Software Requirements and Design Document

For

Group 2

Version 2.0

Authors:
Lance F
Carter M
Matthew O
Jayen L
Jacob N

1. Overview (5 points)

The group 2 team would like to design and build an app that provides a simple, yet fulfilling interface for a user to plan, track, and reflect upon their dieting and fitness objectives. This means having a structure that brings together these components in such a way that they each feel intrical to the user's daily routine, and helps them maintain a balanced lifestyle.

2. Functional Requirements (10 points)

Index, Login, Registration:

1. [High] An index.html page will allow the user to either login or register.
2. [High] A user will start as unauthorized. Register will take the user to a register page.
3. [High] A form will allow the user to enter a username, email, password, and confirmed password into fields.
4. [High] Information is checked, and if allowed, will add the user information to the user table of the database and return the user to index.
5. [Medium] If not allowed, it will provide an appropriate error message and allow the user to enter information again.
6. [High] Login will take the user to a login page.
7. [High] A form will allow the user to enter an email and password, where it will check the user database for email and password match.
8. [Medium] If matched, the user is authorized and taken back to the index, and if not, it will indicate an appropriate error, allowing the user to try again.
9. [High] If the user is authorized by the login page, then index will take the user to hub.html instead of index.html.

Hub:

1. [Medium] Upon entering the hub, the user will be at a home page.
2. [High] The bar at the top will allow the user to traverse to a diet component, workouts component, or progression component of the app.
3. [High] The user will be able to click on their username and select a logout option.
4. [High] If the user logs out, they are unauthorized and returned to the index.

Diet:

1. [High] The bar at the top will allow the user to traverse to the Hub component, fitness component, or progression component of the app.
2. [High] If user is logged in a logout button should be visible.
3. [High] If user is not logged in no information should populate table or fill body.
4. [High] Upon entering the Diet.html page any data related to the user in the Diet table will be loaded.
5. [High] Data related to user will be found using user id in diet table.
6. [High] A table will allow user to enter their calorie intake for the day.
7. [High] A calculator will allow user to calculate their bmr.
8. [High] Calorie table will allow user to input their bmr.

Workouts:

1. [High] If the user is logged in and in the hub, a fitness button will be visible for the user to be redirected.
2. [High] Inside the workout component, the user will be able to determine whether they are a beginner, intermediate, or advanced lifter, depending on how many times they go a week.
3. [High] Two different sections will be available depending on if the user has access to a gym/ gym equipment (dumbbells, barbells, bands, etc.) or is working out at home.
4. [Medium] Determine if the user wants to put on muscle (gain strength vs. aesthetics) or lose weight, will bring a different fitness routine based on what the user wants.

5. [High] Once all the questions are asked, a specific custom fitness plan will be provided to the user
6. [High] Rep ranges and suggested weight for each exercises provided.
7. [Low] Table showing weight (weight for the weights) progress to determine if any changes need to happen.
8. [High] if the user isn't logged in, no data will be shown.

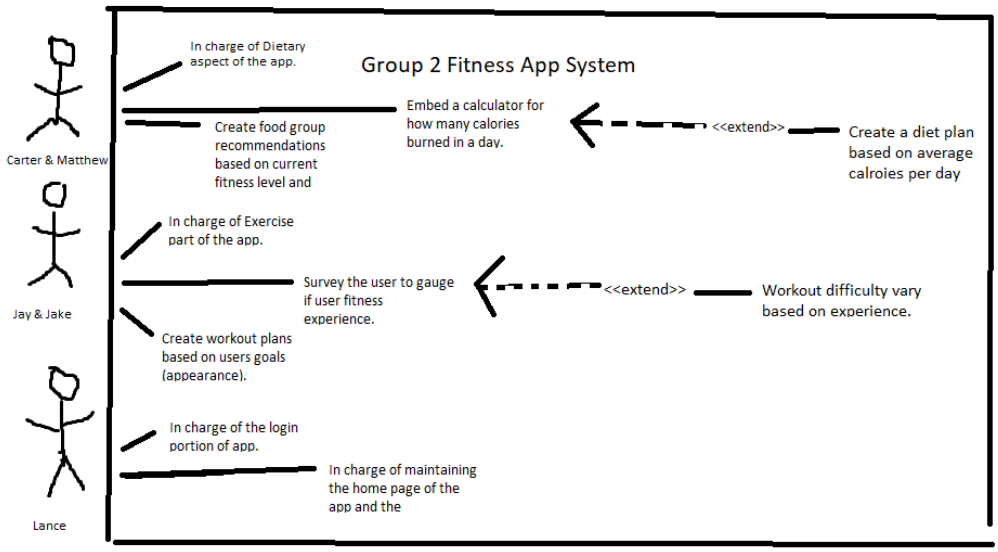
Progression:

1. [Medium] The page will be visually divided into a progress overview, goals, and before/after part.
2. [Low] Progress overview will show statistics of the user's progression, including how many goals were completed, time since registering, and tie-ins to the other components depending on what they may track.
3. [Medium] Goals will allow the user to add, update, delete, and finish goals based on a form where they specify a name, units, current, and goal amount.
4. [Medium] A scroll bar will allow the adding of unlimited goals.
5. [Low] Summary will allow a user to upload a before and after image of themselves and update a date taken and weight amount associated with the image.

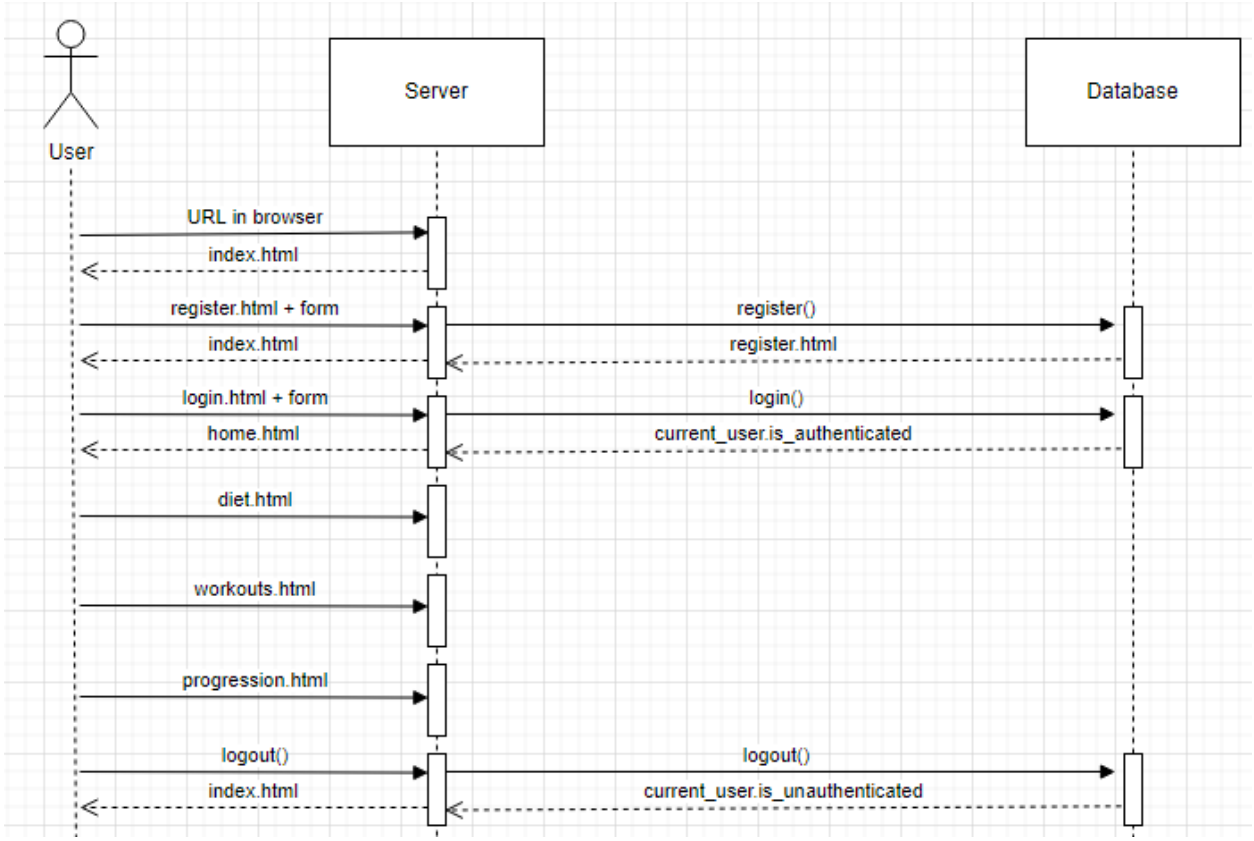
3. Non-functional Requirements (10 points)

User data will be secure to their account. Forms and queries will reliably interact with the database. All user ids generated will be unique.

4. Use Case Diagram (10 points)



5. Class Diagram and/or Sequence Diagrams (15 points)



—

6. Operating Environment (5 points)

The Feelin' Good fitness app is being developed for web-based platforms. It is built and tested using a linux ubuntu virtual machine and through a PC browser, so it is difficult to determine the resulting presentation on ios at this time. The application must be hosted using a variety of python modules, some of which are reliant upon utilization of the linux operating system.

7. Assumptions and Dependencies (5 points)

It is assumed that the program will be demonstrated locally, as there is no hosting being paid for in order to allow access from anywhere. All modules must be installed in the virtual environment in order for the server to run. Other than this, there are no other dependencies at this time.