**Use Cases**

**for**

**<**SC2006**>**

**Version 1.0 approved**

**Prepared by <Lim Weijun>**

**<Fivers>**

**<1st February 2024>**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Lim Weijun | 01/02/2024 | First draft | 1.1 |
|  |  |  |  |

# **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.

# **Use Case Identification**

## **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.

## **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.

## **Use Case History**

### **Created By**

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

### **Date Created**

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

### **Last Updated By**

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

### **Date Last Updated**

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

# **Use Case Definition**

## **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.

## **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.

## **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.

## **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.

## **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.

## **Frequency of Use**

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

## **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.

## **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.

## **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.

## **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.

## **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.

## **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.

## **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 Cases**

**3.1 Sign up**

| Use Case ID: | UC1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Sign up | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(primary) |
| --- | --- |
| Description: | This use case allows the user to sign up an account in our system. |
| Preconditions: |  |
| Postconditions: | The user must be able to sign up for their own account. |
| Priority: | 1 |
| Frequency of Use: | 1 time per application usage |
| Flow of Events: | 1. The system displays the Start-up Page, offering 2 options “Register for New Account” or “Login with Existing Account”. 2. The user clicks on the “Register for New Account” option. 3. The Start-up Page redirects the user to the Registration Page. 4. The user enters the necessary information such as username, email, password, phone number and dietary requirement. 5. The system must send a verification code to the user’s number and prompt verification. 6. The user verifies the registration. 7. The system registers the user’s details into the system’s database and logs the user once authentication is successful. 8. The system opens up the Home Page. |
| Alternative Flows: | AF1.1: At Step 4, if the user did not provide complete information such as weak password/invalid email format/invalid phone number.   1. The system prompts the user to provide complete information. 2. Go to Step 3.   AF1.2: At Step 4, if the user’s email address/username is found in the system.   1. The Registration Page provides the user with an option to “Navigate to the Login Page or “Reattempt registration”. 2. If the user selects “Navigate to the Login Page”, the system should open up the Login Page. 3. If the user selects “Reattempt registration”, go to Step 3.   AF1.3: At Step 7, if registration failed.   1. The system will display an error message indicating the reason why the registration failed. 2. Go to Step 3. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.2 Login**

| Use Case ID: | UC2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows the user to login their account to the system. |
| Preconditions: | The user must already have an account registered in the system’s database. |
| Postconditions: | The user must be able to login their own account. |
| Priority: | 1 |
| Frequency of Use: | 1 time per application usage |
| Flow of Events: | 1. The system displays the Start-up Page, offering 2 options “Register for New Account” or “Login with Existing Account”. 2. The user clicks on the “Login with Existing Account” option. 3. The system displays the Login Page. 4. The user enters the necessary information such as email and password. 5. The system authenticates the user and logs the user once authentication is successful. 6. The system opens up the Home Page. |
| Alternative Flows: | AF2.1: At Step 4, if the user did not provide complete information such as invalid email format/wrong password.   1. The system prompts the user to “Try Again” by providing complete information or “Register for New Account”. 2. If the user selects “Try Again”, go to Step 3. 3. If the user selects “Register for New Account”, the system should open up the Registration Page.   AF2.2: At Step 5, if authentication failed.   1. The system will display an error message indicating the reason why the authentication failed. 2. Go to Step 3. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.3 Home Page**

| Use Case ID: | UC3 | | |
| --- | --- | --- | --- |
| Use Case Name: | Home Page | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select one of the 4 different tabs available on the Home Page. |
| Preconditions: | The user must be logged in to the system. |
| Postconditions: | The system must redirect the user to the Page that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. Upon successful login, the Home Page displays 4 tabs: “Search Page”, “Favourites Page”, “My Booking Page”, “My Profile Page’. 2. The user selects their preferred page. 3. The system displays the Page according to the user’s input. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.4 Search Page**

**3.4.1 Searching Restaurant's Name**

| Use Case ID: | UC4 | | |
| --- | --- | --- | --- |
| Use Case Name: | Searching Restaurant’s Name | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to search for restaurants based on the search query inputted by the user. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants that correspond to the closest keyword which the user has input. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user inputs the restaurant name in the search bar on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s input on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favourited restaurants. |
| Notes and Issues: |  |

**3.4.2 Select Dietary Restrictions**

| Use Case ID: | UC5 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Dietary Restrictions | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select restaurants based on a filter of different dietary restrictions which includes, but is not limited to, halal food, vegetarian food, gluten-free, certain meat restrictions like beef. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants corresponding to the dietary restrictions options that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects their preferred dietary restrictions by ticking the corresponding dietary restrictions choice on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s dietary restrictions choice on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favourited restaurants. |
| Notes and Issues: |  |

**3.4.3 Select Proximity**

| Use Case ID: | UC6 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Proximity | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select restaurants within the user’s selected distance radius. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants that correspond to the proximity that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects the proximity by ticking the range of distance that they would prefer on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s range of distance choice on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favorites restaurants. |
| Notes and Issues: |  |

**3.4.4 Select Waiting Time**

| Use Case ID: | UC7 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Waiting Time | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select restaurants based on a filter of different ranges of waiting time. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants that correspond to the range of waiting time that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects their preferred range of waiting time by ticking the corresponding checkbox of the range on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s range of waiting time choice on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favourited restaurants. |
| Notes and Issues: |  |

**3.4.5 Select Affordability**

| Use Case ID: | UC8 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Affordability | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select restaurants based on a filter of different ranges of prices. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants that correspond to the range of prices that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects their preferred range of prices by ticking the corresponding checkbox of the range on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s range of prices on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favourited restaurants.. |
| Notes and Issues: |  |

**3.4.6 Select Restaurant Ratings**

| Use Case ID: | UC9 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Restaurant Ratings | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to select restaurants based on a filter of different ranges of restaurant ratings. |
| Preconditions: |  |
| Postconditions: | The system must return a list of restaurants that correspond to the range of prices that the user has selected. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects their preferred range of restaurant ratings by ticking the corresponding checkbox of the range on the Search Page. 2. The system updates and displays a list of restaurants based on the user’s range of restaurant ratings on the Search Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | If no options are selected, the system will display top restaurants near the user’s location if GPS is activated, else return the user’s favourited restaurants. |
| Notes and Issues: |  |

**3.5.1 Submit Reservation**

| Use Case ID: | UC10 | | |
| --- | --- | --- | --- |
| Use Case Name: | Submit Reservation | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to submit a reservation for the restaurant by clicking on the “Reserve” button. |
| Preconditions: |  |
| Postconditions: | The system should produce a pop-up page to inform the user of a successful reservation. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user navigates to the desired restaurant and clicks on the “Reserve” button on the Search Page. 2. The system requests mandatory fields such as date of reservation, time of reservation and number of pax. 3. The user enters the necessary information. 4. The system displays a pop-up page to inform the user of a successful reservation and updates the system. |
| Alternative Flows: | AF10.1: At Step 3, if the user inputs incomplete fields.   1. The system will highlight the incomplete field name(s) in red, with an error prompt on top of the field name that says “\*Field is required”. 2. Go to Step 2. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: | There are optional fields that the user need not complete including but not limited to: 1) requires a high stool, 2) request for indoor, 3) request for non-smoking dining area. |

**3.5.2 Favourite Restaurant**

| Use Case ID: | UC11 | | |
| --- | --- | --- | --- |
| Use Case Name: | Favourite Restaurant | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 07/02/2024 | Date Last Updated: | 07/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows the user to favourite a restaurant by clicking on favourite icon. |
| Preconditions: |  |
| Postconditions: | 1.System updates favourite indicator icon to reflect the restaurant has been favourited by the restaurant.  2.System adds favourited restaurant to list of user’s favourited restaurant |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1.User navigates to desired restaurant  2. User clicks on the favourite icon  3. System updates indicator to reflect the restaurant has been favourited  4.. System updated restaurant to list of user’s favourited restaurant |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.6 Favourite page**

**3.6.1 View favourites**

| Use Case ID: | UC12 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Favourites | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 07/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case describes the process by which a user manages their favorite items within the system. |
| Preconditions: | The user must be logged in to the system |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1.User navigates to the "Favorites" section of the website.  2.System displays list of the user favourited restaurants, with each restaurants having an option to remove from list of favourites |
| Alternative Flows: | If user has no favourited restaurants, system displays no favourited restaurant. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.6.2 Remove favourites**

| Use Case ID: | UC13 | | |
| --- | --- | --- | --- |
| Use Case Name: | Remove Favourites | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 07/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case describes the process by which a user manages their favorite items within the system. |
| Preconditions: | The user is in the favourites page  The user has at least one favourited restaurant |
| Postconditions: | System updates the favourited list accordingly |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1.User navigated to the restaurant he wishes to remove from favourites  2. User clicks on the remove favourite  3.System updates and display the remaining favourited restaurants accordingly |
| Alternative Flows: | If user has no favourited restaurants, system displays no favourited restaurant. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.7 My Bookings Page**

**3.7.1 View Successful Reservations**

| Use Case ID: | UC14 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Successful Reservations | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), My Bookings Page(primary), Reservation System(primary), System(secondary), Home Page(secondary) |
| --- | --- |
| Description: | This use case allows the user to view all of his successful reservations on My Bookings Page. |
| Preconditions: |  |
| Postconditions: | The Reservation System should display all of the user’s successful reservations on My Bookings Page. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects “My Booking Page” on the Home Page. 2. The Reservation System should display all of the user’s successful reservations on My Bookings Page. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.7.2 Manage Current Reservations**

| Use Case ID: | UC15 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Successful Reservations | | |
| Created By: | Weijun | Last Updated By: | Weijun |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary), My Bookings Page(primary), Reservation System(primary), System(secondary) |
| --- | --- |
| Description: | This use case allows the user to amend any of his current reservations on My Bookings Page. |
| Preconditions: |  |
| Postconditions: | The Reservation System should amend and display all of the user’s updated current reservatoins on My Bookings Page. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user selects one of the successful reservations on My Bookings Page. 2. The Reservation System prompts the user with 2 options, “Edit Reservations” or “Cancel Reservations”. 3. The user selects the “Edit Reservations” option. 4. The Reservation System will prompt the user to input the newly amended details such as date of reservation, time of reservation and number of pax. 5. The user inputs the required information. 6. The Reservation System will open up My Bookings Page. |
| Alternative Flows: | AF12.1: At Step 2, if the user selects the “Cancel Reservations” option.   1. The Reservation System will cancel the user’s successful reservation in the System. 2. Go to Step 6. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | The user should be allowed to amend or cancel his booking at any time before the scheduled date and time of reservation. |
| Notes and Issues: |  |

**3.7.3 View Completed Reservations**

| Use Case ID: | UC16 | | |
| --- | --- | --- | --- |
| Use Case Name: | View completed reservation | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows user to view reservation they have completed |
| Preconditions: | The user has previously made a reservation |
| Postconditions: | The User is able to submit ratings |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User navigates to view completed reservation  2. System display a list of reservations the user has completed  3. System displays an option to submit ratings and review |
| Alternative Flows: |  |
|  |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

| Use Case ID: | UC17 | | |
| --- | --- | --- | --- |
| Use Case Name: | Submit rating | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows users to submit ratings of restaurants they have previously reserved and completed |
| Preconditions: | The user has previously completed their reservation with the restaurant The user has navigated to the view completed reservation page |
| Postconditions: | The user rating is updated on the restaurant’s page |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user navigates to the restaurant he wishes to rate  2. The user clicks on submit rating option  3. System displays stars from 1 - 5, with optional text box to submit review  4. The user selects the star  5. System updates the restaurant ratings accordingly  6. System updates user’s rating to his profile  7. System awards 2 points to the user |
| Alternative Flows: | If user wants to submit text review proceeds to submit review use case |
|  |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

| Use Case ID: | UC18 | | |
| --- | --- | --- | --- |
| Use Case Name: | Submit reviews | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows users to submit reviews of restaurants they have previously reserved and completed |
| Preconditions: | The user has navigated to submit ratings |
| Postconditions: | The user review is updated on the restaurant’s page |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. System display optional text box  2. User keys text review  3. User clicks submit  4. System updates text review and reflects on restaurant page according  5. System returns user to home page  6. System updates user’s review to user’s rating and review history  7. System awards 3 points to user |
| Alternative Flows: | If text exceeds 250 words, system display review too long and do not allow user to type any further |
|  |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.8 Manage Profile page**

| Use Case ID: | UC19 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Profile | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows user to manage their profile |
| Preconditions: | The user must have an account and is logged in |
| Postconditions: | The user is able to access the various features of manage profile. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user click on the my profile page tab at the bottom of the home page  2. Systems returns a page which display user’s personal information.  3. System also display a few options - edit profile information, view points , view ratings and reviews history. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | update profile information, view points, view ratings and reviews history |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.8.1 Update profile information**

| Use Case ID: | UC20 | | |
| --- | --- | --- | --- |
| Use Case Name: | Update profile information | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows users to update their profile information |
| Preconditions: | The user was in the manage profile page |
| Postconditions: | The updated information is reflected accordingly in the my profile page. User automatically returns to manage profile page. |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User navigates to the edit profile information  2. System presents fields for updating contact number, name, and dietary preference.  3. The user select the information he wants to update  4. The user select contact number and enters in his contact number  5. System sends OTP to his phone number  6. The user enter OTP he received  7. System verifies OTP and updates contact number in the profile  8. The user selects name and enters desired name  9. System updates name in the profile  10. The user selects dietary preference and selects from a few options - Halal, Vegetarian, No preference  11. System updates dietary preference in profile |
| Alternative Flows: | If user did not select update phone number skip from 2 to 8  If user did not select update name skip from 2 to 10 |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.8.2 View points**

| Use Case ID: | UC21 | | |
| --- | --- | --- | --- |
| Use Case Name: | View points | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows users to view the point they accumulated. |
| Preconditions: | The user was in manage profile page |
| Postconditions: | The System displays the point the user has |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user navigates to the view points in the mange profile page  2. System display points the user has.  3. System display points history |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**3.8.3 View Ratings and Review history**

| Use Case ID: | UC22 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Ratings and Review history | | |
| Created By: | Chee Cheng | Last Updated By: | Chee Cheng |
| Date Created: | 05/02/2024 | Date Last Updated: | 05/02/2024 |

| Actor: | User(primary) |
| --- | --- |
| Description: | This use case allows user to view the ratings and reviews they have previously history |
| Preconditions: | The user was in manage profile |
| Postconditions: | The System displays the history of the rating and reviews they have previously given |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user navigates to the view ratings and review given in the manage profile page.  2. System display history of the rating and reviews the user have given |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |