**Use Cases Document**

|  |  |  |
| --- | --- | --- |
| **No.** | **Group Name (SE15)** | **Code** |
| **1** | Mohamed Abdelrahman Aboelkassem | 180113 |
| **2** | Mohamed Gamal Yaseen | 180104 |
| **3** | Mohamed Essam Abdelmoneem | 180114 |
| **4** | Nourhan Mahmoud Hussein | 180146 |
| **5** | Ola Abdallah Mogawer | 180087 |

|  |  |
| --- | --- |
| **Module Name** | Daily Healthy Food (DHF) |
| **Date** | Dec 10th, 2020 |
| **Version** | 1.0 |

**Table of Contents**

Contents

[Features List 4](#_Toc58523627)

[Use Cases List 4](#_Toc58523628)

[***1 Registration system*** 5](#_Toc58523629)

[***1.1 Feature Process Flow / Use Case Model*** 5](#_Toc58523630)

[***1.2 Use Case(s)*** 5](#_Toc58523631)

[2 Account Management 7](#_Toc58523632)

[***2.1 Feature Process Flow / Use Case Model*** 7](#_Toc58523633)

[***2.2 Use Case(s)*** 7](#_Toc58523634)

[3 Calculation of his calories 8](#_Toc58523635)

[***3.1 Feature Process Flow / Use Case Model*** 8](#_Toc58523636)

[***3.2 Use Case(s)*** 8](#_Toc58523637)

[4 Daily Remaining Calories 9](#_Toc58523638)

[***4.1 Feature Process Flow / Use Case Model*** 9](#_Toc58523639)

[***4.2 Use Case(s)*** 9](#_Toc58523640)

[5 Receiving day’s food 10](#_Toc58523641)

[***5.1 Feature Process Flow / Use Case Model*** 10](#_Toc58523642)

[***5.2 Use Case(s)*** 10](#_Toc58523643)

[6 Detecting food item 13](#_Toc58523644)

[***6.1 Feature Process Flow / Use Case Model*** 13](#_Toc58523645)

[***6.2 Use Case(s)*** 13](#_Toc58523646)

[7 Integration of food nutrients 14](#_Toc58523647)

[***7.1 Feature Process Flow / Use Case Model*** 14](#_Toc58523648)

[***7.2 Use Case(s)*** 14](#_Toc58523649)

[8 Recommendation of healthy food. 15](#_Toc58523650)

[***8.1 Feature Process Flow / Use Case Model*** 15](#_Toc58523651)

[***8.2 Use Case(s)*** 15](#_Toc58523652)

[9 Food menus management 16](#_Toc58523653)

[***9.1 Feature Process Flow / Use Case Model*** 16](#_Toc58523654)

[***9.2 Use Case(s)*** 16](#_Toc58523655)

[10 Physical activities management 19](#_Toc58523656)

[***10.1 Feature Process Flow / Use Case Model*** 19](#_Toc58523657)

[***10.2 Use Case(s)*** 19](#_Toc58523658)

[11 Calculation of burnt energy 21](#_Toc58523659)

[***11.1 Feature Process Flow / Use Case Model*** 21](#_Toc58523660)

[***11.2 Use Case(s)*** 21](#_Toc58523661)

[12 Receiving user’s goals 22](#_Toc58523662)

[***12.1 Feature Process Flow / Use Case Model*** 22](#_Toc58523663)

[***12.2 Use Case(s)*** 22](#_Toc58523664)

[13 Reporting food and physical activities 23](#_Toc58523665)

[***13.1 Feature Process Flow / Use Case Model*** 23](#_Toc58523666)

[***13.2 Use Case(s)*** 23](#_Toc58523667)

**Features List**

|  |  |  |
| --- | --- | --- |
| **Feature ID** | **Module Name** | **Feature Description** |
| 1 | Registration system | User can login and register within our app using some data like height, weight, date of birth |
| 2 | Account Management | User can update his personal data |
| 3 | Calculation of his calories | System can calculate the user calories by given weight, age and height. |
| 4 | Daily Remaining Calories | System calculates the remaining calories from given food items to show the user the progress. |
| 5 | Receiving day’s food | User shall enter food by providing food name, amount, date. |
| 6 | Detecting food item | System can detect food item by taking a capture using camera with computer vision modal. |
| 7 | Integration of food nutrients | System will display for each food it’s nutrients to the user. |
| 8 | Recommendation of healthy food. | User after choose a list of daily food, System can calculate the remaining vitamins that required, suggest to increase or decrease suitable food. |
| 9 | Food menus management | User can create menus by giving a menu name, food items, And can add, delete, update food items to this menu, each food display its nutrients. |
| 10 | Physical activities management | User should enter physical activity by writing activity name, duration in minutes containing list contains relevant activities name like exercise, summing, running. |
| 11 | Calculation of burnt energy | System should calculate how much energy a user burned according to user’s height and weight. |
| 12 | Receiving user’s goals | user can enter his goals. |
| 13 | Reporting food and physical activities | User shall display how many calories consumed and burnt, and the balance between them in daily, weekly reports. |

**Use Cases List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature ID** | **Use Case ID** | **Primary Actor** | **Use Cases** |
| 1 | UC1.1 | User | Login |
| 1 | UC1.2 | User | Register |
| 2 | UC2.1 | User | Update user info |
| 3 | UC3.1 |  | Calculate Calories |
| 4 | UC4.1 |  | Calculate remaining vitamins |
| 5 | UC5.1 | User | Enter custom food |
| 5 | UC5.2 | User | Select daily food |
| 5 | UC5.3 |  | Calculate food vitamins |
| 5 | UC4.1 |  | Calculate remaining vitamins |
| 6 | UC6.1 |  | Detect food items |
| 7 | UC7.1 | nutritionix | Read food nutrients |
| 8 | UC8.1 |  | Recommend healthy food |
| 9 | UC9.1 | User | Create menu |
| 9 | UC9.2 | User | Display menu items |
| 9 | UC9.3 | User | Manage menu |
| 9 | UC9.4 | User | Edit menu |
| 9 | UC9.5 | User | Delete menu |
| 10 | UC10.1 | User | Add physical activities |
| 10 | UC10.2 | User | Manage physical activities |
| 10 | UC10.3 | User | Edit activities |
| 10 | UC10.4 | User | Delete activities |
| 11 | UC11.1 |  | Calculate burnt energy |
| 11 | UC11.2 | User | View burnt energy |
| 12 | UC12.1 | User | Get user goals |
| 12 | UC12.2 |  | Display the progress |
| 13 | UC13.1 | User | Manage reports |
| 13 | UC13.2 |  | Display weekly reports |

***1 Registration system***

***1.1 Feature Process Flow / Use Case Model***

***1.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC1.1 | | | |
| **Use Case Name:** | Login | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | In this case of use, the user enters his email and password to be able to log into the application | | |
| **Trigger:** | | The use case begins with the user logging into the app if they registered before | | |
| **Preconditions:** | | 1. The user must have completed registration at the registration stage | | |
| **Post conditions:** | | 1. The user is able to log in to the home page 2. The user is able to update his data and use the application easily | | |
| **Normal Flow:** | | 1. User insert your email that he entered in the registration process 2. User insert your password that he entered in the registration process 3. The user clicks the login button 4. The application displays a message waiting for the user to be able to verify the validity of the data in the database | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 3a. In step 3 of the normal flow, The user can register with Google   1. The user clicks the login by google button 2. The user selects the account he wants to log in with 3. Use Case resumes on step 4   3b. In step 3 of the normal flow, The user can register with Facebook   * + - 1. The user clicks the login by Facebook button       2. The user selects the account he wants to log in with       3. Use Case resumes on step 4 | | |
| **Exceptions:** | | 1a. In step 1 of the normal flow, if the user enters and invalid email   1. login is disapproved 2. Message to customer to re-enter his information 3. user enters correct information 4. Use Case resumes on step 3 of normal flow   2a. In step 2 of the normal flow, if the user enters and invalid password   1. login is disapproved 2. Message to customer to re-enter his information 3. user enters correct information 4. Use Case resumes on step 3 of normal flow | | |
| **Includes:** | | UC1.2 , UC3.1 | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | A high degree of security and privacy | | |
| **Assumptions:** | | Dealing with those who understand well the English language | | |
| **Notes and Issues:** | | Minimum password is 8 items containing at least one character | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC1.2 | | | |
| **Use Case Name:** | Register | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | In this case of use, the user enters his email , password ,phone number , age , Length and the weight to be able to login to the application | | |
| **Trigger:** | | The use case begins with the user logging into the application if they have not registered before | | |
| **Preconditions:** | | 1. The user enters the login page and clicks the Register button | | |
| **Post conditions:** | | 1. A message "Registration completed successfully" appears 2. The application goes directly to the login page 3. The user is able to log in to the home page | | |
| **Normal Flow:** | | 1. User insert your email 2. User insert your password 3. User insert your phone number and length 4. User insert your age and the weight 5. The user clicks the register button 6. The application displays a message waiting for the user to be able to verify the validity of the data that have been entered and recorded in the database | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1a. In step 1 of the normal flow, if the user enters and invalid email   1. register is disapproved 2. Message to customer to re-enter his email 3. user enters correct information 4. Use Case resumes on step 5 of normal flow   2a. In step 2 of the normal flow, if the user enters and invalid password   1. register is disapproved 2. Message to customer to re-enter password 3. user enters correct information 4. Use Case resumes on step 5 of normal flow   3a. In step 3 of the normal flow, if the user enters and invalid phone number or length   1. register is disapproved 2. Message to customer to re-enter phone number or length 3. user enters correct information 4. Use Case resumes on step 5 of normal flow   4a. In step 4 of the normal flow, if the user enters and invalid age or the weight   1. register is disapproved 2. Message to customer to re-enter age or the weight 3. user enters correct information 4. Use Case resumes on step 5 of normal flow | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | A high degree of security and privacy | | |
| **Assumptions:** | | Dealing with those who understand well the English language | | |
| **Notes and Issues:** | | 1. Minimum password is 8 items containing at least one character 2. Minimum age is 12 years 3. Minimum phone number is 11 items 4. Minimum weight is 25 Kg | | |

**2 Account Management**

***2.1 Feature Process Flow / Use Case Model***

***2.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC2.1 | | | |
| **Use Case Name:** | Update user info | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | In this case of use, the user changes his info (age , phone number , length , weight ) | | |
| **Trigger:** | | The use case starts when the user wants to change the data and clicks the "Modify User Data" button from the settings page | | |
| **Preconditions:** | | 1. The user has successfully signed in to the application | | |
| **Post conditions:** | | 1. User data in the database is changed 2. Calculations are performed in the application that depend on this data | | |
| **Normal Flow:** | | 1. The user enter to settings page 2. The user clicks the "Modify User Data" button from the settings page 3. The application goes directly to the data change page 4. The user changes the data he wants 5. He clicks on the "Save Changes" button 6. A waiting message appears on the screen 7. The application goes directly to the settings page | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 4a. In step 4 of the normal flow, if the user enters and invalid phone number or length   1. register is disapproved 2. Message to customer to re-enter phone number or length 3. user enters correct information 4. Use Case resumes on step 5 of normal flow   4a. In step 4 of the normal flow, if the user enters and invalid age or the weight   1. register is disapproved 2. Message to customer to re-enter age or the weight 3. user enters correct information 4. Use Case resumes on step 5 of normal flow | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 1000 per day | | |
| **Special Requirements:** | | A high degree of privacy and the speed of performance | | |
| **Assumptions:** | | Dealing with those who understand well the English language | | |
| **Notes and Issues:** | | 1. Minimum password is 8 items containing at least one character 2. Minimum age is 12 years 3. Minimum phone number is 11 items 4. Minimum weight is 25 Kg | | |

**3 Calculation of his calories**

***3.1 Feature Process Flow / Use Case Model***

***3.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC3.1 | | | |
| **Use Case Name:** | Calculate Calories | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | |  | | |
| **Description:** | | In this case of use, Calories are calculated from the type of food that has been identified in the application | | |
| **Trigger:** | | This use case begins when the user clicks the " Calculate Calories " button on the food selection page | | |
| **Preconditions:** | | 1. The user has successfully signed in to the application 2. The user must specify the group of foods that he will eat | | |
| **Post conditions:** | | 1. The user appears with a message containing the result of the calculation and a number calories 2. Save this result in database To be used in preparing reports | | |
| **Normal Flow:** | | 1. The user enter to food page 2. The user selects the food 3. The user clicks the "Calculate Calories" button 4. A message appears with the number of calories in the selected food 5. He clicks on the "Close" button 6. The message disappears | | |
| **Alternative Flows:** | | 5a. In step 5 of the normal flow, The user can close this window by pressing in empty position in the screen   1. The user clicks in empty position in the screen 2. Use Case resumes on step 6 | | |
| **Exceptions:** | | 2a. In step 2 of the normal flow, if the user enters and invalid phone number or length   1. Calculation process failed 2. Message to user to determine the food 3. user determines the food 4. Use Case resumes on step 3 of normal flow | | |
| **Includes:** | | UC8.1 , UC13.1 | | |
| **Frequency of Use:** | | 1000 per day | | |
| **Special Requirements:** | | A high degree of speed of performance | | |
| **Assumptions:** | | Dealing with those who understand well the English language | | |
| **Notes and Issues:** | | 1. At least one food is chosen | | |

**4 Daily Remaining Calories**

***4.1 Feature Process Flow / Use Case Model***

***4.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC4.1 | | | |
| **Use Case Name:** | Calculate remaining vitamins | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | |  | | |
| **Description:** | | In this case the user is calculating the remaining vitamins (which the user must take) | | |
| **Trigger:** | | When the user selects the food, it is used in math operations | | |
| **Preconditions:** | | 1. The user has successfully signed in to the application 2. The food identification stage is completed 3. The calculation of the selected foods ends | | |
| **Post conditions:** | | 1. Coming up with a result in which the number of remaining vitamins is to complete the user's healthy food | | |
| **Normal Flow:** | | 1. We get the selected foods from memory 2. After calculating the calculations for calories, etc. 3. Calculations of the remaining vitamins are done in the background 4. The result appears on the main page of the application | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 1a. In step 1 of the normal flow, If there are no foods selected   1. The calculation will give incorrect results ) null ) 2. Show the message "Choose food first" 3. The user clicks "select food" in the message 4. The application goes directly to the food selection page 5. The user selects a food 6. Use Case resumes on step 2 of normal flow | | |
| **Includes:** | | UC5.3 , UC7.1 | | |
| **Frequency of Use:** | | 10 per day | | |
| **Special Requirements:** | | A high degree of the speed of performance | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

**5 Receiving day’s food**

***5.1 Feature Process Flow / Use Case Model***

***5.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC5.1 | | | |
| **Use Case Name:** | Enter custom food | | | |
| **Created By:** | Mohamed Abdelrahman | | **Last Updated By:** | Mohamed Abdelrahman |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | The user can enter food by providing food name, amount, date if he didn’t find in our system. | | |
| **Trigger:** | | When click on search bar list to find his food. | | |
| **Preconditions:** | | 1. User must be registered to our app. 2. User must have active account. 3. User has entered his personal data like weigh, height, date of birth. 4. User Type in search bar the food name. 5. User had selected which meal will this food item belongs to. | | |
| **Post conditions:** | | 1. After entered his custom food, he should choose the amount of food (g). 2. System will display this food to daily user’s dashboard. 3. System calculate the calories of this food. 4. System update the consumed calories in this day. | | |
| **Normal Flow:** | | 1. User enters password and email to login to our app. 2. System display user’s daily dashboard which contains the food he eat today, physical activities, goal. 3. User clicks on add food button. 4. System prompts user to select main meal. 5. User enters the food name he wants to add. 6. System presented autocomplete list which contains large food names. 7. System must prioritize food names which are already in user’s history in autocomplete list. 8. User choose enter custom food if he did not find the target food in our system. 9. User chooses the amount of food he eat. 10. System calculate the calories of this food. 11. System display the nutrients of this food item in the dashboard. 12. System update the consumed calories of each food he eats and remaining calories per day. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 3a. In step 3 of the normal flow, if the user did not want to click this button   1. Choose search bar to search the food 2. The food exists into our app. 3. Add this food to current day. 4. Use case resumes on step 4   3b. In step 3 of the normal flow, if the user did not want to click this button   1. Choose search bar to search the food 2. The food does not exist in our app. 3. Use case resumes on step 8 | | |
| **Exceptions:** | | 1a. In step 1 of the normal flow, if the user enters and invalid email, password.   1. System display to the user to enter correct email, password. 2. User enters valid email, password. 3. Use Case resumes on step 2 in normal flow   10a. In step 10 of the normal flow, if the user enters the very small amount of food.   1. System prompt to the user that he should enter the amount by gram 2. User entered correct amount of food 3. System calculate the calories of this food 4. Use Case resumes on step 11 in normal flow | | |
| **Includes:** | | UC5.2 | | |
| **Frequency of Use:** | | 10000 Per day | | |
| **Special Requirements:** | | 1. Manage Food operations 2. The system shall be both resource-efficient and scalable 3. The system should be usable. 4. The system shall push and encourage its users | | |
| **Assumptions:** | | 1. User must write the name of food in English or Arabic languages. 2. User must know the food names he eats of his country. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC5.2 | | | |
| **Use Case Name:** | Select daily food | | | |
| **Created By:** | Mohamed Abdelrahman | | **Last Updated By:** | Mohamed Abdelrahman |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | Allow user to select the food he wants to eat from our system in the 4 main meals and see which nutrients contains. | | |
| **Trigger:** | | When user click on add food button | | |
| **Preconditions:** | | 1. User must be registered to our app. 2. User must have active account. 3. User has entered his personal data like weigh, height, date of birth. 4. User had chosen his current meal. | | |
| **Post conditions:** | | 1. After entered his custom food, he should choose the amount of food (g). 2. System will display this food to daily user’s dashboard. 3. System calculate the calories of this food. 4. System update the consumed calories in this day. | | |
| **Normal Flow:** | | 1. User enters password and email to login to our app. 2. System display user’s daily dashboard which contains the food he eat today, physical activities, goal. 3. User clicks on add food button. 4. System prompts user to select main meal. 5. User enters the food name he wants to add. 6. System presented autocomplete list which contains large food names. 7. User chooses the amount of food he eat. 8. System calculate the calories of this food. 9. System display the nutrients of this food item in the dashboard. 10. System update the consumed calories of each food he eats and remaining calories per day. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 4a. In step 4 of the normal flow, if the user does not select to his main meal.   1. System put this food item in default meal (breakfast) 2. Use case resumes on step 5   8a. In step 8 of the normal flow, if the user enters food name not exist.   1. User add custom food name, amount (g), nutrients. 2. System calculate the calories of this food. 3. Use case resumes on step 9 | | |
| **Exceptions:** | | 5a. In step 5 of the normal flow, if the user enters numbers or invalid food name.   1. System show message that he needs to enter only text and correct food name. 2. User enters correct and exist food name 3. Use Case resumes on step 7 in normal flow | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 50000 Per day | | |
| **Special Requirements:** | | 1. The system shall expose an HTTP RESTful API which supports 2. The system should be usable. 3. Manage Food operations 4. The system shall be both resource-efficient and scalable | | |
| **Assumptions:** | | 1. User must select the name of food in English or Arabic languages. 2. User must know the food names he eats of his country. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC5.3 | | | |
| **Use Case Name:** | Calculate food vitamins | | | |
| **Created By:** | Mohamed Abdelrahman | | **Last Updated By:** | Mohamed Abdelrahman |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | |  | | |
| **Description:** | | User have maximum daily vitamins that his body needs, so the system calculate each food it’s vitamins to keep the body healthy. | | |
| **Trigger:** | | When user select/enter his daily food. | | |
| **Preconditions:** | | 1. User completed login process. 2. User Selected food item. 3. System received food nutrients from external system. 4. System Calculated daily vitamins and calories for the user | | |
| **Post conditions:** | | 1. System calculated consumed daily food vitamins. 2. System calculated remaining food vitamins. 3. System Recommended a health food to eat. 4. System Display these calculations on user’s daily dashboard. | | |
| **Normal Flow:** | | 1. Users enter password and email to login to our app. 2. System display user’s daily dashboard which contains the food he eat today, physical activities, goal. 3. User clicks on add food button. 4. System prompts user to select main meal. 5. User enters the food name he wants to add. 6. System presented autocomplete list which contains large food names. 7. User chooses the amount of food he eats. 8. System gets the nutrients of the food item from the API 9. System calculate the calories and vitamins of this food. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 8a. In step 8 of the normal flow, if the system cannot get nutrients of the food   1. System use alternative API to get food nutrients 2. System calculate nutrients depending on amount of food user entered. 3. Use case resumes on step 9 | | |
| **Exceptions:** | | 8a. In step 8 of the normal flow, if the system returns an error from external systems.   1. System push notification telling user that there is an error 2. User sends report back to the system 3. System show message to refresh the app or try again later. 4. System terminate adding food item. | | |
| **Includes:** | | UC7.1 | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | 1. Recommend Healthy Food 2. Calculate User calories 3. The system shall be both resource-efficient and scalable 4. The system shall expose an HTTP RESTful API which supports | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

**6 Detecting food item**

***6.1 Feature Process Flow / Use Case Model***

***6.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC6.1 | | | |
| **Use Case Name:** | Detect food items | | | |
| **Created By:** | Mohamed gamal yaseen | | **Last Updated By:** | Mohamed gamal yaseen |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | |  | | |
| **Description:** | | The camera is used to take pictures of food and it is recognized in the application so that all operations in the application are completed on it | | |
| **Trigger:** | | This use case is used when the user clicks the " Detect food " button on the food page | | |
| **Preconditions:** | | 1. The user has successfully signed in to the application 2. The app takes permission to access the camera | | |
| **Post conditions:** | | 1. User data in the database is changed 2. Calculations are performed in the application that depend on this data | | |
| **Normal Flow:** | | 1. The user enter to food page 2. The user clicks the " Detect food " button from the settings page 3. If you are the first time, the app must obtain permission to access the camera 4. The application goes directly to the camera 5. User takes a photo of the food 6. He clicks on the "Done " button 7. A waiting message appears on the screen 8. The app detects the food 9. The application goes directly to the food page 10. After identifying the food, the name and information of the food appear | | |
| **Alternative Flows:** | | 3a. In step 3 of the normal flow, The user can choose photo from gallery   1. User allows app access to gallery 2. Select the image 3. He presses the " Done " button 4. Use Case resumes on step 7 | | |
| **Exceptions:** | | 3a. In step 3 of the normal flow, if the user If the user does not allow the application to access the camera   1. Detection fails 2. A message appears to the user that the operation failed 3. The user allows the application to access the camera 4. Use Case resumes on step 4 of normal flow   8a. In step 8 of the normal flow, If the app doesn't recognize the food   1. Detection fails 2. A message appears to the user that the operation failed 3. User retakes the photo 4. Use Case resumes on step 9 of normal flow | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 10 per day | | |
| **Special Requirements:** | | A high degree of privacy and the speed of performance | | |
| **Assumptions:** | | 1. User must have a camera in his device. 2. User must have real food item to be detected. | | |
| **Notes and Issues:** | | 1. To complete the process, the application must obtain permission to access the camera 2. To complete the process, the internet must be active | | |

**7 Integration of food nutrients**

***7.1 Feature Process Flow / Use Case Model***

***7.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC7.1 | | | |
| **Use Case Name:** | Read food nutrients | | | |
| **Created By:** | Mohamed Abdelrahman | | **Last Updated By:** | Mohamed Abdelrahman |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | Nutritionix | | |
| **Description:** | | Allow user to know the nutrients of each food item to be tracking and calculated into our app, | | |
| **Trigger:** | | When user add or select food item | | |
| **Preconditions:** | | 1. User must be registered to our app. 2. User must have active account. 3. User must select or write a food item name. | | |
| **Post conditions:** | | 1. User receives the nutrients of each food item. 2. System stores these nutrients in our database. 3. System display food name and it’s nutrients to user’s dashboard | | |
| **Normal Flow:** | | 1. User login to our app using email and password. 2. System display user’s daily dashboard which contains the food he eats today, physical activities, goal. 3. User clicks on add food button. 4. User writes food name. 5. External System presented autocomplete list with food names including their nutrients. 6. User chooses one of them to add into our app. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 5a. In step 5 of the normal flow, if the system returns an error from external systems.   1. System push notification telling user that there is an error 2. User sends report back to the system 3. System show message to refresh the app or try again later. 4. System terminate adding food item. | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 50000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system shall expose an HTTP RESTful API which supports 3. The system shall use and contain only open source technologies, libraries, and tools | | |
| **Assumptions:** | | 1. User should write the food name item in English or Arabic language. 2. User should write a correct food name. | | |
| **Notes and Issues:** | |  | | |

**8 Recommendation of healthy food.**

***8.1 Feature Process Flow / Use Case Model***

***8.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC8.1 | | | |
| **Use Case Name:** | Recommend healthy food | | | |
| **Created By:** | Mohamed Abdelrahman | | **Last Updated By:** | Mohamed Abdelrahman |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | |  | | |
| **Description:** | | In this use case the system recommends healthy food for user depending of remaining vitamins | | |
| **Trigger:** | | When selected a daily food and calculated remaining vitamins. | | |
| **Preconditions:** | | 1. User must be login to our app 2. User selects daily food 3. System calculated remaining vitamins. | | |
| **Post conditions:** | | 1. System suggest list of food suitable to his status. 2. User keep stay healthy. | | |
| **Normal Flow:** | | 1. User selected his daily food. 2. System gets the calorie and vitamin data from memory 3. System process this data and using artificial intelligence, it suggests some foods to the user. 4. These suggestions appear on the home page. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 3a. In step 3 of the normal flow, if the result suggestions not suitable to users needs.   1. User asks to load more suggestions. 2. System retrieve user’s data 3. System suggest another food items. 4. Use Case resumes on step 4 in normal flow | | |
| **Exceptions:** | | 2a. In step 2 of the normal flow, If there is a problem obtaining the information on calories and vitamins.   1. The suggestion process will not work 2. System prompts a message to select daily food items. | | |
| **Includes:** | | UC5.2 | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | 1. Recommend Healthy Food 2. The system shall be both resource-efficient and scalable 3. The system shall push and encourage its users | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

**9 Food menus management**

***9.1 Feature Process Flow / Use Case Model***

***9.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9.1 | | | |
| **Use Case Name:** | Create menu | | | |
| **Created By:** | Mohamed essam | | **Last Updated By:** | Mohamed essam |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | Let user to enter many food items for one menu. | | |
| **Trigger:** | | The use case starts when the user clicks on create menu button | | |
| **Preconditions:** | | 1. User login to our system 2. System contains at least one more food item in database. | | |
| **Post conditions:** | | 1. System store menu of the user`s menu including food items. 2. System calculate Food vitamins, calories. 3. System display food items nutrients in the menu. | | |
| **Normal Flow:** | | 1. User login into the system 2. User click create menu button 3. User adds food items to the menu 4. Click done to add the menu to database | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 3a. In step 3 of the normal flow, if the user want adds the food items with another way.   1. User enters food items name, amount manually. 2. Use Case resumes on step 4 | | |
| **Exceptions:** | | 3a. In step 3 of the normal flow, if the user had entered invalid food item name or not exist.   1. System show a message to enter valid name item. 2. System remove the previous name 3. User reentered valid name 4. Use Case resumes on step 4   4a. In step 4 of the normal flow, if the user clicked done button without entered food items to the menu   1. System show a message to select the food items into the menu. 2. System prompts options for the user to back. 3. System redirect dashboard. | | |
| **Includes:** | | UC5.2 | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system shall expose an HTTP RESTful API which supports 3. The system shall push and encourage its users | | |
| **Assumptions:** | | User must know Arabic or English languages to create new data. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9.2 | | | |
| **Use Case Name:** | Display menu | | | |
| **Created By:** | Mohamed essam | | **Last Updated By:** | Mohamed essam |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | Allow user to display the menu items and it’s details. | | |
| **Trigger:** | | When the user clicks on any menu name. | | |
| **Preconditions:** | | 1. User has logged to our system 2. User has created a menu | | |
| **Post conditions:** | | 1. User can see his/her menu details. | | |
| **Normal Flow:** | | 1. User login to the system 2. User navigates to menus page 3. User clicks on any menu name. 4. System display the details of the menu. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | |  | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system should be usable. 3. The system shall push and encourage its users | | |
| **Assumptions:** | | User must know Arabic or English languages to read menu details. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9.3 | | | |
| **Use Case Name:** | Manage menu | | | |
| **Created By:** | Mohamed essam | | **Last Updated By:** | Mohamed essam |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | A process to facilitate dealing with menus to the user | | |
| **Trigger:** | | When user click on manage menu button | | |
| **Preconditions:** | | 1. User Login to the system | | |
| **Post conditions:** | | 1. User can create menu 2. User can display menu 3. User can edit menu 4. User can delete menu | | |
| **Normal Flow:** | | 1. User login to the system 2. User pressed manage menus button 3. System enable the user to manage foods menus. 4. System allow the user to select any of its operations like display menu, edit menu and delete menu. | | |
| **Alternative Flows:** | | 3a. In step 3 of the normal flow, if the user did not want to manage any of his menus.   1. User goes back to home dashboard. | | |
| **Exceptions:** | | 4a. In step 4 of the normal flow, if the system has no menus to display, edit, delete.   1. System show a message that there are not menus to manage. 2. System prompts an option to create a menu. | | |
| **Includes:** | | UC9.1, UC9.3, UC9.4, UC9.5 | | |
| **Frequency of Use:** | | 100000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system should be usable. | | |
| **Assumptions:** | | User must know Arabic or English languages to manage menus. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9.4 | | | |
| **Use Case Name:** | edit menu | | | |
| **Created By:** | Mohamed essam | | **Last Updated By:** | Mohamed essam |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | Allow user to update the menu by deleting existing food item, adding another. | | |
| **Trigger:** | | When the user navigates to “manage menu page”, hover to menu name click “Edit menu” | | |
| **Preconditions:** | | 1. User had logged to our system. 2. User had created a menu 3. User must be navigated to “manage menu page” | | |
| **Post conditions:** | | 1. User able to delete existing food item. 2. User able to add a new food item to the menu. 3. System display the updated menu with new details. | | |
| **Normal Flow:** | | 1. User login to the app 2. User navigates to “manage menu page” 3. User hover to menu name 4. User clicks to edit menu button. 5. User deletes existing food items or adding new food items. 6. User saves new updated data | | |
| **Alternative Flows:** | | 3a. In step 3 of the normal flow, if the user clicked on menu name to display menu details.   1. Use Case resumes on step 4 | | |
| **Exceptions:** | | 6a. In step 6 of the normal flow, if the user clicked on back without saving the data   1. System prompts window to confirm if he wants to leave without saving new data. 2. System shows a new button to save or discard. | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 10000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system should be usable. 3. The system shall push and encourage its users | | |
| **Assumptions:** | | User must know Arabic or English languages to update new data. | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC9.5 | | | |
| **Use Case Name:** | Delete menu | | | |
| **Created By:** | Mohamed essam | | **Last Updated By:** | Mohamed essam |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** | Dec 10th, 2020 |
| **Actors:** | | User | | |
| **Description:** | | Allow user to delete menu including it’s food items. | | |
| **Trigger:** | | When the user navigates to “manage menu page”, hover to menu name click “Delete menu” | | |
| **Preconditions:** | | 1. User had logged to our system. 2. User had created a menu 3. User must be navigated to “manage menu page” | | |
| **Post conditions:** | | 1. System delete the menu from dashboard. 2. System delete the menu item from daily dashboard. 3. System recalculated the daily food items calories and vitamins. | | |
| **Normal Flow:** | | 1. User login to the app 2. User navigates to “manage menu page” 3. User hover to menu name 4. User clicks to delete menu button. 5. System show confirmation message | | |
| **Alternative Flows:** | | 3a. In step 3 of the normal flow, if the user clicked on menu name to display menu details.   1. Use Case resumes on step 4 | | |
| **Exceptions:** | |  | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 1000 per day | | |
| **Special Requirements:** | | 1. Manage Food menu operations 2. The system should be usable. 3. The system shall expose an HTTP RESTful API which supports. 4. The system shall push and encourage its users | | |
| **Assumptions:** | | User must know Arabic or English languages to delete the menu. | | |
| **Notes and Issues:** | |  | | |

**10 Physical activities management**

***10.1 Feature Process Flow / Use Case Model***

***10.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC10.1 | | | |
| **Use Case Name:** | Add physical activities | | | |
| **Created By:** | Ola Abdallah | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | | User | | |
| **Description:** | | User should enter physical activity by writing activity name, duration in minutes containing list contains relevant activities name like exercise, summing, running. | | |
| **Trigger:** | | When the user navigates to “manage activities page”, click “Add physical activities” | | |
| **Preconditions:** | | 1. User must login with valid username and password. 2. User must navigate to “Manage activities page” | | |
| **Post conditions:** | | 1. System save the physical activities 2. System calculate burnt energy of the physical activities. | | |
| **Normal Flow:** | | 1. User navigates to “Mange activities page” 2. User clicks on Add new physical activities button 3. User adds the number of activities. 4. User selects the duration of each activity. 5. System calculate burnt energy. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | 3.1 select his activity from existing physical activities | | |
| **Exceptions:** | | 3a. In step 3 of the normal flow, if the user did not find any target activities.   1. User enters his custom activity 2. Use case resumes to step 4 | | |
| **Includes:** | | UC11.1, UC10.2 | | |
| **Frequency of Use:** | | 100000 per day. | | |
| **Special Requirements:** | | 1. Manage physical activity operations 2. The system should be usable. 3. The system shall be both resource-efficient and scalable | | |
| **Assumptions:** | | 1. User has knowledge of English and Arabic languages 2. User has knowledge of sports activities and how it played. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC10.2 | | | |
| **Use Case Name:** | Manage activities | | | |
| **Created By:** | Ola Abdallah | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | | User | | |
| **Description:** | | Allow user to manage his physical activities like summing, sports, running …etc to calculate the burnt energy. | | |
| **Trigger:** | | When user want to edit or delete physical activities. | | |
| **Preconditions:** | | Add physical activities. | | |
| **Post conditions:** | | Edit or delete activities | | |
| **Normal Flow:** | | 1- select the activity wanted to be delete or update.  2- update the activity list. | | |
| **Alternative Flows:** | | 2.1- remove the activities from activity list.  2.2- edit the activity wanted to update.  2.3- replace activity with another new one. | | |
| **Exceptions:** | | Ignore the process if the activity list is empty. | | |
| **Includes:** | | UC10.3, UC10.4 | | |
| **Frequency of Use:** | | 100000 per day. | | |
| **Special Requirements:** | | Recalculate the burnt energy. | | |
| **Assumptions:** | | View the burnt energy. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC10.3 | | | |
| **Use Case Name:** | Edit activities | | | |
| **Created By:** | Nourhan Mahmoud | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | | User | | |
| **Description:** | | User can edit the activity list and replace any with another, according to his daily activities. | | |
| **Trigger:** | | When the user navigates to “manage activities page”, click “Edit activity” | | |
| **Preconditions:** | | 1. User had logged to our system. 2. User had created a physical activity 3. User must be navigated to “manage activities page” | | |
| **Post conditions:** | | 1. User able to delete existing physical activity. 2. User able to add a new physical activity to the app. 3. System display the updated physical activity with new details. | | |
| **Normal Flow:** | | 1. User login to the app 2. User navigates to “manage physical activities page” 3. User clicks to edit physical activity button. 4. User saves new updated data | | |
| **Alternative Flows:** | | 2.1 select the wrong activity.  2.2 remove the wrong activity.  2.3 put the correct activity. | | |
| **Exceptions:** | | Ignore the process if the activity list is empty. | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 1000000 per day. | | |
| **Special Requirements:** | | 1. Manage physical activity operations 2. The system should be usable. 3. The system shall expose an HTTP RESTful API which supports | | |
| **Assumptions:** | | 1. User has knowledge of English and Arabic languages 2. User has knowledge of sports activities and how it played. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC10.4 | | | |
| **Use Case Name:** | Delete activities | | | |
| **Created By:** | Nourhan Mahmoud | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | | user | | |
| **Description:** | | User can delete the activity list, according to his daily activities. | | |
| **Trigger:** | | When user find a mistake in his activities. | | |
| **Preconditions:** | | Add activities | | |
| **Post conditions:** | | Activity list is updated. | | |
| **Normal Flow:** | | 1-Select the mistake in activity list  2- remove it. | | |
| **Alternative Flows:** | | 2.1 select the wrong activity.  2.2 remove the wrong activity. | | |
| **Exceptions:** | | Ignore the process if the activity list is empty. | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 100000 per day. | | |
| **Special Requirements:** | | Recalculate the burnt energy. | | |
| **Assumptions:** | | View the burnt energy. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

**11 Calculation of burnt energy**

***11.1 Feature Process Flow / Use Case Model***

***11.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC11.1 | | | |
| **Use Case Name:** | Calculate burnt energy | | | |
| **Created By:** | Ola Abdallah | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | |  | | |
| **Description:** | | System should calculate how much energy a user burned according to user’s height, weight and his activity list. | | |
| **Trigger:** | | User should approve the activity list. | | |
| **Preconditions:** | | The activity list isn’t empty. | | |
| **Post conditions:** | | View the burnt energy. | | |
| **Normal Flow:** | | 1-iterate over all activities.  2-calculate burnt energy in each activity.  3-accumulate burnt energy in all activity. | | |
| **Alternative Flows:** | | 2.1 in each activity calculate the burnt energy according to the amount of this activity. | | |
| **Exceptions:** | | Ignore this process if there is no activities in activity list. | | |
| **Includes:** | | UC11.2 | | |
| **Frequency of Use:** | | 10000 per day. | | |
| **Special Requirements:** | | Store the amount of burnt energy. | | |
| **Assumptions:** | | View the burnt energy. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC11.2 | | | |
| **Use Case Name:** | View burnt energy | | | |
| **Created By:** | Nourhan Mahmoud | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | |  | | |
| **Description:** | | User shall display how much energy burnt in particular activity. | | |
| **Trigger:** | | When system ends calculation. | | |
| **Preconditions:** | | The activity list isn’t empty and the burnt energy is calculated. | | |
| **Post conditions:** | | Evaluate the amount of burnt energy and recommend the user according the result. | | |
| **Normal Flow:** | | 1-view the amount of burnt energy then accumulate all of them. | | |
| **Alternative Flows:** | | * 1. for each activity view the amount of burnt energy.   2. View the summation of all of them. | | |
| **Exceptions:** | | Ignore this process if there is no activities in activity list. | | |
| **Includes:** | | UC11.1 | | |
| **Frequency of Use:** | | 100000 per day. | | |
| **Special Requirements:** | | Evaluate the amount of burnt energy. | | |
| **Assumptions:** | | Recommend the user according the result. | | |
| **Notes and Issues:** | | Follow up with the physical activities and the burnt energy. | | |

**12 Receiving user’s goals**

***12.1 Feature Process Flow / Use Case Model***

***12.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC12.1 | | | |
| **Use Case Name:** | Get user goals | | | |
| **Created By:** | Nourhan Mahmoud | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | | User | | |
| **Description:** | | User should determine his/her goal throw using this app if it’s (getting enough nutrients) or (burning calorie) or (losing weight).  User shall display his/her daily progress in a charts. | | |
| **Trigger:** | | When the user logged in. | | |
| **Preconditions:** | | Valid username and password. | | |
| **Post conditions:** | | Display the progress. | | |
| **Normal Flow:** | | 1. Login 2. Determine his goals. | | |
| **Alternative Flows:** | | 2.1 select the number of goals he will enter.  2.2 enter his goals one by one. | | |
| **Exceptions:** | | Ignore the process if Invalid username or password. | | |
| **Includes:** | | UC12.2 | | |
| **Frequency of Use:** | | 100000 per day | | |
| **Special Requirements:** | | Store the data. | | |
| **Assumptions:** | | Display the progress. | | |
| **Notes and Issues:** | | Follow up with the user’s progress. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC12.2 | | | |
| **Use Case Name:** | Display the progress | | | |
| **Created By:** | Ola Abdallah | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | |  | | |
| **Description:** | | System should calculate how much energy a user burned according to user’s height and weight.  User should determine his/her goal throw using this app if it’s (getting enough nutrients) or (burning calorie) or (losing weight).  User shall display his/her daily progress in a charts. | | |
| **Trigger:** | | When user enter his goals. | | |
| **Preconditions:** | | Get user’s goals. | | |
| **Post conditions:** | | Evaluating user’s progress. | | |
| **Normal Flow:** | | 1-System shall display user’s progress. | | |
| **Alternative Flows:** | | 1.1 calculate how much burned energy according to user’s height and weight.  1.2 calculate amount of nutrients and burning calorie.  1.3 accumulate all burnt energy and display the progress. | | |
| **Exceptions:** | | Ignore the process if no goals selected. | | |
| **Includes:** | | UC12.1 | | |
| **Frequency of Use:** | | 100000 per day. | | |
| **Special Requirements:** | | Store user’s data. | | |
| **Assumptions:** | | User can evaluate his progress and improve his activity. | | |
| **Notes and Issues:** | | Follow up with the user’s progress. | | |

**13 Reporting food and physical activities**

***13.1 Feature Process Flow / Use Case Model***

***13.2 Use Case(s)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC13.1 | | | |
| **Use Case Name:** | Manage reports | | | |
| **Created By:** | Ola Abdallah | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | |  | | |
| **Description:** | | In this case of use, weekly reports are managed and displayed | | |
| **Trigger:** | | When user click on manage reports button | | |
| **Preconditions:** | | 1. User Login to the system 2. Calculate the remaining vitamins, view the burnt energy. | | |
| **Post conditions:** | | Display the weekly reports. | | |
| **Normal Flow:** | | 1. calculate the burnt energy and vitamins and calories in food. 2. Make a report about the result. | | |
| **Alternative Flows:** | | 2.1 report calories amount.  2.2 report vitamins amount.  2.3 report the burnt energy. | | |
| **Exceptions:** | | Ignore the calories report if doesn’t exist.  Ignore the vitamins report if doesn’t exist.  Ignore the burnt energy report if doesn’t exist. | | |
| **Includes:** | | UC3.1, UC4.1 | | |
| **Frequency of Use:** | | 10000 per day. | | |
| **Special Requirements:** | | Evaluate the user’s daily routine. | | |
| **Assumptions:** | | Recommend user to improve his daily routine. | | |
| **Notes and Issues:** | | Follow up with the user’s progress. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC13.2 | | | |
| **Use Case Name:** | Display weekly reports | | | |
| **Created By:** | Nourhan Mahmoud | | **Last Updated By:** |  |
| **Date Created:** | Dec 9th, 2020 | | **Last Revision Date:** |  |
| **Actors:** | |  | | |
| **Description:** | | In this case, weekly reports are displayed on the number of calories, vitamins, and activities performed during the week | | |
| **Trigger:** | | When the user navigates to “manage reports page”, click “Show week report” | | |
| **Preconditions:** | |  | | |
| **Post conditions:** | | Evaluate the report and compare it with the previous reports. | | |
| **Normal Flow:** | | 1. Evaluate report of the current week. 2. Evaluate reports of the previous weeks. 3. Compare it with them. 4. Give the user the final report and recommend him. | | |
| **Alternative Flows:** | | 4.1 report user with the weekly report.  4.2 evaluate the result of this week.  4.3 recommend the user to improve from his weekly routine activities. | | |
| **Exceptions:** | | Ignore the process if there is no activities in the current week. | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | 10000 per day. | | |
| **Special Requirements:** | | Evaluate the user’s weekly routine. | | |
| **Assumptions:** | | Recommend user to improve his routine. | | |
| **Notes and Issues:** | | Follow up with the user’s progress. | | |