation
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.3.1 Submit Reservation

Use Case ID:	UC003.1		
Use Case Name:	Submit Reservation		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to submit a reservation for the restaurant by clicking on the “Book” button.
Preconditions:	The patron must be logged in to our application and navigated to the Home Page.
Postconditions:	The system should produce a pop-up page to inform the patron of a successful reservation.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron searches for his desired restaurant from the list of restaurants at the Home Page. 2. The patron clicks on the “Book” button. 3. The system produces a pop-up window requesting for mandatory fields such as date of reservation, time of reservation and number of pax. 4. The patron enters the necessary information. 5. The patron clicks on the “Make Reservation” button. 6. The system displays a pop-up page to inform the patron of a successful reservation and updates the database.
Alternative Flows:	<u>AF3.1: At Step 5 if the patron inputs incomplete fields.</u> 1. The system will highlight the incomplete field name(s) in purple, with an error prompt at the bottom of the field name that says, “Please fill out this field”. 2. Go to Step 4. <u>AF3.2: Cancellation of reservation by closing the pop-up window tab.</u> 1. No reservation was made.
Exceptions:	<u>EX3: Patron inputs date/time from the past.</u> 1. The system will return an error message “You cannot make a reservation in the past!”
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.4 View Active Reservations

Use Case ID:	UC004		
Use Case Name:	View Active Reservations		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to view all his active reservations on Active Bookings Page.
Preconditions:	The patron has previously made a reservation.
Postconditions:	The system should display all the patron's current active reservations on Active Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the patron navigates to the Active Bookings Page on the Navbar. 2. The system should display all the patron's current active reservations on Active Bookings Page, with details like date, time, number of pax and the restaurant name.
Alternative Flows:	
Exceptions:	
Includes:	UC004.1: Edit Active Reservation UC004.2: Cancel Active Reservation
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.4.1 Edit Active Reservation

Use Case ID:	UC004.1		
Use Case Name:	Edit Active Reservation		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to edit any of his active reservations on Active Bookings Page.
Preconditions:	The patron has previously made a reservation.
Postconditions:	The system should display all the patron's edited active reservations on Active Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron clicks on the "Edit" button on the Active Bookings Page. 2. The system redirects patron to the Edit Reservation Page and request for the same mandatory fields such as date, time, and number of pax. 3. The patron enters the necessary information. 4. The patron clicks on the "Save Changes" button. 5. The system redirects patron back to the Active Bookings Page and displays all the patron's edited active reservations.
Alternative Flows:	AF4.1: At Step 4 if the patron inputs incomplete fields. 1. The system will highlight the incomplete field name(s) in purple, with an error prompt at the bottom of the field name that says, "Please fill out this field". 2. Go to Step 3.
Exceptions:	EX4: Patron clicks on the back arrow at the top left of the Edit Reservation Page. 1. No editing of reservation was made and patron is redirected back to Active Bookings Page.
Includes:	UC004.2: Cancel Active Reservation
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.4.2 Cancel Active Reservation

Use Case ID:	UC004.2		
Use Case Name:	Cancel Active Reservation		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	05/02/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to cancel any of his active reservations on Active Bookings Page.
Preconditions:	The patron has previously made a reservation.
Postconditions:	The system should display all the patron's current active reservations on Active Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron clicks on the "Cancel" button on the Active Bookings Page. 2. The system produces a pop-up window saying "Are you sure you want to cancel this reservation?" 3. The patron clicks on the "Ok" button. 4. The system produces a pop-up window saying "Your reservation has been cancelled." 5. The system redirects patron back to the Active Bookings Page and displays all the patron's current active reservations.
Alternative Flows:	
Exceptions:	EX4: At Step 2, Patron clicks on the "Cancel" button at first pop-up window. 1. No cancelling of reservation was made.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.5 View Completed Reservations

Use Case ID:	UC005		
Use Case Name:	View Completed Reservations		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to view all his completed reservations on Completed Bookings Page.
Preconditions:	The patron has previously made and completed a reservation.
Postconditions:	The system should display all the patron's completed reservations on Completed Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the patron navigates to the Completed Bookings Page on the Navbar. 2. The system should display all the patron's completed reservations on Completed Bookings Page, with details like date, time, number of pax and the restaurant name. 3. The system will also display the patron's feedback details like number of stars given and review text if patron has given feedback.
Alternative Flows:	
Exceptions:	
Includes:	UC004.1: Give Feedback UC004.2: Edit Feedback
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.5.1 Give Feedback

Use Case ID:	UC005.1		
Use Case Name:	Give Feedback		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to provide feedback on his reservations on Completed Bookings Page.
Preconditions:	The patron has previously made and completed a reservation.
Postconditions:	The system should display all the patron's completed reservations and feedback details on Completed Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron clicks on the "Give Feedback" button on the Completed Bookings Page. 2. The system redirects the patron to Feedback Page and requests for mandatory fields such as rating and price paid per pax, but the review text field is optional. 3. The patron completes the fields and clicks on the "Submit Review" button. 4. The system redirects patron back to Completed Bookings Page and displays additional information such as the patron's feedback details.
Alternative Flows:	
Exceptions:	EX5: Patron clicks on the back arrow at the top left of the Feedback Page. 1. No feedback was given and patron is redirected back to Completed Bookings Page.
Includes:	UC004.2: Edit Feedback
Special Requirements:	
Assumptions:	
Notes and Issues:	The review text field is optional, if patron does not input anything on it, it will display "" on Completed Bookings Page. If text review exceeds 250 characters, system will not allow patron to type any futher.

3.5.2 Edit Feedback

Use Case ID:	UC005.2		
Use Case Name:	Edit Feedback		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows the patron to edit feedback on his reservations on Completed Bookings Page.
Preconditions:	The patron has previously made and completed a reservation.
Postconditions:	The system should display all the patron's completed reservations and edited feedback details on Completed Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron clicks on the "Edit Feedback" button on the Completed Bookings Page. 2. The system redirects the patron to Feedback Page and requests for mandatory fields such as rating and price paid per pax, but the review text field is optional. 3. The patron completes the fields and clicks on the "Submit Review" button. 4. The system redirects patron back to Completed Bookings Page and displays the patron's edited feedback details.
Alternative Flows:	
Exceptions:	EX5: Patron clicks on the back arrow at the top left of the Feedback Page. 1. No feedback was edited and patron is redirected back to Completed Bookings Page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	The review text field is optional, if patron does not input anything on it, it will display "" on Completed Bookings Page. If text review exceeds 250 characters, system will not allow patron to type any further.

3.6 View Patron Profile Information

Use Case ID:	UC006		
Use Case Name:	View Patron Profile Information		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows patron to view his profile information on Profile Page.
Preconditions:	Patron must be logged into our application.
Postconditions:	The system should display all the patron's profile information such as name, mobile number, email address and dietary requirements.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the patron navigates to the Profile Page on the Navbar. 2. The system displays all the patron's profile information such as name, mobile number, email address and dietary requirements.
Alternative Flows:	
Exceptions:	
Includes:	UC006.1: Edit Patron Profile Information
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.6.1 Edit Patron Profile Information

Use Case ID:	UC006.1		
Use Case Name:	Edit Patron Profile Information		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Patron
Description:	This use case allows patron to edit his profile information.
Preconditions:	Patron must be logged into our application and navigated to the Profile Page.
Postconditions:	The system should display all the patron's edited profile information on Profile Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The patron clicks on the "Edit Profile Button" on Profile Page. 2. The system redirects patron to Edit Profile Page and displays current profile information and allows patron to change the relevant information. 3. The patron changes the profile information. 4. The patron clicks on "Save Profile" button. 5. The system redirects patron back to Profile Page and displays all the patron's edited profile information.
Alternative Flows:	
Exceptions:	EX6: Patron clicks on the "Save Profile" button at the right-hand side of the Edit Profile Page without changing any fields. 1. The profile information was not changed.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.7 View Restaurant Bookings

Use Case ID:	UC007		
Use Case Name:	View Restaurant Bookings		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	19/02/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to view information about his patrons and the bookings for his restaurant.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should display the information about patrons for every booking on the Restaurant Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the restaurant owner navigates to the Restaurant Bookings Page on the Navbar. 2. System displays the bookings information such as name of patron, phone number, date of booking, time of booking and number of pax.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.8 Toggle Restaurant Availability

Use Case ID:	UC008		
Use Case Name:	Toggle Restaurant Availability		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	19/02/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to toggle his restaurant availability to allow patrons to book for his restaurant.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should change the restaurant availability to red if unavailable and green if available for booking.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the restaurant owner navigates to the Restaurant Bookings Page on the Navbar. 2. The restaurant owner toggles the restaurant availability to available for booking by clicking on the "Unavailable for Booking" button. 3. The system will return a pop-up window saying "Are you sure you want to change the availability to available?" 4. The restaurant owner clicks "OK". 5. The restaurant availability is now available for booking and the button is now green on the Restaurant Bookings Page.
Alternative Flows:	
Exceptions:	EX8: At Step 3, Patron clicks on the "Cancel" button on the pop-up window. 1. No toggling of restaurant availability was made.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	When a new restaurant owner account is registered, by default, the restaurant is unavailable for booking.

3.9 Update Restaurant Waiting Time

Use Case ID:	UC009		
Use Case Name:	Update Restaurant Waiting Time		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	19/02/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to update his restaurant estimated waiting time.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should reflect the restaurant waiting time to what the restaurant owner input at the Restaurant Bookings Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the restaurant owner navigates to the Restaurant Bookings Page on the Navbar. 2. The restaurant owner updates the restaurant waiting time by inputting on the bar provided. 3. The restaurant owner clicks on the “Confirm Time” button. 4. The system will return a pop-up window saying “Are you sure you want to update the waiting time to XX minutes?”, where XX is the waiting time inputted by the restaurant owner at Step 2. 5. The restaurant owner clicks “OK”. 6. The system will reflect the restaurant waiting time on the Restaurant Bookings Page and to the patrons booking the restaurant.
Alternative Flows:	
Exceptions:	EX9: At Step 4, Patron clicks on the “Cancel” button on the pop-up window. 1. No updating of restaurant waiting time was made.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	When a new restaurant owner account is registered, by default, the waiting time is “not set”. The range of the waiting time is set from 0 to 90 minutes.

3.10 View Restaurant Reviews

Use Case ID:	UC010		
Use Case Name:	View Restaurant Reviews		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	19/02/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to view his restaurant reviews made by patrons for his restaurant.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should display the reviews made by patrons on the Restaurant Reviews Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the restaurant owner navigates to the Restaurant Reviews Page on the Navbar. 2. System displays the reviews information such as name of patron, stars given, date, time, and the review text.
Alternative Flows:	
Exceptions:	
Includes:	UC010.1: Sort Restaurant Reviews
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.10.1 Sort Restaurant Reviews

Use Case ID:	UC010.1		
Use Case Name:	Sort Restaurant Reviews		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	19/02/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to sort his restaurant reviews for his restaurant.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should display the reviews made by patrons on the Restaurant Reviews Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The restaurant owner selects the filter based on what he wants to sort it by. 2. The system returns the reviews based on the filters that the restaurant owner selected in Step 1.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.11 View Restaurant Profile Information

Use Case ID:	UC011		
Use Case Name:	View Restaurant Profile Information		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows the restaurant owner to view his profile information on Restaurant Profile Page.
Preconditions:	Restaurant Owner must be logged into our application.
Postconditions:	The system should display all the restaurant owner's profile information such as name, email address, dietary types offered and address.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Upon successful login, the restaurant owner navigates to the Restaurant Profile Page on the Navbar. 2. The system displays all the restaurant owner's profile information such as name, email address, dietary types offered and address.
Alternative Flows:	
Exceptions:	
Includes:	UC011.1: Edit Restaurant Profile Information UC011.2: Edit Restaurant Address
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.11.1 Edit Restaurant Profile Information

Use Case ID:	UC011.1		
Use Case Name:	Edit Restaurant Profile Information		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows restaurant owner to edit his profile information.
Preconditions:	Restaurant Owner must be logged into our application and navigated to the Restaurant Profile Page.
Postconditions:	The system should display all the restaurant owner's edited profile information on Restaurant Profile Page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The restaurant owner clicks on the "Edit Restaurant Details" on Restaurant Profile Page. 2. The system redirects restaurant owner to Edit Restaurant Profile Page and displays current profile information and allows restaurant owner to change the relevant information. 3. The restaurant owner changes the profile information. 4. The restaurant owner clicks on "Save Changes" button. 5. The system redirects restaurant owner back to Restaurant Profile Page and displays all the restaurant owner's edited profile information.
Alternative Flows:	
Exceptions:	EX11: Patron clicks on the "Cancel" button at the left-hand side of the Edit Profile Page. 1. The restaurant profile information was not changed.
Includes:	UC011.2: Edit Restaurant Address
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.11.2 Edit Restaurant Address

Use Case ID:	UC011.2		
Use Case Name:	Edit Restaurant Address		
Created By:	Weijun	Last Updated By:	Weijun
Date Created:	20/04/2024	Date Last Updated:	20/04/2024

Actor:	Restaurant Owner
Description:	This use case allows restaurant owner to edit his profile information.
Preconditions:	Restaurant Owner must be logged into our application and navigated to the Restaurant Profile Page.
Postconditions:	The system should return a pop-up message saying “Address updated successfully” to restaurant owner.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The restaurant owner clicks on the “Edit Address” button on Edit Restaurant Profile Page. 2. The system redirects restaurant owner to Address Page and requests for address field. 3. The restaurant owner inputs the address and clicks on the “Save Address” button. 4. The system returns a pop-up message saying “Address updated successfully”.
Alternative Flows:	
Exceptions:	EX11: Patron clicks on the “Cancel” button on the Address Page. 1. The restaurant address information was not changed/saved.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	