**Use Cases**

**for**

**TrainTogether**

**Version 3.2 approved**

**Prepared by** Lee Yen Foong Ernest, Low Zhan Long,

Solomon Duke Tneo Yruan Rui, Tey Li Zhang Edmund, Wan Kai Jie

**SC2006 A26 – Coding Pigeons**

**13 April 2023**

# **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 User 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 Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

## **Use Case Descriptions**

| Use Case ID: | 001 | | |
| --- | --- | --- | --- |
| Use Case Name: | Register | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 18 February 2023 | Date Last Updated: | 6 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows users to create an account in the system to access the application’s functions. The user has to enter a unique email address for registration and cannot use an email that has already been registered. |
| Preconditions: | 1. User must be in the Register page |
| Postconditions: | 1. System creates an account 2. System displays the profile page |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User inputs their username, name, email address, password and re-enter their password into the form 2. User submits the form 3. System validates the username, name, email address, password and the re-entered password 4. System creates account 5. System triggers Set Up use case 6. System displays profile page |
| Alternative Flows: | 1. AF-S1: User already has account and clicks on login    1. System triggers Login use case |
| Exceptions: | 001.EX.1 User provides incomplete information during registration   1. System will display “Missing information. Please enter the required fields.” 2. System prompts the User to enter their details 3. Use case resumes at main flow step 1   001.EX.2 User provides an email address that has been used   1. System will display “Email has been used. Please register with another email or login with an existing account.” 2. System prompts the User to enter their details 3. Use case resumes at main flow step 1 |
| Includes: | Set Up, Register with Google Account |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User enters in a valid phone number and email address |
| Notes and Issues: | NIL |

| Use Case ID: | 002 | | |
| --- | --- | --- | --- |
| Use Case Name: | Set Up | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 6 April 2023 | Date Last Updated: | 6 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows users to fill in additional information for their account. |
| Preconditions: | 1. User must have already registered for an account 2. User must not have set up their account |
| Postconditions: | 1. System insert additional information into the User’s account |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. System displays a form which contains Date of Birth, Phone Number, Height in cm, Weight in kg, dropdown menu for gender and file input for Profile Picture. 2. User inputs Date of Birth, Phone Number, Height, Weight, select their gender and upload their Profile Picture. 3. User submits the form. 4. System validates the form. 5. System insert additional information in the form to the User’s account |
| Alternative Flows: | NIL |
| Exceptions: | 002.EX.1 User submitted invalid inputs in the form   1. System displays error message for the invalid input 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1 |
| Includes: | NIL |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User entered their actual Phone Number, Date of Birth, Weight, Height, Gender |
| Notes and Issues: | NIL |

# 

# 

| Use Case ID: | 003 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login | | |
| Created By: | Low Zhan Long | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 2 February 2023 | Date Last Updated: | 6 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to login into the system to access the relevant functions of the application. To login to the system, all users have to enter their unique email address and their password. |
| Preconditions: | 1. User does not have an existing session cookie 2. User has a valid account |
| Postconditions: | 1. System displays the profile page |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User inputs his email into the System 2. User inputs his password into the System 3. User submits email and password to the System 4. System verifies the email and password 5. System displays the user’s profile page |
| Alternative Flows: | 1. AF-S1: User forgets password and choose Forget Password    1. System trigger Forget Password use case |
| Exceptions: | 003.EX.1 User enters Invalid email or password   1. System displays “Invalid email and/or password” message 2. System prompts for email and password 3. Use case resumes at main flow step 1. |
| Includes: | Forget Password |
| Extends | NIL |
| Special Requirements: | * System should hash and salt the password entered by the User to verify if the hashed passwords match |
| Assumptions: | * Users should not be able to brute force and try out all the emails and passwords to gain entry into the system |
| Notes and Issues: | NIL |

| Use Case ID: | 004 | | |
| --- | --- | --- | --- |
| Use Case Name: | Forget Password | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 11 April 2023 | Date Last Updated: | 11 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to reset their password. A password reset link will be sent to the account that the user registered with. |
| Preconditions: | 1. User must have an account registered 2. User is in Forgot Password Page |
| Postconditions: | 1. Password reset link is sent to User 2. System update password of User |
| Priority: | Medium |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. System displays Forgot Password 2. User enters their email address 3. User submits their email address 4. System validates the email address 5. System verifies the email address 6. System send reset link to the email address 7. System displays “A password reset link has been sent to your email.” 8. User click on reset link in email 9. User is redirected back to the TrainTogether 10. User enters their new password and re-enters their password for confirmation 11. User submit password 12. System validates the password 13. System updates the password for the User 14. System directs user to log in page |
| Alternative Flows: | NIL |
| Exceptions: | 004.EX.1 User submitted invalid email format   1. System displays “Invalid Email Format” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1   004.EX.2 User submitted email that is not registered   1. System displays “Email is not registered to an account” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1   004.EX.3 User submitted password that does not contain at least one Uppercase, Lowercase, one number and special character   1. System displays “Password must contain at least one Uppercase, Lowercase, one number and special character” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 9 |
| Includes: | NIL |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

# 

| Use Case ID: | 005 | | |
| --- | --- | --- | --- |
| Use Case Name: | Locate Exercise Facility | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 6 April 2023 | Date Last Updated: | 6 April 2023 |

| Actor: | User, Google Map, Data Gov API |
| --- | --- |
| Description: | This use case allows users to locate exercise facilities in Singapore and receive information about the name, location, opening hours and website of the exercise facility. |
| Preconditions: | 1. User must have an account 2. User must have been logged in 3. User must be in the Exercise Facility Tab |
| Postconditions: | NIL |
| Priority: | High |
| Frequency of Use: | 5/session |
| Flow of Events: | 1. System triggers the View Map use case 2. User selects an Exercise Facility for more information 3. System query Data Gov API to get the Exercise Facility name, location, opening hours and website. 4. System displays the queried data |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | View Map |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

| Use Case ID: | 006 | | |
| --- | --- | --- | --- |
| Use Case Name: | Locate Food Facility | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Solomon Tneo |
| Date Created: | 5 February 2023 | Date Last Updated: | 11 April 2023 |

| Actor: | User, Google Map, Data Gov API |
| --- | --- |
| Description: | This use case allows users to locate healthy food facilities in Singapore and receive information about the store name, food type, dining guidelines, venue type and health category |
| Preconditions: | 1. User must have an account 2. User must be logged in 3. User must be in the Hungry Go Where tab |
| Postconditions: | 1. System shows details of the food store |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User clicks on HungryGoWhere in the Header 2. System triggers the View Map use case 3. User can filter Food Facility from the food type, dining guidelines, venue type and health category 4. User selects a location for more information 5. System query Data Gov API to get the Food Facility name, location, opening hours and website 6. System displays the store name, food type, dining guidelines, venue type and health category |
| Alternative Flows: | NIL |
| Exceptions: | 006.EX.1 User does not enter any food requirements   1. System will display all available food stores |
| Includes: | NIL |
| Extends | View Map |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * There are 5 filters for users to use to filter food stores, including, health category, food type, dining guideline, venue and regions. * For health category, the available options are: lower in sugar, higher in wholegrains, lower in calories, no added sugar, source of dietary fibre, eat 2+2 serving of fruits, vegetables daily * For food type, the available options are: drinks, bread, cakes, cookies, pastries, Western, desserts, snacks and kuih, bubble tea, Chinese, fast food, Malay, Thai, cut fruit, Japanese, Korean, local flavours, Fusion, Indian * For dining guideline, the available options are Halal and Non-Halal * For venue, the available options are: hawker centre, bakery, cafe, kiosk, restaurant, fast food, coffee shop, dessert, f&b on digital platforms * For regions, the available options are: North, East, Central, North-East, West * User can click on ‘Reset’ to reset the filter |

| Use Case ID: | 007 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Map | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Solomon Tneo |
| Date Created: | 5 February 2023 | Date Last Updated: | 11 April 2023 |

| Actor: | User, Google |
| --- | --- |
| Description: | Able to navigate, scroll and zoom on location of exercise facility or food facility |
| Preconditions: | 1. User must have an account 2. User must have been logged in 3. User must be in the Exercise Facility Tab or Hungry Go Where Tab |
| Postconditions: | 1. System shows Map |
| Priority: | High |
| Frequency of Use: | 5/session |
| Flow of Events: | 1. User clicks on Exercise facilities tab 2. System displays a map of the user’s current location 3. System request Google for the Google Maps 4. User zoom in or out using punch gestures or using their mouse gestures 5. System updates map based on gestures 6. User scroll around the map by dragging it with their mouse 7. System updates map based on gesture 8. User can query for a specific location on the map by typing into the search bar 9. System updates map based on query. |
| Alternative Flows: | 1. AF-S1: User clicks on Hungry Go Where tab    1. System displays a map of the user’s current location    2. System request Google for the Google Maps    3. User zoom in or out using punch gestures or using their mouse gestures    4. System updates map based on gestures    5. User scroll around the map by dragging it with their mouse    6. System updates map based on gestures    7. User can query for a specific location on the map by typing into the search bar    8. System updates map based on query. |
| Exceptions: | 007.EX.1 User query a location that does not exist on the map   1. System will display “Location does not exist” 2. Use case resumes at the main flow at step 4 |
| Includes: | NIL |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

| Use Case ID: | 008 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Exercises | | |
| Created By: | Low Zhan Long | Last Updated By: | Solomon Tneo |
| Date Created: | 2 February 2023 | Date Last Updated: | 11 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to view exercises’ instructions and remarks |
| Preconditions: | 1. User must have an account 2. User must login 3. User must have rights, privilege and access |
| Postconditions: | 1. System displays exercise’s instructions and remarks |
| Priority: | Medium |
| Frequency of Use: | 1/Session |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. User able to view Exercises 5. User selects an exercise 6. System displays exercise’s instructions and remarks |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * System is to sort exercises according to body parts, which are: Arms, Back, Cardio, Core and Legs |

| Use Case ID: | 009 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Exercise Plan | | |
| Created By: | Low Zhan Long | Last Updated By: | Solomon Tneo |
| Date Created: | 2 February 2023 | Date Last Updated: | 13 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows users to manage their exercise plans. Users are able to add, edit and delete their exercise plan. |
| Preconditions: | 1. User must have an account 2. User must login 3. User must have rights, privilege and access |
| Postconditions: | 1. System displays exercise plans that user desires |
| Priority: | Medium |
| Frequency of Use: | 1/Session |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. System displays list of exercise plans and add exercise plan button 5. User selects an exercise plan to view more information 6. System displays the name of the exercise plan and the list of exercises tagged in the exercise plan 7. System displays edit exercise plan button and delete exercise plan button 8. User can select to add, edit and delete exercise plan |
| Alternative Flows: | 1. AF-S7: User selects add exercise plan    1. System displays add exercise plan page    2. User enters name of exercise plan    3. User uploads a picture for the exercise plan    4. User selects on exercises to be added    5. System trigger Add Exercise use case    6. User clicks “Save”.    7. System adds the new exercise plan to the database    8. Use case resumes at main flow step 3 2. AF-S7: Users selects delete exercise plan    1. System triggers Delete Exercise use case    2. Systems deletes the exercise plan from the database    3. Use case resumes at main flow step 3 3. AF-S7: Users selects edit exercise plan    1. System display edit exercise plan page    2. User enters new name of exercise plan    3. User uploads a new picture for the exercise plan    4. System triggers Edit Exercise use case    5. User selects submit    6. System saves exercise plans detail and updates database    7. Use case resumes at main flow at step 3 |
| Exceptions: | 009.EX.1. Exercise Plans have no exercises   * System will display “no exercise” message   009.EX.2. Users did not create any exercise plans   * System will display list of default exercise plans |
| Includes: | Login, Add Exercise, Edit Exercise, Delete Exercise |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * System has a list of default exercise plans for each fitness corners shown on the map |

| Use Case ID: | 010 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add Exercise | | |
| Created By: | Low Zhan Long | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to add exercise into an exercise plan |
| Preconditions: | 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan |
| Postconditions: | 1. Exercise is added into the exercise plan |
| Priority: | Medium |
| Frequency of Use: | 2/Session |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. User clicks on add exercise plan 5. User enters name of exercise plan 6. User chooses an exercise group 7. System displays list of exercises tagged under the chosen exercise group 8. System selects the “plus icon” beside the exercise name 9. User edits the exercise’s reps and sets 10. User selects save 11. System adds the new exercise plan into the list of exercise plans as well as into the database and displays added exercises in the exercise plan. |
| Alternative Flows: | 1. AF-S3 Users selects edit exercise plan    1. System displays list of exercises tagged under the chosen exercise group    2. User adds exercise to the exercise plan    3. User edits the exercise’s reps and sets    4. User selects submit    5. System adds new exercise plan to the database    6. System displays added exercises in the exercise plan. |
| Exceptions: | 008.EX.1. Users did not create any exercise plans   * User is unable to add exercise |
| Includes: | Login, Manage Exercise Plan |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User’s desired exercise is available in TrainTogether App * User is able to carry out the exercise add at user’s desired location |
| Notes and Issues: | * User does not have rights to add exercise for default exercise plans |

| Use Case ID: | 011 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Exercise | | |
| Created By: | Low Zhan Long | Last Updated By: | Tey Li Zhang Edmund |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to edit exercise in an exercise plan |
| Preconditions: | 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan 5. User must have added at least 1 exercise |
| Postconditions: | 1. Exercise edited in the exercise plan |
| Priority: | Medium |
| Frequency of Use: | 0-1/Session |
| Flow of Events: | 1. User clicks on Workout 2. System displays Workout page 3. User clicks on an exercise plan 4. System displays list of exercises tagged under the chosen exercise plan 5. User selects edit exercise beside the exercise name 6. System displays list of exercises tagged under the chosen exercise group 7. User selects the “plus icon” beside the exercise name 8. User edits the exercise’s reps and sets 9. User selects submit 10. System updates the exercise plan in the database 11. System displays updated exercises in the exercise plan. |
| Alternative Flows: | NIL |
| Exceptions: | 009.EX.1. Users did not create any exercise plans   * User is unable to edit exercise   009 EX 2. Users did not add exercise in exercise plans   * User is unable to edit exercise |
| Includes: | Login |
| Extends: | Manage Exercise Plan |
| Special Requirements: | NIL |
| Assumptions: | * User’s desired exercise is available in TrainTogether App * User is able to carry out the exercise add at user’s desired location |
| Notes and Issues: | * User does not have rights to edit exercise for default exercise plans |

| Use Case ID: | 012 | | |
| --- | --- | --- | --- |
| Use Case Name: | Delete Exercise | | |
| Created By: | Low Zhan Long | Last Updated By: | Solomon Tneo |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to delete exercise from an exercise plan |
| Preconditions: | 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan 5. User must have added at least 1 exercise |
| Postconditions: | 1. Exercise deleted in the exercise plan |
| Priority: | Medium |
| Frequency of Use: | 0-1/Session |
| Flow of Events: | 1. User clicks on Workout 2. System displays Workout page 3. User clicks on an exercise plan 4. System displays list of exercises tagged under the chosen exercise plan 5. User selects the edit exercise besides the exercise name 6. System displays list of exercises tagged under the chosen exercise group 7. User selects the delete exercise plan button beside the exercise name 8. System remove exercise from the exercise plan 9. User selects submit 10. System displays updated exercise plans. |
| Alternative Flows: | NIL |
| Exceptions: | 010.EX.1. Users did not create any exercise plans   * User is unable to delete exercise   010 EX 2. Users did not add exercise in exercise plans   * User is unable to delete exercise |
| Includes: | Login |
| Extends: | Manage Exercise Plan |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * User does not have rights to delete exercise for default exercise plans |

| Use Case ID: | 013 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Review | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 13 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to Manage Review about Exercise |
| Preconditions: | 1. User must have selected the Exercise |
| Postconditions: | NIL |
| Priority: | Low |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. System displays a list of Exercises under the Exercise Group 3. User selects an Exercise 4. System displays details about the selected Exercise 5. User clicks to ‘Review’ to view the reviews of the selected Exercise 6. System displays the list of reviews tagged to the particular Exercise |
| Alternative Flows: | 1. AF: User view reviews from dashboard    1. User clicks on reviews from dashboard    2. System displays all the reviews from all exercises 2. AF-S5: User selects add review    1. System displays add review page    2. System trigger Add Review use case    3. System adds the new review to the database    4. Use case resumes at main flow step 5 3. AF: Users selects delete review    1. User clicks on Profile    2. User clicks on Reviews component    3. System triggers Delete Review use case    4. Systems deletes the review from the database    5. Use case resumes at alternative flow step 3 |
| Exceptions: | NIL |
| Includes: | Login, Add Review, Delete Review |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

| Use Case ID: | 014 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add Review | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 13 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to Add Review about Exercise |
| Preconditions: | 1. User must have selected the Exercise 2. User must be logged in |
| Postconditions: | 1. The new review is added to database |
| Priority: | Low |
| Frequency of Use: | 1/Exercise Plan |
| Flow of Events: | 1. User clicks on Workout 2. System displays Workout page 3. User selects an Exercise Group 4. User select an Exercise 5. User clicks view Review on selected Exercise 6. User clicks add Review on selected Exercise 7. User input details of Review Details 8. User clicks Submit 9. System generates a Review for Exercise |
| Alternative Flows: | NIL |
| Exceptions: | 014.EX.1 User provides does not give a rating when submitting a review   * System will display an error message and prompts the user to give a rating |
| Includes: | NIL |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

| Use Case ID: | 015 | | |
| --- | --- | --- | --- |
| Use Case Name: | Delete Review | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to Delete Review about Exercise |
| Preconditions: | 1. User must have done a review about an exercise before 2. User must be logged in |
| Postconditions: | 1. The particular review is deleted from database |
| Priority: | Low |
| Frequency of Use: | 1/Review |
| Flow of Events: | 1. User clicks on Profile 2. User clicks on Reviews component 3. User selects the Review 4. User clicks the delete button to delete the review 5. System deletes Review |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User is sure of the choice to delete the review |
| Notes and Issues: | * There is no editing of reviews available, hence a user is required to delete the review before adding a new one if there are any edits the user wants to make to a review |

| Use Case ID: | 016 | | |
| --- | --- | --- | --- |
| Use Case Name: | Schedule Exercise | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to schedule an exercise plan |
| Preconditions: | 1. There must be at least one exercise plan available to be scheduled for |
| Postconditions: | 1. Newly scheduled exercise plan is added to database |
| Priority: | Medium |
| Frequency of Use: | 1/Exercise Plan |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks onto the Schedule Exercise button 3. User input schedule details 4. User clicks Submit 5. System updates newly scheduled exercise |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User only schedules an exercise plan that is on a future date and time |
| Notes and Issues: | * User is able to reset the Exercise Plan to the default one when inputting schedule details |

| Use Case ID: | 017 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Scheduled Exercise | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | Able to view Scheduled Exercise from profile |
| Preconditions: | 1. User must have an account |
| Postconditions: | NIL |
| Priority: | Medium |
| Frequency of Use: | 1/Session |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks Profile 3. User selects Scheduled Exercise component 4. System displays all Scheduled Exercises belonging to the user |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * Scheduled Exercises are ordered according to date from the closest to the furthest date, in the Scheduled Exercise component page |

| Use Case ID: | 018 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Profile | | |
| Created By: | Lee Yen Foong Ernest | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to view his account from the settings function |
| Preconditions: | 1. User must have an account |
| Postconditions: | 1. User is able to view his account 2. User is unable to login into the same account |
| Priority: | Medium |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. User clicks Profile 3. System displays User dashboard 4. User clicks on the Settings button 5. User clicks on Account button 6. System displays basic profile details 7. If user selects edit profile, System triggers the Edit Profile use case 8. If user selects delete account, System triggers the Delete Profile use case |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login, Edit Profile, Delete Account |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | * User is able to view other types of profile details in the dashboard page, which is demonstrated in other use cases |

| Use Case ID: | 019 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Profile | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to edit and update his own profile information |
| Preconditions: | 1. User must have an account 2. User must be logged in |
| Postconditions: | 1. New Details have been updated |
| Priority: | Medium |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User clicks onto Profile 2. System displays User dashboard 3. User clicks on Account Settings 4. User edits information about Account 5. User clicks Done Button 6. System update Profile |
| Alternative Flows: | NIL |
| Exceptions: | 019.EX.1 User enters invalid username, mobile number or email address   * System will display an error message and prompt user to re-enter their particulars again |
| Includes: | NIL |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | * User should enter in a valid username, mobile number and email address |
| Notes and Issues: | * User is also able to switch to Notifications segment to update notification settings * User is also able to switch to Security segment to change their password |

| Use Case ID: | 020 | | |
| --- | --- | --- | --- |
| Use Case Name: | Delete Account | | |
| Created By: | Low Zhan Long | Last Updated By: | Wan Kai Jie |
| Date Created: | 2 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to delete his own account through the Account settings function |
| Preconditions: | 1. User must have an account 2. User must be logged in |
| Postconditions: | 1. User’s account has been deleted 2. User is unable to login into the same account 3. User will be redirected back to the login page |
| Priority: | Medium |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User clicks onto Profile 2. System displays User dashboard 3. User clicks on Account Settings 4. User clicks on Security segment 5. User clicks on Delete Account Button 6. System sends warning message to user 7. User clicks on Confirm Delete Account Button 8. System notifies user through email that account have been deleted 9. System logs User out of account and remove account |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Extends: | NIL |
| Special Requirements: | NIL |
| Assumptions: | * All relevant databases will also delete details related to the User * User is sure of the choice to delete their account, and is not accidental |
| Notes and Issues: | NIL |

| Use Case ID: | 021 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Fitness Goals | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Lee Yen Foong Ernest |
| Date Created: | 5 February 2023 | Date Last Updated: | 13 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to view all the fitness goals that he has created. User is able to add, edit and delete his fitness goals. |
| Preconditions: | 1. User must have an account 2. User must have at least one fitness goal when editing or deleting |
| Postconditions: | 1. User is able to view his updated fitness goals |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘Manage Fitness Goal Page’ component 5. System retrieves User’s fitness goals 6. System displays user’s fitness goals in ‘Fitness Goal’ component |
| Alternative Flows: | 1. AF-S7: User decides to add a fitness goal    1. User clicks on ‘+’ button to add a new fitness goal    2. System displays ‘Add Fitness Goal Page’    3. User enter details of fitness goal    4. User clicks Submit button    5. System generates the updated fitness goals list    6. User clicks “Save”.    7. System adds the new fitness goal to the database 2. AF-S7: Users decides to edit fitness goal    1. User clicks on ‘Edit’ button to edit a fitness goal    2. System retrieves user’s fitness goals    3. System displays user’s fitness goals in ‘Edit Fitness Goal Page’    4. User changes details of the fitness goal and submit    5. System updates the fitness goal and displays updated fitness goals list in ‘Fitness Goal’ component in Dashboard Page 3. AF-S7: Users decides to delete fitness goal    1. User clicks on ‘Delete’ button to delete a fitness goal    2. System deletes the fitness goal    3. System generates the updated fitness goals list |
| Exceptions: | 021.EX.1 User provides incomplete details when adding fitness goal   * System will display an error message and prompts the user to enter the details correctly   021. EX.2 User provides incorrect details when adding fitness goal   * System will display an error message and prompts the user to enter the details correctly   021. EX.1 User provides incorrect details when editing fitness goal   * System will display an error message and prompts the user to enter the details correctly |
| Includes: | Login |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | User enters all details of fitness goals correctly   1. Target value and current value must be an integer 2. Target date cannot be in the past 3. Target date cannot be more than 100 years in the future |
| Notes and Issues: | NIL |

| Use Case ID: | 022 | | |
| --- | --- | --- | --- |
| Use Case Name: | Calculate BMI | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Lee Yen Foong Ernest |
| Date Created: | 5 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to calculate BMI using a height and weight value |
| Preconditions: | 1. User must have an account |
| Postconditions: | 1. User is able to view his BMI index value |
| Priority: | Medium |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘BMI Calculator’ component 5. System retrieves User’s height and weight and calculates BMI index 6. System displays BMI index value in ‘BMI Calculator’ component |
| Alternative Flows: | 1. AF6: User can input custom height and weight value    1. User inputs a different height or weight value    2. System displays new BMI index value in ‘BMI Calculator’ component |
| Exceptions: | NIL |
| Includes: | Login |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User must input a Height and Weight value that is greater than 0 |
| Notes and Issues: | NIL |

| Use Case ID: | 023 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Calories History | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Lee Yen Foong Ernest |
| Date Created: | 5 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to be able to view his calories burnt history |
| Preconditions: | 1. User must have an account |
| Postconditions: | 1. User can see his calories burnt in the past |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘Daily Calories’ component 5. System retrieves User’s calories history 6. System displays calories history in ‘View Calories History’ component |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

| Use Case ID: | 024 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Daily Calories | | |
| Created By: | Tey Li Zhang Edmund | Last Updated By: | Lee Yen Foong Ernest |
| Date Created: | 5 February 2023 | Date Last Updated: | 12 April 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows the user to be able to view his daily calories burnt |
| Preconditions: | 1. User must have an account |
| Postconditions: | 1. User can see his current calories burnt for the day |
| Priority: | High |
| Frequency of Use: | 1/session |
| Flow of Events: | 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘View Daily Calories’ component 5. System retrieves User’s calories burnt today 6. System displays calories burnt in ‘View Daily Calories’ component |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Login |
| Extends | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

## 