

# **Software Implementation and Testing Document**

**For**

**Group 11**

Version 1.0

**Authors:**

Isaiah Alex

## **1. Programming Languages (5 points)**

The programming languages used in this project will be python for handling the back end and algorithms. Python was chosen as it is a simple language semantically and seems feasible to use for a web application. HTML and CSS for the front end to handle design layout. These languages are easy to pick up and can handle our needs to make our design of the web application. Lastly, SQL for the database for we are using SQL as our database will be an SQLite database.

## **2. Platforms, APIs, Databases, and other technologies used (5 points)**

For our database we will be using a SQLite database to handle multiple users as well as managing their different roles and their accounts. Users will be able to store their own recipes and keep track of their caloric intake and macronutrient intake as well. We will also be using Recipe Search API which entails a recipe database while also allowing users to be able to search for recipes based off given ingredients. This API also will provide us with the macronutrient breakdown that we will be able to keep track of. We also used Flask to handle the front end and able to link the html pages to the main python application, while also able to execute functions with information entered. We also adapted code from an API, [https://github.com/JarbasAI/py\\_edamam/blob/master/py\\_edamam/\\_init\\_.py](https://github.com/JarbasAI/py_edamam/blob/master/py_edamam/_init_.py), to help us send requests for recipe searches.

## **3. Execution-based Functional Testing**

We were able to test user creation through the usage of the console app. We were able to verify that the database was created with a user account as we were able to close the program and tried logging in again with the credentials we made, and it showed that we successfully logged in. We were also able to test the viewing of recipe detail functionality as we were able to create a file that sends requests to the Edamam API and retrieve back recipes as well as the ingredients for the recipe and the macronutrients of the recipe. We weren't able to test calorie tracking, create recipe, create shopping list, edit recipe, edit shopping list, admin ingredient management, admin recipe management, and delete recipe as it has not been implemented yet.

## **4. Execution-based Non-Functional Testing**

We were able to test the security of our program by viewing the database and it shows that the users account is stored as a hashed password and not the user's password in plain text. In the event that our database gets breached our user's account passwords are safe. The performance of our programs is great, as all our requests are met in milliseconds and there is no wait time for retrieval of information. Data backup has also been tested as the database with user accounts is still stored after the program is exited. The usability of the program has also been tested to ensure that anyone would be able to use it as it is straightforward and easy to use. We were not able to test for. We weren't able to test for scalability and accessibility at this stage of our project but we will in the next increment.

## **5. Non-Execution-based Testing**

Our group met multiple times to run through the code after we coded each part and walked through the code and see how it was running and if it was running as planned. We all also looked through each other's code to ensure that everyone understood what was going on as well as checking to make sure that it is accomplishing what we were expecting.