Software Implementation and Testing Document for Group 8

Version 1.0

Authors:

Ryan Rowe
Spencer Choy
Liam Salem

1. **Programming Languages**
   We use python in the Flask framework to build our backend. This includes retrieving data from the database, manipulating data and providing it to the frontend, and creating graphs and charts using python libraries. We chose python because we are already familiar with building web apps using Flask. We use html, css, and javascript for the frontend portion of our project because we plan to migrate to React, a javascript library.
2. **Platforms, APIs, Databases, and other technologies used**
   For the database, we are using MySQL because it is a relational database system that we have experience with. The website is already deployed on AWS beanstalk, the database on AWS RDS, and we have to set up a new codepipeline to the new repository to ensure future changes show up on our website immediately. To get nutrition information for the foodlogger feature, we will use USDA FoodData Central API, a database from the U.S. Department of Agriculture. This will be necessary for some features we want to implement, like allowing users to have saved meals.
3. **Execution-based Functional Testing**
   We have tested all the basic features (user authentication, CRUD operations on habits & macros, navigating pages, displaying graphs & charts, searching for and sending friend requests to life coaches) by simply using the website. We first tested these features locally and then tested them again when the website was deployed. We will follow the same procedure with features implemented in the future.
4. **Execution-based Non-Functional Testing**
   We have not performed any tests on the non-functional requirements, but we will in the future, and we will make efficiency-based improvements based on the testing.
5. **Non-Execution-based Testing**
   We have not done any code walkthroughs/inspections yet, but we will after each feature is implemented and pushed to the repo.