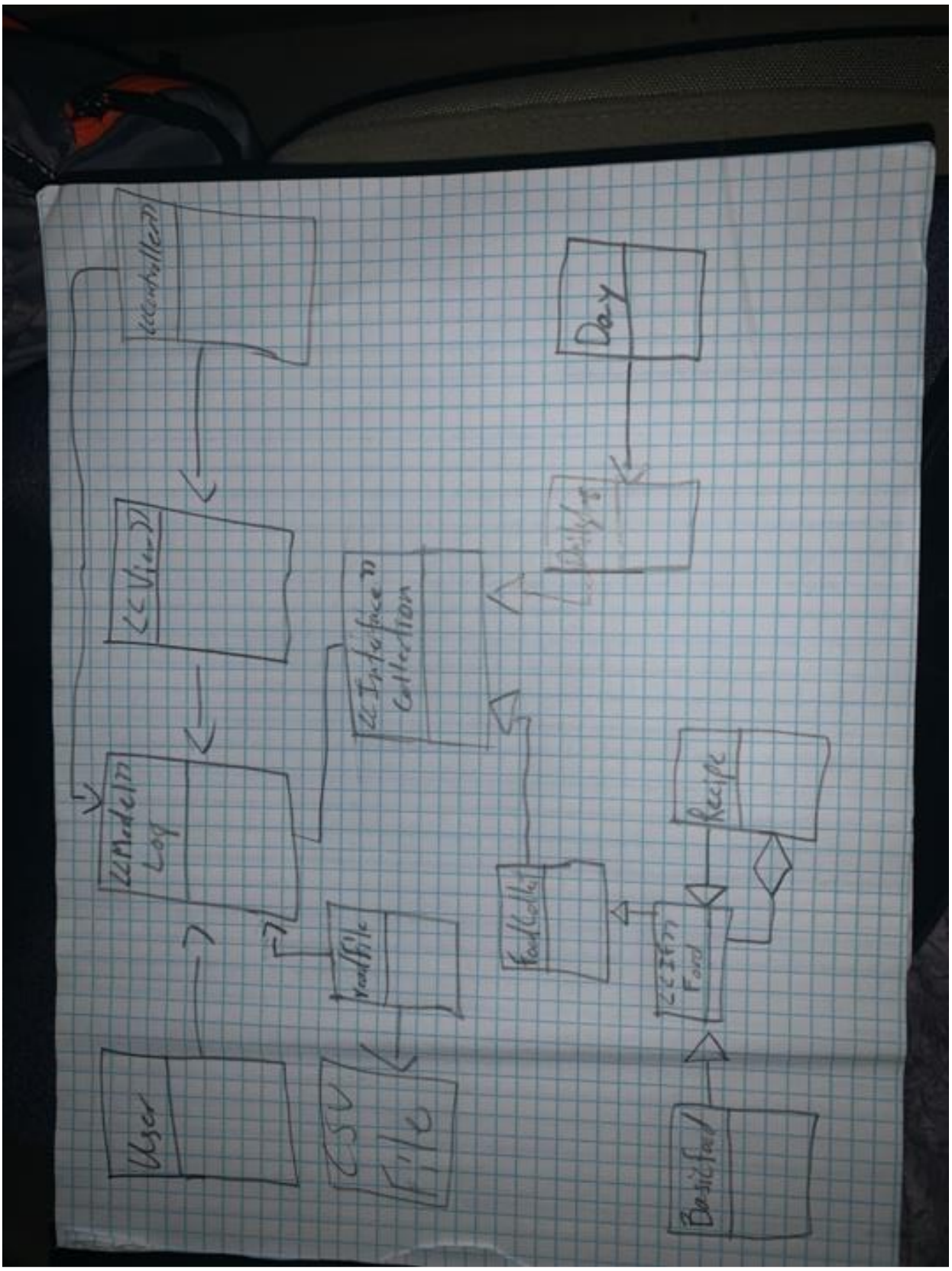
Design Sketch
10/16/2020
Team D

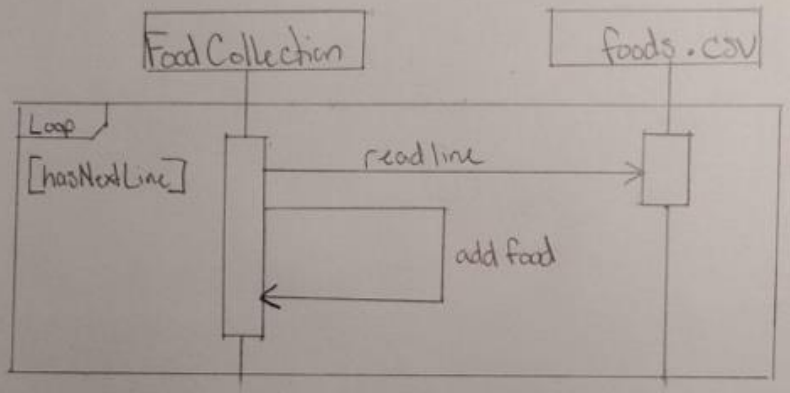
Seth Landers
Steve Jackling
Catherine Liu
Azfal Ali
Steven Morrissey

Class Diagram:

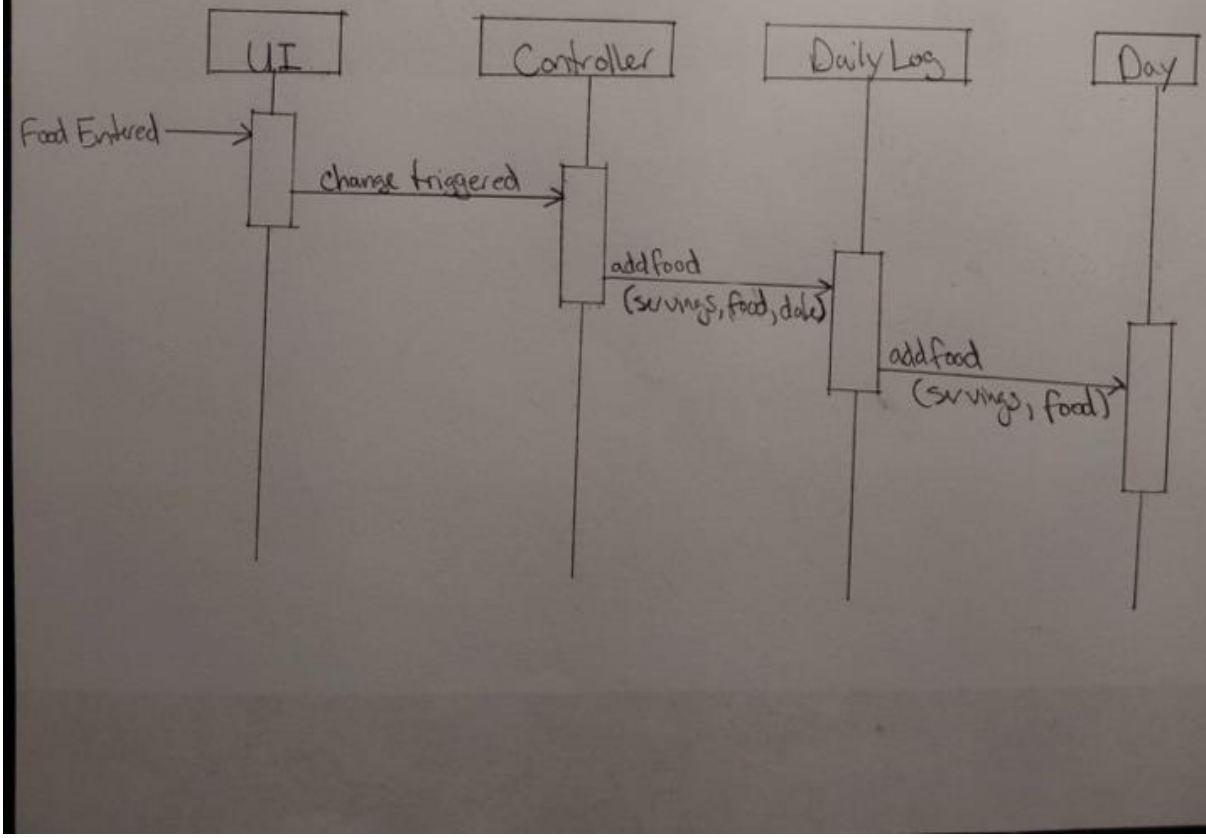


Sequence Diagrams:

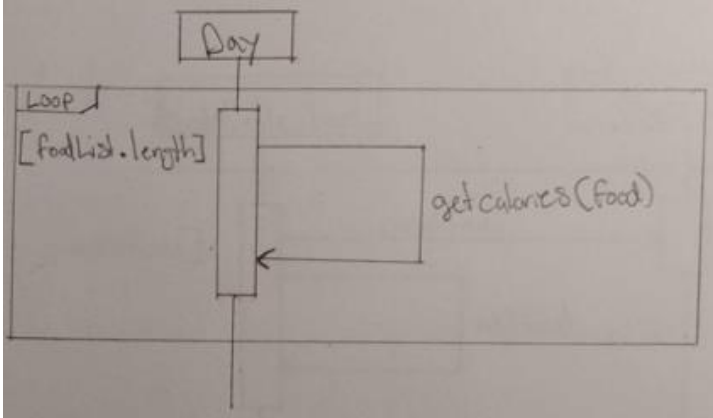
1. Read in a food database consisting of three basic foods, a recipe that contains two of the basic foods, and a recipe that contains the first recipe and the remaining food.



2. Add two servings of a food to the log entry for the current date.



3. Compute the total number of calories for the current date assuming the log consists of a basic food and a recipe consisting of two basic foods



Descriptions:

Log – The Log class acts as a single point of contact for the controller to interact with the model.

View - The view displays the UI for human interaction with the wellness manager.

Controller - The controller updates both the model and the view. It accepts the user input and updates the model and view accordingly.

User - The User class allows for users to be able to maintain an individual profile on the wellness manager.

FoodCollection - The FoodCollection class is a container holding Food records. Extends Collection, has access to foods.csv.

BasicFood - The BasicFood class consists of all the data for the food including the name, calories, grams of fat, grams of carbs, and grams of protein.

Recipe- The Recipe class consists of all data for the recipe including the name of the recipe followed by pairs of foods/sub-recipes and their corresponding number of servings contained in the recipe.

DailyLog - The DailyLog class is a container holding Day records. Extends Collection, has access to log.csv.

Day – The Day class contains the data for a given day including weight, calorie limit, and a list of foods/servings consumed.

Rationale:

In first designing our system, we started building out the classes for patterns we will be implementing in our design. In doing this, it gave us a solid and organized foundation in which to build the rest of our infrastructure upon. We applied the Dependency Inversion Principle with the collection interface which separates the log - a high level class- from two lower level classes: FoodCollection and DailyLog. With the reading of a csv file in the FoodCollection class.

An advantage of our design is that it is easy for the food and recipes in the csv to be read. It is also easy to make them into objects as the FoodCollection class is injected with the component interface for the composite design in which we have or basic food and recipes. Another advantage is our low coupling of our system. A disadvantage at the moment is our log class. The log class has a heavy amount of functionality that we are thinking of refactoring to improve cohesion. Since our design has low coupling, refactoring in the future should be relatively easier if not designed this way.