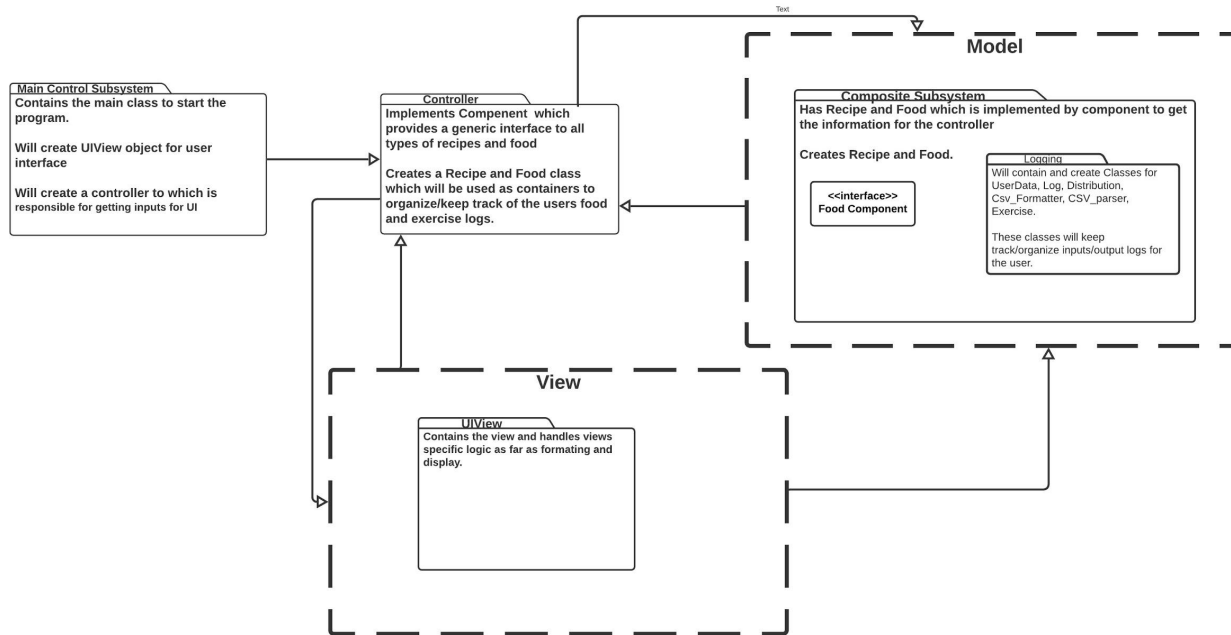


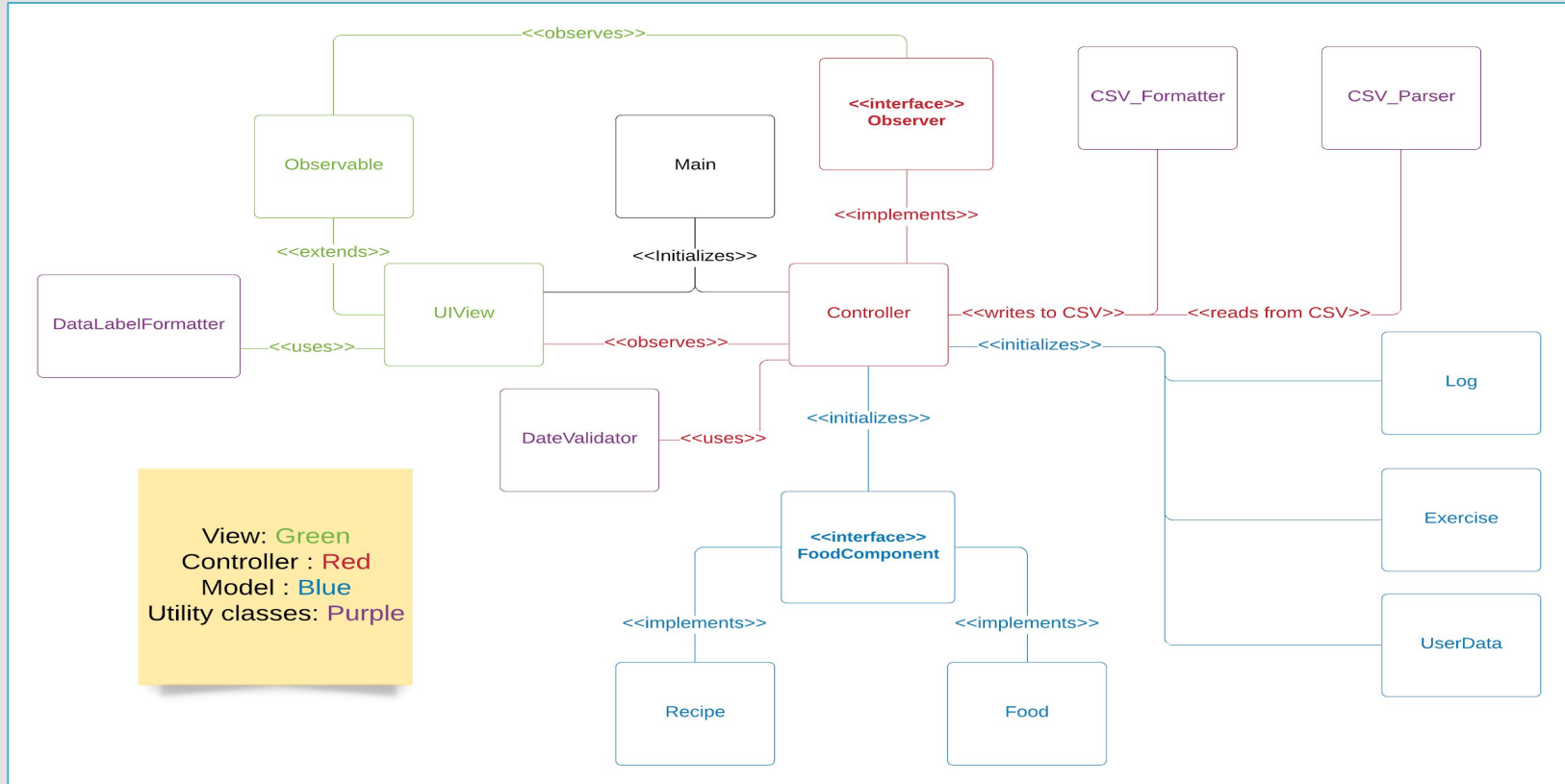
CTeam

John Mule
Justin Nauta
Miguel Galan
Dev Bhat
Andrea Pallotta

Overview of system design

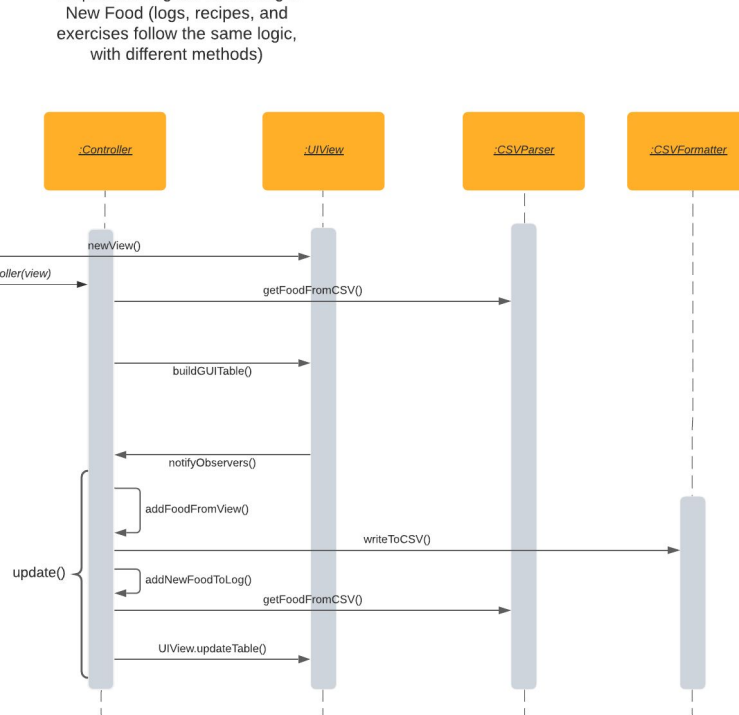


State Diagram

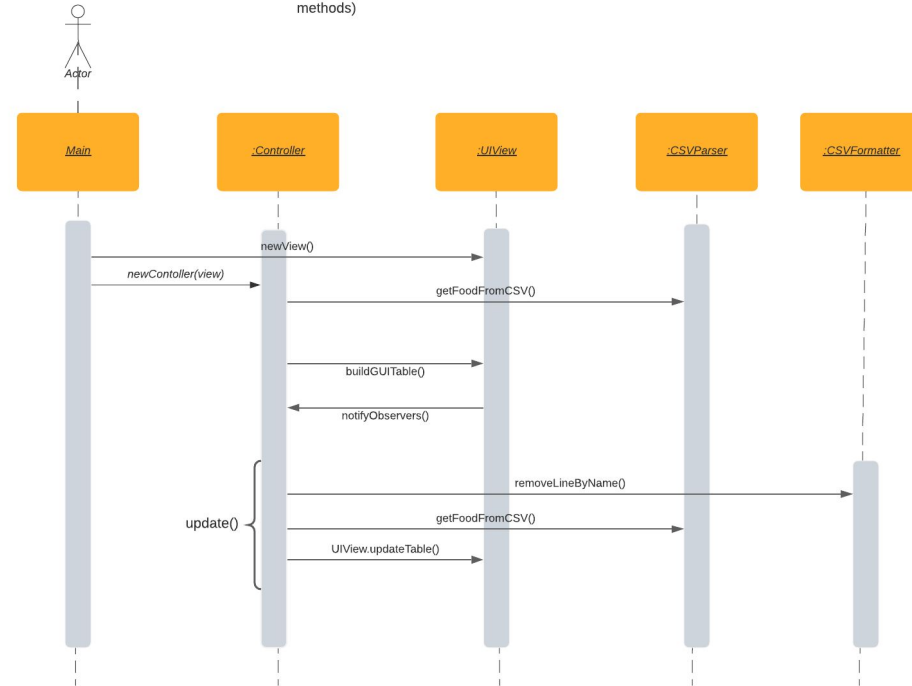


Sequence Diagrams

Sequence Diagram for Adding a New Food (logs, recipes, and exercises follow the same logic, with different methods)

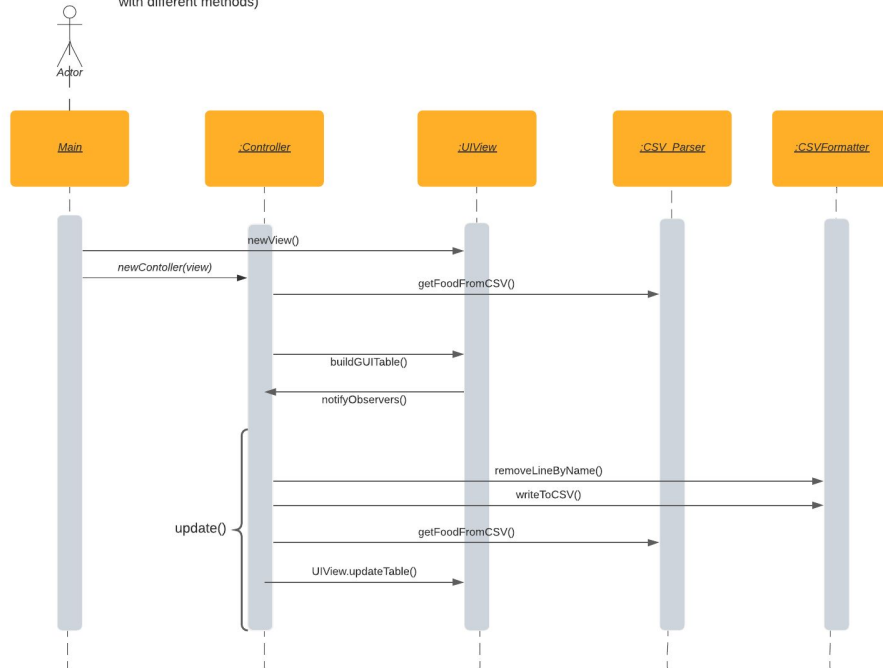


Sequence Diagram for Removing Food (logs, recipes, and exercises follow the same logic, with different methods)

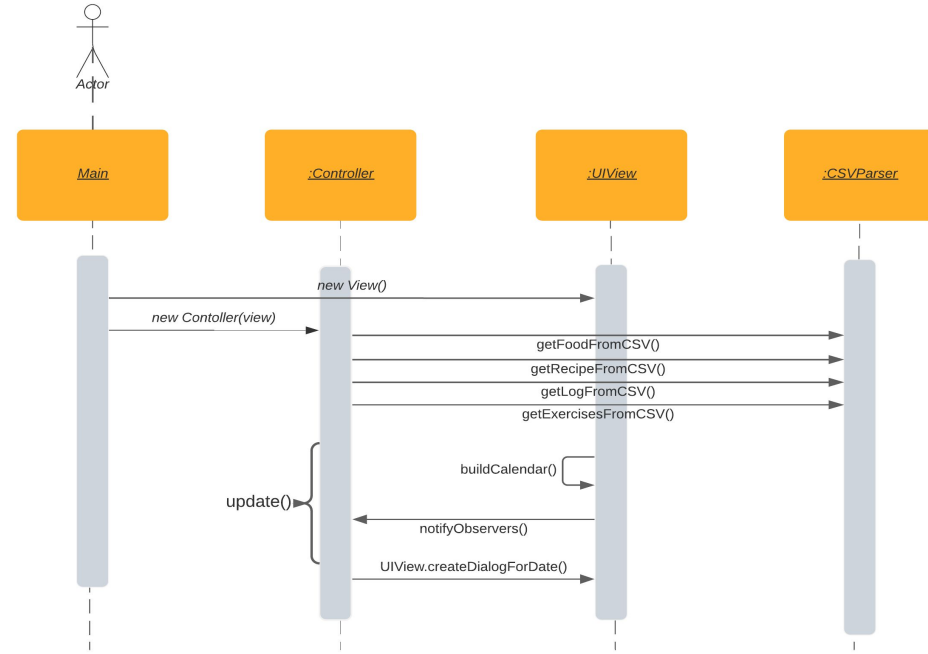


Overview of system design

Sequence diagram for changing any value in a food entry (logs, recipes, and exercises follow the same logic, with different methods)



Sequence Diagram for Showing Data for Specific Date.



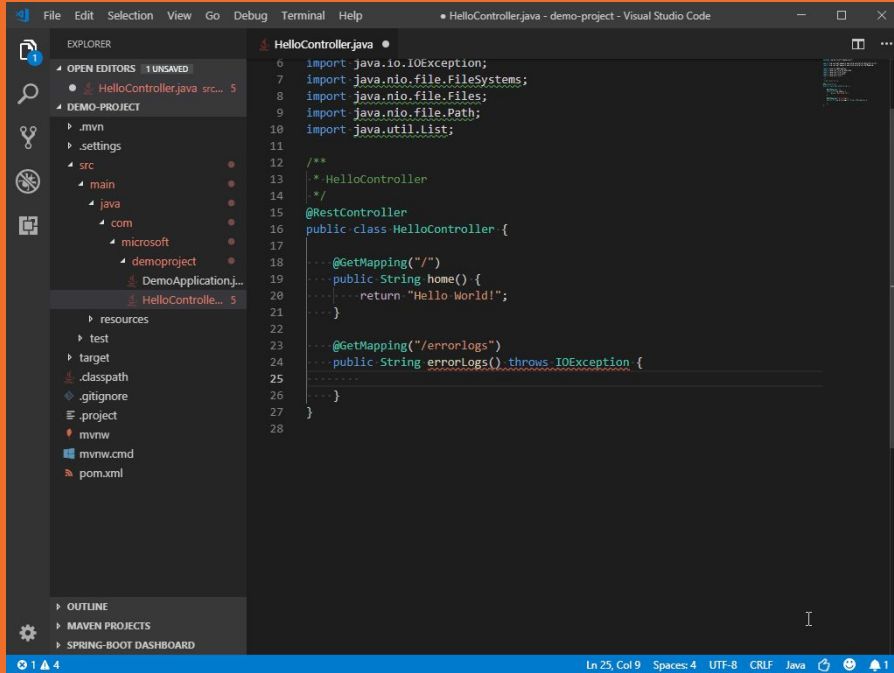
Design Implementation (Good)

- Our strong points of the design would be using
 - MVC - Properly modeled.
 - View requests info, which is supplied by the model
 - Controller handles interaction between View and Model
 - Model handles all the business logic (“meat & potatoes”)
 - (Added in phase 2) Observer pattern - Allows the Controller to be notified of changes in the View
 - Composite pattern - Allows us to handle the recursive nature of food and recipe objects while being easily maintainable
- Medium cohesion but very low coupling



Design Implementation (Bad)

- Controller will eventually become a god class if more features continue to be added.
 - would be forced to add additional controller classes
- Miscommunication
 - getting our code to work together properly could be challenging
- Better Generalizations / Abstractions
 - could've made some of our functions more generic (lazy coding).
- We wish we had higher cohesion
 - With more time we could've made the code much more scalable for future releases / views



```
File Edit Selection View Go Debug Terminal Help • HelloController.java - demo-project - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
HelloController.java src... 5
DEMO-PROJECT
.mvn
.settings
src
  main
    java
      com
        microsoft
          demoproject
            DemoApplicationJ...
            HelloControle... 5
resources
test
target
.classpath
.gitignore
.project
.mvnw
mvnw.cmd
pom.xml

OUTLINE
MAVEN PROJECTS
SPRING-BOOT DASHBOARD

HelloController.java
6 import java.io.IOException;
7 import java.nio.file.*;
8 import java.nio.file.Files;
9 import java.nio.file.Path;
10 import java.util.List;
11
12 /**
13  * HelloController
14  */
15 @RestController
16 public class HelloController {
17
18     ... @GetMapping("/")
19     ... public String home() {
20     ...     return "Hello World!";
21     ... }
22
23     ... @GetMapping("/errorlogs")
24     ... public String errorLogs() throws IOException {
25     ...
26     ... }
27
28 }
```

Ln 25, Col 9 Spaces: 4 UTF-8 CRLF Java



Demonstration!