# Design Document (TP2)

Keren Huang Mentor: Alex Xie

# TP2 Update

## Structural Plan

- Realizing that having every single line of code I wrote in a single file was not a good idea for organizational purposes, I have split up the code accordingly by mode:
    - tp_v3.py (main file, will rename to **wit_on_wellness.py**)
    - splashscreen.py
    - instructions.py
    - sandbox.py
    - puzzlemode.py
    - creditspage.py

## Timeline Plan

- December 5 → TP2 due
- POST-MVP (rough idea)
    - December 6 → work on user data import / export feature
    - December 7 → look into smart suggestion search system
    - December 8 → finish up intended post-MVP features
    - December 9 → TP3 due
- Running list of potential post-MVP features (ranked from most likely to implement to least likely):
    - User data import / export feature with .csv files
    - Smart suggestion search system (nltk)
    - AI diet suggestions (scikit-learn)
    - Using Kaggle datasets to train
    - Login system for online community for users

# Project Proposal

## Project Description

- **Name**: Wit on Wellness (WoW)
- **Description**: An Edutainment app that teaches nutritional habits and provides a sandbox to test diets and make food plans

## Competitive Analysis

- Part of my inspiration comes from a similar project idea I saw called Deep Learning for Stocks - Using LSTM/RNNs (Roshan Ram) where he created an edutainment app for stocks; my original idea was to create an app similar to MyFitnessPal or FitBit to let people simulate a food plan and how it affects their health. After running the idea through with Professor Kosbie, it was decided that the app should provide a more cogent user experience and essentially allow it to become an edutainment app.
- My app will be similar to MyFitnessPal / FitBit, because I will allow the user to input the foods they may decide to eat in a day and my app will discuss whether the food plan is sufficient for the person's goals. A way that I possibly plan to make it slightly different (shoutout Professor Kosbie) is to add a problem solving mode where I can try providing the user with a case scenario where they have to figure out what to feed an avatar to keep them happy and healthy, while overcoming adversaries (roommate eating all the food, some food not available) to help a user learn to look at macronutrients when picking foods.

## Structural Plan

- I plan on having most, if not all, of my code in one file, where it can be run. I'm using the subclass ModalApp to organize the different modes for my project, and I'm creating methods to perform certain tasks, such as pulling data from the FDC API in Sandbox Mode through the getFoodDict method.
- SandboxMode
    - Methods:
        - takeUserInputData
        - getFoodDict
        - getCachedPhotoImage
        - displayFoods
        - displayUserFoods
        - calculateTDEE
        - calculateQuantities
    - Subclasses:
        - Results
            - Methods:
                - findProportions
                - checkProportions
                - drawBarGraph
                - drawLinePlot
- PuzzleMode
    - Subclasses:
        - Puzzle1 (Easy) → Find the food with the lowest total calorie count
            - displayFoods
            - calculateCalories
        - Puzzle2 (Hard) → Restaurant menu of foods, pick the most fulfilling plate while under spending limit
            - ***Implement linear programming to get results***
            - simplexAlgorithm
- Instructions
- Credits

## Algorithmic Plan

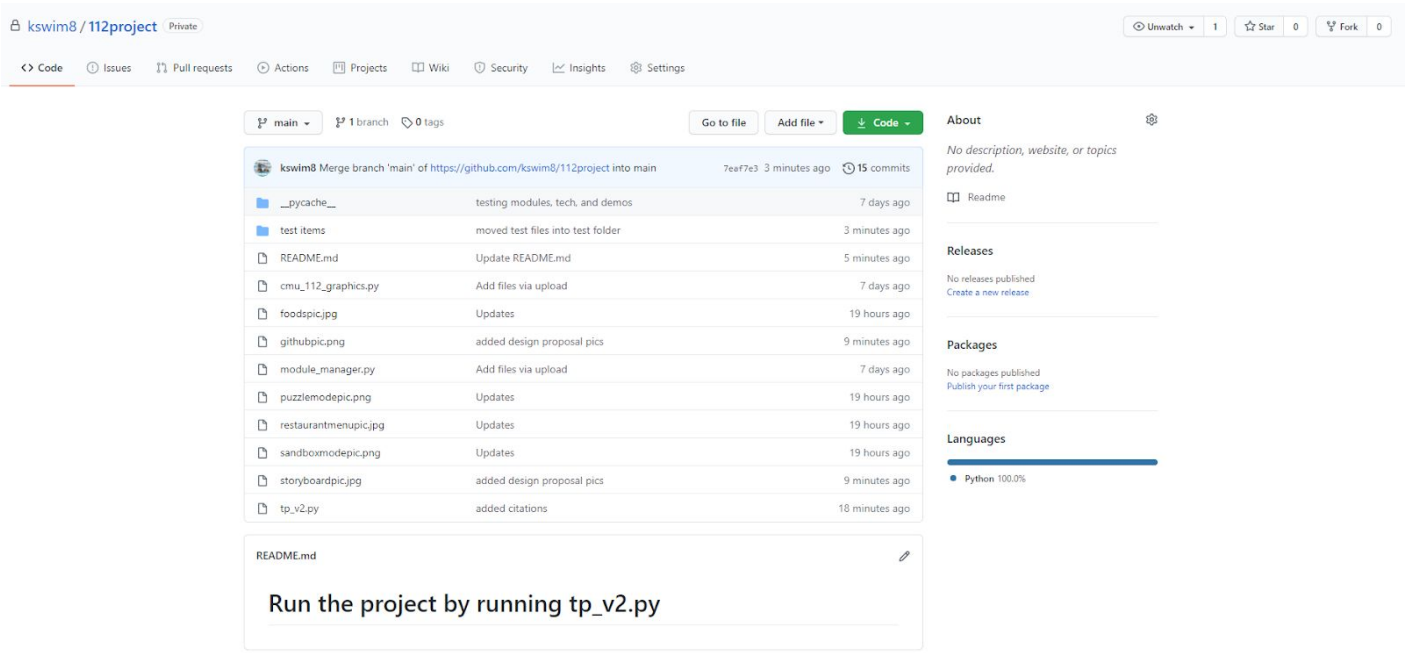
- Trickiest parts of project: Puzzle mode and linear programming implementation, webscraping and displaying images, API use
    - Puzzle mode and linear programming implementation
        - Use the simplex algorithm for optimization - https://medium.com/@jacob.d.moore1/coding-the-simplex-algorithm-from-scratch-using-python-and-numpy-93e3813e6e70 / https://www.cs.cmu.edu/~15451-f17/handouts/simplex.pdf
        - Apply algorithm for Puzzle Mode 2 to find optimal combination of foods and prices
    - Webscraping and displaying images
        - Using bs4 to scrape the first image from Google images, then embedding a y coordinate in the food dict to avoid MVC violation from changing app state
    - Using API to pull food data
        - Using a post request to get a response with a simple query, then pulling the proper data to use in project

## Timeline Plan

- November 30 → TP1 due
- December 1 → figure out how simplex algorithm works
- December 2 → finish Puzzle Mode 1 and start Puzzle Mode 2
- December 3 → work on Puzzle Mode 2
- December 4 → work on and finish Puzzle Mode 2
- December 5 → TP2 due
- December 6 → decide post-MVP features, AI diet suggestions (scikit-learn) or login system or search autocomplete

- December 7 → implement one of the above post-MVP features
- December 8 → finish up the post-MVP feature
- December 9 → TP3 due

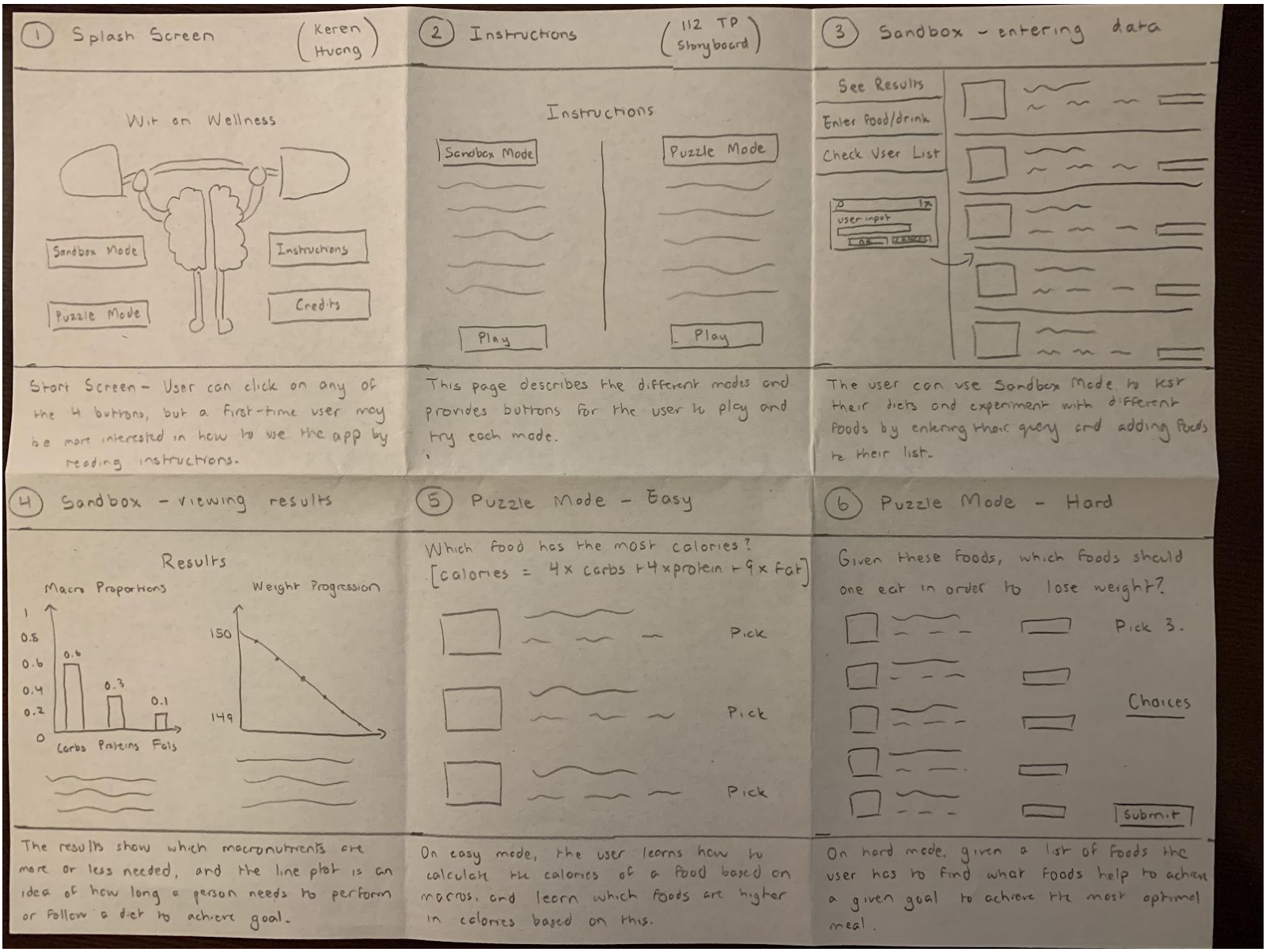
## Version Control Plan

- GitHub, private repository
- Image:
  - 


## Module List

- cmu_112_graphics
- requests
- json
- bs4
- pillow
- For post-MVP later: scipy, nltk

# Storyboard


1 Splash Screen
Keren Huang
Wit on Wellness
Sandbox Mode
Instructions
Puzzle Mode
Credits
Start Screen - User can click on any of the 4 buttons, but a first-time user may be more interested in how to use the app by reading instructions.
2 Instructions
112 TP Storyboard
Instructions
Sandbox Mode
Puzzle Mode
Play
Play
This page describes the different modes and provides buttons for the user to play and try each mode.
3 Sandbox - entering data
See Results
Enter food/drink
Check User List
user input
The user can use Sandbox Mode to test their diets and experiment with different foods by entering their query and adding foods to their list.
4 Sandbox - viewing results
Results
Macro Proportions
Weight Progression
1
0.8
0.6
0.4
0.2
0
0.6
0.3
0.1
Carbs Proteins Fats
150
149
The results show which macronutrients are more or less needed, and the line plot is an idea of how long a person needs to perform or follow a diet to achieve goal.
5 Puzzle Mode - Easy
Which food has the most calories?
[calories = 4x carbs + 4x protein + 9x fat]
Pick
Pick
Pick
On easy mode, the user learns how to calculate the calories of a food based on macros, and learn which foods are higher in calories based on this.
6 Puzzle Mode - Hard
Given these foods, which foods should one eat in order to lose weight?
Pick 3.
Choices
Submit
On hard mode, given a list of foods the user has to find what foods help to achieve a given goal to achieve the most optimal meal.