Software Requirements and Design Document

For

Group 11

Version 2.0

Authors:
Eli Bendavid
Charlie Penner
Gordon Leadbetter
Isaiah Alex

1. Overview (5 points)

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

The proposed system is a food and nutrition management application designed to assist users with tracking their daily calorie intake, creating, and managing personalized recipes, and efficiently generating shopping lists. Users will be prompted to select their goals upon account creation. The system will offer users the ability to maintain a record of their daily calorie consumption to monitor their progress towards their dietary goals.

Users will have the ability to create, modify and delete recopies. This provides a flexible and customizable was to manage their culinary preferences. Users will also be granted the option to add the necessary ingredients for a recipe to a shopping list automatically. Admins will be able to manage ingredients within the app through a database. We hope to accomplish a simpler way to meal plan and adopt healthier eating habits.

2. 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. *User Registration (High Priority): The system will allow users to register to an account. Users will be prompted to fill in an email address, a username, and a password. Registration for an account will also collect valuable information, such as weight goal and calorie-per-day goal. This requirement is high priority because it contains all the information that the other requirements depend on.*
2. *Calorie Tracking (High Priority): Users will have the ability to track their daily calorie intake. Calorie tracking will allow users to track their breakfast, lunch, dinner, and snacks. When meals are logged, the system will subtract the calories ate from the total calorie balance and display the number of calories the user has left for the day. This is a high priority requirement as it is the main point of the system.*
3. *Create Recipe (High Priority): Users will have the ability to create new recipes. They will be able to name their recipes, give them a description, and list the ingredients that are required. Optionally, the user will also be able to add instructions for cooking the recipe. This is a high priority requirement as it is required to implement the shopping list requirement.*
4. *Edit Recipe (Medium Priority): Users will have the ability to edit existing recipes, whether they be ones that they have created, or their takes on existing recipes in the database. They can change the ingredients, quantities, or the instructions. This is a medium priority requirement as it helps users personalize their experience with the system.*
5. *View Recipe Details (Medium Priority): Users will be able to view detailed information about a recipe. This includes ingredients, quantities, and instructions. This is a medium priority as it is important for users to view a recipes information in terms of calorie tracking.*
6. *Admin Ingredient Management (Low Priority): Admin users will be able to add, edit, or delete ingredients in the ingredient database. This is a low priority as many ingredients will be hardcoded into the database and is not a necessary functional requirement.*
7. *Admin Recipe Management (Low Priority): Like the Admin Ingredient Management, this functional requirement will allow admin users to add, edit, or delete recipes in the recipe database. This is low priority because most recipes will be created by users or hardcoded into the database.*

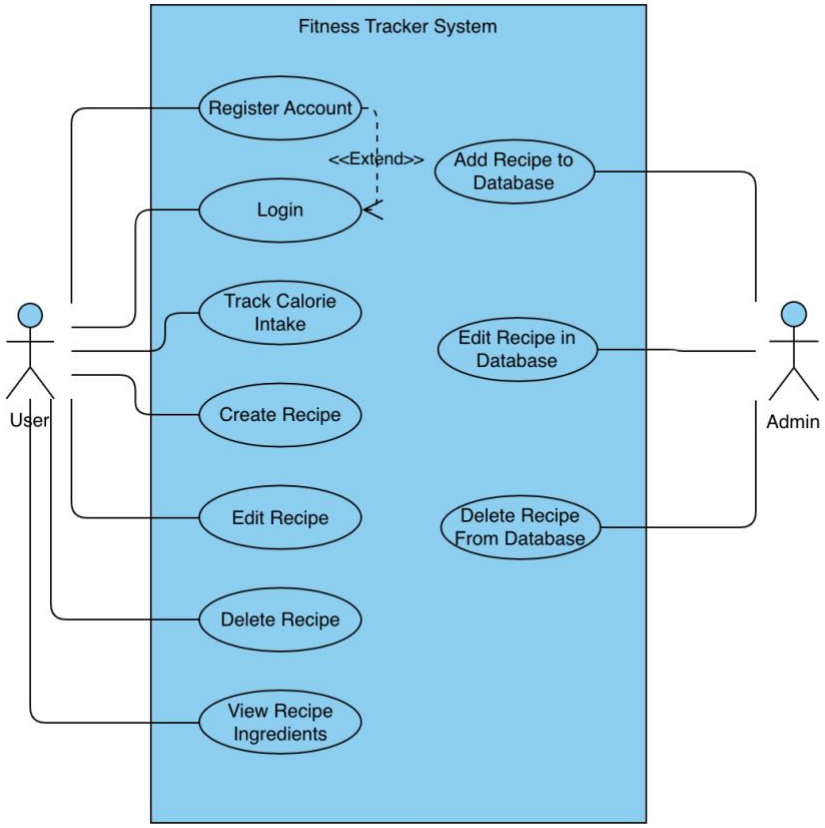
8. *Delete Recipe (Low Priority): Users will have the ability to delete recipes from their collection. This is low priority because the program is still functional without giving the user the ability to delete a recipe. In other words, it is a quality-of-life feature.*

3. 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) *Security (High Priority): The system will ensure user data privacy and security. Passwords and other sensitive information will be encrypted. This is high priority to protect user data.*
2) *Performance (High Priority): The system has a fast response time. It must respond to user interactions promptly. This is essential for a positive user experience; thus, it is high priority.*
3) *Reliability (High Priority): The system must be available with no issues 24/7. Maintenance downtime must be minimal. This is crucial for a system like this, as users need to log in daily to track their calories.*
4) *Data Backup (Medium Priority): Data backup is extremely important in the situation where data loss occurs. This system relies on user inputted data, such as custom recipes and calorie goals. Data Backup is medium priority to mitigate data loss risk.*
5) *Software Quality (Medium Priority): The system will follow good coding practices. Regular testing must be conducted to ensure high software quality. This is medium priority to ensure a well-built system.*
6) *Usability (Medium Priority): The system will have a user-friendly interface. This will maximize user experience. The interface must be simple and straightforward to use (even for people who are not good with technology). Medium priority because front end will be simpler to implement later.*
7) *Scalability (Low Priority): Because this is a capstone project and not many people will access the system, scalability is low priority. Scalability refers to allowing the system to handle an increasing number of users.*
8) *Accessibility (Low Priority): Accessibility refers to having the system accessible to people with users that have disabilities. This is low priority as most users won't be impacted by it not being included.*

4. Use Case Diagram (10 points)

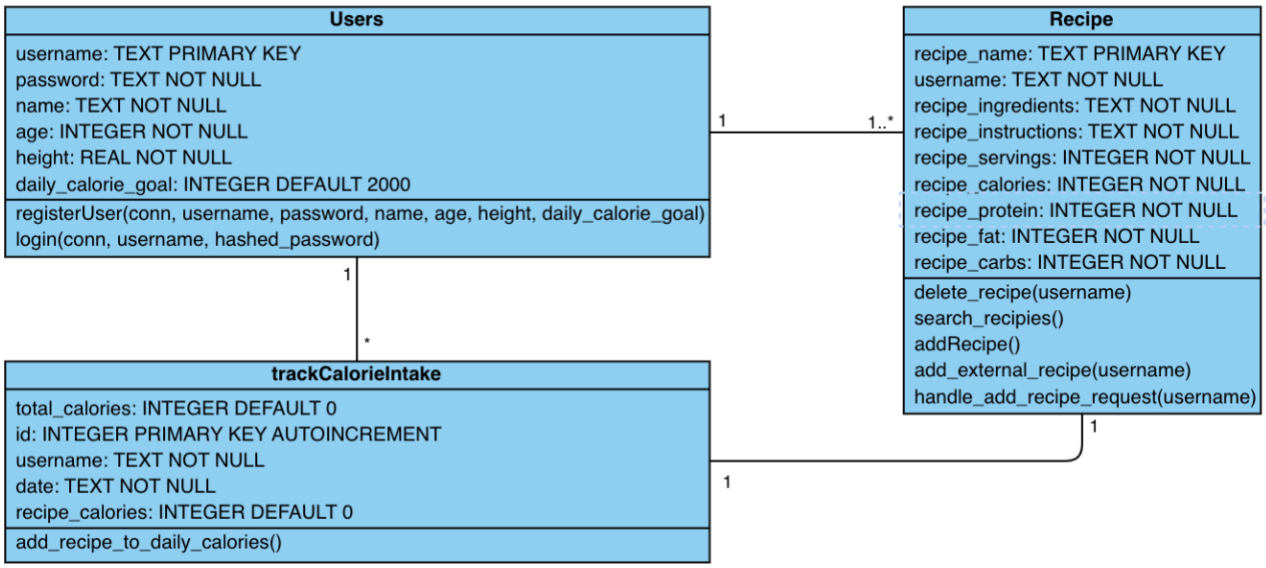


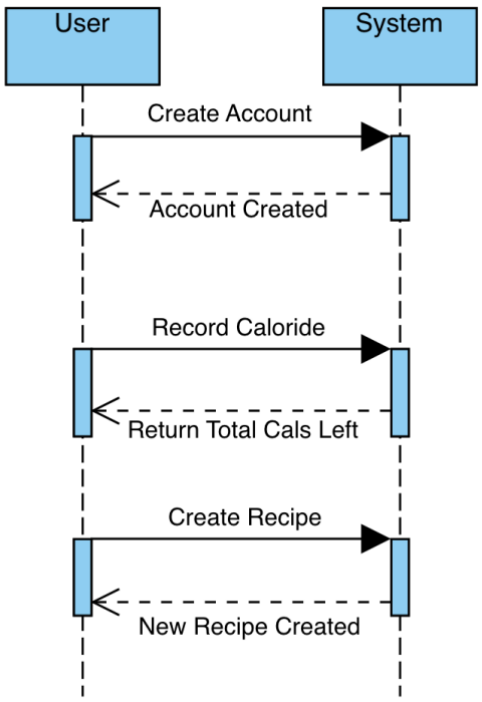
Textual Descriptions:

- Register: This function allows users to register for an account. They are asked for a username, password, name, age, height, and calorie goal. The new users are stored in a database (users.db)
- Login: Allows users to log in to their account if it was created
- Track calorie intake: this function allows the user to select recipes or ingredients from the database that they have consumed to track their calorie intake for the day.
- Create recipe: this function allows the user to create a custom recipe using the ingredient database. This custom recipe will be added to the recipe database allowing the user to conveniently save it for the next time they want to track it.
- Edit recipe: this function allows the user to edit previously created recipes.
- View recipe ingredients: This will allow users to view ingredients in the recipes from the database or from their custom-made recipes.
- Delete recipe: This function allows users to delete previously created recipes (users will not be able to delete pre-set recipes).
- Add ingredient to database: This function allows admin accounts to add ingredients to pre-set database.

- Edit ingredient to database: This function allows admin accounts to edit ingredients to pre-set database.
- Delete ingredient to database: This function allows admin accounts to delete ingredients to pre-set database.

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





6. 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.

The software will be designed to operate on personal computing hardware, specifically desktops and laptops. It will be compatible with the operating systems Windows (7, 8, 10) and macOS (High Sierra). This Is subject to change. It will work with common web browsers such as google chrome and safari. It must coexist with a database system.

7. 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.

Assumed Factors:

- Internet connection
- Device combability
- 3rd party sources (database for ingredients and recipes)

Dependencies:

- Database Management System