**Calorie Counter App**

**Software Requirements Specification**

**Version #2.0**

**Team Number: 6**

**Project Manager: Andrea Gaietto**

**Mentor: Lisa Schenkewitz**

**Team Members:**

* Andrea Gaietto
* Andrew Jenkins
* Victor Lameda Rojas
* Jose Reyes
* Zachary Walker
* Larry Webb
* Xavier Roca Hernandez

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| 1.0 | Andrea Gaietto, Andrew Jenkins, Victor Lameda Rojas, Jose Reyes, Zachary Walker, Larry Webb, Xavier Roca Hernandez | First draft | 10/16/2022 |
| 1.1 | Andrea Gaietto, Andrew Jenkins, Victor Lameda Rojas, Jose Reyes, Zachary Walker, Larry Webb, Xavier Roca Hernandez | Post mentor review adjustments | 10/22/2022 |
| 2.0 | Andrea Gaietto, Andrew Jenkins, Victor Lameda Rojas, Jose Reyes, Zachary Walker, Larry Webb, Xavier Roca Hernandez | First draft of high level design | 10/30/2022 |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
| Lisa Schenkewitz | 1.0 | 10/19/2022 |
|  |  |  |

**Table of Contents (version 1)**

1. Introduction
   1. Project Objectives
   2. Project Scope
   3. Project Overview
2. General Description
   1. Project Features / Functions
   2. User Stories
   3. Use Cases
3. Team Collaboration and Documentation Tools
4. Project Management Plan
5. Requirements Specification
   1. Business Requirements
   2. User Requirements
   3. Functional Requirements
   4. Non-Functional Requirements
   5. Implementation (Performance) Requirements (Optional)
6. High-level Design

6.1. Security (Required)

6.2. Hardware (Required)

6.3. User Interface (Required)

6.4. Architecture (Required)

6.5. Database (Required)

6.6. Top-level Classes (Required)

6.7. Data Flow and States (Required)

6.8. Reports (Required)

1. **Introduction**
   1. Project Objectives

## This project will create a Calorie Counter Application.

* 1. Project Scope

The scope of this project is to create a simple Calorie Counter Application that allows users to perform the following actions:

* Calculate calories needed for weight loss
* Input calories burned
* Input food and amount
* Input weight
* Login/account tracking/profile creation
* View past entries
* View past weight
* View total calories for the day
  1. Project Overview

This project will create a Calorie Counter Application. Users will be able to input food and calculate calories, as well as track weight loss progress.

1. **General Description**
   1. Project Features / Functions

* Calculate calories needed for weight loss
* Input calories burned
* Input food and amount
* Input weight
* Login/account tracking/profile creation
* View past entries
* View past weight
* View total calories for the day
  1. User Stories

As a user

I want to calculate my BMR (Basal Metabolic Rate)

So that I know how many calories I need to lose, maintain or gain weight.

As a user

I want to input the food and the amount that I am eating.

So that I can be sure that I am keeping within my allotted calories for the day based on my height, weight, age, gender.

As a user

I want to be able to view my calories eaten previously

So that I can compare entries on a weekly basis to make predictions or to track my progress.

As a user

I want to see my measurements like waist, arm, chest, thigh, hips

So that I can check my fat loss progress.

As a Customer

I want to view my total daily calories intake

So that I can make sure I am not going over my daily target

* 1. Use Cases

**Enter Food and Amount**

**ID:** 1

**Brief Description:** Enter type of food eaten and the amount of food.

**Primary Actors:** User who is a calorie counter

**Secondary Actors:** none

**Preconditions:**

* The user is logged into the dieting app

**Main flow:**

* On the food entry page, the actor will enter the type of food eaten in the food input box and then the amount eaten in the amount input box and will enter type of amount by selecting the appropriate amount from a pull-down list (ounces, grams, tablespoons, etc.)
* As the user inputs the food type and amount, the application retrieves total calories for the day and calculates calories for the food type and amount just added. It then adds them together and displays the new total calories consumed for the day.
* The system will store this information to use in other parts of the application such as calculating total calories consumed for the day, tracking water consumption, etc.

**Post conditions:**

* Information stored in application’s database.

**Alternative flows:**

* If a food type has been previously entered, the application remembers and will attempt to auto fill the food type for the user

**Calculate BMI**

**ID:** 2

**Brief Description:** User wants to calculate their BMI. This calculation is done as follows: BMI = (w + h2) \* 703.

**Actors:** Application user

**Secondary Actors:** None

**Preconditions:**

* The user is logged into the dieting app.

**Main Flow:**

* On the BMI calculation page, the actor will enter their weight in the weight input box and then their height in the height input box.
* The user will select the calculate button and the application will calculate through the BMI algorithm.
* The system will store this information to use in other parts of the application.

**Post Conditions:**

* Information stored in the application’s database

**Alternative Flows:**

* None

**User Profile**

**ID :** 3

**Brief description:** User will create a profile with their personal information

**Actors:** User

**Secondary Actors:** None

**Preconditions:**

* The user is logged into the dieting app

**Main Flow:**

* User goes to profile page
* User enters name, age, height, current weight, email, user ID, and password in the input fields provided

**Post Conditions:**

* The information is stored in the application database

**Alternative Flows:**

* If the password entered is a different password from the one already stored in the database, it triggers an email to the user from which they must verify their account.

1. **Team Collaboration and Documentation Tools**

Team Collaboration will take place using Microsoft Teams. Team Documentation and version control will take place in GitHub.

1. **Project Management Plan** (optional)

Team PM is using a Gantt chart for project tracking.

1. **Requirements Specification**
   1. **Business Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| BR1 | The software must have the ability to allow for different types of food to be entered. | M |
| BR2 | The software must have the ability to add the number of calories that are burned by the user. | M |
| BR3 | The software should have the ability to calculate what times are needed for weight loss | S |
| BR4 | The software should have the ability to view the total calories in a day. | S |
| BR5 | The software could have the ability to track the intake of water. | C |
| BR6 | The software should have the ability to show the past entries for food and calories. | S |

* 1. **User Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| UR1 | The Login Screen takes user information, such as a profile name and password, to verify the user's identity | M |
| UR2 | The Main Screen allows users a choice of inputting daily tracking or viewing past inputs | M |
| UR3 | The daily tracking system keeps count of how many calories the user has consumed, and how much they have burned, as inputted by the user. | M |
| UR4 | The daily tracking system should also be able to calculate how many calories an inputed meal would be worth by comparing food items to a database | S |
| UR5 | A tracking system should hold on to daily entries, showing the user progress in a long term diet plan | S |

* 1. **Functional Requirements**

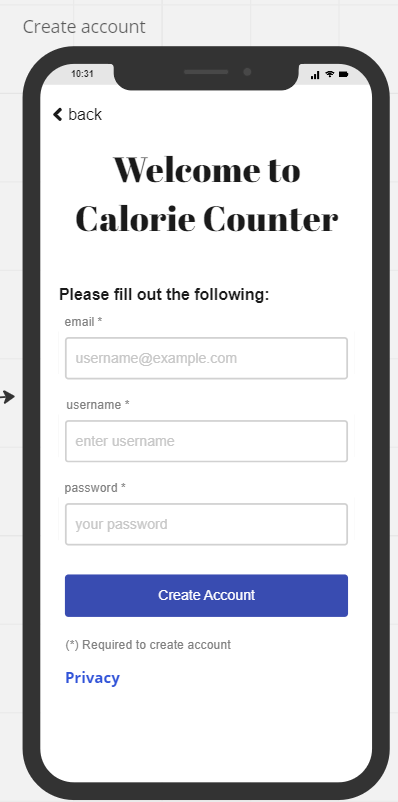
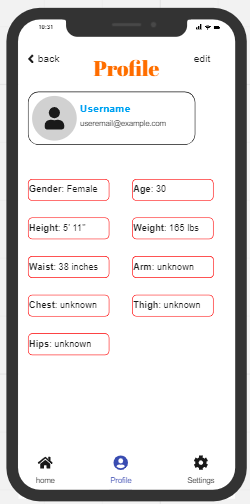
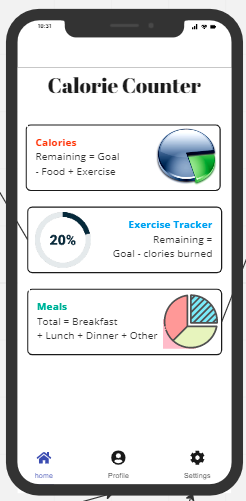
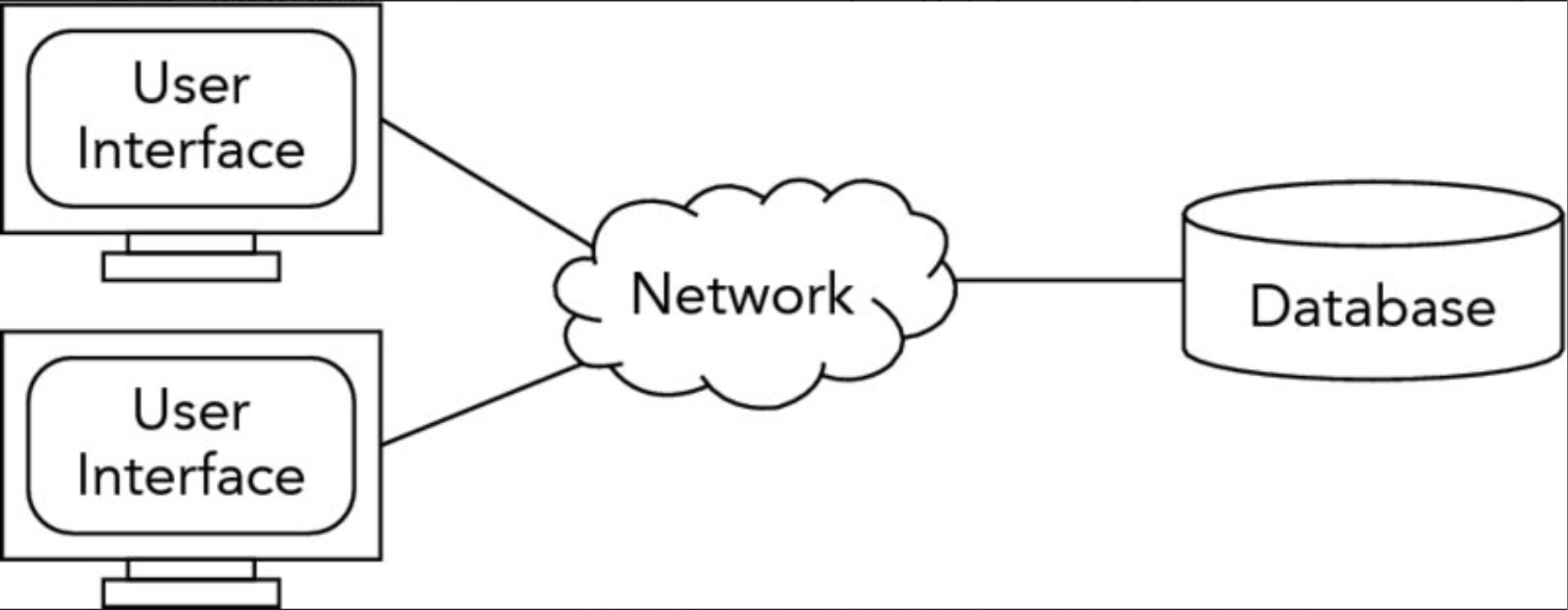
| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| FR01 | User can log in, sign up for an account on the app | M |
| FR02 | User uses E-mail or phone number to sign up + password | M |
| FR03 | Protect from duplicate accounts | M |
| FR04 | Make sure account is protected by two factor identification methods. (Login credentials + phone text verification code) | S |
| FR05 | Verify user creates strong password (letters, numbers, and special characters) | M |
| FR06 | Make sure user’s data is saved so they can check their calories intake from days before. | S |
| FR07 | Give user ability to reset password using an email link | S |
| FR08 | Provide user with an account profile to add name, gender, age, height, and weight. | M |
| FR09 | Give user the ability to update account’s profile and save changes. | M |
| FR10 | Application will save all data in a Database | M |
| FR11 | Give user “Calculate my BMR” option in the app. App will use user’s age, gender, height, and weight to do the calculation. The output will return the number of daily calories the user needs to consume. | M |
| FR12 | App lets user input the food and amount to calculate current number of calories consumed | M |
| FR13 | App will provide user with the ability of tracking the number of calories eaten previously | M |
| FR14 | App can add and edit waist, arm, chest, thigh, and hips measurement. | M |
| FR15 | App can track old measurements and compare to new measurements and return number of inches gained or lost. | M |
| FR16 | App provides user with the ability to see the total number of calories consumed. | M |
| FR17 | App lets user set up a calorie intake goal | S |
| FR18 | App notifies user when a certain number of calories have been consumed | S |
| FR19 | App lets user add number of calories burned due to exercise | C |
| FR20 | App calculates the total number of calories burned using all previous data. | C |
| FR21 | There should be regularly scheduled updates to maintain bug issues | M |
| FR22 | The application should notify users of large updates to the application one hour in advance | S |

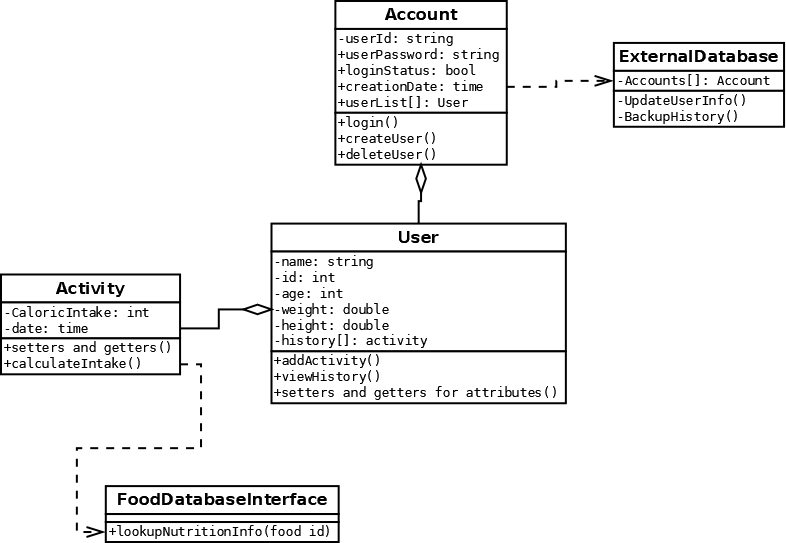
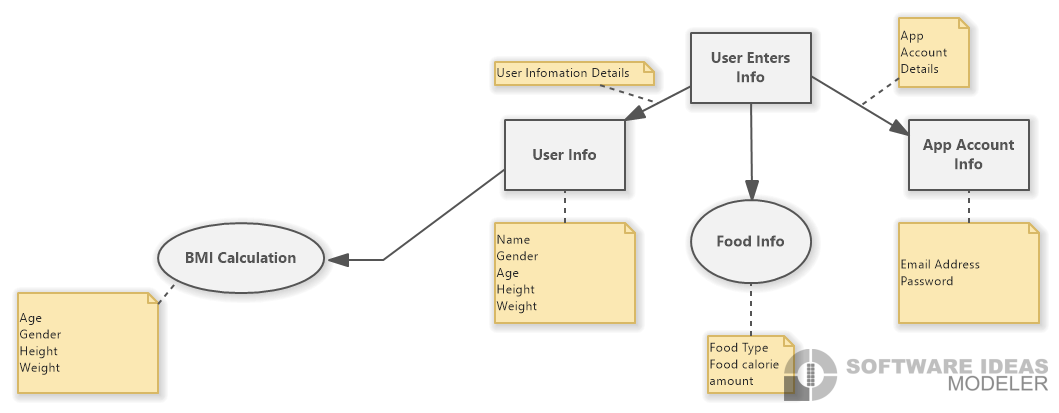
* 1. **Non-Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| NR1 | Thirty concurrent users should be able to access the food calorie database with no noticeable system delay | M |
| NR2 | Notify the user if incorrect login credentials have been used | S |
| NR3 | The application should be compatible with both android and iOS devices | S |
| NR4 | In the event of system failure, the system should not be down for more than 15 minutes | S |
| NR5 | Each page within the application should load in 2 seconds or less | S |
| NR6 | Verification emails should be sent to user’s email within 2 minutes after initial sign up | S |
| NR7 | Application shall encrypt user’s passwords | M |
| NR8 | Notification and reset password emails shall be sent to the user within 2 minutes of reset password request | S |
| NR9 | Calculate my BMR should calculate the user’s BMR within 5 seconds | S |
| NR10 | The application should have a 99.999% uptime | S |
| NR11 | Large updates should be implemented during slow use hours of the application with the exception of fixing major bugs | S |

* 1. **Implementation (Performance) Requirements (Optional)**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| IR1 | Any existing user data will be maintained in a separate database until successful migration has been confirmed. | M |
| IR2 | User education materials should be prepared and available by product launch. | S |

1. **High Level Design**
   1. **Security**
      1. Application
         1. Users must create an account in order to use the application.
         2. The account must use a unique username or email.
         3. The password must be at least seven characters, with at least one number, one letter, and one special character.
         4. Application will utilize two factor authentication to perform password resets.
      2. Data
         1. Passwords will be stored in an encrypted format.
         2. Users will have the option to permanently delete or de-identify their data.
         3. Any APIs used will be secured and require authentication.
   2. **Hardware**
      1. Mobile Devices
         1. The application shall be compatible with iOS and Android devices and OS versions released within the last five years.
         2. Application can store data locally, with back ups when connected to a network.
      2. Network
         1. Application shall be able to be used without an active network connection.
         2. Application shall be able to be used while device is connected to wifi, 3G, 4G, etc.
   3. **User Interface**
      1. Link to full set of User Interface screens at Miro: <https://miro.com/app/board/uXjVPIqkmZI=/?share_link_id=188576087118>
      2. Selection of designs:
         1. User Registration
            1. 
         2. User Profile
            1. 
         3. Home Screen
            1. 
         4. Report Export
            1. 
   4. **Architecture**
      1. 
      2. The user interface is the part the user interacts with and could be a phone, laptop, tablet, etc. A relational database will be the server and will contain all the information entered by the user as well as calculations made. The clients will communicate with the server via a LAN. Since there will be multiple users and calculations and we want to keep response time at a good level, we chose the client/server methodology. However, a third tier would probably be overkill for this application, so we went with a simple client/server architecture.
   5. **Database**Diagram, schematic

      Description automatically generated
   6. **Top-level classes**
   7. **Data Flow and States  
      **
   8. **Reports**
      1. **User Reports**
         1. User Progress Report
            1. Users will be able to export report as a csv directly to their device.

Users will be able to choose to export weight, steps, chest, hips, calories burnt, waist, thigh, and arm records, as well as date recorded.

Users will be able to enter a name for the file export.

* + - * 1. Users will also have the option to export the User Progress report as a pdf.

Pdf will display selected metrics and date recorded in chart format.