

Design Sketch

Beep_Boop

Oct 18, 2019

Aaron Kelly, Justin Newton, Aaron “Joel” Wade Parker, Edward Riley, Vincent Venutolo

The Explanation of the Classes

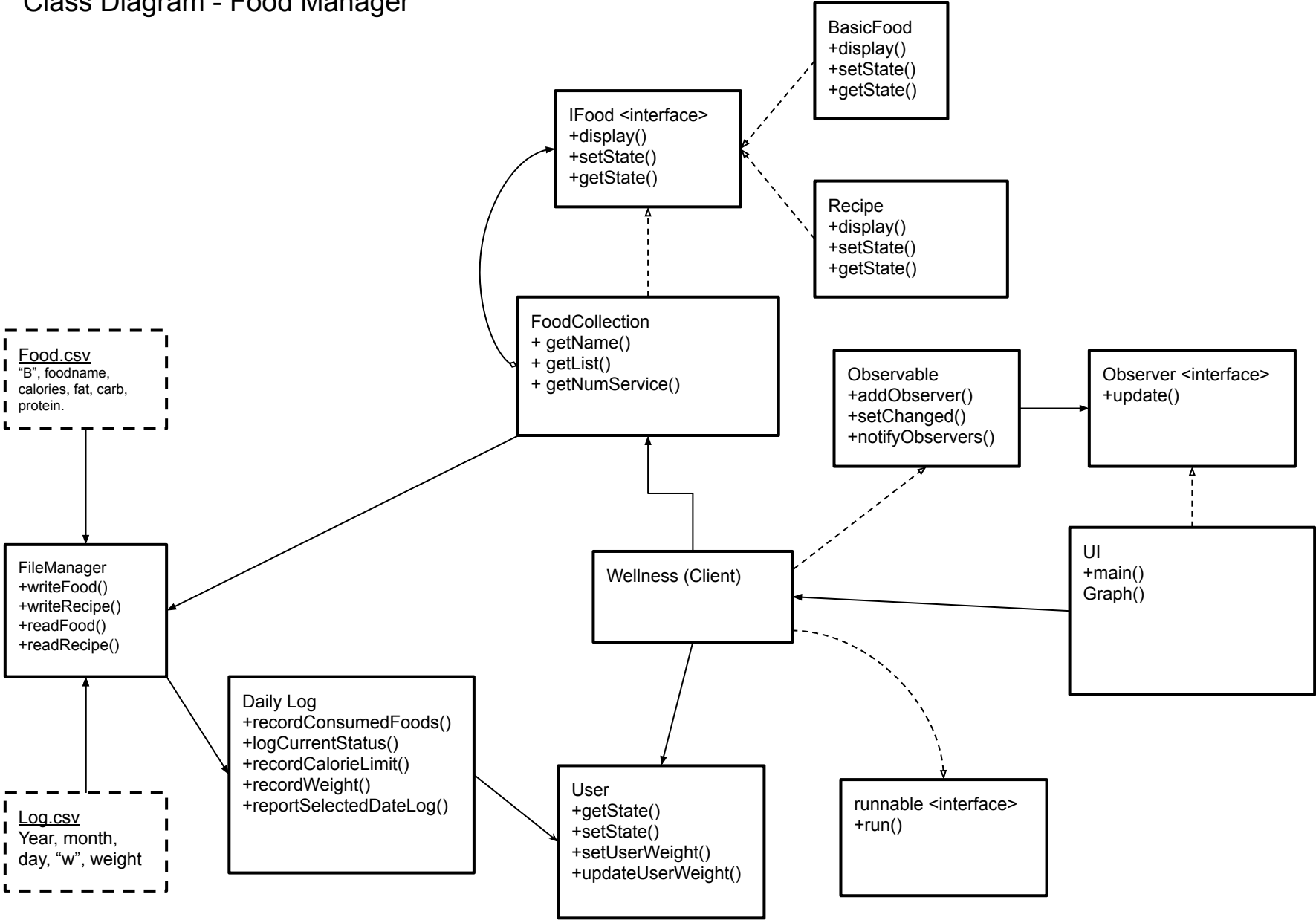
- Food Collection
 - An ArrayList collection of IFood and acts as a controller of the BasicFood and Recipe classes.
- IFood <interface>
 - A shell class for any class that is implemented by it and define those methods. Will be use for dependency inversion.
- Recipe
 - A class that act as composite for basic foods by adding basic foods or remove it.
- BasicFood
 - A single food that is only a leaf and can be defined as pepperoni.
- Wellness
 - An observable class that will notify any changes in information or data.
- User
 - A user that will be using our app and what information they put down about themselves as weight, height will be recorded.
- Daily Log
 - A class where all the user's activities with foods will be recorded and then write to log.csv.
- FileManager
 - A class that handles creating file or existing file and writing to that file. It also handles reading from that file.

The Organization of the System

The system is operating in accord with two implementation of interfaces and spread ways in which one starts with the main station class *Wellness* and initially traverse down step by step its own inheritances and relationship with other classes. *Wellness*'s two important interfaces are *IFood* and *Recipes* are configured to operate through the class which is *FoodCollection* (composite class) used to treat both classes as a simple object. While making sense, the whole project is designed to create and organize a significant amount of data and structure in a mix of unidirectional and multidirectional which composed of a straight linear used for the purpose of obtaining data from files and is multidirectional for updating and getting information to its *Wellness* from its neighbor classes, *User*, *FileManager* and *FoodCollection*.

The project has advantages and disadvantages for the implementation and ongoing maintenance of the program. The advantage of this program design is that it starts with following the requirements and analyzing the situations based on what a client needs. Another advantage while following this design approach, with the trials and errors, the revised design will yield better results. However, there are some disadvantages is that to follow such a design approach to implement a program for the first time will have some errors as in not running smooth and have bugs to fix. Also, we haven't come up with a preventable measure against the user's input error such as "typo" and as well as systems may crash due to its missing exceptions.

Class Diagram - Food Manager



Sequence Diagram - Wellness Beep-Boop

