NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_GROUP 5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Diet Manager V2 – Rubric

SWEN.383 SW Design Principles and Patterns

Implementation Functionality Grade for Diet Manager V2

(20% of Overall Project Grade)

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Tasks** | **Points** | **Score** |
| ☐ 1 | Load a foods.csv file with only basic foods and an empty log.csv file.  User can see an empty log for today with default calories and weight.  User can view the basic foods loaded.  This works and it’s done via the SecondScreenController->Controller who calls the loadFood and loadDaily on the Loader which fills in the data for the foods and the current Daily Log info. Afterwards the GUI is updated. Everything here works. | 50 (50) |  |
| ☐ 2 | Level 1 plus add basic food(s) in the database to today's log with appropriate  updating of the nutrients consumed.  This is done vis the AddFoodController->Controller which adds the food using the WriterFacade and afterwards loads with the Loader and updates the GUI based on the new state. Everything works here. | +10 (60) |  |
| ☐ 3 | Level 2 plus add new basic food(s) to the food database. This implies the  ability to then add such basic food(s) to today's log as in level 2.  After you choose the food the SecondScreenController->Controller gets the food from the Loader’s collection and adds it with it’s amount onto the DailyInformation and updates the GUI. Everything works here. | +5 (65) |  |
| ☐ 4 | Level 3 plus loading and viewing a foods.csv file with recipes.  With the observer patter this was done nicely. The Loader loads food regardless of what they are. Firstly, all the basics and then goes on to create loaded recipes from the already existing basics. | +5 (70) |  |
| ☐ 5 | Level 4 plus add recipe(s) as well as basic food(s) to today's log with  appropriate updating of nutrients consumed.  This works via the SecondScreenController->Controller. The user has a dynamically populated drowpdown which if he chooses a recipe gets the recipe from the Loader’s collection and adds it onto the DailyInformation. This works. | +5 (75) |  |
| ☐ 6 | Level 5 plus add new recipe(s) to the food database. This implies the ability to  then add such recipe(s) to today's log as in level 5.  This also works via the AddRecipeController->Controller which allows the user to stack the basic foods and their amounts onto an abbreviated string which is written afterwards using the WriterFacade and then loaded by the Loader. This works. | +5 (80) |  |
| ☐ 7 | Level 6 plus the ability to read a non-empty log.csv file, to navigate to  different days in the log, and to add foods to days other than today.  All of this is done via Loader which populates the DailyInformation based on the day chosen. After another day is chosen it does it again. This works. | +5 (85) |  |
| ☐ 8 | Level 7 plus the ability to save the log and food database back to the log.csv  and foods.csv files, quit the program, and they restart it by loading the  information in the saved files without error.  This also works via Loader->DailyInformation. After saving the Façade saves the DailyInformation and the Loader loads it again. It’s the same process on save and on open. This works. | +5 (90) |  |
| ☐ 9 | Level 8 plus the ability to support the new program feature - daily exercises  The exercise works using all the new Exercise classes like the factory etc. Controller/Loader/DailyInformation are updated to handle the Exercise and there is a new screen controller for adding the exercises to the .csv file. | +10 (90) |  |
| SUBTOTAL: | | **100** |  |

To receive any credit for level N, the preceding levels must be sufficiently functional to test level N. In general, this means previous levels must work without failure when the user enters normal (non-error) data.

Remember, the solution should apply design patterns as required. The final grade will be adjusted based on how good you have applied the pattern. You may loose up to 25% of the final project grade in this regard.