Software Requirements and Design Document

For

Group <13>

Version 1.0

Authors:
Jacob Siegel
Gam San
Dean Arriveiello
Braden Seidl
Declan Murphy

Overview (5 points)

Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).

An application that gives access to a database of meals and recipes that are categorized by their ingredients, diet, and overall nutrition. The app will also allow users to keep track of what ingredients they currently have and search for recipes based on that information. If a recipe contains ingredients that the user does not have, they will be able to add those ingredients to a 'shopping list' to keep track of. The app will also contain functionality for tracking calories consumed and exercise done on a specific day.

The purpose for this application is to give users ease of access to a database of recipes to choose from. Users will also be able to track what foods/meals they have eaten in a day as well as the amount of calories consumed. This will allow users to plan their diet ahead as well as keep track of how far along they are in their dietary goals, making it easier for everyone to stay healthy.

Functional Requirements (10 points)

*List the **functional requirements** in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. Display current stats for the day (meals planned, fitness data, calories)
    a. Display user's information for the day. This will show what meals they have planned, their current fitness stats and progress towards goals, option to view pantry, and option to view shopping list. This page will be a general overview that will allow the user to enter submenus in order to see specific information or edit goals/stats.
2. List of recipes that updates based on given criteria
    a. Users will be able to search through a list of recipes pulled from the API. User can be given the option to modify their search options based on: Dietary preference, recipes in the pantry, food category, time to prep
3. Update/Display ingredients the user currently owns
    a. This section will be called the recipes. These are the ingredients the user currently has. This list will be used to give the option of searching for meals based on ingredients already owned.
4. Update/Display ingredients in 'recipes'
    a. These will be food items added to a list through two different methods:
        i. User can add ingredients manually (either through searching the spoonacular database or by manually inputting food items)
        ii. Users can add ingredients they are missing from a specific recipe (When they save a recipe, they will be given the option to add missing ingredients to shopping list)
    b. 'Shopping List' will be one of several pages the user may navigate/edit. Other pages will be 'Pantry', 'Recipes', and a general wellness/stats page.
5. Fitness tracking: User calorie count
    a. Display to user how many calories they have consumed so far in a day
    b. Display calories and nutritional information on a specific meal
    c. Allow users to set goals for maximum/minimum calories per day.
6. Meals planned for the day

   a. Show list of all meals planned for the current day. When a meal is selected, allow the user to view ingredients/recipes/information about the meal. Additional ability for users to add personal notes to a meal.

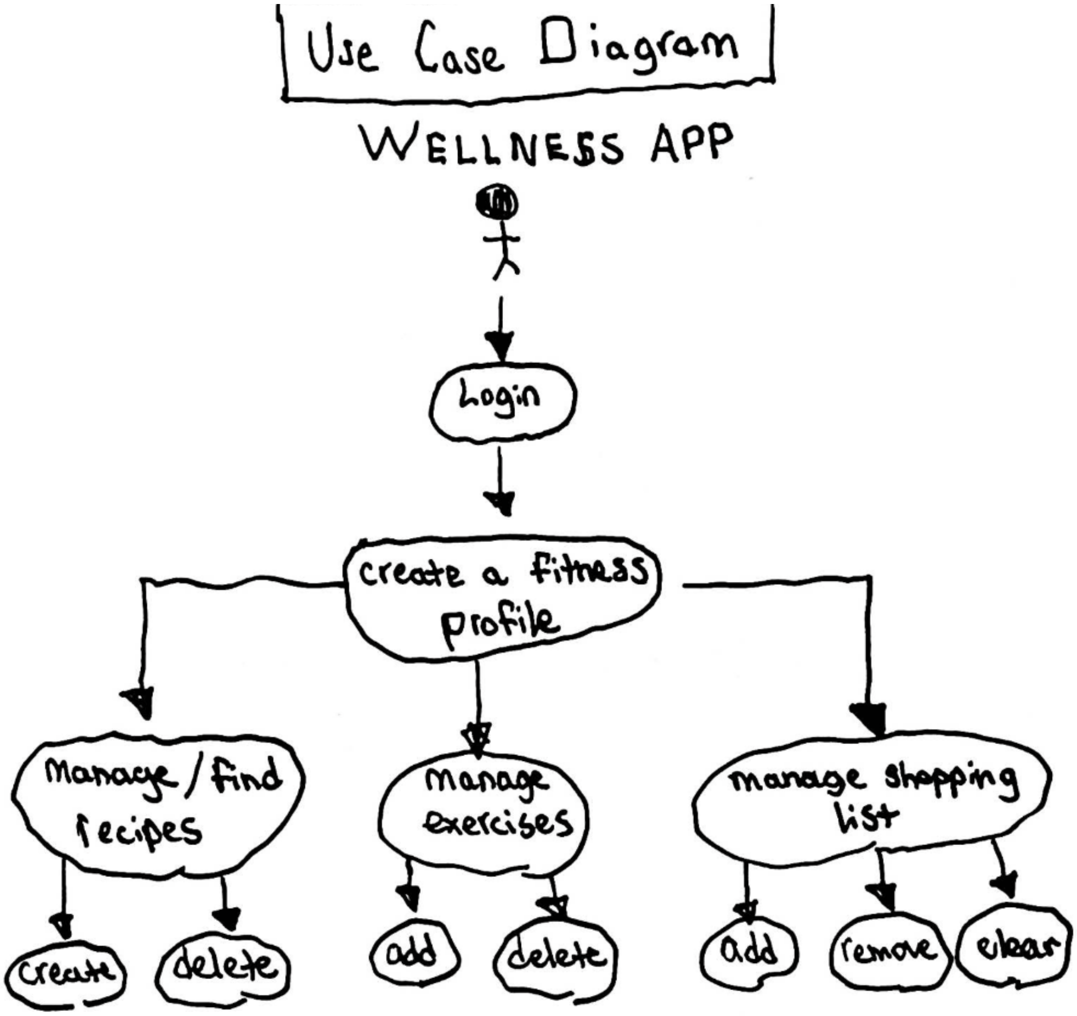
7. Meal/Food/Recipe database
   a. Utilizing the spoonacular API, we will have to develop a UI to display all meals the user can add to their plan. We can base this off of something like https://www.allrecipes.com/ where the user searches for a general term and will be displayed results related to it. This can be an ingredient or the name of a specific meal.
8. Personal meal recommendations
   a. Users can add their interests/dietary preferences and will be displayed recommended recipes based on this data.

Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

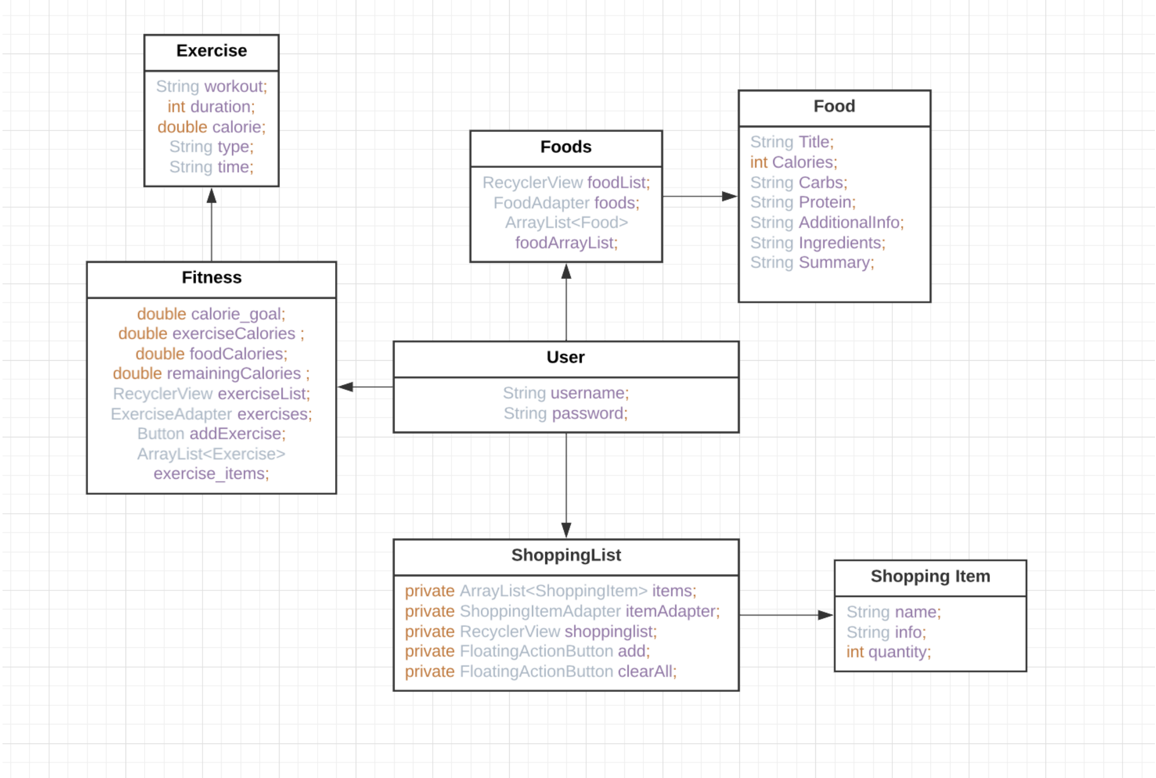
1. Add functionality to parse food database
2. Data persistence (Firebase)
3. Implement web service to allow for API calls
4. Import spoonacular SDK into android studio
5. User Authentication
   a. Allow for an account to be created for data persistence across devices
   b. Create a new account for new users

Use Case Diagram (10 points)



Class Diagram and/or Sequence Diagrams (15 points)

Very rudimentary class diagram, will be more developed and implemented in increment 2.



Operating Environment (5 points)

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

Android Lollipop 5.1 Minimum
Virtual Android Device (Pixel 6)

Assumptions and Dependencies (5 points)

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

- We are assuming we are able to access the fitness metrics on an android phone however since we are operating under a virtual phone at the moment this may lead to a feature problem
- We are assuming we will be able to implement objects/data from the spoonacular API into a form tangible to the user. This is limited to a certain amount of calls to the API, if we go over the amount of calls, they charge a fee

- We are currently assuming that we will not run into a limit for API calls to spoonacular once we have multiple users on the platform.
- We are assuming due to differing tasks between each member in the group, the app would eventually be able to integrate one feature with another although it might have a differing design pattern.
- We are assuming there won’t be a problem when integrating databases when we interact with data that is stored on cloud vs the data that is stored locally on the device.