

---

# Use Cases

for

# MealMate

Prepared by:

Chew Zhen Yi (U2122697E)

Kunthamas Donchada (U2123242J)

Ng Ding Hei Ryan (U2121023D)

Tar Sreeja (U2123104B)

Yeoh Wei Yang (U2121112A)

Nanyang Technological University

20.02.2023

## 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.

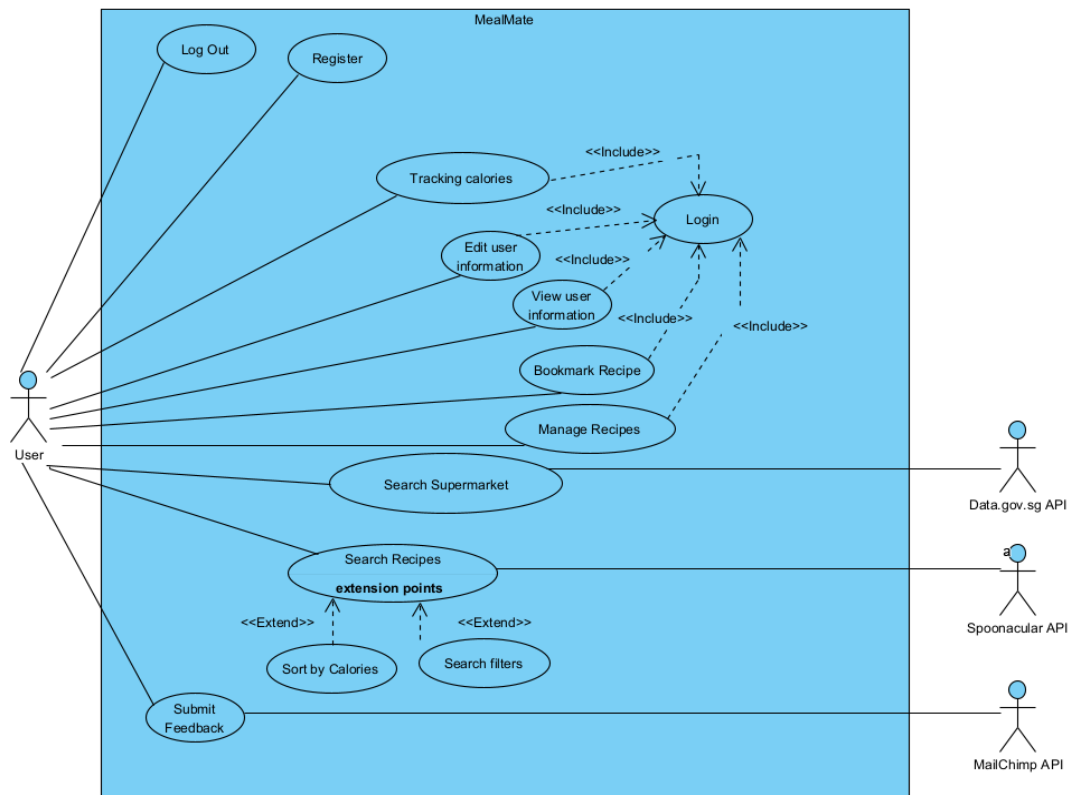
## 2.12. Assumptions

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.

## 3. Use Case Diagram



## 4. Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Yeoh Wei Yang	Last Updated By:	Ng Ding Hei Ryan
Date Created:	31/1/2023	Date Last Updated:	5/2/2023

Actor:	User
Description:	Authenticate account details (username and password) before user accesses any other functions of the system
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has previously registered for an account.</li> <li>2. The user must be connected to the internet (Wifi, mobile data, etc.).</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The user must be able to search for recipes.</li> <li>2. The user must be able to search for nearby supermarkets.</li> <li>3. The user must be able to view their profile.</li> </ol>
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Sign In” button.</li> <li>2. The system prompts the user for a username and password.</li> <li>3. The user inputs their username and password.</li> <li>4. The user clicks on the “Log In” button.</li> <li>5. The system authenticates the user to log in upon successful validation.</li> </ol>
Alternative Flows:	<p>AF-S4: The system detects empty username or password.</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Username or password fields cannot be empty.”</li> <li>2. Return to step 3 in the main flow.</li> </ol> <p>AF-S3: Invalid username / password</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Username not found” or “Invalid password”.</li> <li>2. Return to step 3 in the main flow.</li> </ol>
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> <li>1. The system queries the database correctly to validate against the current list of users.</li> </ol>
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Register Account		
Created By:	Yeoh Wei Yang	Last Updated By:	Ng Ding Hei Ryan
Date Created:	31/1/2023	Date Last Updated:	5/2/2023

Actor:	User
Description:	Registers a new user account
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must not have an existing account.</li> <li>2. The user must be connected to the internet.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The user will be able to access their profile information.</li> <li>2. An email will be sent to the user to notify them about the account creation.</li> </ol>
Priority:	High
Frequency of Use:	Once
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Sign In” button.</li> <li>2. The user clicks on “Click here to sign up!” hyperlink on sign in page.</li> <li>3. The system brings the user to the registration page.</li> <li>4. The system prompts the user to enter 4 items: <ol style="list-style-type: none"> <li>4.1. Username</li> <li>4.2. Email</li> <li>4.3 Password</li> <li>4.4 Confirm password</li> </ol> </li> <li>5. The user clicks the “Sign Up” button.</li> <li>6. The system adds user information to its database of current users and authenticates the login of the user.</li> </ol>
Alternative Flows:	<p>AF-S4-1. The system detects an empty field.</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Username/Password/Email fields cannot be empty”.</li> <li>2. Return to step 4 in the main flow.</li> </ol> <p>AF-S4-2. Duplicate username/email.</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Username/Email taken.”</li> <li>2. Return to step 4 in the main flow.</li> </ol> <p>AF-S4-3. Mismatch between password and confirm password fields.</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Password and Confirm password mismatch. Please check your password and enter again.”</li> <li>2. Return to step 4 in the main flow.</li> </ol> <p>AF-S4-4. Invalid email address</p> <ol style="list-style-type: none"> <li>1. The system will output an error message “Incorrect email address format. Please enter a valid email address.”</li> <li>2. Return to step 4 in the main flow.</li> </ol>

Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Search for Recipes		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	1/2/2023	Date Last Updated:	6/2/2023

Actors:	User, Spoonacular API
Description:	Searches for recipes
Preconditions:	1. The user must be connected to the internet.
Postconditions:	The user gets the desired recipe or receives an appropriate message in the absence of the desired recipe from the system's database.
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system displays the home page.</li> <li>2. The user selects the 'Recipes' button.</li> <li>3. The system displays the 'Browse Recipes' page.</li> <li>4. The user enters the name of a dish/recipe.</li> <li>5. The user clicks on the right arrow (search button).</li> <li>6. The system displays the list of recipes matching the search string.</li> <li>7. The user selects the desired recipe from the list by clicking on it.</li> <li>8. The system displays the selected recipe.</li> </ol>
Alternative Flows:	<p>AF-S4. The user wants to search recipes by ingredients instead of dish/recipe title.</p> <ol style="list-style-type: none"> <li>1. The user enters <math>\leq 5</math> ingredients in the search field, separated by commas.</li> <li>2. The system accepts each entered ingredient as a filter.</li> <li>3. The user clicks on the right arrow (search button).</li> <li>4. The system applies the ingredient filters and displays a list of recipes that contain the input ingredients.</li> <li>5. Continue with step 7 in the main flow.</li> </ol> <p>AF-S6. The user's desired recipe is not available in the system's database.</p> <ol style="list-style-type: none"> <li>1. The system displays the message "Sorry! This recipe is not available."</li> <li>2. The user clicks on the 'OK' button on the message.</li> <li>3. Return to step 3 in the main flow.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. The system does not accept <math>&gt; 5</math> ingredients. Further typing displays a message "Up to 5 ingredients allowed!".</li> </ol>
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> <li>1. The system queries the database correctly to validate against the current list of recipes.</li> </ol>
Notes and Issues:	-



Use Case ID:	4		
Use Case Name:	Sort by calories		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	12/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	Sorts search results (recipes) by calories
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> </ol>
Postconditions:	The user sorts the displayed list of recipes based on calorie information.
Priority:	Low
Frequency of Use:	Ad-hoc
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Search for Recipes to sort the displayed list of recipes by calories.</li> <li>2. 'Calories': The user can slide on the calorie scale to determine a calorie range.</li> <li>3. The system applies the selected calorie range and displays the appropriate list of recipes.</li> </ol>
Alternative Flows:	<p>AF-S3. The database does not contain any recipes in the selected calorie range.</p> <ol style="list-style-type: none"> <li>1. The system displays the error message "Sorry, no recipe in this calorie range! Try again."</li> <li>2. Return to step 2 in the main flow.</li> </ol>
Exceptions:	-
Includes:	3. Search for Recipes
Special Requirements:	
Assumptions:	<ol style="list-style-type: none"> <li>1. The system contains at least one recipe.</li> <li>2. The system queries the database correctly to validate against the current list of recipes.</li> </ol>
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Apply search filters		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	6/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	Applies search filters such as diet type and meal type
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> </ol>
Postconditions:	The user applies search filters to narrow down the displayed list of recipes based on certain criteria such as diet type and meal type.
Priority:	Low
Frequency of Use:	Ad-hoc
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Search for Recipes to apply search filters.</li> <li>2. The user applies one or both of the following search filters. <ol style="list-style-type: none"> <li>2.1 'Diet Type' which has the following options: 'Dairy-Free', 'Gluten-Free', 'Vegan' and 'Vegetarian'</li> <li>2.2 'Meal Type' which has the following options: 'Breakfast', 'Lunch', 'Dinner' and 'Dessert'.</li> </ol> </li> <li>3. The system applies the selected filters and displays the appropriate list of recipes.</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	3. Search for Recipes
Special Requirements:	
Assumptions:	<ol style="list-style-type: none"> <li>1. The system contains at least one recipe for each diet type and each meal type.</li> <li>2. The system queries the database correctly to validate against the current list of recipes.</li> </ol>
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	Bookmarking a recipe		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	2/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	Bookmarks a recipe to save it as a favourite
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> <li>4. The user has found a desired recipe on the website.</li> </ol>
Postconditions:	The user marks the recipe as a favourite and the recipe gets added to the list of favourites on the user's profile.
Priority:	Low
Frequency of Use:	Ad-hoc
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use cases Login and Search for Recipes to display the desired recipe.</li> <li>2. The user reads the recipe.</li> <li>3. The user marks it as a favourite by clicking on the 'Add to Favourites' button.</li> <li>4. The system adds the recipe to the user's list of favourite recipes.</li> </ol>
Alternative Flows:	AF-S4. The recipe is already on the user's list of favourite recipes. <ol style="list-style-type: none"> <li>1. The system displays a button as 'Remove from Favourites'.</li> <li>2. The user clicks on the button to remove it from their favourites list.</li> <li>3. The system removes the recipe from the user's list of favourite recipes.</li> </ol>
Exceptions:	-
Includes:	<ol style="list-style-type: none"> <li>1. Login</li> <li>3. Search for Recipes</li> </ol>
Special Requirements:	
Assumptions:	<ol style="list-style-type: none"> <li>1. The system contains at least one recipe.</li> <li>2. The system queries the database correctly to validate against the user's current list of favourite recipes.</li> </ol>
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Search Nearby Supermarkets		
Created By:	Tar Sreeja	Last Updated By:	Ng Ding Hei Ryan
Date Created:	2/2/2023	Date Last Updated:	5/2/2023

Actors:	User, Data.gov.sg Data API
Description:	Searches for nearby supermarkets
Preconditions:	1. The user must be connected to the internet.
Postconditions:	The user finds information about the desired supermarket.
Priority:	High
Frequency of Use:	Once a week
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system displays the home page.</li> <li>2. The user selects the 'Store Finder' button.</li> <li>3. The system displays the 'Browse Stores' page.</li> <li>4. The user enters the postal code of their current location.</li> <li>5. The user clicks on the right arrow (search button).</li> <li>6. The system displays the list of supermarkets within a 5km radius from the user's current location.</li> <li>7. The user selects the desired supermarket from the list by clicking on it.</li> <li>8. The system displays information about the selected supermarket.</li> </ol>
Alternative Flows:	<p>AF-S2. Skip to step 4 in the main flow.</p> <p>AF-S4. The user wants to search for a supermarket by name.</p> <ol style="list-style-type: none"> <li>1. The user enters the name of a supermarket.</li> <li>2. The user clicks on the right arrow (search button).</li> <li>3. The system displays the list of supermarkets with names that match the search string.</li> <li>4. Continue with step 7 in the main flow.</li> </ol> <p>AF-S6. There are no supermarkets within a 5km radius from the postal code entered by the user.</p> <ol style="list-style-type: none"> <li>1. The system displays the message "Sorry! No nearby supermarkets. Try another location."</li> <li>2. The user clicks on the 'OK' button on the message.</li> <li>3. Return to step 3 in the main flow.</li> </ol>
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	1. The system queries the database correctly to validate against the current list of nearby supermarkets.
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	View User Information		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	1/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	User to access his/her own profile
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> </ol>
Postconditions:	The user can view and edit information on their profile.
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Login to access their own profile.</li> <li>2. The system displays the home page.</li> <li>3. The user selects the button with their username on it.</li> <li>4. The system displays a dropdown with two options - 'View Profile' and 'Log Out'.</li> <li>5. The user selects 'View Profile'.</li> <li>6. The system displays the 'Profile' page which includes the following information. <ol style="list-style-type: none"> <li>3.1 Username</li> <li>3.2 Height</li> <li>3.3 Weight</li> <li>3.4 BMI</li> <li>3.5 Caloric intake information <ol style="list-style-type: none"> <li>3.5.1 Caloric intake in the past 31 days</li> <li>3.5.2 Recommended caloric intake (daily)</li> <li>3.5.3 Calories consumed</li> <li>3.5.4 No. of calories away from the recommended daily limit</li> </ol> </li> <li>3.6 Favourites - List of favourite recipes</li> <li>3.7 My Recipes - List of recipes the user has created</li> </ol> </li> <li>7. The system also displays an 'Edit Profile' button.</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	1. Login
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> <li>1. The system queries the database correctly to validate against the user's information.</li> </ol>
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Manage Recipes		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	4/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	Manages their own recipes (add a recipe, edit recipe, delete recipes, etc.)
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. A recipe is added to the system's database which can be accessed and bookmarked by the user who uploaded the recipe as well as other users. OR</li> <li>2. A recipe is deleted from the system's database. OR</li> <li>3. An edit is made to a previously uploaded recipe.</li> </ol>
Priority:	High
Frequency of Use:	Once in two weeks
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Login to be allowed to manage recipes.</li> <li>2. The system displays the home page.</li> <li>3. The user selects the 'Recipes' button.</li> <li>4. The system displays a dropdown with 2 options - 'Add a Recipe' and 'Browse Recipes'.</li> <li>5. The user selects 'Add a Recipe'.</li> <li>6. The system displays a recipe template which includes the following information. <ol style="list-style-type: none"> <li>4.1 Recipe Title</li> <li>4.2 Preparation Time</li> <li>4.3 Serving Portions</li> <li>4.4 Calories</li> <li>4.5 Description</li> <li>4.6 Ingredients List</li> <li>4.7 Steps</li> </ol> </li> <li>7. The user fills one or more of the above fields.</li> <li>8. The user clicks on 'Save and Upload'.</li> <li>9. The system adds the recipe to its database.</li> </ol>
Alternative Flows:	<p>AF-S3-1. The user wants to edit a previously uploaded recipe.</p> <ol style="list-style-type: none"> <li>1. The system uses the included use case View User Information to allow the user to edit a previously uploaded recipe.</li> <li>2. The user selects the desired recipe under the 'My Recipes' section on their profile page.</li> <li>3. The system displays the selected recipe.</li> <li>4. The user selects the 'Edit Recipe' button.</li> <li>5. The system displays a cursor next to one of the editable fields in the recipe.</li> <li>6. The user makes the desired changes to the recipe.</li> </ol>

	<ol style="list-style-type: none"> <li>7. The user clicks on the 'Save annd Upload' button.</li> <li>8. The system updates the recipe in its database.</li> </ol> <p>AF-S3-2. The user wants to delete a recipe.</p> <ol style="list-style-type: none"> <li>1. The system uses the included use case View User Information to allow the user to edit a previously uploaded recipe.</li> <li>2. The user selects the desired recipe under the 'My Recipes' section on their profile page.</li> <li>3. The system displays the selected recipe.</li> <li>4. The user selects the 'Delete Recipe' button.</li> <li>5. The system displays a message "Are you sure you want to delete this recipe?" with two options 'Yes' and 'No'.</li> <li>6. The user selects 'Yes'.</li> <li>7. The system deletes the recipe from its database.</li> </ol>
Exceptions:	-
Includes:	<ol style="list-style-type: none"> <li>1. Login</li> <li>8. View User Information</li> </ol>
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	Edit User Information		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	4/2/2023	Date Last Updated:	12/2/2023

Actor:	User
Description:	Edits profile information
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> </ol>
Postconditions:	The system saves the user's updated profile in its database.
Priority:	High
Frequency of Use:	Once a week
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Login to allow the user to edit their profile information.</li> <li>2. The system displays the home page.</li> <li>3. The user selects the button with their username on it.</li> <li>4. The system displays a dropdown with two options - 'View Profile' and 'Log Out'.</li> <li>5. The user selects 'View Profile'.</li> <li>6. The system displays the user's 'Profile' page.</li> <li>7. The user selects the 'Edit Profile' button.</li> <li>8. The system displays a cursor next to one of the editable fields (username, weight and height).</li> <li>9. The user edits one or more of the editable fields on their profile.</li> <li>10. The user clicks on 'Save Profile'.</li> <li>11. The system updates the user's profile in its database.</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	1. Login
Special Requirements:	-
Assumptions:	-
Notes and Issues:	An editable field is a field that can be changed by the user. BMI is a non-editable field, i.e. the user cannot edit BMI as it is calculated by the system based on the weight and height inputs entered by the user.





Use Case ID:	11		
Use Case Name:	Add Meal		
Created By:	Tar Sreeja	Last Updated By:	Chew Zhen Yi
Date Created:	15/2/2023	Date Last Updated:	15/2/2023

Actor:	User
Description:	Adds a meal (i.e. Breakfast, Lunch, Dinner and/or Snacks) to the user's food diary to track their daily calorie consumption.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> </ol>
Postconditions:	The system saves the user's caloric intake in its database to track and report the user's calorie consumption.
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system uses the included use case Login to allow the user to add meals to track their calorie consumption.</li> <li>2. The user clicks the 'Add New Meal' button on the home page.</li> <li>3. The user clicks the 'Browse Recipes' dropdown option.</li> <li>4. The system redirects the user to the Browse Recipes page.</li> <li>5. The user browses for a specific recipe and opens the recipe page.</li> <li>6. The user clicks the 'Add to Meal' button on the recipe page.</li> <li>7. The system displays a modal for the user to choose which meal type the recipe should be assigned to.</li> <li>8. The user selects the meal type and clicks on 'Add'.</li> <li>9. The system saves the recipe as the specified meal type and adds it into the user's food diary.</li> <li>10. The system displays a success toast message, "Meal successfully added!"</li> </ol>
Alternative Flows:	<p>AF-S3. The user wants to add a custom meal.</p> <ol style="list-style-type: none"> <li>1. The user clicks the 'Create Custom Meal' dropdown option.</li> <li>2. The system displays a modal for the user to input: <ol style="list-style-type: none"> <li>a. Meal Type</li> <li>b. Meal Name</li> <li>c. Calories (kcal)</li> </ol> </li> <li>3. The user enters the aforementioned inputs and clicks on 'Add'.</li> <li>4. The system saves the recipe as the specified meal type and adds it into the user's food diary.</li> </ol>
Exceptions:	-
Includes:	1. Login
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	Log Out		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	5/2/2023	Date Last Updated:	5/2/2023

Actor:	User
Description:	Logs out of the website
Preconditions:	<ol style="list-style-type: none"> <li>1. The user must be connected to the internet.</li> <li>2. The user must have a registered account on the website.</li> <li>3. The user must be successfully logged in to the website.</li> </ol>
Postconditions:	The user is successfully logged out of the website. Subsequent use of the website would require logging in.
Priority:	Low
Frequency of Use:	Once a week
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system displays the home page.</li> <li>2. The user selects the button with their username on it.</li> <li>3. The system displays a dropdown with three options - 'View Profile', 'Edit Profile' and 'Log Out'.</li> <li>4. The user selects 'Log Out'.</li> <li>5. The system displays a message "You have been successfully logged out!"</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	13		
Use Case Name:	Submit feedback/query		
Created By:	Tar Sreeja	Last Updated By:	Tar Sreeja
Date Created:	6/2/2023	Date Last Updated:	6/2/2023

Actors:	User, MailChimp API
Description:	Submits feedback on the website or a query
Preconditions:	1. The user must be connected to the internet.
Postconditions:	The user submits a query or feedback on the website.
Priority:	Low
Frequency of Use:	Ad-hoc
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system displays the home page.</li> <li>2. The user selects the 'Contact Us' button.</li> <li>3. The system displays a form with the following fields. <ol style="list-style-type: none"> <li>3.1 Query/Feedback</li> <li>3.2 Email address</li> </ol> </li> <li>4. The user fills in the above fields.</li> <li>5. The user clicks on the 'Submit' button.</li> <li>6. The system displays the message 'Thank you for your query/feedback!'.</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-