Nutritional Calculator

**Iteration 3 | 04/15/2018**

Hector Garza

Alexander Kermani

Songyang Li

Ryan Lira

Thiphavanh Phoutthivongsack

Parijat Singh

Contents

1. Project Description
2. Organization/Responsibilities 3
3. List of Requirements 3
4. Iteration Plan 3
5. Domain Model
   1. Brainstorming 4
   2. Diagram 4
6. Architectural Design
7. Use Case Model
   1. High Level Use Case 5
   2. Use Case Diagram 6
   3. Extended Use Cases 7
   4. Traceability Matrix 11
8. Sequence Diagrams 12
9. Class Diagram 16
10. Scenario Tables/Testing 17
11. GUI for Iteration 3 18
12. **Project Description**

A nutritional calculator to check daily calorie intake, manage a user’s diet, and notify the user when they are failing/succeeding at reaching their calorie goals

1. **Organization/Responsibilities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tasks** | **Reqs** | **Responsible** | **Iteration 1**  2/23/2018 | **Iteration 2**  3/23/2018 | **Iteration 3**  4/15/2018 |
| R3 Food Controller |  | Thiphavanh | **X** |  |  |
| R3 Food Item |  | Songyang | **X** |  |  |
| R3 Food Entry |  | Songyang | **X** |  |  |
| R2 Goal Controller | R1 | ­­Ryan | **X** |  |  |
| R2 Goal | R1 | Alexander | **X** |  |  |
| R2 Calculate | R1 | Parijat | **X** |  |  |
| User View |  | Hector |  | **X** |  |
| R2 User |  | Hector |  | **X** | **R** |
| R2 Database Manager | R2,R3 | Thiphavanh |  | **X** | **R** |
| R2 Login View |  | Parijat |  | **X** | **R** |
| R2 Save/Load | R2,R3 | Parijat |  | **X** | **R** |
| R2 Add/Delete Items | R3 | Ryan |  | **X** |  |
| GUI |  | Alexander |  |  | **X** |
| Testing |  | Ryan |  |  | **X** |

**X** – finished task

**R** – revised elements

|  |  |  |
| --- | --- | --- |
| **Name** | **Responsibility** | **Meet-up Day** |
| *Hector Garza* | Sequence diagram designer | Wednesday |
| *Alexander Kermani* | Organizer for meetings / Plan and requirements doc | Tuesday |
| *Songyang Li* | Plan and requirements doc / Coder | Wednesday |
| *Ryan Lira* | Class diagram designer / Coder | Tuesday |
| *Thiphavanh Phoutthivongsack* | Sequence diagram designer / Class diagram designer | Tuesday/ Wednesday |
| *Parijat Singh* | Use case diagram designer / Coder | Tuesday/ Wednesday |

1. **List of Requirements for SUD**

* R1: The system will allow the user to **create a new account**.
  + R1.1: The system will allow the user to **log into their account** and view their previously added entries and their goal
  + R1.2: The user will also be able to **log out of their account**.
  + R1.3: The system will allow the user to **save account details**, such as the previous items and goals of the account.
  + R1.4: The system will allow the user to **load the last account details** saved.
* R2: The system will allow users to **add a new goal**.
  + R2.1: Setting a new goal will **calculate** **the recommended caloric intake** (how many calories the user should aim for daily).
  + R2.2: The user can **edit their goal**, which will make the old goal deprecated.
* R3: The user will be able to **add a new food entry** to their food tracker.
  + R3.1: The user can **search for a food item** from the database to make a new entry they wish to add to the food tracker.
  + R3.2: The user can **add a new food item to the database** and that will be used to create a new food entry in the food tracker
  + R3.3: The user will be able to **delete food entries** from their food tracker.

1. **Iteration Plan**

* 1st iteration (February 23): we will allow the user to add items and calculate the daily nutritional requirements to reach their goals.
* 2nd iteration (March 23): we will add a database that users can add items from, instantly including all of the nutritional data from it.
* 3rd iteration (April 15): we will add an interactive GUI to the system, as well as automated testing of program features

1. **Domain Model - Brainstorming**

* Food
  + Food Entries - When did they eat it/how much
  + Food Items – What food is it
* Goal
* User
  + Login/save data in database

1. **Domain Model – Classification**

A close up of a map

Description generated with high confidence

1. **System Architecture Design Diagram**

A close up of a piece of paper

Description generated with very high confidence

1. **Use Case Model – High Level Use Case**

* Use Case 1: **Sign Up**

TUCBW User selects the “Sign Up” option on the login view

TUCEW User seeing confirmation/decline of registration

* Use Case 2: **Log In**

TUCBW User selects the “Log in” option on the login view

TUCEW User seeing the website main page or an error if the login was unsuccessful

* Use Case 3: **Log Out**

TUCBW User selects the “Log Out” option

TUCEW User sees the login view

* Use Case 4: **Manage Goal**

TUCBW User selects the “Manage Goal” option

TUCEW User can see their goal and daily caloric intake recommendation

* Use Case 5: **Add Food Entry**

TUCBW User selects the “Add Food Entry” option

TUCEW User can see a new entry in the food tracker with an item name, quantity, and calories

* Use Case 6: **Delete Food Entry**

TUCBW User selects the “Delete Food Entry” option

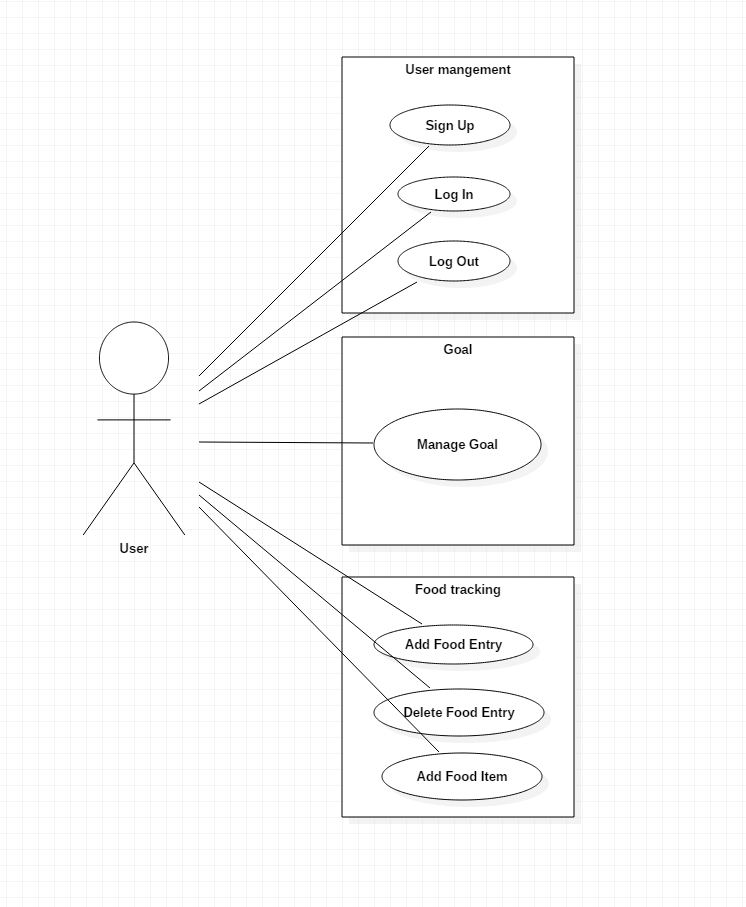
TUCEW User can see an updated entry in the food tracker (name, quantity, and calories)

* Use Case 7: **Add Food Item**

TUCBW User selects the “Add Food Item” option

TUCEW User sees confirmation that the food item has been added to the database

1. **Use Case Model – Diagram**



1. **Use Case Model – Extended Use Case**

Use Case 1: **Sign up**

|  |  |
| --- | --- |
|  | 0. User sees the Login view. |
| 1. TUCBW User selects the “Sign Up” option on the login view | 2. The login view displays the fields for username and password. |
| 3. User inputs their username into the fields password, and selects the confirm option | 4. the login view passes the user’s information to the system to either:  a. create the user’s account in the database and notify the user of the success  b. notify the user an account with that username already exists |
| 5. TUCEW User seeing confirmation/decline of registration |  |

Use Case 2: **Log In**

|  |  |
| --- | --- |
|  | 0. User sees the Login view |
| 1. TUCBW User selects the “Log in” option on the login view | 2. The login view displays the fields for username and password. |
| 3. User inputs their username/password into their respective fields and clicks “Log In” option | 2. The account details are checked against the accounts in the database, and the website GUI displays either:  a. The main website  b. Incorrect login info error |
| 3. TUCEW User sees either:  a. The website main page.  b. “Wrong username/password” if login was unsuccessful |  |

Use Case 3: **Log Out**

|  |  |
| --- | --- |
|  | 0. User sees the user/main view. |
| 1. TUCBW User selects the “Log Out” button | 2. The website GUI prompts if the user is sure if they want to log out |
| 3. The user clicks the “Yes” option | 4. The system logs out the user and displays the login page |
| 5. TUCEW User sees the login view |  |

Use Case 4: **Manage Goal**

|  |  |
| --- | --- |
|  | 0. User sees the user/main view. |
| 1. TUCBW User selects the “Manage goal” option. | 2. The user view shows input fields for the goal calorie count and goal duration |
| 3.User enters the calorie count and the time they wish to achieve it | 4. The user view displays that a goal has been set and displays the recommended daily caloric intake |
| 5. TUCEW User can see their goal and daily caloric intake recommendation |  |

Use Case 5: **Add Food Entry**

|  |  |
| --- | --- |
|  | 0. User sees the user/main view. |
| 1. TUCBW User selects the “Add Food Entry” option | 2. The user view displays all available food items and a field for the user to select one |
| 3. User enters the food item they wish to add to the calculator | 4. The user view prompts the user for the quantity of the item to add |
| 6. User inputs the quantity into the field | 7. The user view displays options to either confirm the entry addition or decline the entry addition. |
| 5. User either:  a. Confirms the addition of the entry  b. Declines the addition of the entry | 6. The system adds the entry to the database and updates the food tracker table with the new entry |
| 7. TUCEW User can see a new entry in the food tracker with an item name, quantity, and calories |  |

Use Case 6: **Delete Food Entry**

|  |  |
| --- | --- |
|  | 0. User sees the user/main view |
| 1. TUCBW User selects the “Delete Food Entry” option | 2. The user view displays a list of food entries they have entered and a field for the user to select one |
| 3. User enters the food entry the wish to delete from the calculator | 4. The user view displays options to either confirm the entry deletion or decline the entry deletion. |
| 5. User either:  a. Confirms the deletion of the entry  b. Declines the deletion of the entry | 6. The system deletes the entry to the database and updates the food tracker table |
| 7. TUCEW User can see their food tracker without the entry they wish deleted |  |

Use Case 7: **Add Food Item**

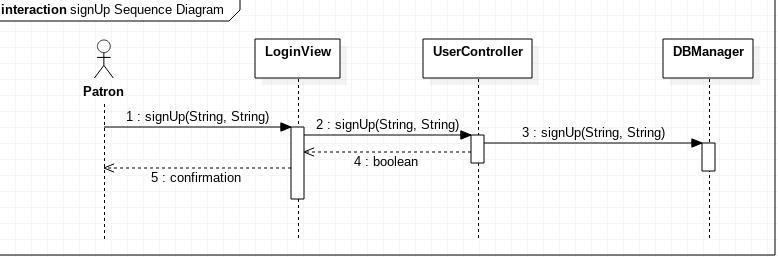
|  |  |
| --- | --- |
|  | 0. User sees the user/main view |
| 1. TUCBW User selects the “Add new food item” option | 2. The user view displays input fields for food item name and calories |
| 3. The user types the name and calories into the fields | 4. The System adds a new food item it to the database |
| 5. TUCEW User sees confirmation that the food item has been added to the database |  |

1. **Use Case Model – Traceability Matrix**

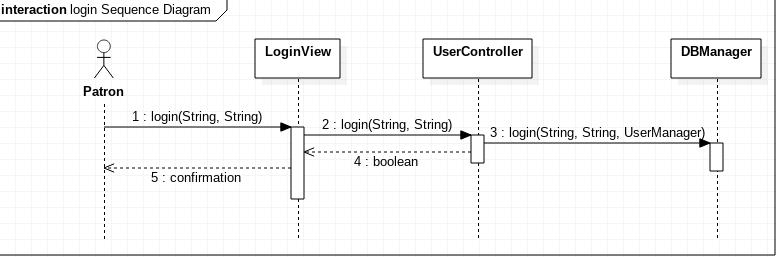
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Weight | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 |
| R1 | 3 | X | X | X |  |  |  |  |
| R2 | 1 |  |  |  | X |  |  |  |
| R3 | 2 |  |  |  |  | X | X | X |
|  | Score | 3 | 3 | 3 | 1 | 2 | 2 | 2 |

1. **Use Case Model – Sequence Diagrams**

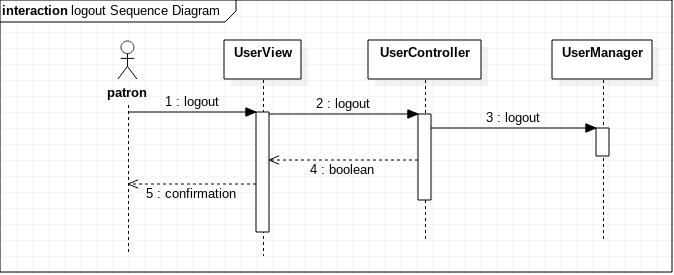
* Use Case 1: **Sign Up**



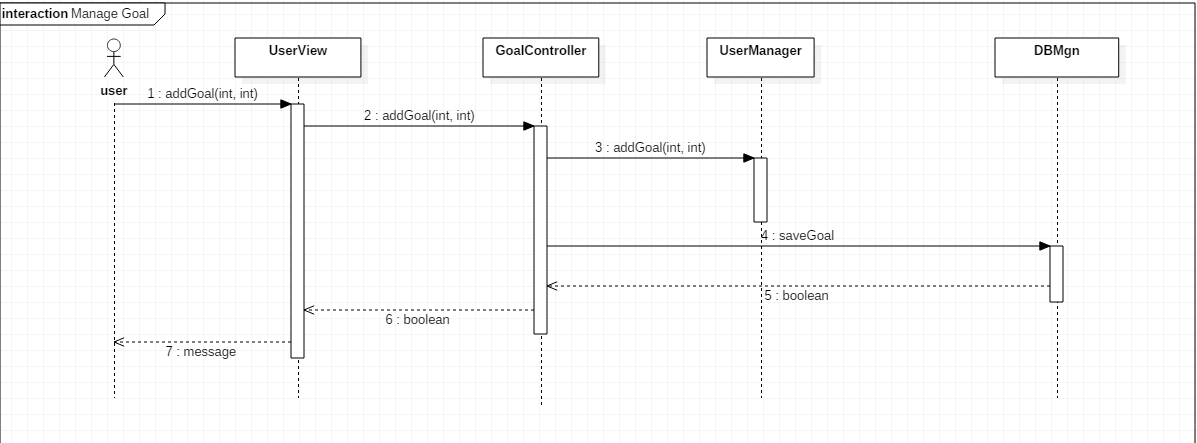
* Use Case 2: **Log In**



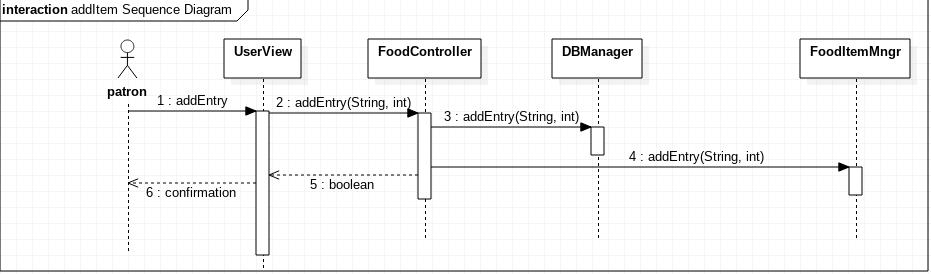
* Use Case 3: **Log Out**



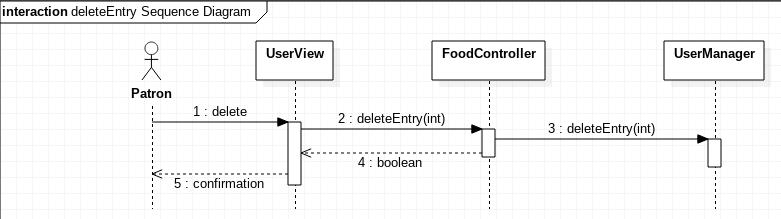
* Use Case 4: **Manage Goal**



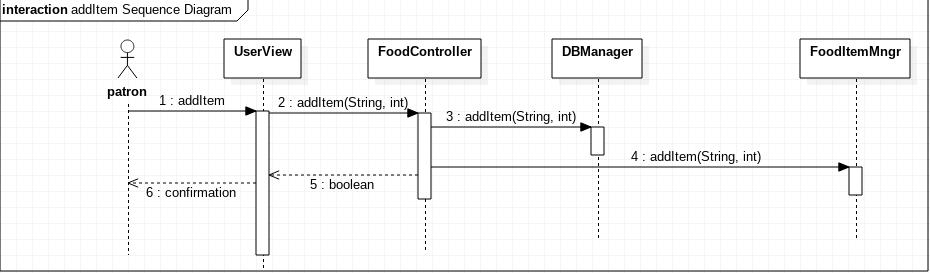
* Use Case 5: **Add Food Entry**



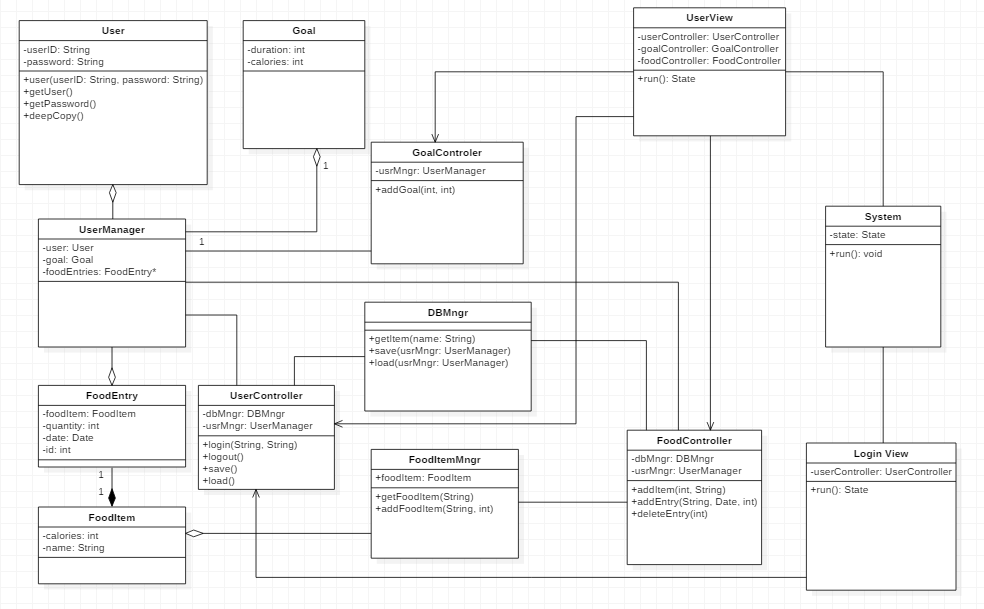
* Use Case 6: **Delete Food Entry**



* Use Case 7: **Add Food Item**



**11. Design Class Diagram**



**12. Testing**

* Use Case 1: **Sign Up**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Username | Password | Exp. Result |
| TC1 | Successful sign up | V | V | Sent to login page |
| TC2 | Invalid username | I | N/A | Error message; asked for username again |
| TC3 | Invalid password | V | I | Error message; asked for password again |

* Use Case 2: **Log In**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Username | Password | User Matches | Exp. Result |
| TC1 | Successful log in; User exists in DB | V | V | T | Sent to user page |
| TC2 | Unsuccessful log in; User not found | V | V | F | Error message; Sent to login page |

Won’t send an error even if it is not possible for a username/password to match (contains whitespaces); will instead attempt to match and fail.

* Use Case 3: **Log Out**

|  |  |  |
| --- | --- | --- |
| Test Case ID | Scenario | Exp. Result |
| TC1 | Successful log out | Sent to login page |

* Use Case 4: **Manage Goal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Day | Calories | Exp. Result |
| TC1 | Successful goal replace | V | V | Sent to user page |
| TC2 | Invalid number of days | I | N/A | Error message; asked for number of days again |
| TC3 | Invalid number of calories | V | I | Error message; asked for number of calories again |

* Use Case 5: **Add Food Entry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Food Item | Quantity | Exp. Result |
| TC1 | Successful food entry | V | V | Sent to user page |
| TC2 | Food item not found | I | N/A | Error message; asked for food item again |
| TC3 | Invalid quantity | V | I | Error message; asked for quantity again |

* Use Case 6: **Delete Food Entry**
* Use Case 7: **Add Food Item**

**13. GUI for Iteration 3**

