Isabella Fons
Project: Free Of

TP3 Update
Two big changes were made in TP3. First, if a product is not found in the database, it doesn't crash. Rather, it will allow the user to type in ingredients and will test for allergens based off of those. Secondly, if a user can't eat an item, there is a similar products button. This comes up with a slide scroller of possible products that are safe to eat and if you click on the picture and the product is available online, it will take you to a link to purchase.

TP2 Update
Some updates I have made to my program mainly include being able to type in your own ingredient that you are allergic to and then add it to your set of allergens. This way, you can customize your allergens entirely and the program will still work. There is also a "rescan" feature for when the item isn't found. Moving forward, I want to add a recommended items feature for when items are not safe to eat. I will do this by web scraping Amazon Prime Now which contains UPC codes then adding it into the not safe interface.

Project Description: Free Of is an app in which you can input your allergies and scan the barcode of a food and it will tell you whether it is safe to eat or not!

Competitive Analysis:
Similar projects I've seen that have some sort of "Scanning" feature include Makeup Magic, Emotion Detector, and Jason Derulifier. These are all very good projects, but the main issue I see in the videos is that they all have quite a bit of lag. This could be because the algorithms they use in their projects are very lofty, making their whole program and their computer lag. To make my program not lag, I plan on using sets and dictionaries (mapping allergies to things that are that allergen) to make my algorithm more efficient.

Other similar projects use data analysis, such as Tennis Insights or PyStocks. While because of data analysis you can do many features, this often makes the interface look messy or discomboulated. I want to keep my interface very clean, yet still have multiple cool features and methods of analysis. I will try to do this by having multiple windows!

Structural Plan:
Files that need to be included include:

- Barcode scanner file
- File that sorts through all the USDA information (finder algorithim)
- Barcode CSV file that gets added to
- Init file

- Python files for separate pages
- Perhaps - a file to break down the item that was recently scanned

Algorithmic Plan:
The trickiest part of my project would definitely be identifying which item is scanned with the barcode reader. The way I plan to do this is by storing each scan into a CSV that clears itself when the program is exited. I will turn the items in the CSV into a set of tuples of the barcode and then a blank spot so that there are no duplicate items because that wouldn't really be needed. Going through the set of tuples and the CSV from the USDA, then I can match the barcode number to the name of the product in the CSV from the USDA and identify it that way. If I find I need more information later (such as ingredients), I can create more blank spots in the tuple I create to fill it with information that I need.

Timeline Plan:

4/17 - Basic template for product with buttons and whatnot
4/18 - Scanner reveals what item was scanned
4/20 - Perliminary Allergy Detector done (shows ingredients)
4/22 - Image upload feature
4/23 - Minimum Viable Product

Version Control Plan:
Email to myself whenever I reach a new feature
Also my hard drive that I use always not since I spilled water on my laptop over spring break!

Module List:
open CV