

Design Sketch

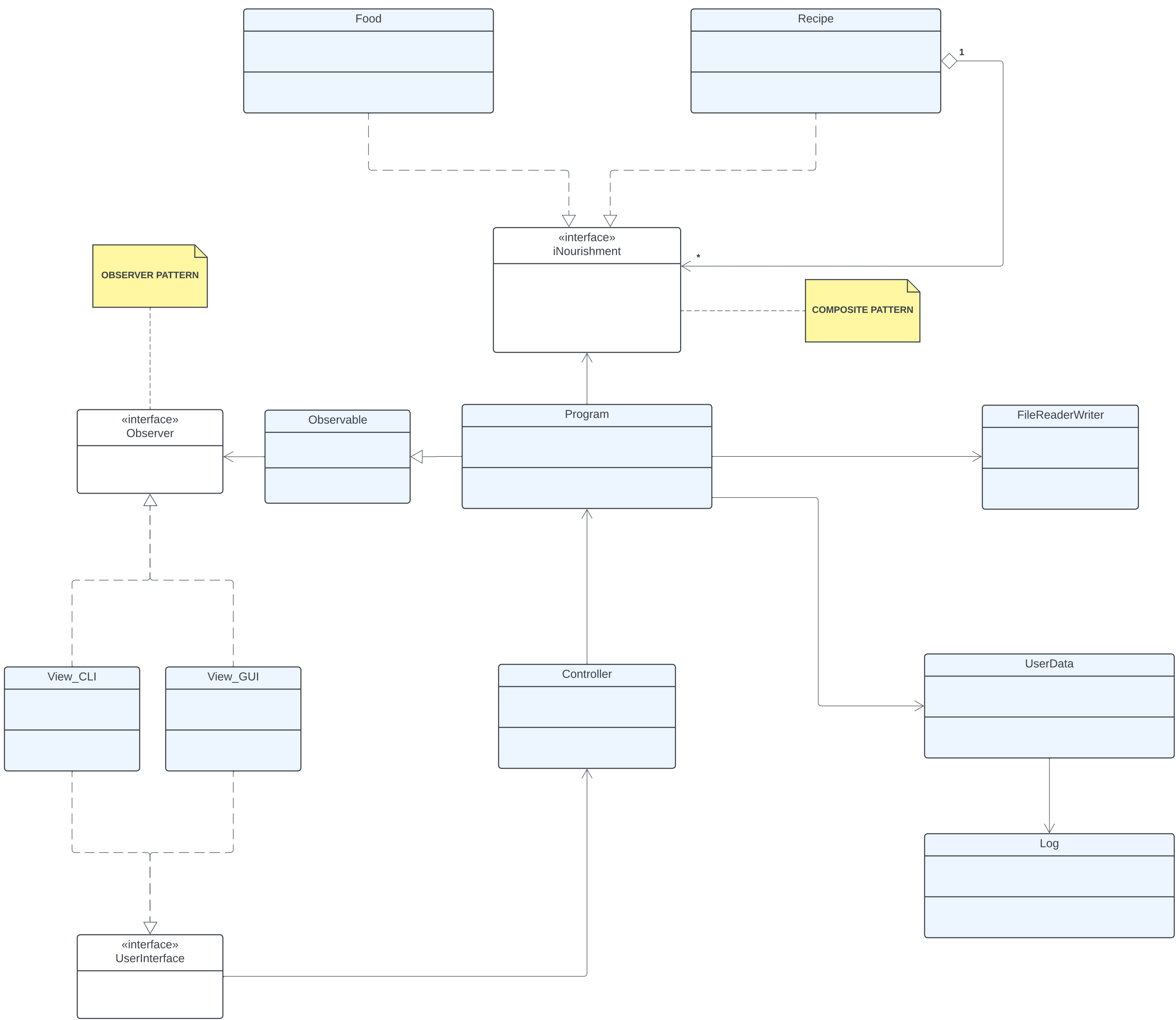
October 17th, 2024

Team Dakota (1A)

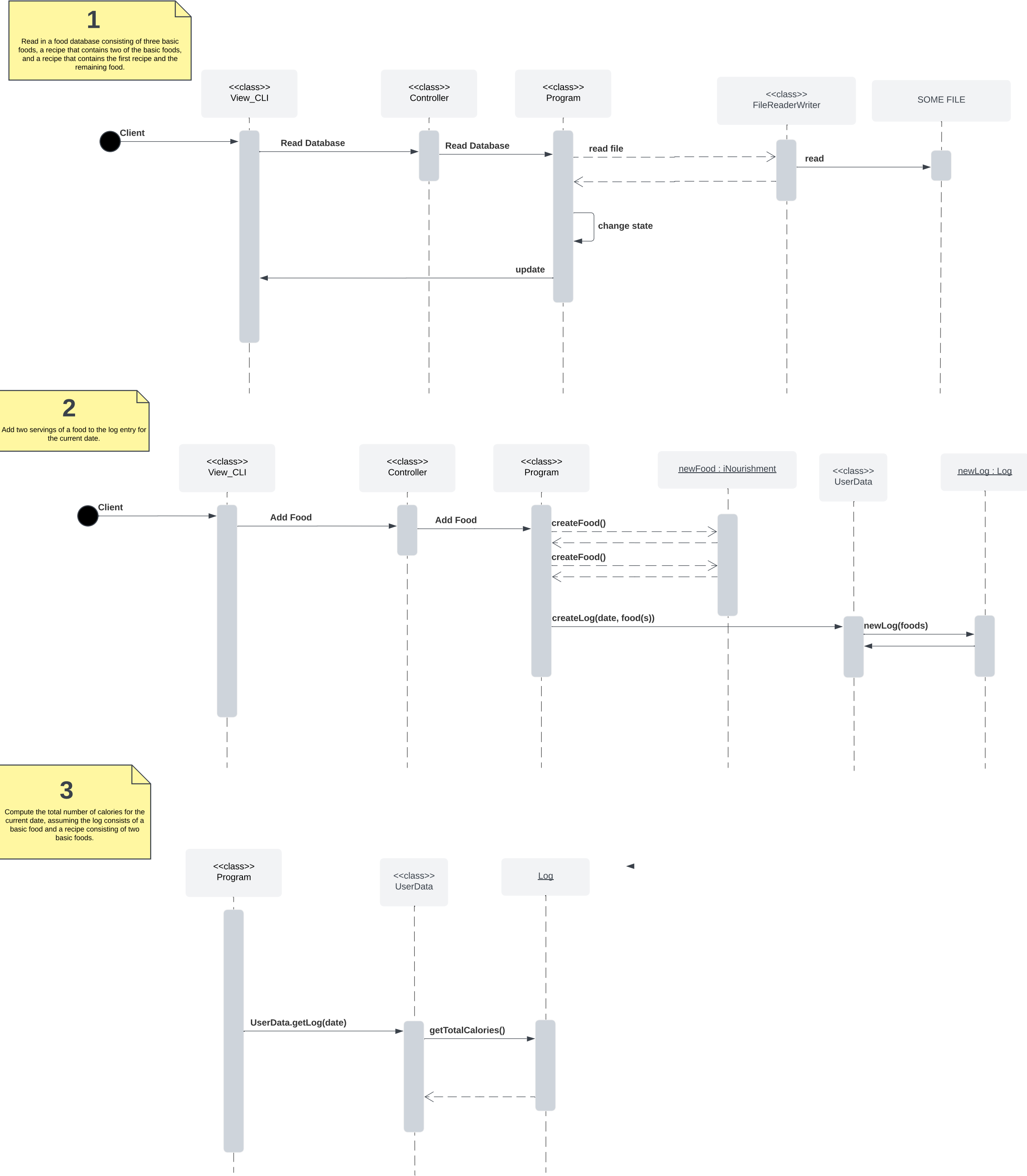
Daniel McKee, Kyle Weishaar, Luke Picciano, Leeon Noun,

Sahand Nowshiravani

UML CLASS DIAGRAM



UML SEQUENCE DIAGRAM



Team 1A

1. Classes:

Food: The Food class creates a food object to be added to logs by the user. The food knows: Name, Calories, Fat, Protein, Carbs, Sodium. It can do: Create Food

Recipe: The Recipe class creates a recipe for the user. The recipe will have a name and is made up of ingredients. A recipe can also use other recipes already created.

Nourishment: The nourishment interface is a set of guidelines and acts as a component in a composite pattern. Each food and recipe class implements the Nourishment operations such as getting the name or calories of each item.

Program: Initializes the components. The Program class loads both user logs and food data from files, sets up the UI, and starts the main event loop. The Program also has the ability to save data as well.

FileReaderWriter: Handles the input and outputs with the file system. User data, food logs, and recipes, for example, are read and written in a CSV format.

UserData: The UserData class is where the logs created by the user for their food are stored. This class knows: All logs made. It can do: Create a new log and Delete log.

Log: The log class is where and food the user eats is recorded. Users can choose to log a food on the date they ate it, and can remove food from the log as well. It Knows All food in log, Total calories, Total fat, Total protein, Total carbs, and Total sodium. It can Do: Get all food, get dates, add food, and remove food.

Controller: The Controller class works in between the user's view and program classes. Interaction through the UI is carried through the controller which in turn, makes the Program class do things.

2. Each section of the program is an attempt to separate concerns.

- The data wing, which handles the user's data (Logs of iNourishment items) which can be calculated for total stats (carbs, fat, protein, etc.).
- The Interaction wing, through a controller class which directly interacts with the program for user interaction through the UI classes.
- The Nourishment wing, which handles the creation of food/recipe items to be used throughout the program.

The program would be easy to modify, have low coupling, and decent cohesion for the implementation. A disadvantage would be having to make sure each section of the program is able to find/use what they need.