Project Description
Name: Balance CMU
Description: Balance CMU is a Tkinter-based nutrition app designed to help students of Carnegie Mellon plan more balanced meals while using CMU Dining Services. The app allows students to plan balanced meals on a personalized basis based on their BMI, allergies, etc., and provides recommendations on future meals based on these factors and on what they have already eaten in the day. It can be hard to find healthy meals on campus, so the goal of this app would be to help users remain healthy and to recommend and promote balanced meal options.

Competitive Analysis
One similar project is Recommend4U. Recommend4U recommends restaurants based on voice recognition. My program will be different from this because it is focused on recommending foods offered by CMU Dining services, and it focuses on recommending foods based on the personalized needs of the user. On the other hand, Recommend4U is meant to be much more generalized, recommending foods in a particular city and sorting by price. If someone was looking to go out to eat, they might use Recommend4U; however, if someone was looking to eat a balanced meal at CMU, they might use my app. Therefore the algorithms for recommendation will probably be a little different; while both are based on user input, his is based on more immediate user input, while mine is based on accumulated user input (BMI, what they've eaten so far in the day, etc.). In addition, a major difference between Recommend4U and my project is that Recommend4U does not include a nutrition calculator.

Another project that is similar to mine is Kitchen Helper Bot, which is a chatbot that recommends recipes, emails them, and learns what the user likes. A major difference between my project and hers is that, once again, mine has to do with getting food at CMU, while hers has to do with getting recipes to make at home. She incorporates machine learning to learn what the user likes, which is something I have been considering incorporating into my term project. It could also be interesting to incorporate a Chat Bot into my project, especially with the recommendation feature.

Structural Plan
After logging in/creating an account, the user is taken to a home page with 9 different features displayed, including:

1. User's Stats (Height, weight, BMI, how much calories/fat/sugar/carbs they should eat in a day)
2. CMU Nutritional Data (acquired via Web Scraping)
3. Your Trajectory (a plot of how many calories/fat/sugar/carbs they have eaten per recorded day since starting)
4. Your Meal History (a log of what user has eaten each recorded day since starting)
5. Today's Meal Log (user can see what they have eaten today, how many calories/fat/sugar/carbs they have eaten today, and add as they go)
6. Recommendation feature (recommend food based on user preferences and today's meal log)

7. Share trajectory on Facebook
8. TBD
9. TBD

For the TBD features, some ideas I have in mind include a Chat Bot feature and a voice assistant feature (kind of like Siri). I have considered incorporating it into the recommendation feature, or keeping it separate altogether. For instance, the app right now is entirely food based, so the chatbot / voice assistant could recommend other components of a balanced lifestyle, like a type of exercise to do, that is based off of what they have eaten so far today, their stats, their trajectory, etc. The reason that I am keeping this feature open for the moment is because I want to develop a workable program first, and based on how it works see what features would contribute best to making it the best and most well-rounded of a program it can be. The folder will include a main file, which is the original file that must be run. Other files in the folder include a homePage file, which is imported into the main file and sets up the home page once the user logs in, and 9 different files, one for each of the 9 features, which are imported in the homePage.py file and the functions in these files are called whenever the user clicks on the particular feature.

Algorithmic Plan

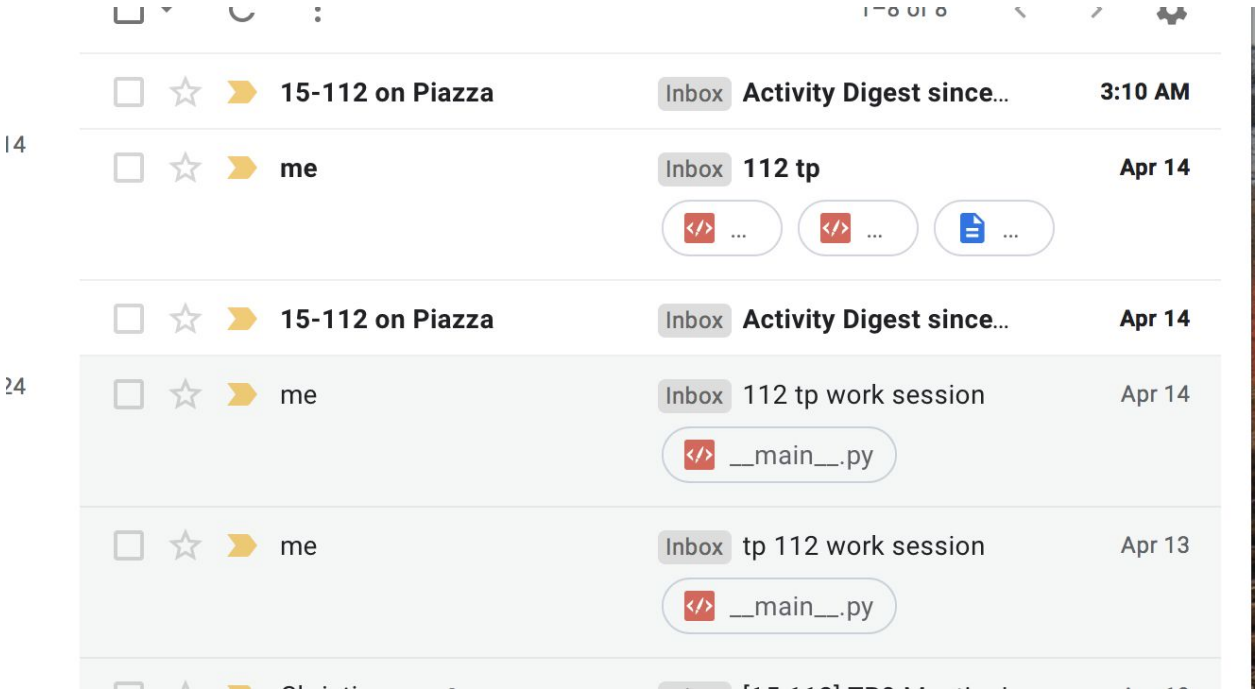
I anticipate that the most algorithmically intense part of the project is going to be the Recommendation feature. I want to write my own pseudo-ML (I know it's not real ML) algorithm that takes into account user preferences based on what recommendations the user has chosen so far. Before the user has used the recommendation feature, the feature will select each CMU food item that fits under or at the remaining of the daily nutritional requirements, and select three from different places (e.g. Underground, Resnik, Gallo de Oro). Every time a user selects a particular food item or place (e.g. burrito at Gallo, Gallo de Oro), the food item gets a higher score. Then, the feature will prefer the foods and places with higher scores, and the next time that the user uses the recommendation feature, the program will opt for the foods/places with higher scores, and if there are multiple foods/places with the same score, the feature will randomly select three. Every time the user uses the feature, the feature displays three food items (I have also been toying with the idea of having this number be adjustable, so if the user wants just one recommendation, or if the user wants four recommendations, the feature can adjust to meet these preferences). In every scenario, however, the food item will first be chosen on the basis of whether it fits the nutritional requirements of the user (so if the user has 300 calories left to consume in the day and the food is 500 calories, the food will be weighted less than a food that has 400 calories). In the end, the foods will be chosen based on a combination of their preference score and their nutritional score, with the nutritional score being weighted heavier because, while user preference is important, this is meant to be a nutritional app.

Timeline Plan

- By 4/17: Use Web Scraping, have data for CMU nutritional info set up
- By 4/19: Have "Your Stats", "About Us", "Your Meal History", "Today's Meal Log", "Your Trajectory" features set up
- By 4/20: Have "Food Recommendations" set up

- By 4/22: Have “Share On Facebook” set up
- By 4/23: Choose additional feature(s) based on how program looks so far (e.g. chatbot? Voice assistant? Both?) and begin to develop these
- By 4/27: Finish additional feature(s)
- By 5/1: All bugs fixed

Version Control
I am emailing the code to myself after each work session.



Module List
Beautiful Soup (for Web Scraping)

TP2 Update
The food data is being stored in nested food dictionaries instead of being acquired by Web Scraping. I am not including height, weight, and BMI on the User Stats page because from a design perspective it doesn’t make much sense when I look at it (the useralready knows his/her height/weight, and the BMI is pretty irrelevant to the rest of the information on the page). In its place, I put the remaining amount eaten in the day and the percent of the daily amount that has been eaten. The TBD features include an About section and a Voice Mode section that the user can turn on and off, such that the user can speak to the app and the app will open certain icons instead of clicking (I will do this after TP2). **Updated Modules:** fbchat for Facebook Messenger feature, datetime to keep track of dates, PIL for images, matplotlib for trajectory graphs, and after TP2 I will be using Speech Recognition API and pyaudio for Voice Mode feature.

TP3 Update

I implemented the voice feature, so now the user can choose whether to click on the voice icon and speak a command to the program or to click on an icon as usual. In addition, I added a new feature where a user can add new foods that are not CMU-affiliated. I did this using the USDA's API. The user can search for non-CMU foods and look at the nutrition information. If they want, they can add it to their meal log, and the non-CMU food will incorporate itself into the meal history, trajectory, user Info, and Facebook features (so basically all of the features that keep track of meals).