Name

Alex Benasutti

VM

`csc415-server21.hpc.tcnj.edu`

Project name

Nutrify

Github

https://github.com/Jamboii/Nutrify

Social issue being addressed

This project is intended to combat the social issue of obesity/anorexia.

Option 1 or 2 with rationale

Food nutrition is something that interests me because it plays a key part in personal health, and it's something that I want to know how to improve at. In addition, I'm much more interested in creating a project of my own rather than adding on to an existing one that I may or may not be passionate about. It is for these reasons that I chose to pursue "Option 2" of this assignment.

Web-based or Mobile-based, languages, frameworks, platform

The project will be implemented as a mobile application, for on-the-go usage. The application will be built using the iOS platform using Swift, utilizing macOS's Xcode IDE, and FileMaker Pro relational database application.

Project idea in one sentence

The premise of the overall project is a mobile application that will act as a self-help and nutritional guide to achieving a desired weight goal in a certain amount of time.

Why is it innovative and interesting, addressing the social issue

The application aims to discipline, motivate, and engage users in an environment that rewards them for completing progress towards their goal. It objectively should be an intermediary device that sits between the user and their weight goals, and measures itself based on self-evaluation. Calorie counting is a process that can easily be improved with automations that are easy to use on-the-go and provide a lot more information to the user. By additionally incentivizing usage by tracking weekly progress, being informative with the user, and entertainment such as games, the application will be able to actively influence the user to make better nutritional choices to combat the issue of obesity or anorexia.

Algorithms and rationale

- Register user and begin with a survey to determine their track
    - Ask for their meal habits
        - breakfast
        - lunch
        - dinner
        - snacks
        - beverages
        - condiments as a subset for each of the 3 meals
    - Ask for their height, weight, age, gender, exercise habits (light and hardcore)
        - Statistics that could determine their ideal calorie count for the week
    - Ask for a weight goal and a specified amount of time to do it in
        - Set constraints for unreasonable goals (e.g. don't allow "lose 50 pounds in a week")
        - Set to gain or lose weight
- User inputs the food they eat for certain meals of the day to gauge them as they go about their day
    - Checkbox select and confirm foods from database
        - Sort by breakfast, lunch, dinner, snacks, beverages, condiments+additions
        - Process should only take a few clicks
        - Recommend food items to user that were previously selected
        - Total food eaten will add to calorie total for the day, will check against desired calorie total
    - Can add new food to database with picture, calories, ounces
    - Give notifications at certain intervals of the day for when they eat a meal so they can input it into the app
    - Give notifications if they go above or below their desired calorie count for the day
- Incentivize the user to work toward weight goal
    - Entertainment
        - Game about calorie prediction and education
            - Similar to a "The Price is Right" format
    - Be informative
        - Display different lists of foods that will get the user to their specified calorie total for the day
            - Draw these lists first from the user's commonly consumed foods
            - Create different lists to keep user diet interesting
    - Track weekly progress by self-evaluation
        - Tell user to input their weight by the end of each week
        - Graph data points of user's weight changing over time
        - Congratulate user when they complete their goal
    - Give user "points" by consistent use of the app + seeing progress in goals
        - Users can redeem "points" for coupons for healthy food products (i.e. Beyond Meat) or some similar reward

        - Would be dependent on utilization of advertisements within the app to create revenue and dependent on partnership with companies that would see profit in this form of advertisement
        - (User honesty could also pose an issue)

Data structures and rationale

- Utilize a food database for intuitive look-up and organization
    - Manually create a smaller food database to start with or use Nutritionix nutrition database
        - If using a smaller database, allow users to add their own food and data to the database with human validation. Partner with food and restaurant establishments to add their products and calorie counts
        - https://www.nutritionix.com
    - Food information
        - name
        - picture
        - calories
        - ounces

Software engineering concepts I expect to learn and enforce

In aspects of development, I expect to learn a lot more about the Swift programming language in addition to the Xcode IDE for mobile application development. I also must learn the different aspects of database creation and management with the use of Apple's FileMaker Pro and the Nutritionix database.

In aspects of software engineering concepts and principles, I expect to learn to focus on quality at every step of the process, but also be ready to adapt at any point. The end product should be polished from a user's perspective, but what the idea of the project looks like now will more than likely not be what the project turns out to be in the long run. I also hope to enforce version control through use of a private github repository to track changes I make to my program. Managing change is an important principle that is established on the mechanisms of change requests, assessments, approval, and implementations.

Other software engineering principles I expect to practice involve the idea of project divide and conquer strategy. A large problem is easier to solve if it is subdivided into a collection of elements, and for a long-term project like this mobile application, it is imperative to break each sub-problem down to its core components and solve them from a lower level first. However, it is also important to first learn to express the project at a high level of abstraction before breaking those components down. Described in my algorithm and my use case diagram are many high level aspects of the Nutrify application. Approaching each abstracted use case on its own and dividing it into its smaller components would be one way to go about this project.

I want to be able to build software that exhibits effective modularity. The modules I build should be simply interconnected in some manner, and exclusively focus on a single aspect of the system. Having effective modules combined with effective documentation will make for an easily interpretable product that will also be easy to maintain over the long term. The use case diagram in Figure 1below exhibits many different interconnected cases that grab information from others.

Striving for consistency is also an important concept to enforce. When considering the user-interface of a mobile app, front-end aspects like menu option placement and overall color scheme are necessary to keep consistent to keep user-experience as seamless and understandable as possible. In the case of Nutrify, the experience of entering in your meals throughout the day should take no more than a couple clicks. If the process is difficult to understand in any aspect, the application will lose consumer retention.

Use case diagram

Figure 1 shows the use case diagram for the Nutrify application. There are two primary actors, one being the human user and the other being the client itself. While the user will manually input their own biological and nutritional data as they use the app, the client will take this information and make sense of it. Example use cases of the client include creating recommendations for the user with respect to their goals, keeping track of and displaying their weight data over time, and sending notifications throughout the day such as reminders to input the meals they ate.

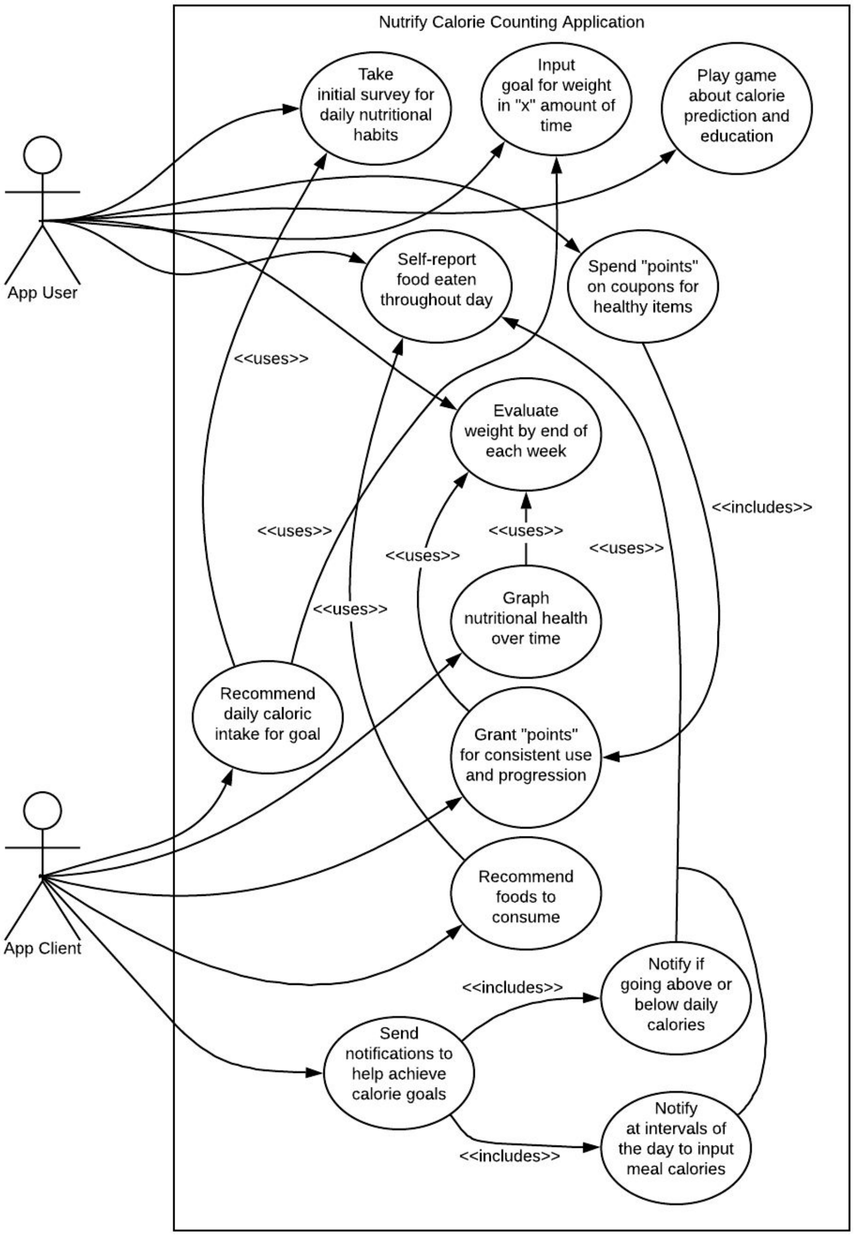
Nutrify Calorie Counting Application
App User
App Client
Take initial survey for daily nutritional habits
Input goal for weight in "x" amount of time
Play game about calorie prediction and education
Self-report food eaten throughout day
Spend "points" on coupons for healthy items
Evaluate weight by end of each week
Graph nutritional health over time
Recommend daily caloric intake for goal
Grant "points" for consistent use and progression
Recommend foods to consume
Notify if going above or below daily calories
Send notifications to help achieve calorie goals
Notify at intervals of the day to input meal calories
<<uses>>
<<uses>>
<<uses>>
<<uses>>
<<uses>>
<<uses>>
<<includes>>
<<includes>>
<<includes>>

Figure 1. Nutrify Use Case UML Diagram

Proposed timeline

Figure 2 shows a proposed timeline for the development of the Nutrify application. This timeline was created without any previous knowledge of deadlines and expectations within the CSC-415 class, but still keeps in mind the amount of time necessary to learn the different platforms and languages used for this project. About 10 days at the beginning of March 2020 will be utilized to learn as much as possible about the Swift coding language, Xcode IDE, and iOS platform, with 5 days overlapping to write out proper requirements for the completion of the project. 5 days will also be considered for educating myself in FileMaker Pro, a cross-platform relational database application that will make management of the database used within this application a lot more friendly to the developer in terms of prototyping.

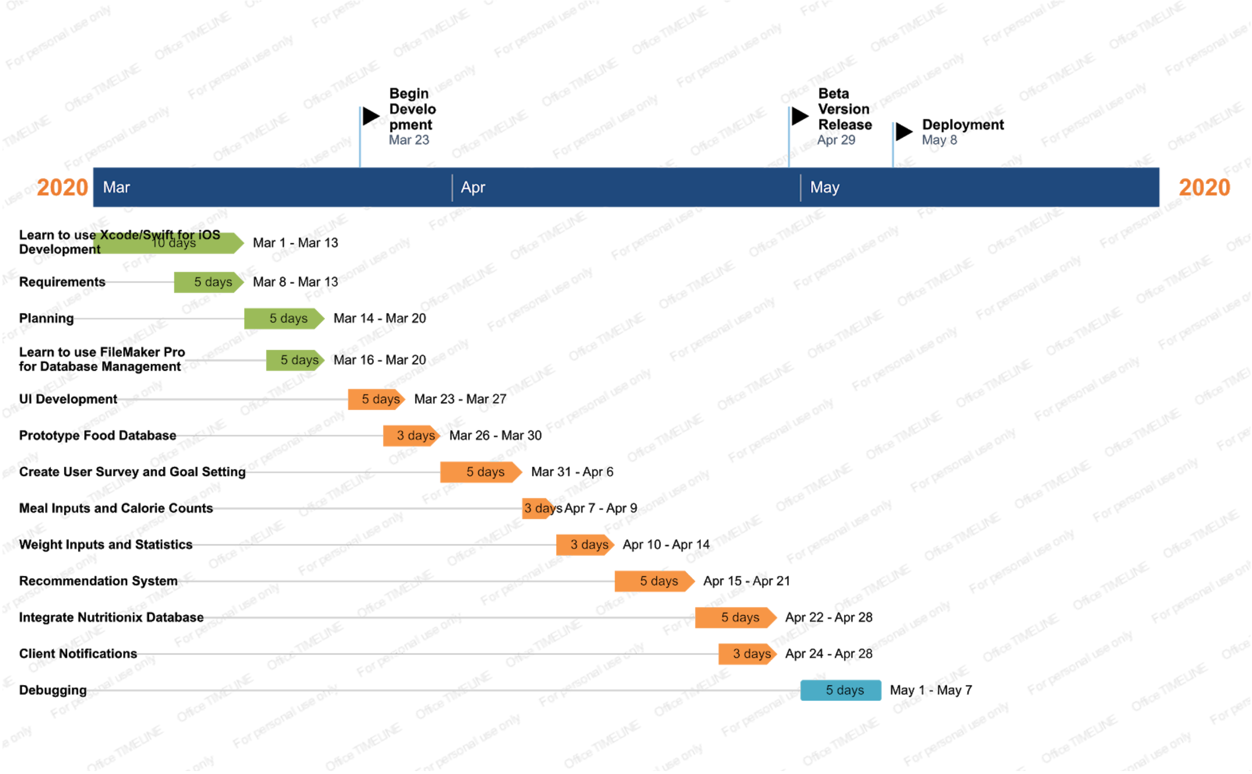

Figure 2. Proposed Timeline for Nutrify Application

Resources to Utilize

- Xcode: https://developer.apple.com/documentation/xcode/
- Swift: https://swift.org/documentation/
- FileMaker Pro: https://www.filemaker.com/support/product/documentation.html