Tiffany Yu
Class: CS 521 - Fall 2

Dinner Diary is a program that I created from my past experiences. Everyday I am the one to cook dinner for my family, and I write down and log in a notebook what I've made for dinner. In doing so, it allowed me to look back for ideas on what to make if I was stuck thinking about what meal I should make. In addition, I use the log to make sure I don't remake a recipe that I recently made. The program I have created can be used to keep track of other meals such as breakfast or lunch; personally, it helps me keep track in a more organized way. When I write it down in a notebook, I tend to forget some nights and end up logging in several nights in one entry, sometimes almost forgetting what I've made and eaten. Having the program online makes things easier because almost everyday I am using my laptop so it is convenient.

The program Dinner Diary consists of two files: the classes.py file that contains the two classes, and dinner_diary.py, which prompts the user and runs the program. When the program is run, a prompt will appear in the terminal asking the user if they want to login or register. Each user has their own file in the same directory as the programs that correspond with the username they chose; users are able to open the file if they wish to look at all their past meals. If the user chooses to login, the program checks if there is any file in the directory with the name they have inputted; if there is no such file, then the user is prompted once more and told the user does not exist. The user can then choose the second option to register, which creates a new file in the directory.

After the user successfully logs in, they are prompted with 5 different options they can choose from: adding a single meal, choosing from past meals, adding meals via a text file, getting a dinner recommendation, or exiting the program. When adding a single meal, the user is prompted to enter a meal, usually the most recent one, and add it to their dinner log. The program opens the user's file and appends the meal they have just added. When choosing from the past meals, an indexed dictionary with all of the meals previously logged in will be printed out. The user can then type in the corresponding number to choose which one they would like to add to their dinner log; obviously, if the user is new and they have not added anything, nothing will be printed out. When adding meals via a text file, the user will be prompted to enter the name of the text file that contains a list of meals they want to add. It might be annoying for the user to add one meal at a time, so this option is if the user wants to add a lot of meals at once. The text file has to be in a certain format: each meal is on its own separate line.

For the dinner recommendations, I thought it would be a cool and useful feature if I was able to get a recommendation based on what I've eaten in the past. A lot of times I have trouble deciding what to eat, so I look back to past meals and choose from them sometimes. So the program prompts the user to input a number because the program will not suggest a meal that the user has eaten recently. So if the user inputs 4 days, the program will not recommend a meal they have eaten in the last 4 days. It takes all the meals the user has ever eaten, removes the meals they have eaten in the past # days, and then randomly chooses a meal and then prints it for the user. The last option is to exit the program once the user is done.

In my final project zip, I have included the classes.py, dinner_diary.py, two unit test files (named unit_test1.py and unit_test2.py). I have also included a test file named test that contains three meals in order to use the adding meals via a text file option. I also included a file named "Tiffany" that is used in the unit testing. There are no third-party modules, and to run the code depending on which IDE is used: if using Spyder IDE, while the user is on the dinner_diary.py file, just press the run button. While creating the program at home, I am used to working in a linux environment, so working in the directory where all the files are, I type in the command "python3 dinnery_diary.py" in the terminal.