SCSJ2203: Software Engineering

**Software Requirements Specification**

Let’s Get Hot! (LeGeT)

Version 1.0

01/05/2016

Department of Software Engineering

Faculty of Computing

Prepared by team LeGeT

**Revision Page**

**a.** **Overview**

This is an initial draft version 1.0 documenting intrinsic subsystems of Let’s Get Hot! (LeGeT) apps. Subsystems included are User Profile, BMI Calculator, Daily Water Intake, Workout Tips, Pharmacy Locator and Healthy Food Suggestion.

**b.** **Target Audience**

1. Client
2. System Developer
3. Programmer

**c.** **Project Team Members**

1. Mohd Hakimi Iqmall Bin Mohd Zolkifly - Workout Tips

### Amy Natasha Binti Abdul Rashid - Daily Water Intake

### Afif Wahdi Bin Mustafa - Pharmacy Locator

### Amirah Binti Adnan - BMI Calculator & User Profile

1. Yap Yoong Seng - Healthy food Suggestion

**d.** **Version Control History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | LeGeT group members | Initial draft | 29/4/2016 |

**Note:**

This Software Requirements Specification (SRS) template is based on IEEE Std 830-1998, organized by modules according to system features (Appendix A.5 of the IEEE Std, 830-1998, Section 5) and customized to meet the need of SCSJ2203 course at Faculty of Computing, UTM. Compiled and checked by Shahida Sulaiman, PhD on 20 March 2016. Examples of models are from Satzinger (2011).

**Table of Contents**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Introduction** | | | | | 1 |
|  | 1.1 | Purpose | | | | 1 |
|  | 1.2 | Scope | | | | 1 |
|  | 1.3 | Definitions, Acronyms and Abbreviations | | | | 1 |
|  | 1.4 | References | | | | 2 |
|  | 1.5 | Overview | | | | 2 |
| **2** | **Overall Description** | | | | | 3 |
|  | 2.1 | Product Perspective | | | | 3 |
|  |  | 2.1.1 | System Interfaces | | | 3 |
|  |  | 2.1.2 | User Interfaces | | | 4 |
|  |  | 2.1.3 | Hardware Interfaces | | | 5 |
|  |  | 2.1.4 | Software Interfaces | | | 5 |
|  |  | 2.1.5 | Communication Interfaces | | | 6 |
|  |  | 2.1.6 | Memory | | | 6 |
|  |  | 2.1.7 | Operations | | | 6 |
|  |  | 2.1.8 | Site Adaptations Requirements | | | 7 |
|  | 2.2 | Product Functions | | | | 8 |
|  | 2.3 | User Characteristic | | | | 8 |
|  | 2.4 | Constraints | | | | 9 |
|  | 2.5 | Assumption and Dependencies | | | | 9 |
|  | 2.6 | Apportioning of Requirements | | | | 9 |
| **3** | **Specific Requirements** | | | | | 10 |
|  | 3.1 | External Interface Requirements | | | | 11 |
|  |  | 3.1.1 | | User Interfaces | | 11 |
|  |  | 3.1.2 | | Hardware Interfaces | | 12 |
|  |  | 3.1.3 | | Software Interfaces | | 13 |
|  |  | 3.1.4 | | Communication Interfaces | | 13 |
|  | 3.2 | System Features | | | | 13 |
|  |  | 3.2.1 | | Module of User Profile Subsystem | | 13 |
|  |  |  | | 3.2.1.1 | UC001: Use Case for User Profile | 14 |
|  |  |  | | 3.2.1.2 | UC002: Use Case for Update Profile | 17 |
|  |  |  | | 3.2.1.3 | UC003: Use Case for View Current Profile | 17 |
|  |  | 3.2.2 | | Module of BMI Calculator Subsystem | | 17 |
|  |  |  | | 3.2.2.1 | UC004: Use Case for Enter Weight and Height | 20 |
|  |  |  | | 3.2.2.2 | UC005: Use Case for View BMI Value | 20 |
|  |  | 3.2.3  3.2.4  3.2.5  3.2.6 | | Module of Pharmacy Locator Subsystem  3.2.3.1 UC006: Use  Case for Searching Pharmacies  3.2.3.2 UC007:Use Case for Locating the  Selected Pharmacy  Module of Water Intake Subsystem  3.2.4.1 UC008: Use Case for Water Volume  Summation    3.2.4.1 UC009: Use Case for View Total  Water Volume Taken  Module of Food Suggestion System  3.2.5.1 UC010: Use Case for View Food  Suggestion  Module of Workout System  3.2.6.1 UC011: Use Case for Workout  Instruction  3.2.6.2 UC012: Use Case for Workout  Timer | | 18  20  20  20  22  22  23  24  24  26  26 |
|  | 3.3 | Performance Requirements  3.3.1 how to edit? | | | | 27 |
|  | 3.4 | Design Constraints | | | | 27 |
|  | 3.5 | Software System Attributes | | | | 28 |
|  | 3.6 | Other Requirements | | | | 28 |
|  |  |  | |  | |  |
|  |  |  |  |  |  |  |

**1.** **Introduction**

**1.1** **Purpose**

The purpose of this document is to define the requirements for creating an Android apps for the Let’s Get Hot! (LeGeT) apps. This document will outline all of the necessary information to start development.

**1.2** **Scope**

The scope for this document is to create an Android apps that called Let’s Get Hot (LeGeT). This apps can tell user about their Body Mass Index (BMI) by calculate the height and weight, and will show the estimated water needed in one day that should be drink base on the BMI. User can set alarm to remind them to drink the water. LeGeT apps also provide the guide of workout routine to help users keep their health and body fit.Future more, this apps will show the suggestion of healthy foods to eat and provide the pharmacy locator.

**1.3** **Definitions, Acronyms and Abbreviation**

* **Username** - An identification used by a person with access to a computer,network, or online service.
* **BMI** - Body Mass Index
* **LeGeT** - Let’s Get Hot!
* **Database** - Computer software apps that interact with user, other apps, and it capture and analyse the data.
* **HTTPS** - A protocol for secure communication over a computer network which is widely used on the internet.
* **GPS** - Global Positioning System

**1.4** **References**

* *Workflow materials: The Umlet was used in the creation of creating workflow of diagrams of this document. The umlet open sourse software may be accessed at creately.com*
* BMI Formula: The formula of calculating BMI is taken from the website <http://www.epic4health.com/bmiformula.html>
* Body Categorization: The body categorization is taken for the website http://www.cdc.gov/healthyweight/assessing/bmi/adult\_bmi/

**1.5** **Overview**

The rest of the SRS examines the specifications of the LeGeT in detail. Section 2 of the SRS presents the overall description of the product. The design of system, user, hardware, software and communication interfaces of the product, and also the constraint of the product. Section 3 outline detailed about the specific requirements, the external interface of the product. The state diagram of the LeGeT application is explained in this section. Next, the system feature is explain by using the use-case and the diagram like activity and sequence diagram. Furthermore, there will be a follow up for the design constraint, software system attribute and other requirements of the product.

**2.** **Overall Description**

**2.1** **Product Perspective**

The purpose of this product is for general user to get clear guides for their healthy lifestyle. User can easily get the apps without a single cent out. This apps is divided into a few subsystems such as User Profile, BMI Calculator, Daily Water Intake, Healthy Food Suggestion, Workout Tips, and Pharmacy Locator.

**2.1.1** **System Interfaces**

1. BMI Calculator
   1. Displaying Body Mass Index(BMI).
      1. User needs to key-in their informations such as weight and height for BMI calculation and the system will show the results of level of BMI.
   2. Displaying daily water intake needed.
      1. System will calculate and display suggested daily water intake needed for user.
2. Daily Water Intake
   1. Calculating rate of water intake for user
      1. System will calculate base on user’s weight to detect the rate of water intake everyday for user.
      2. User can update amount of water consumed everyday to achieve daily goal.
3. Healthy Food Suggestion
   1. Displaying healthy food suggestions with different categories of meal
      1. User will be provided with information about calories of any particular meal that have been set in the system.
4. Workout Tips
   1. Time taken for each of exercise
      1. System will provide a timer in seconds for each exercise.
   2. Displaying image as guideline
      1. System provide images as guideline for user to follow workout steps.
5. Pharmacy Locator
   1. Displaying location of the pharmacies
      1. The system will show the location of pharmacies nearby using kilometre radius.

F. User Profile

1. First interface of the apps

i. When the user first tap on the apps, the first thing will be displayed

would be the user profile.

ii. Record user’s username, height and weight for initial calculation.

**2.1.2** **User Interfaces**

1. Button
   1. The uses of button in our interfaces is for make user feel easy to use by putting function for every button provided for example “Submit” button.
2. Functional Image
   1. The uses of functional image is for make the interface more interesting and require little time for user to learn the system.
3. Input Text
   1. The uses of this input text is for user to enter any required information or keyword especially for inserting weight and height information for our system.
4. Timer
   1. The uses of timer in our system is for Workout Tips use case to count the time taken for the exercises.
5. Message Box
   1. The uses of message box is for giving important information for the user.
6. Progress Bar
   1. The uses of progress bar is for giving information about the system progress to the user.

**2.1.3** **Hardware Interfaces**

1. Android Mobile Phone
2. Android Tablet

**2.1.4** **Software Interfaces**

1. Just In Mind
   1. Name : JustInMind Prototyper
   2. Version Number : 7.1.1
   3. Source : [www.justinmind.com](http://www.justinmind.com)
   4. Function for project : We use this software for making mock-up interface
2. Android Studio
   1. Name : Android Studio
   2. Version Number : 2.1
   3. Source : developer.android.com
   4. Function for project : We use this software for developing mobile apps
3. Database

**2.1.5** **Communication Interfaces**

* User will use GPS to find location

**2.1.6** **Memory**

* The apps uses main memory when running the system in order to store all the information needed.
* The apps also use secondary memory to allow user to backup their information.

**2.1.7** **Operations**

Product will be deployed to Google PlayStore as an android apps suitable for any android devices. Basically Let’s Get Hot! (LeGeT) apps have sub-apps consisting User Profiling, Daily Water Intake, BMI Calculator, Food Suggestions, Pharmacy Locator and Workout Steps. User is free to choose any of the sub-apps to proceed.

First and foremost, right when the user taps on LeGeT apps, a message will be displayed seeking for user’s username, height and weight to be saved for initial calculation for recommendation of daily water intake. The user then can edit the informations in User Profiling menu.

If user chooses to go with Daily Water Intake, user can update daily consumed water based on recommended water intake, from the first interface. User will be able to add and view the remaining water to be consumed for the day. From the info, the apps will help the user to monitor their target at the home display. We also include notifications to help user to set reminder according to their preferences of time. User can choose whether to use metric or imperial U.S. as their unit of measurement.

Next we have BMI Calculator, user will have to enter their weight (kg) and height (cm). Based from the inputs, the apps will manipulate the data for calculation of BMI with suggested daily water intake. The apps will also display user’s BMI value, level, and range of normal weight perpendicular to user’s height. The calculation process are all written in coded instructions.

For Food Suggestions, the apps provides user with variety of choices from breakfast to dinner, soup, drink and even side dish. Each food will have a short description such as the name, calories content, amount of protein, carbohydrates and fats, and star rating.

For Pharmacy Locator, the apps will help user to find the nearest pharmacy in particular radius, specifically only in Taman Universiti.

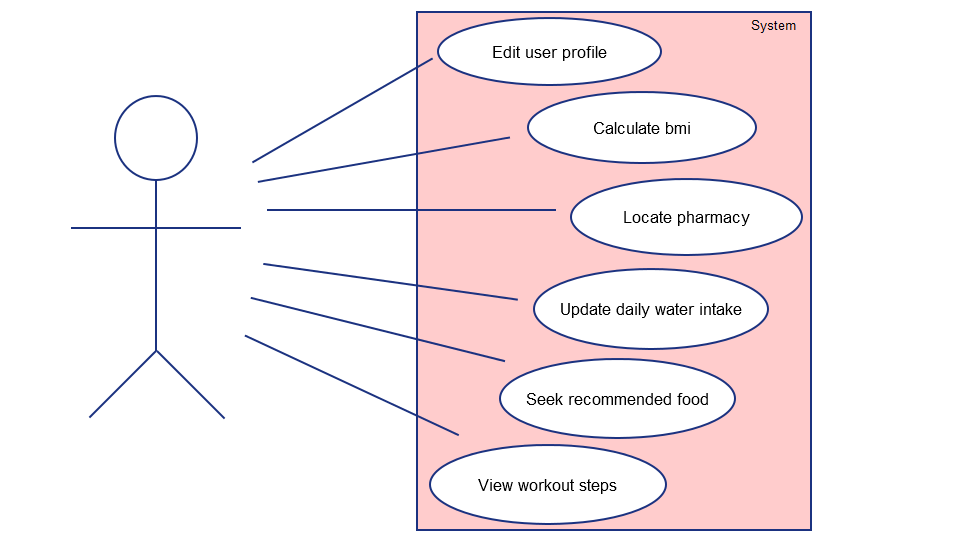
Lastly, Workout Tips. On this one, the apps will provide several steps of basic exercises along with short description and images. Each exercise will be provided with fixed timer that user have to get it done with.

**2.1.8** **Site Adaptation Requirements**

For Daily Water Intake, user can choose whether to have metric (kg/ml/l) as their unit of measurement or imperial U.S. (oz/lb). So the apps will have to adapt to the changes made as it needs to do calculation of the water intake.

For Pharmacy Locator, the apps needs to adapt to certain region. Because it will have to look for the nearest pharmacy based on user options.

**2.2** **Product Functions**



All use cases below :

**Edit user profile** : User may update their weight and height.

**Calculate BMI** : User may calculate their bmi based on height and weight entered.

**Locate pharmacy** : User will have to search for pharmacy using our Pharmacy Locator app.

**Update daily water intake** : User can update their daily water intake everyday.

**Seek recommended food** : User may look for many types of food based on the

suggestions from our product.

**View workout steps** : User can view many type of workout steps provided by the app.

**2.3** **User Characteristics**

a) Android phone user

b) Dieter

c) Gym trainer

d) Pharmacy customer

e) Patient

**2.4** **Constraints**

a) Higher-order language requirements

b) Hardware restrictions

c) Interfaces to other apps

d) Control functions

e) Reliability requirements

f) Criticality of the apps

g) Cost (time)

h) Lack of technical skills

**2.5** **Assumption and Dependencies**

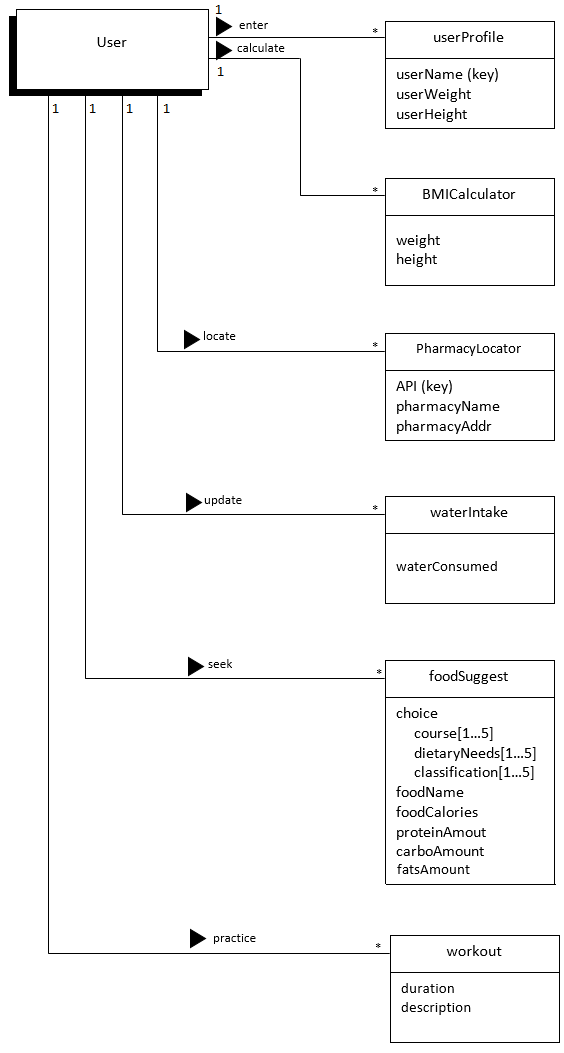
Some changes may be made different from the SRS. User interface and some functionalities can change during the development process of the project. There might be some addition to the product. And also new functionalities can be added which is able to change during the dependent system requirements.

**2.6** **Apportioning of Requirements**

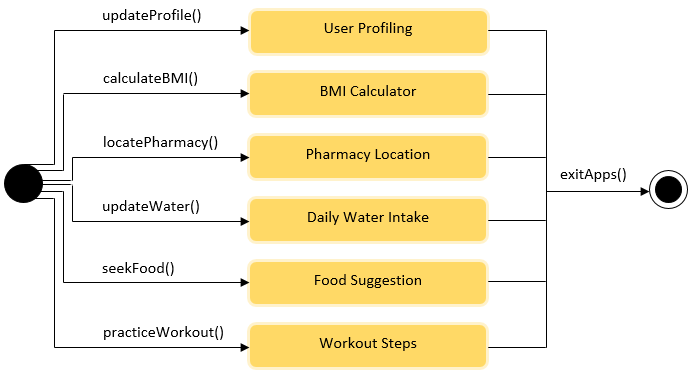
The requirements described in sections 1 and 2 of this document are referred to as preliminary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both preliminary and functional specifications, the apps will be built from functional specification since it is more detailed.

‘Essential requirements’ (referred to in section 3) are to be implemented for this version of LeGeT apps. ‘Desirable requirements’ are to be implemented in this release if possible, but are not committed to by the developers. It is anticipated that they will be part of future release. ‘Optional requirements’ will be implemented at the discretion of developers.

**3.** **Specific Requirements**



**Figure 3.1: Domain Model of LeGeT**



**Figure 3.2: State Machine Diagram of LeGeT**

**3.1** **External Interface Requirements**

**3.1.1** **User Interfaces**

The user interface for the system will be an apps on the Android smartphone. A prototype has been created that represents the final interface for the system in terms of look and feel. The user interface will be limited to the types of controls that can be generated using Java language. The user interface code will be generated by individual developers, as well as using Android Studio as a platform of development.

There will only be one interface; for the smartphone user. The user can access all features provided such as User Profile, BMI Calculator, Pharmacy Locator, Daily Water Intake, Food Suggestions, and Workout Steps.

UI-1: The user shall be able to enter username, weight and height for user profiling.

UI-2: The user shall be able to display BMI level and suggested daily water intake.

UI-3: The user shall be able to locate nearest pharmacy in Taman Universiti, along with the pharmacies’ names and addresses.

UI-4: The user shall be able to update his/her consumed water to achieve daily goals and view balance amount of water.

UI-5: The user shall be able to view suggested healthy food with a few classifications, tagged along with some images, name, calories content, protein, carbohydrate and fats amount, description, and food ratings.

UI-6: The user shall be able to play workout steps, along with images, workout description, and timer.

**3.1.2** **Hardware Interfaces**

This will be an Android phone apps, and as such will be designed to interface with the hardware present on the Android phone. In theory the apps will be able to run by other devices that can emulate the Android, but this will not be a consideration during design.

As this is a mobile device, it will be using the Android network to connect to the internet, which will allow it to communicate with the database servers. This means that it will be using the infrastructure, be it wireless communication points or physical lines, of the network in order to perform properly.

**3.1.3** **Software Interfaces**

This product will be connecting to a database that is already set. This allows for use in exercises by owner (user) of the phone. The operating system the software runs on will be the operating system the Android phone runs on, which comes with a software framework that will be utilized, including many prepackaged components to do things like create menus, hookup buttons, and other common functions expected of a mobile device. The only communication will be between the phone, server housing the database, and Global Positioning System (GPS) which will be sending queries or updates and receiving the information back. The logic associated with the apps will be duplicated on the phone, so there will be little in the way of a server side component performing logic.

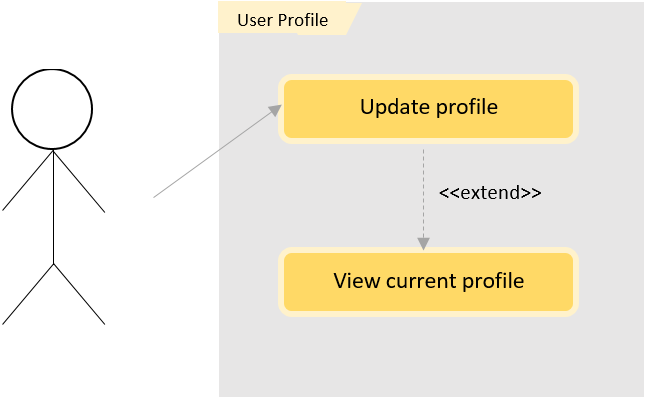
**3.1.4** **Communication Interfaces**

This will be an Android apps and will not be linked to web pages that are not necessary to duplicate. As described above, this will be communicating with a database server, and so will be making use of the Android network and HTTPS in order to communicate. The primary forms of communication will be database transactions or requests. The apps will need to be synchronized so that the information displayed to the user is always up to date.

**3.2** **System Features**

**3.2.1** **Module User Profile Subsystem**

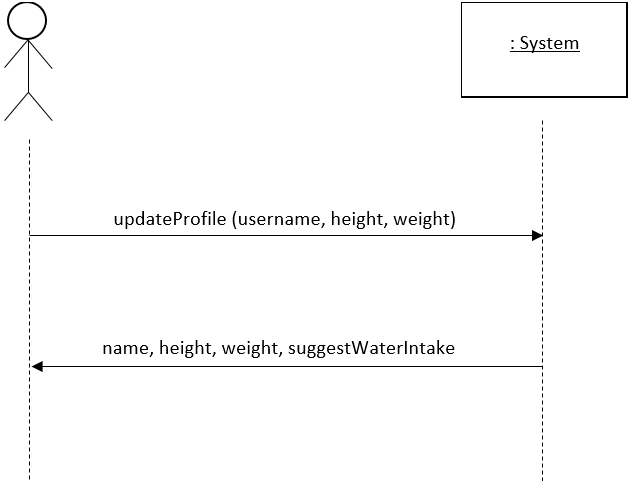
The functional requirements that are available in this module is only to update user profile such as username, height and weight for auto-calculation of daily water intake. Below is the diagram of User Profile Subsystem from the overall use case diagram in Figure 2.1.



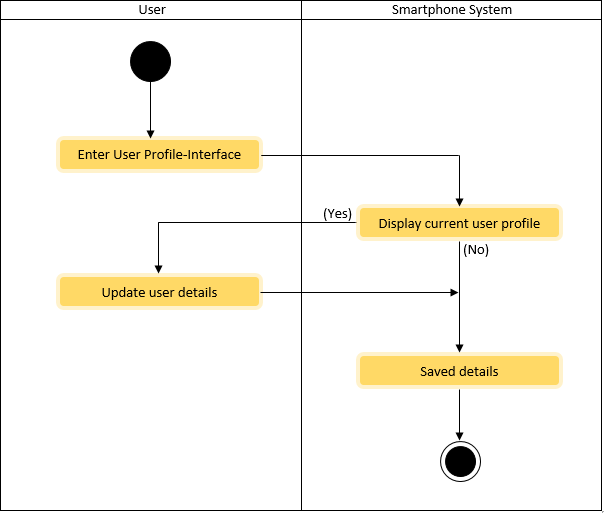
**Figure 3.2.1.1.1: User Profile Subsystem**

**3.2.1.1 UC001: Use Case Update Profile**

The user profile should be empty at the beginning and after some updates, must be able to display current username, weight and height of the user, including recommended amount of daily water intake. The recommended amount daily water intake should automatically be computed from users’ inputs; weight and height.



**Figure 3.2.1.1.2: System Sequence Diagram of Update Profile**



**Figure 3.2.1.1.3: Activity Diagram of Update Profile**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Update Profile | |
| **Scenario:** | User update profile details | |
| **Triggering Event:** | User wants to change his/her information | |
| **Brief Description:** | User wants to update current height and weight for a new calculation of daily water intake. | |
| **Actors:** | Android phone user | |
| **Related Use Case:** | Extend: *View Current Profile* | |
| **Stakeholder:** | User: to provide profile details such as name, weight and height to be updated. | |
| **Precondition:** | Details of profiling must exist before updates can be done. | |
| **Postcondition:** | Profiling details has been updated by the user. | |
| **Flow of Events:** | Actors | User Profile System |
| 1. The user clicks “User Profile”.    2. If the user taps on username space,  2.1 The user enters new  username    3. If the user taps on weight space,  3.1. The user enters new weight    4. If the user taps height,  4.1. The user enters new height  5. If the user taps “Save”,  5.1. The user taps “OK” | 5.0.1. Display a message if the  data have been updated.  5.1.1 Back to the main menu |
| **Exception Conditions:** | None | |

**Figure 3.2.1.1.3:Use Case Description for Update Profile Use Case**

**3.2.1.2 UC002: Use Case Update Profile**

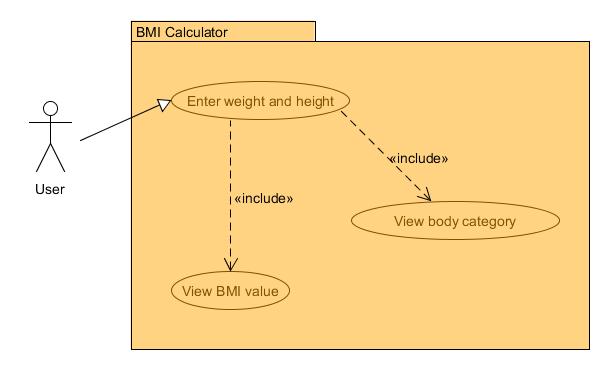
The user can do some changes upon user profiling such as username, weight and height. If the details have been saved before, the interface will display the last updated user details. After filling new details, the data will be saved in database.

**3.2.1.3 UC003: Use Case View Current Profile**

User current profile will be viewed first when user enters User Profile interface. Display details are as such; username, weight and height. The details can be edited later by updating profile manually.

**3.2.2** **Module of BMI Calculator System**

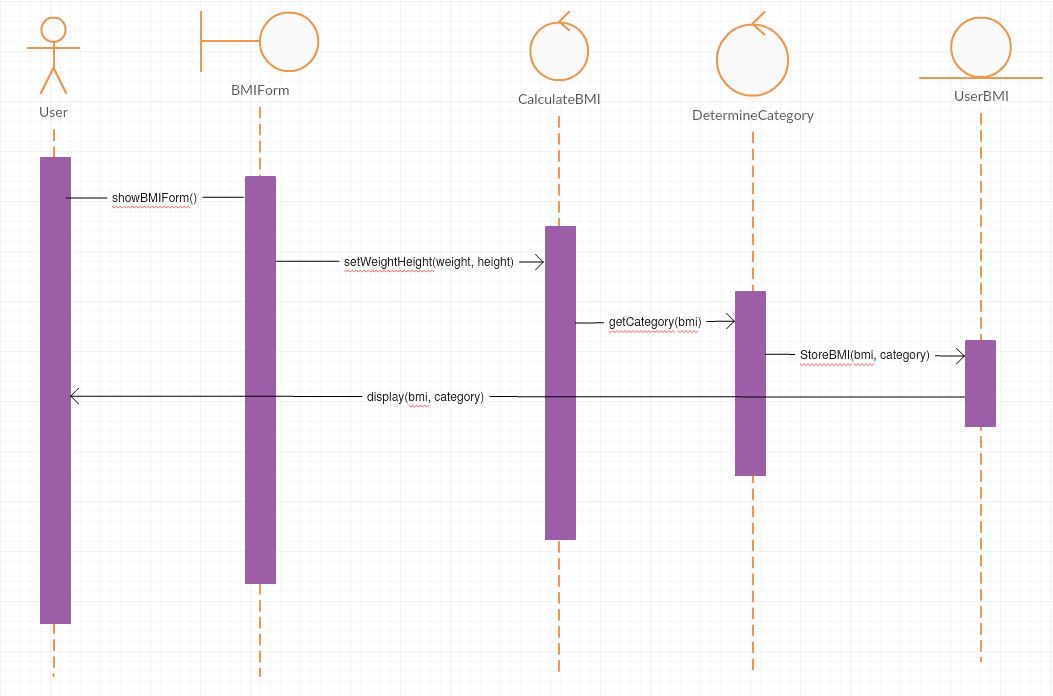
The functional requirement of BMI Calculator is obtain the BMI value and body category of the user. The user is required to key in own weight and height. The system then determines the body category of the user based on BMI value. The below is the general use case diagram, use case specification, sequence diagram and activity diagram of BMI Calculator system.



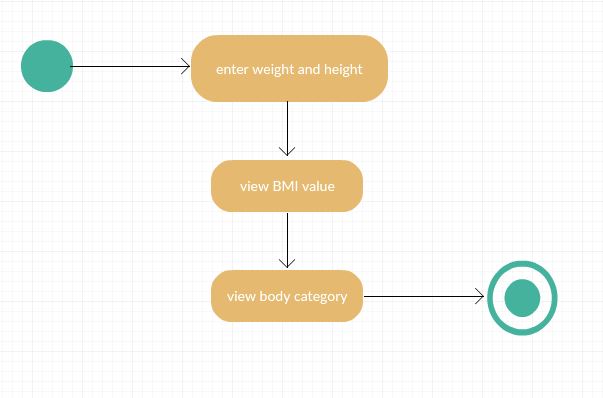
**Figure 3.2.2.1.1: BMI Calculator Subsystem**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | BMI Calculator | |
| **Scenario:** | Enter weight and height  View BMI value  View body category | |
| **Triggering Event:** | User wants to view the physical body condition. | |
| **Brief Description:** | When user enters own weight and height, the data will be computed and BMI value is produced. The produced BMI value determines the body category. The BMI value and body category are displayed to user. | |
| **Actors:** | User | |
| **Related Use Case:** | none | |
| **Stakeholder:** | User | |
| **Precondition:** | User must enter the own weight and height to get BMI value.  User must enter the own weight and height to get body category. | |
| **Postcondition:** | The BMI value and body category are produced. | |
| **Flow of Events:** | **User** | **BMI Calculator System** |
| 1. The user enters own weight and height.  2. The user views the BMI value.  3. The user views the body category. | 1.1. The system collects the data of weight and height..  2.1. The system displays the BMI value.  2.2. The system determines the body category based on the BMI value.  3.1. The system displays the BMI value. |
| **Exception Conditions:** | 1. No input of weight and height   The system would not display the BMI value and body category | |

**Use Case Description of BMI Calculator System**



**Figure 3.2.2.1.2: Sequence Diagram of BMI Calculator**



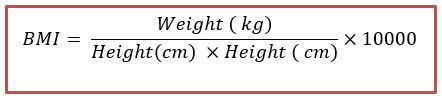
**Figure 3.2.2.1.3: Activity Diagram of BMI Calculator**

**3.2.2.1 UC004: Enter Weight and Height Use Case**

The system will first requests the user to input own weight and height. The values entered affect the BMI value and body category.

**3.2.2.2 UC005: View BMI Value Use Case**

The values of weight and height of user are null initially. Once the user entered the values of weight and height, the system will calculate the BMI value according to BMI formula as shown at below.



After the calculation, the system will display the results on the same interface.

(note: BMI formula reference is inserted in 1.4 References)

**3.2.2.3 UC006: View Body Category Use Case**

Consequently the BMI value is shown, the system will also determine the category of user’s physical condition. Below is the table of body categorization that based on BMI value according to standardised global weight status categories.

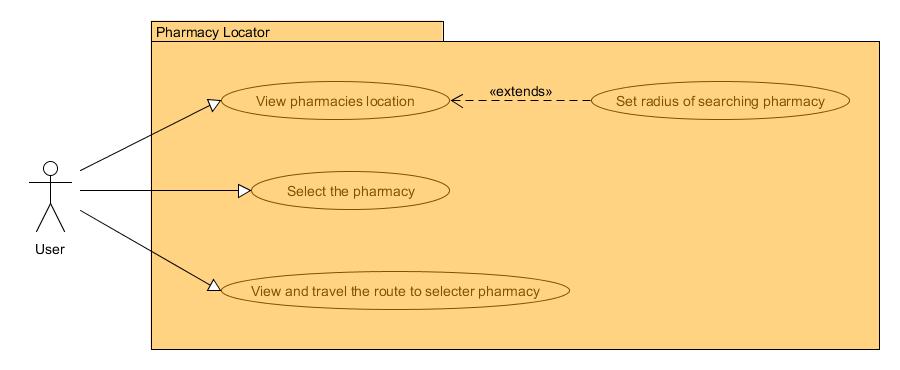
|  |  |
| --- | --- |
| **BMI value** | **Category** |
| <18.5 | underweight |
| 18.5 - 24.9 | normal |
| 25.0 - 29.9 | overweight |
| > 30.0 | obese |

Therefore, the body status will be shown on the same interface.

(note: body categorization reference is inserted in 1.4 References)

**3.2.3** **Module of Pharmacy Locator Subsystem**

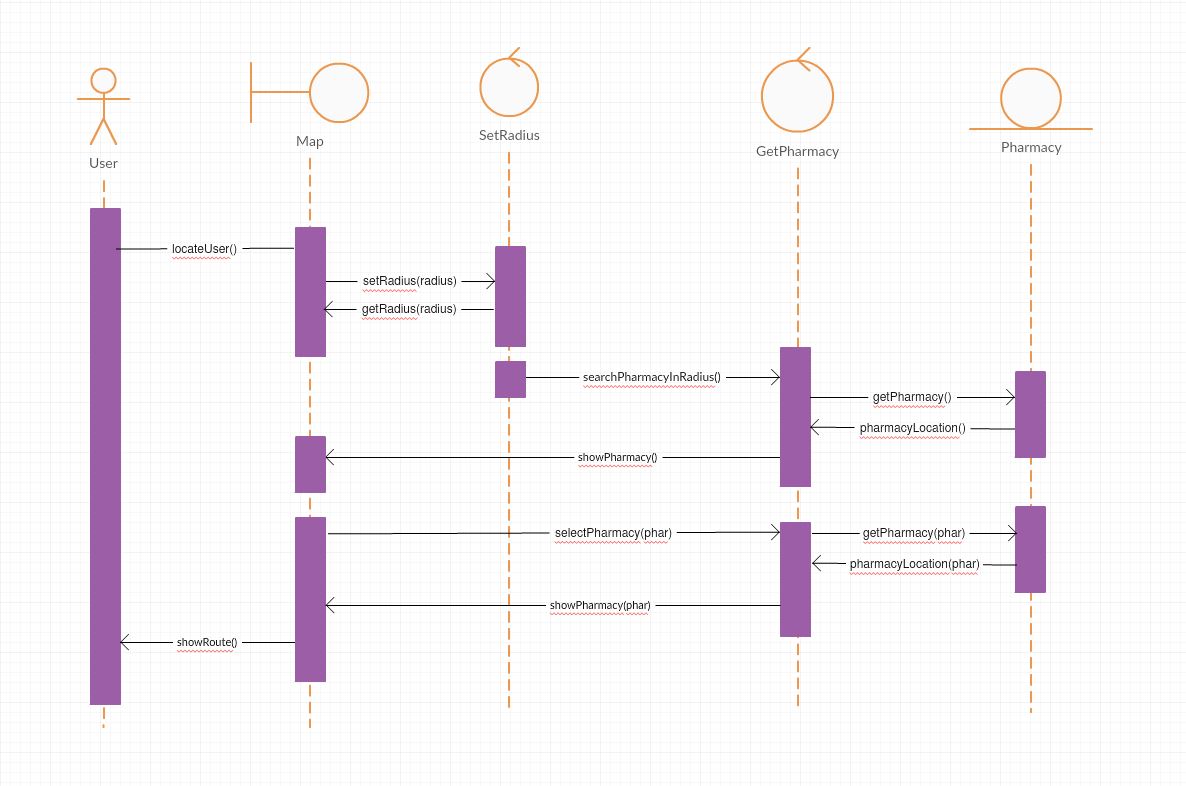
In this pharmacy locator system, it will show the current exact location of the user. Hence the system will shows the nearest pharmacies around the location of user, the user can set the radius of searching pharmacies if needed. The user can individually chooses the pharmacy that user wishes to go. After the selection, the route that leading to the chosen pharmacy will be shown in apps. The below is the general use case diagram, use case specification, sequence diagram and activity diagram of BMI Calculator system.



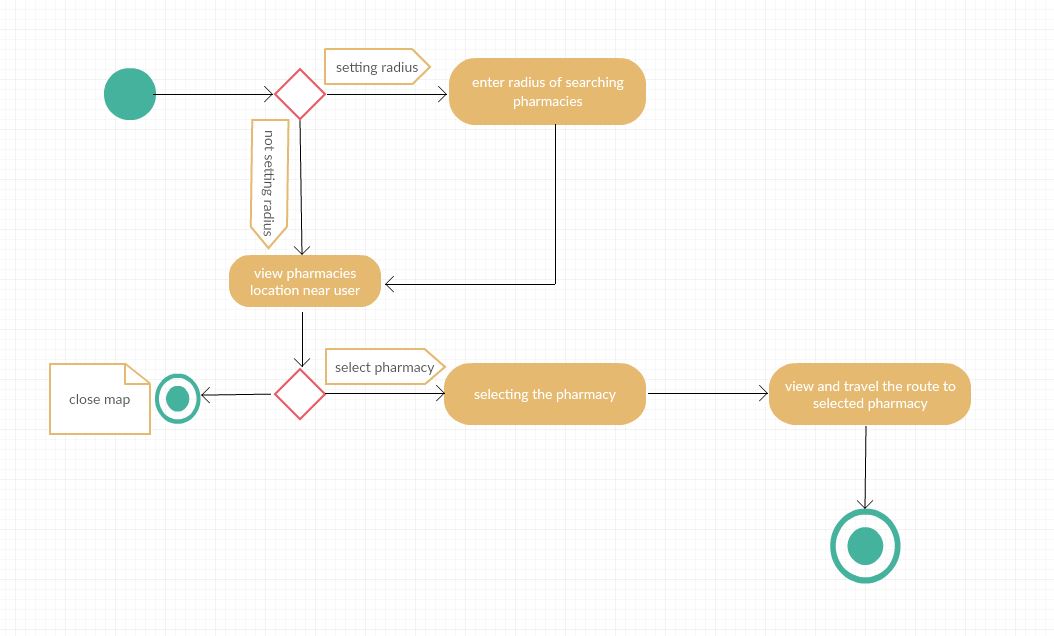
**Figure 3.2.3.1.1: Pharmacy Locator Subsystem**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Pharmacy Locator | |
| **Scenario:** | View pharmacies location  Set radius of searching pharmacy  Select the pharmacy  View and travel the route to selected pharmacy | |
| **Triggering Event:** | User wants to locate the nearest pharmacy or desired pharmacy. | |
| **Brief Description:** | When user sets the radius of searching the pharmacy, the pharmacies that within the circle will be displayed. The user is then select one of the pharmacies and starts to navigate.. | |
| **Actors:** | User | |
| **Related Use Case:** | Extend: set radius of searching pharmacy | |
| **Stakeholder:** | User | |
| **Precondition:** | User should enters the radius of searching pharmacies.  User must selects one of the pharmacies to start navigation. | |
| **Postcondition:** | The route that leading to the selected pharmacy will be shown. | |
| **Flow of Events:** | **User** | **Pharmacy Locator System** |
| 1. The user views the location of pharmacies that near to user.  2. The user sets the radius of searching pharmacies.  3. The user views the location of pharmacies that within the searching area.  4. The user selects one of the pharmacies.  5. The user starts navigation. | 1.1. The system collects the data of pharmacies location.  2.1. The system collects the radius of searching pharmacies.  2.2. The system returns the pharmacies location that within the searching area.  4.1. The system collects the data of selected pharmacy location; name and address. |
| **Exception Conditions:** | Sets radius not more than 5 kilometres | |

**Use Case Description of Pharmacy Locator System**



**Figure 3.2.3.1.2: Sequence Diagram of Pharmacy Locator**



**Figure 3.2.3.1.3: Activity Diagram of Pharmacy Locator**

**3.2.3.1 UC007: View Pharmacies Location Use Case**

At the beginning, the user has to click on “Locate me” to enable the GPS. The map and the current location of user will be shown. Besides that, the pharmacies that near to location of user will be displayed.

**3.2.3.2 UC008: Set Radius of Searching Pharmacies Use Case**

The user also could enters the radius of searching the pharmacies in kilometers. As a results, the pharmacies that within the radius will be displayed in the map.

**3.2.3.3 UC009: Select the Pharmacy Use Case**

The user obtains several pharmacies location in the map, the user can choose one of the pharmacies based on own distance or desirable.

**3.2.3.4 UC010: View and Travel to the Selected Pharmacy Use Case**

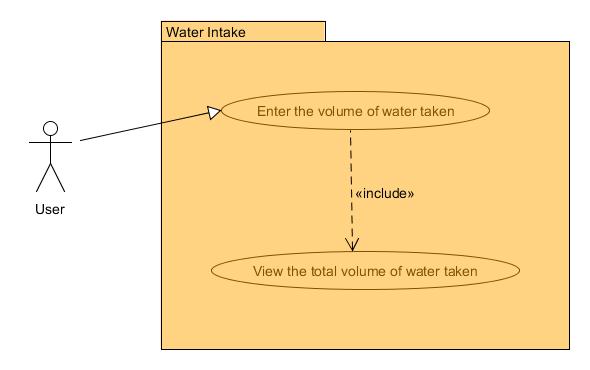
After searching the nearest pharmacies, the user can locates the desired pharmacy by selecting the particular pharmacy. The system will calculate the distance and display the route to the selected pharmacy so it helps the user in navigating.

**3.2.4** **Module of Water Intake Subsystem**

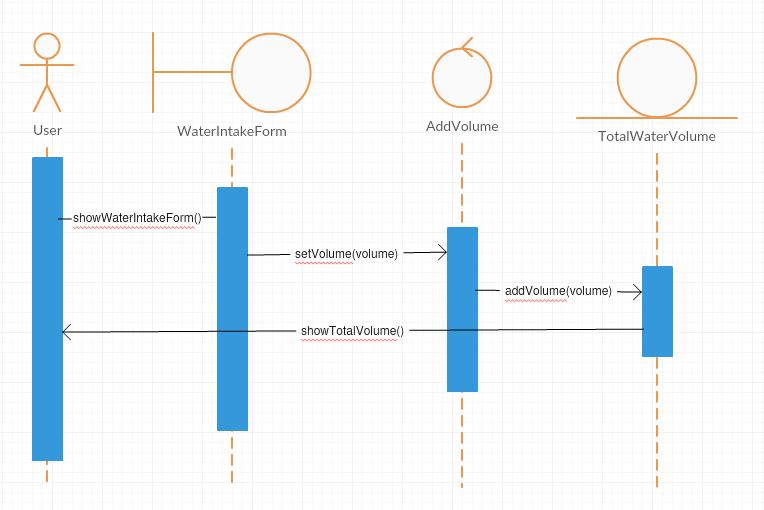
The Water Intake System is the subsystem that assists the user in summing the volume of water taken in a day. The user needs to enter the volume of water taken once water is consumed. Consequently, the system will store and sum up the input value. It will reveal the total volume of water take by user from the starting of the day until the latest water intake. The below is the general use case diagram, sequence diagram and activity diagram of Water Intake system.

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Water Intake | |
| **Scenario:** | User update amount of water consumed | |
| **Triggering Event:** | User wants to update amount of water consumed each time | |
| **Brief Description:** | User wants to top-up amount of water consumed to achieve every day goal. | |
| **Actors:** | Android phone user | |
| **Related Use Case:** | Include: *View the total volume of water taken* | |
| **Stakeholder:** | User to provide amount of water consumed for updation of daily water intake to achieve daily goal. | |
| **Precondition:** | Amount of suggested daily water intake must first be existed. | |
| **Postcondition:** | User updated amount of water consumed. | |
| **Flow of Events:** | **Actors** | User Profile System |
| 1. The user clicks “Daily Water Intake”.    2. If the user taps on addition button “+”,  2.1 The user enters amount of water consumed    3. If the user taps on available amount of water intake “100”, “50” or “10”,    5. If the user taps “Menu” or back button, | 1.0 Displays today’s goal of water  intake.    2.0 Displays an input message of  total water consumed and  remaining water needed to be  consumed.    3.0 Displays updated amount of  water consumed and  remaining amount of water  needed to be consumed.  5.0 Return back to main menu |
| **Exception Conditions:** | None | |

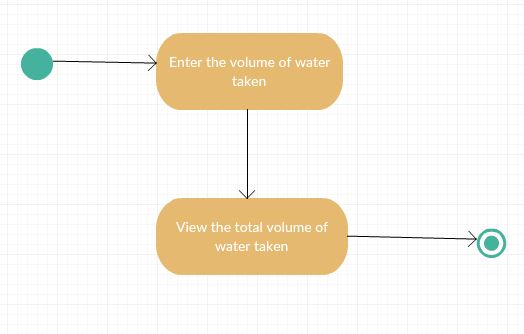
**Figure 3.2.4.1 Use Case Description of Water Intake System**



**Figure 3.2.4.1.1: Water Intake Subsystem**



**Figure 3.2.4.1.2: Sequence Diagram of Water Intake**



**Figure 3.2.4.1.3: Activity Diagram of Water Intake**

**3.2.4.1 UC011: Enter Volume of Water Taken Use Case**

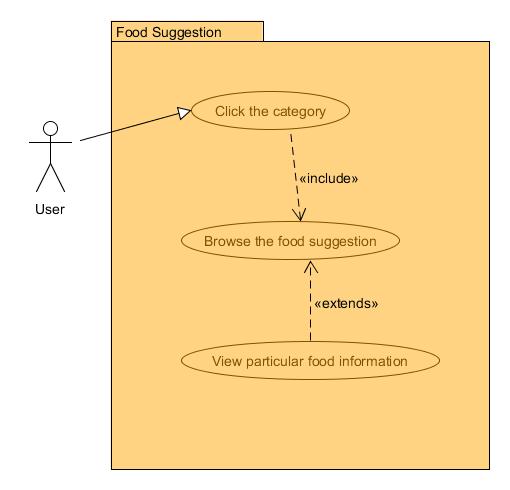
The system will sum up the volume of water taken by user once the user just consumed water and entered the value into the system. The value is then stored into the system, and the total water volume taken from starting of the day until the latest water intake is shown in the apps.

**3.2.4.2 UC012: View Total Water Volume Taken Use Case**

The total volume of water taken is displayed in the apps. The results can either encouraging the user to consume more water or indicates the user that the water volume taken is adequate.

**3.2.5** **Module of Food Suggestion Subsystem**

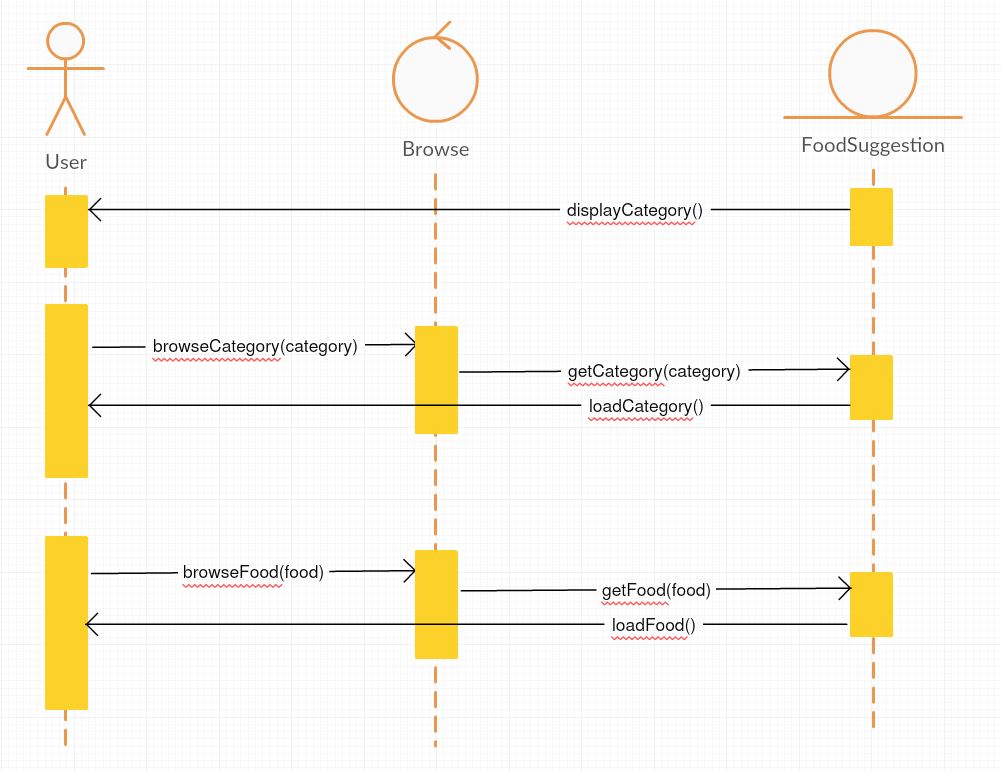
This system should suggests the user the healthy food intake regarding of course, classification and dietary needs. Each food suggestion contains the image, description, rating, calorie value and other information. The user has great diversity of food choices to get inspired by Food Suggestion System. The below is the general use case diagram, use case specification, sequence diagram and activity diagram of Food Suggestion system.



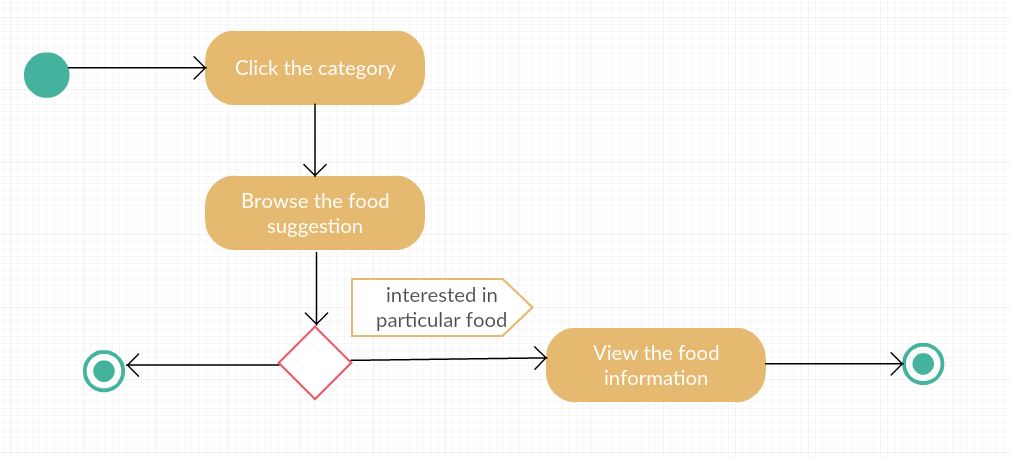
**Figure 3.2.5.1.1: Food Suggestion Subsystem**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Food Suggestion | |
| **Scenario:** | Click the category  Browse the food suggestion  View particular food information | |
| **Triggering Event:** | User has more idea of healthy food intake. | |
| **Brief Description:** | The user can browse all healthy food suggestion along with its nutrition information across all categories and types of food. | |
| **Actors:** | User | |
| **Related Use Case:** | Extend: view particular food information | |
| **Stakeholder:** | User | |
| **Precondition:** | User should clicks through categories to view food suggestion. | |
| **Postcondition:** | The food information is shown. | |
| **Flow of Events:** | **User** | **Food Suggestion System** |
| 1. The user clicks the categories to browse the food suggestion.  2. The user clicks on particular food. | 1.1. The system collects the request of user.  1.2 The system retrieves the information from database.  2.1. The system collects the request of user.  2.2. The system retrieves the information from database. |
| **Exception Conditions:** | none | |

**Use Case Description of Food Suggestion System**



**Figure 3.2.5.1.2: Sequence Diagram of Food Suggestion**



**Figure 3.2.5.1.3: Activity Diagram of Food Suggestion**

**3.2.5.1 UC013: Click the Category Use Case**

With variety of categories, the user has more choice of choosing healthy food based on course like breakfast, lunch or dinner.

**3.2.5.2 UC014: Browse Food Suggestion Use Case**

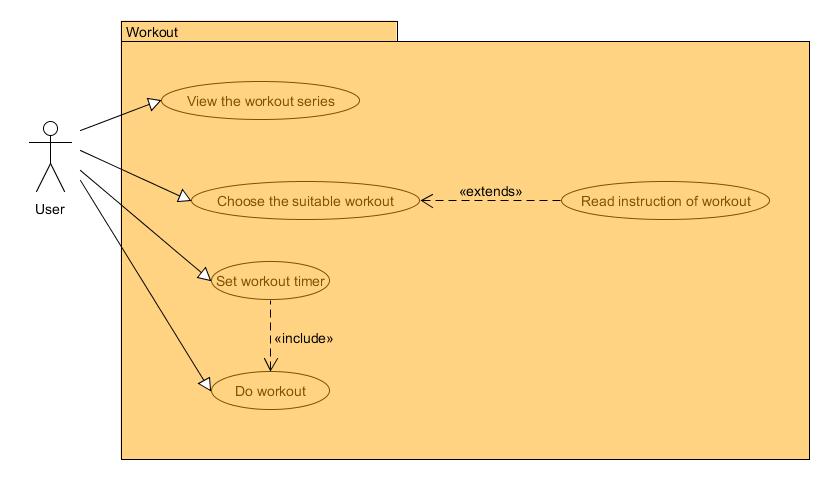
The user views the food suggestion by browsing with categories. Along with images, the user feels interest on certain food can click into it.

**3.2.5.3 UC015: View Particular Food Information Use Case**

Besides the image, the user also can views the information like star rating, description, calorie value and nutrition facts like total fat, carbohydrates and protein.

**3.2.6** **Module of Workout Subsystem**

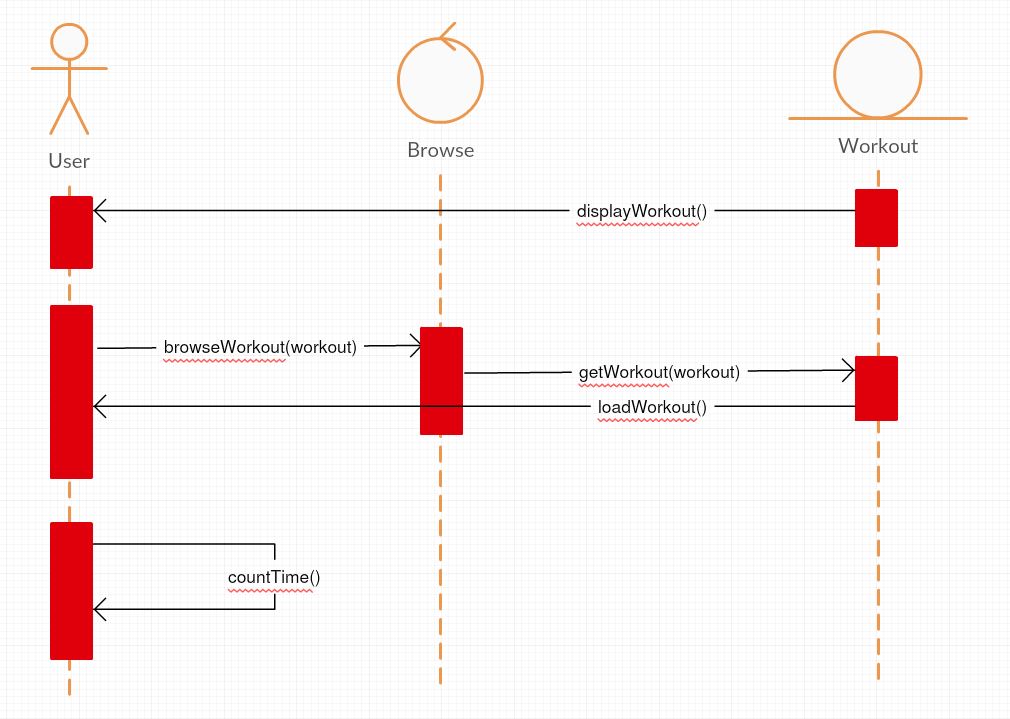
The purpose of Workout System is to provide the user the instructions of workout as well as figures for better illustration. The user can choose the variety of different exercises. Besides that, the timer will be included to count the time for user to execute the workout. The below is the general use case diagram, use case specification, sequence diagram and activity diagram of Workout system.



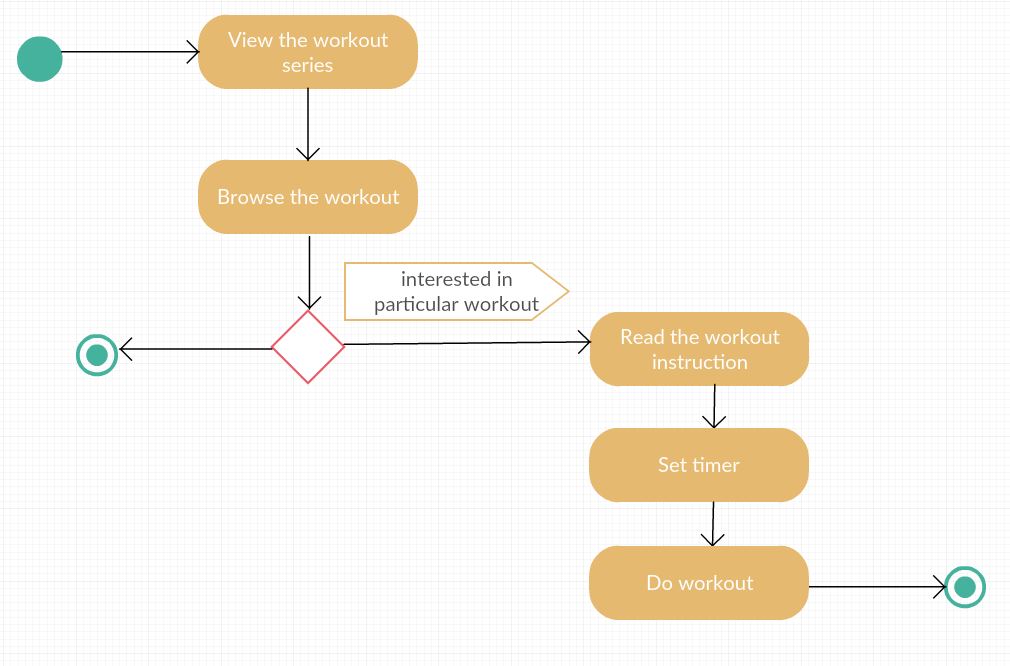
**Figure 3.2.6.1.1: Workout Subsystem**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Workout | |
| **Scenario:** | View the workout series  Choose the suitable workout  Read instruction of workout  Set workout timer  Do workout | |
| **Triggering Event:** | User learns the workout steps. | |
| **Brief Description:** | The user can browse all healthy food suggestion along with its nutrition information across all categories and types of food. | |
| **Actors:** | User | |
| **Related Use Case:** | Extend: Read instruction of workout | |
| **Stakeholder:** | User | |
| **Precondition:** | User should clicks through workout series to view particular workout instruction. | |
| **Postcondition:** | The workout instruction is shown. | |
| **Flow of Events:** | **User** | **Workout System** |
| 1. The user clicks the workout series to browse the workout.  2. The user clicks on particular workout to get instruction.  3. The user sets the workout timer and starts exercise. | 1.1. The system collects the request of user.  1.2 The system retrieves the information from database.  2.1. The system collects the request of user.  2.2. The system retrieves the information from database. |
| **Exception Conditions:** | none | |

**Use Case Description of Workout System**



**Figure 3.2.6.1.2: Sequence Diagram of Workout**



**Figure 3.2.6.1.3: Activity Diagram of Workout**

**3.2.6.1 UC016: View Workout Series Use Case**

The user views the workout instruction by browsing the workout series. The user can chooses either with mild workout or heavy workout.

**3.2.6.2 UC017: Choose Suitable Workout Use Case**

The user can choose the workout based on own desirable. The particular workout also shows the instruction to teach the user the proper way to carry out workout.

**3.2.6.3 UC018: Set Workout Timer Use Case**

The timer is given for counting the time to do workout. For example, the user is given 2 minutes to do sit-up. Once the time is finished, the apps will remind the user.

**3.2.6.4 UC019: Do Workout Use Case**

The user does the workout with the helps of workout instruction and timer.

**3.3** **Performance Requirements**

**3.3.1 Responding Time**

The system shall respond to the user input within 2 seconds. The system shall be able to handle the requests of user, as would be critical in workout timer and pharmacy locator with GPS.

**3.3.2 Speed**

When GPS is running, speed is important. If locating the user will take more than a few seconds, the user shall be warned of this as the user is mobile and always changing the location, the system needs high speed of data transfering to get the latest location.

**3.4** **Design Constraints**

In BMI Calculator System, the BMI formulae must be exactly same as the global standardised BMI formulae. The obtained BMI value from formulae determines the body category of the user. As for the attributes, especially for height, user can only enter height above 147 cm and below 191 cm to display range of normal weight perpendicular to user’s height.

In Food Suggestion System, the food that contains too high calorie value is not recommended to insert into the system. The food needs to be analyzed according to nutrition information that whether it is suitable for healthy food intake.

In Workout System, the workout instructions are needed follow the gym training standard. The standardised gym training constraints the frequency of workout. For example, sit-up workout for 20 times in 1 set is limited to do 3 sets each day, which means sit-up workout for total 60 times in a day.

In Water Intake System, the volume of water taken is limited to maximum 8 litres in a day.

In Pharmacy Locator System, user can only locate nearby pharmacies at most of 5 kilometres in radius from current location. User can tap on the pharmacy listed to get the name of the pharmacy and it’s address but not yet to navigate to the place.

**3.5** **Software System Attributes**

**3.5.1 Simplicity**

In the Workout System, the figures of illustrating the workout are needed so the user like workout beginner can understands the workout instruction easily.

**3.5.2 Discoverability**

In the Food Suggestion System, the images of food are shown in great diversity as well as its nutrition information, therefore the health keeper has the good idea of healthy food intake.

**3.5.3 Determinability**

In BMI Calculator System, the dieter only needs to enter own weight and height to get the BMI value results and its categorization. The body category results is displayed to remind the dieter to keep the dietary life going on until a healthy body is obtained.

**3.5.4 Accuracy**

In Pharmacy Locator System, the radius of searching pharmacies can be adjusted so the traveler can reach the selected pharmacy without going too far.

**3.5.5 Correctness**

In Daily Water Intake System, user can adjust the measurement of input water (whether in ounce of mililitre) without triggering error.

**3.6** **Other Requirements**

**3.6.1** **Maintainability Requirement**

The ease with which the system can be maintained in order to maximize efficiency, reliability, and safety besides making future maintenance easier.

**3.6.2** **Reliability Requirement**

The system needed to be reliable so the user can get consistent data in function such as pharmacy locator. The system also should not crash or hang, other than as the result of operating system error.

**3.6.3** **Portability Requirement**

The system is installed in mobile phone Android operating system as it could be portable to any places and can be used by worldwide Android users.

**3.6.4** **Legal Requirement**

The copyright laws and license agreements must be respected for any third party software used in the creation of this system.