Calorize

Use-case Specification   
Document

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 05/11/2017 | 1.0 | Initial use-case | Nguyễn Quốc Bảo  Lâm Hoàng Dũng  Huỳnh Xuân Khánh  Nguyễn Trần Quốc Khánh  Nguyễn Lê Tâm |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 4

2.1 Use-case: Add food intake 4

2.2 Use-case: Edit food intake 5

2.3 Use-case: Share results to Facebook 5

2.4 Use-case: Take a photo 6

2.5 Use-case: List out food types 6

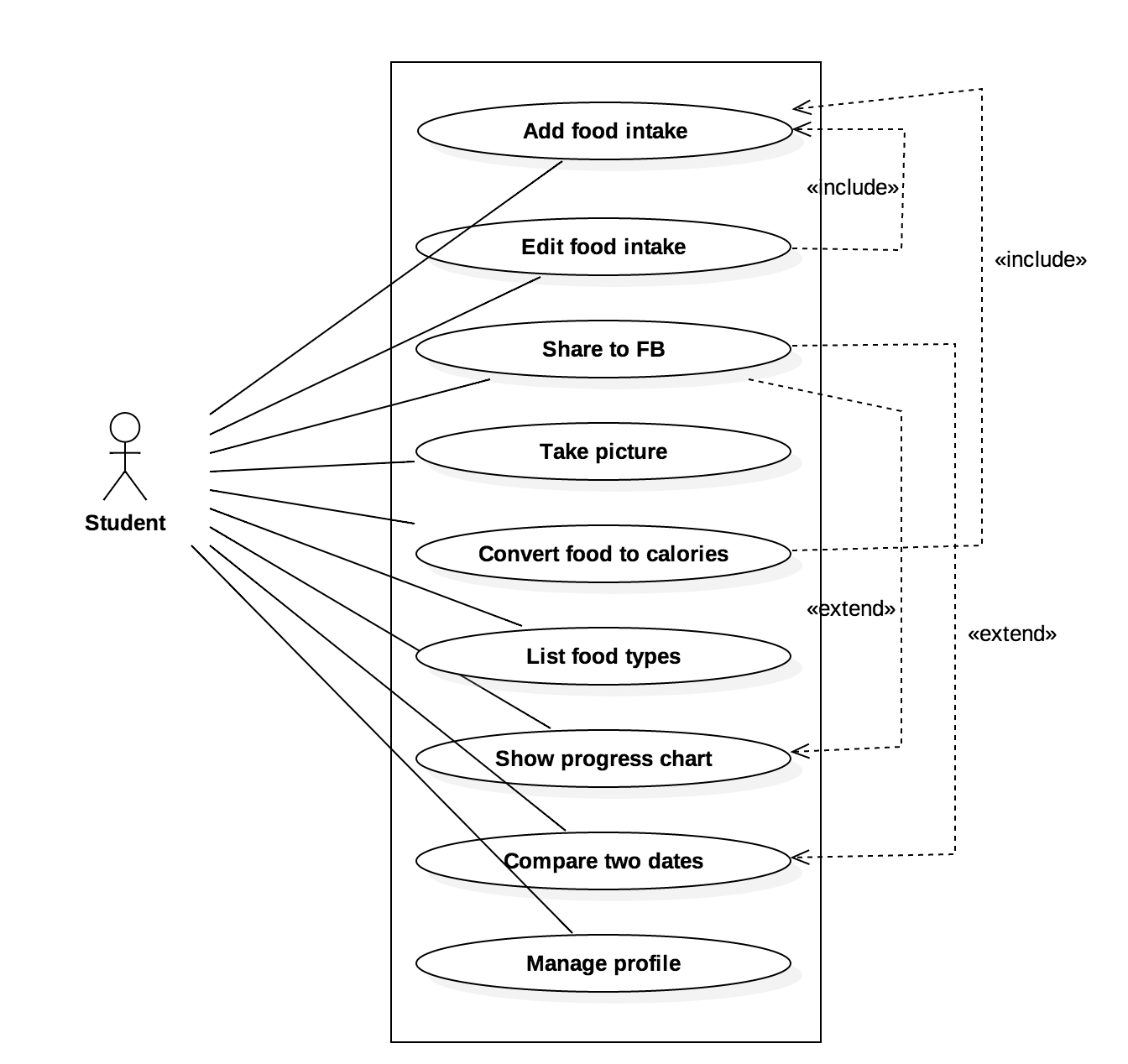
2.6 Use-case: Convert food to calories 6

2.7 Use-case: Show progress chart: 7

2.8 Use-case: Compare results of 2 dates 7

2.9 Use-case: Update/Modify Profile 7

# Use-case Model



# Use-case Specifications

## Use-case: Add food intake

|  |  |
| --- | --- |
| Use case Name | Add food intake |
| Brief description | This use case describes how the user can add food they have eaten. |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “Add food intake”. 2. System displays a list of food types and quantity box. 3. User choose type of food and input the quantity. 4. User clicks “add” button. 5. System calculate the number of calories and add it to the current amount. 6. System display the changes that the user just made. |
| Alternative Flows | **Alternative flow 1: User don’t want to continue adding food**   1. In #2 or #3, user clicks cancel.   Go back and continue step #1 in the basic flow. |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | The current number of calories is increased. |

## Use-case: Edit food intake

|  |  |
| --- | --- |
| Use case Name | Edit food intake |
| Brief description | This use case describes how the user can modify and delete food |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “Edit food intake” 2. System displays a list of food that the user has eaten and their quantities. 3. User modify the type of food of an item or its quantity, or choose delete item. |
| Alternative Flows | **Alternative flow 1: User don’t want to continue editing food intake.**   1. In #2, user clicks cancel. 2. Go back and continue step #1 in the basic flow. |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | A type of food or its quantity is modified, or a food and its quantity are removed. |

## 2.3 Use-case: Share results to Facebook

|  |  |
| --- | --- |
| Use case Name | Share results to Facebook |
| Brief description | This use case describes how the user can share the result to Facebook |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “Share to Facebook” 2. User choose one of two options that the system lists out: share in days and share in months. 3. User choose one or two days (or months, depends on which option that the user chose in step #2). If two days/months are chosen, the system will perform a comparison. 4. User clicks on button “share”. 5. The system shares the results to Facebook. |
|  | **Alternative flow 1: User don’t want to continue sharing results to Facebook.**   1. In #2 or #3, user clicks cancel. 2. Go back and continue step #1 in the basic flow. |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | Results are shared to Facebook. |

## 2.4 Use-case: Take a photo

|  |  |
| --- | --- |
| Use case Name | Take a photo |
| Brief description | This use case describes how the user can capture or remove photos about their muscle in their progress gallery. |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “Camera”. 2. User can take a photo more than one. 3. In the camera interface, user click on button “Gallery” to see their photos. 4. User can delete any photos. 5. Go back and exit Camera. |
| Alternative Flows | **Alternative flow 1: There is an option for user to take a photo.**   1. After taking a photo, user goes back to the main screen without reviewing their images. 2. There is probably a notification about reminding user to capture a photo each day. |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | Their mobile phone must have a camera. |

## 2.5 Use-case: List out food types

|  |  |
| --- | --- |
| Use case Name | List out food types |
| Brief description | This use case describes how the user can see the list of food type stored in the app. |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “List food types” 2. A new activity will be loaded, all kind of food (with following calories) will be represented on this screen. 3. User can scroll up and down to view all the information of the list food. 4. User clicks back button to go back the main screen. |
| Alternative Flows | **Alternative flow 1: There is an option for user to view information of food  (this task looks like a dictionary).**   1. User can skip this task. |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | Make sure the list of food type can be shown. |

## 2.6 Use-case: Convert food to calories

|  |  |
| --- | --- |
| Use case Name | Convert food to calories |
| Brief description | This use case help students evaluate the calories of food intake |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on button “Covert food” 2. Application will show a progress circle illustrate the current calories and the number of total calories daily inside that circle. 3. User click the ‘x’ button on the right corner to go back to main screen |
| Alternative Flows | **Alternative flow 1: There is an option for user to view food intake**   1. User click button “Show food” 2. A pop up list comes out to represent each food – calories which user consumes |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | Make sure application has information of user (weight, height) to calculate the calories in need. |

## Use-case: Show progress chart:

|  |  |
| --- | --- |
| Use case Name | Show progress chart |
| Brief description | This use case help students keep track of their calories intake and eating habit |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on tab “Progress” 2. Application will show a calendar 3. User chooses the starting and ending date, then click “Show chart” |
| Alternative Flows | **Alternative flow 1: The starting date is later than the ending date**   1. Application shows an Error message 2. Go back to step #2   **Alternative flow 2: User clicks cancels**   1. Go back to main screen |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions | Starting date has to be earlier than ending date. |

## Use-case: Compare results of 2 dates

|  |  |
| --- | --- |
| Use case Name | Compare results of 2 dates |
| Brief description | This use case help students compare the results of 2 selected dates |
| Actors | Student |
| Basic Flow | 1. In the main screen, user clicks on tab “Progress” 2. Application will show a calendar 3. User chooses the starting and ending date, then click “Compare” |
| Alternative Flows | **Alternative flow 1: User clicks cancels**   1. Go back to main screen |
| Pre-conditions | User logs in and goes to the main screen of the app. |
| Post-conditions |  |

## Use-case: Update/Modify Profile

|  |  |
| --- | --- |
| Use case Name | Update/Modify Profile |
| Brief description | This use case allows student to change their profile information including height and weight |
| Actors | Student |
| Basic Flow | 1. In the main screen, click the symbol on the toolbar to show up Drawer. This can also be done by swiping from left to right 2. User clicks Profile Icon 3. User Profile activity is displayed 4. User edits weights, heights. 5. User clicks “Ok” 6. “Do you want to save changes” Dialog shows up 7. User clicks “Yes” to save changes and go back to main screen |
| Alternative Flows | **Alternative flow 1: User presses back button**   1. At step #4 2. User presses back button 3. “Warning that changes will be discarded, and ask user if he/she really wants to finish” Dialog shows up 4. User clicks “Yes” to discard all changes and go back to main screen 5. User clicks “No” to stay in the activity and continue to edit   **Alternative flow 2: User clicks “No” in Dialog**   1. At step #6 2. User clicks “No” 3. All changes will be discarded and user will get back to the main screen   **Alternative flow 3: User presses back button**   1. At step #6 2. User presses back button 3. User will be able to continue editing |
| Pre-conditions | User must sign up and sign in  User logs in and goes to the main screen |