Final Project Report

Student Name: 박지연(Jiyeon Park)
Student ID: 2021311867

1. Briefly describe the project purpose:

The program I made is "Shelf Life Manager". We check the shelf life of ingredients only when we shop at the mart, and don't usually check it. So, in the end, the ingredients are often thrown away. To solve this problem, this program was devised. By entering the shelf life of purchased ingredients into the program only once, you can easily manage the shelf lifes of ingredients without constantly checking the refrigerator. It also provides recipes for use with ingredients that are nearing their shelf life.

2. Draw the logic flow of the program (with flowchart):

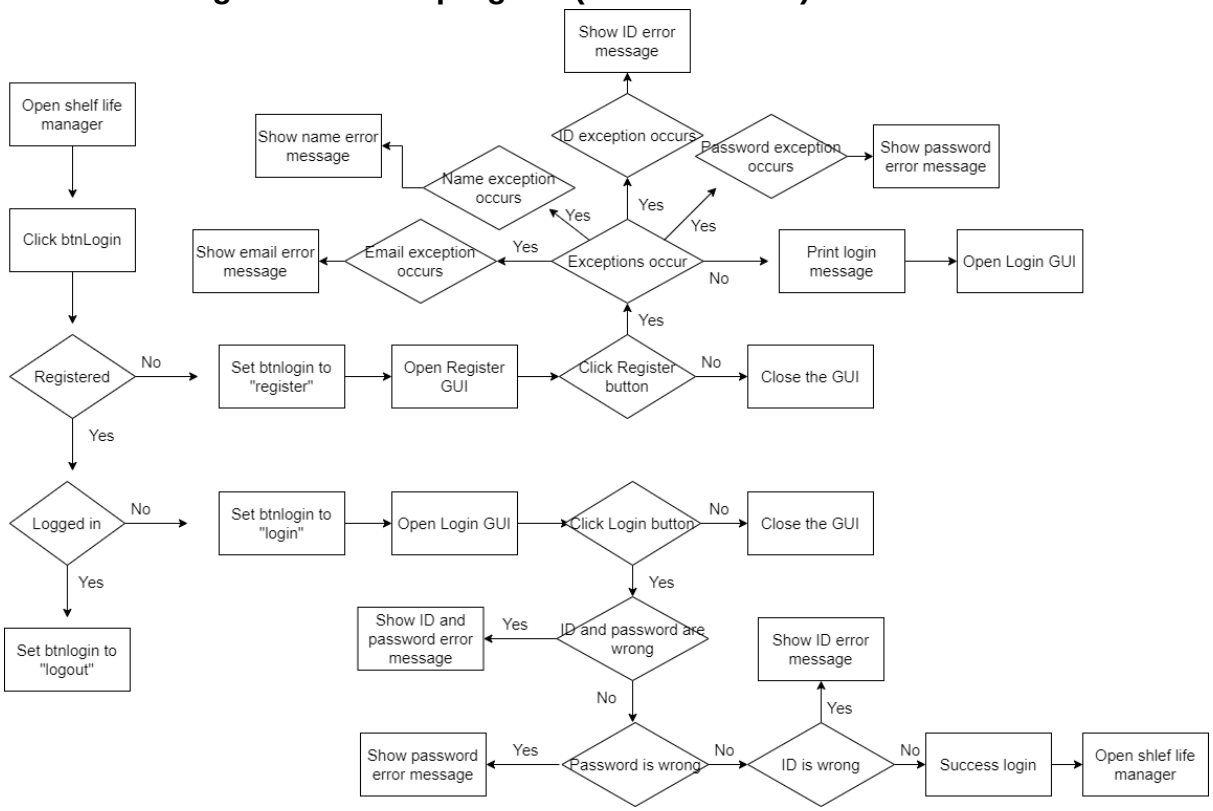

Figure 1: Sign up and Log in

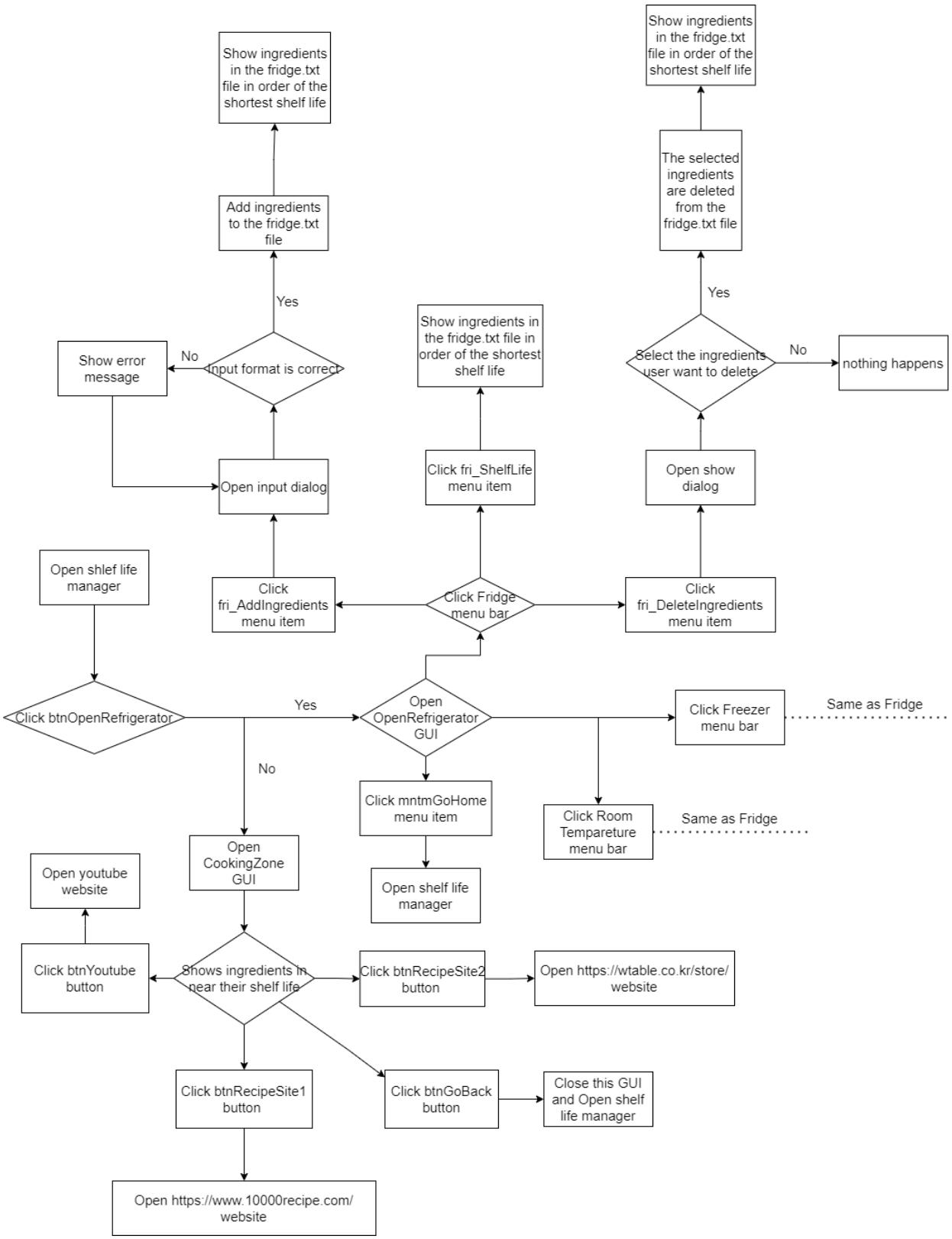

Figure 2: Main Functionality

3. Provide screenshots for each screen with brief description:

When you start the program for the first time, a screen like figure3 appears. In the upper left corner it says "Register", which means that user have not yet registered as a member. In this state, if you press the button that says "Open the

refrigerator" or "Cooking with Ingredients in the Refrigerator", a dialog like figure4 appears. This is to restrict the use of services by non-members. If user click the Register button, the member registration window as shown in figure5 appears. If Email, Name, ID format is wrong, or password is not entered or entered differently, a dialog like figure6 is displayed. When the Cancel button is clicked, a dialog like figure 7 is displayed, and when login is completed, a dialog like figure 8 is displayed and a login window like figure 9 is opened. If login fails, cancel button is pressed, or login is successful, a dialog like figure10 is displayed. When user log in, the button in the upper left corner of the main home also changes to "logout", and the user name appears as shown in Figure 11. In this state, if user press the logout button, a dialog as shown in figure 12 appears. And if user try to use the service in the logout state, a dialog as shown in figure 13 appears and use is restricted.

If you click the "Open the refrigerator" button while logged in, a window like figure 14 appears. The fridge, freezer, and room temperature menu bar has three menu items: shelf life, add ingredients, and delete ingredients. If user click the "Add ingredients" menu, a dialog as shown in figure 15 will appear. If the input format is incorrect or the month or day is out of the input range, a dialog as shown in figure 16 appears. If user add ingredients, the added ingredients will be displayed in order of shelf life as shown in Figure 17. If user click the delete ingredients button, a dialog as shown in Figure 18 appears. If user click the ingredients that user want to delete and click the OK button, the ingredients are deleted. If user press the shelf life button, the ingredients currently stored in the refrigerator are displayed. When the user presses the go main home menu which is an item in the go back menu, it returns to the main home.

Pressing the "cooking with ingredients in the refrigerator" button opens Figure 19 and the CookingZone GUI. Here, ingredients with an shelf life of equal to or less than 7 days are displayed, and there is a button that links to the recipe website or YouTube, so user can cook and eat with the remaining ingredients.

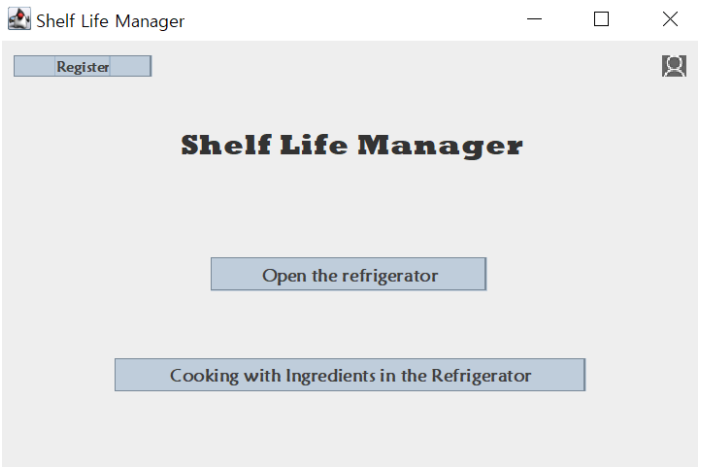

Figure 3: Screen when running the program for the first time

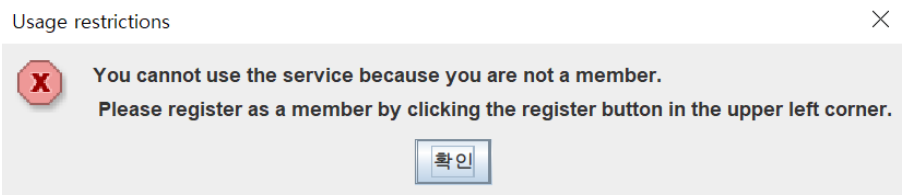

Figure 4: Restrictions on use for non-members

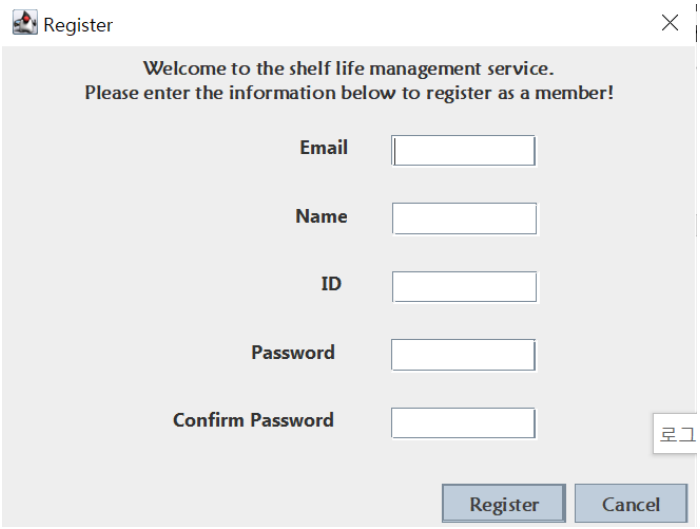

Figure 5: Refister GUI

Figure 6: Register Format Error

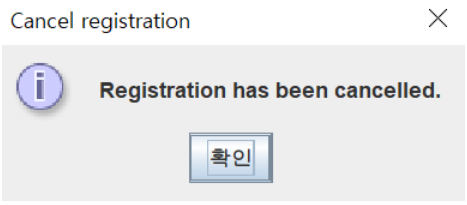

Figure 7: Cancel Registeration

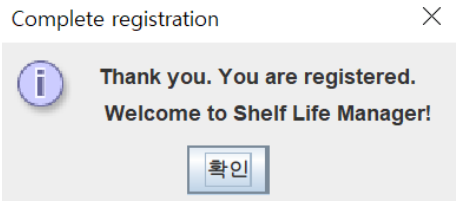

Figure 8: Complete Registration

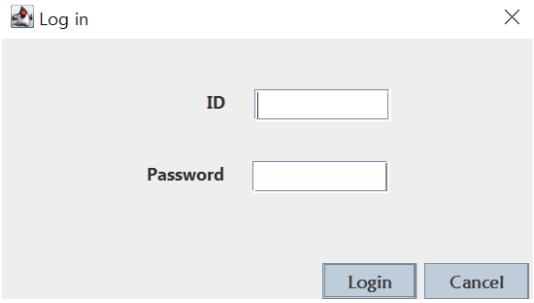

Figure 9: Login GUI

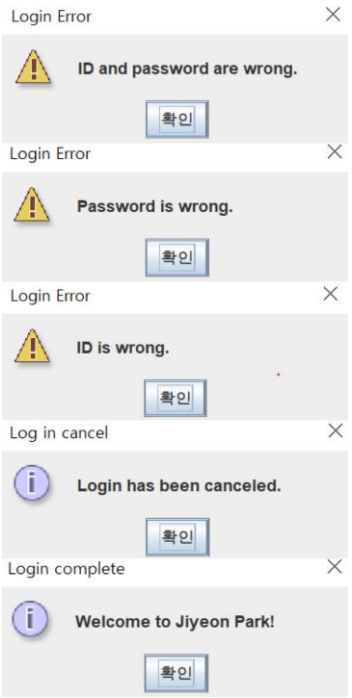

Figure 10: Collection of Login Dialog

Figure 11: Logged in

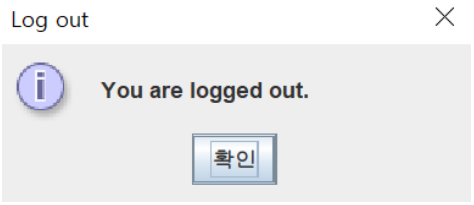

Figure 12: Logout Dialog

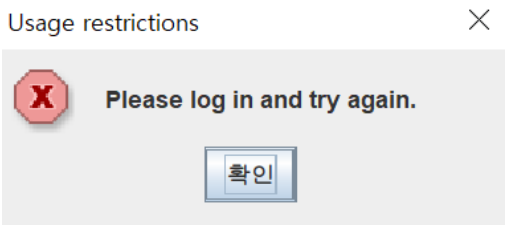

Figure 13: Restrict use to people who are not logged in

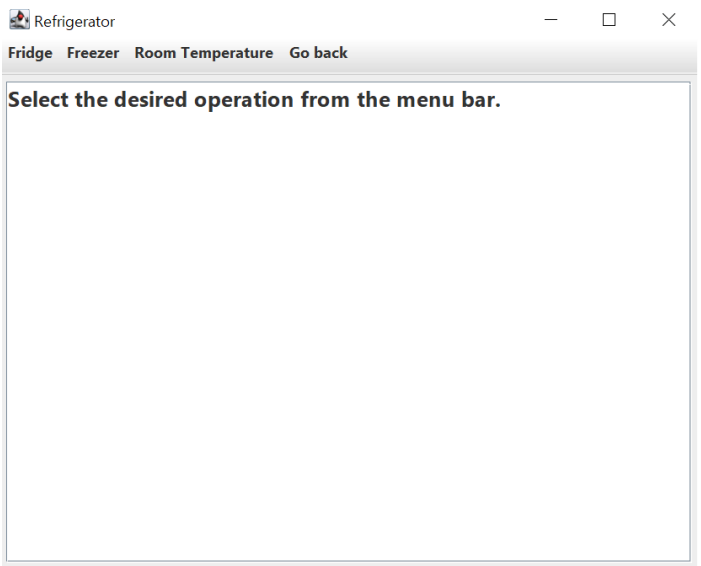

Figure 14: Refrigerator GUI

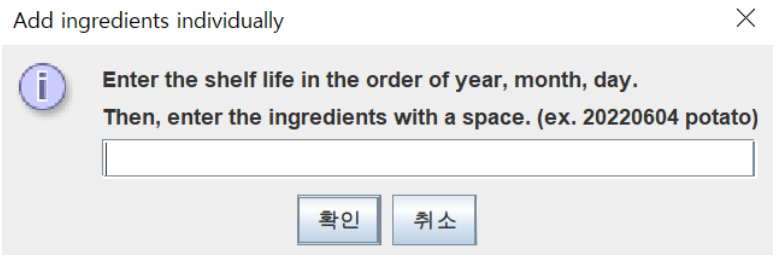

Figure 15: Add ingredients

Figure 16: Incorrect input format

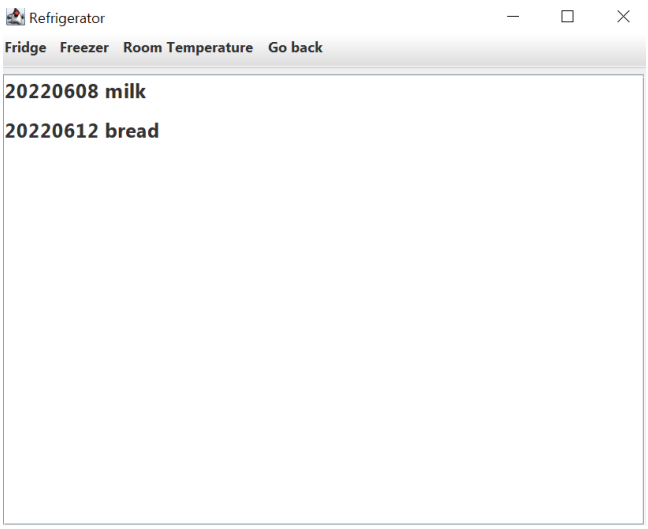

Figure 17: Add ingredient

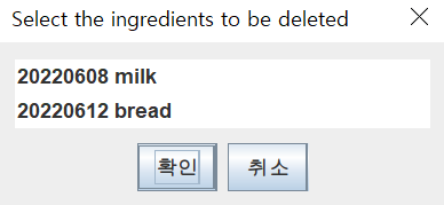

Figure 18: Delete ingredients

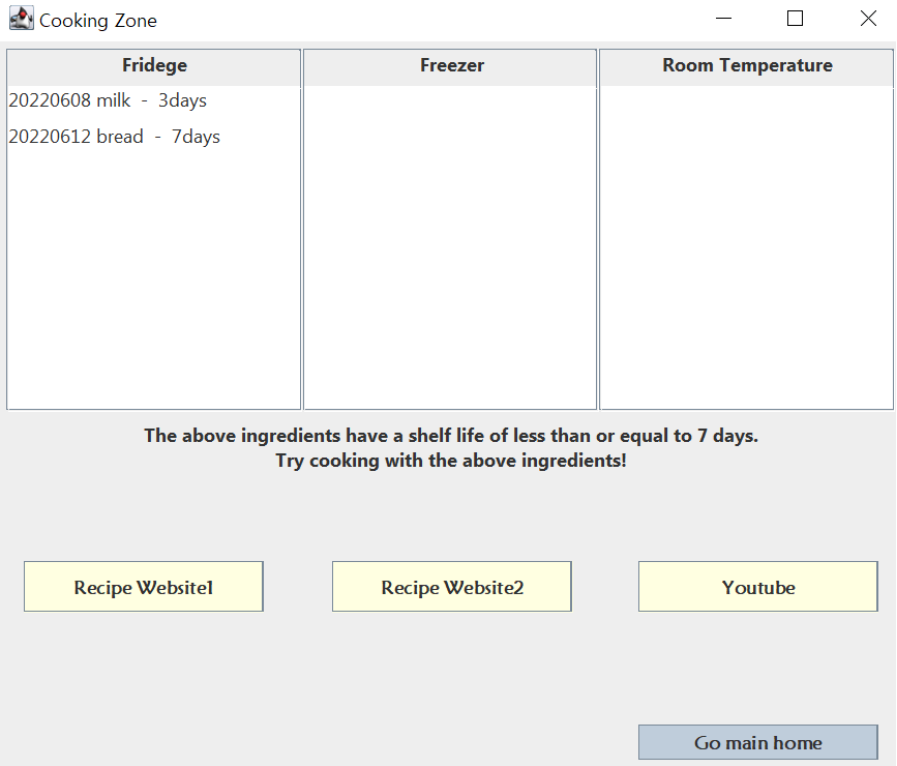

Figure 19: CookingZone GUI

4. Explain the code of the main functionalities

The main functionality of my program is input and output of files. It reads the files corresponding to fridge, freezer, and room temperature and puts it in the array list. Then, the array list is sorted in ascending order and the sorted list is written back to the file.

```java
/**
 * Read shelf life and ingredients written on the file
 */
@Override
public void readFile() {
    try {

        // To read a file whose name is the return value of calling function
        // getSubclassFileName()
        FileInputStream fileObject1 = new FileInputStream(getSubclassFileName());
        Scanner scanner = new Scanner(fileObject1);

        // reads the value from the file and puts it in the array list
        while (scanner.hasNext()) {
            ingredient.add(scanner.nextLine());
        }
        scanner.close();
        Collections.sort(ingredient); // sort by shelf life in ascending order
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

/**
 * Write the shelf life and ingredients on the file
 */
@Override
public void writeFile() {
    try {

        // To write a file whose name is the return value of calling function
        // getSubclassFileName()
        FileOutputStream fileObject2 = new FileOutputStream(getSubclassFileName(), false); // overwrite the file
        PrintWriter writer = new PrintWriter(fileObject2);

        Iterator iterator = ingredient.iterator(); // for visiting through all values in an arraylist
        // write the values of the array list to a file
        while (iterator.hasNext()) {
            writer.println(iterator.next());
        }
        writer.close();

    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Figure 20: The Code of Main Functionality

5. Explain what is included in your project and why it is used (Polymorphism, Inheritance, File I/O, etc)

1) **GUI**
   Used to implement the Shelf Life Manager. I made a program using various GUIs such as the login GUI, register GUI, refrigerator GUI, and cooking GUI.
2) **Inheritance**
   Since the places where ingredients can be stored are fridge, freezer, and room temperature, a refrigerator that shows common characteristics of them was created. And fridge, freezer, and room temperature classes inherited refrigerator class.
3) **Polymorphism**
   I had to use fridge, freezer and room temperature in the OpenRefrigerator GUI. Therefore, polymorphism is used for convenience.
4) **File I/O**
   - It was used to store user information when registering as a member.

- It was used to check whether the ID and password match when logging in.
- Used to figure out whether or not to log in.
- It was used to show the user's name on the main home page.
- Used to put ingredients into the refrigerator and delete them.
- Used to sort ingredients by expiration date.
- It was used to show how much of an ingredient's expiration date is left.