Refrigerator Raider
by Ant Cassetta

1. Summary Description

Project Description:

This project will provide customers with a helpful tool for managing their home or work refrigerators with helpful reminders for when items run low. It will be a web-based application so that users can access the application on almost any device and on-the-go. The estimated cost for this project is $50,000, but will generate revenue through future partnerships with brands and advertising.

Product Characteristics and Requirements:

- Inputting of Items: Users will be able to input the contents of their refrigerator and add to it as new items are purchased
- Decrementing of Quantity: Users will be able to reduce the quantity of items as they are consumed or expire
- Multiple Refrigerators: Users will be able to view multiple refrigerators for their families or businesses
- Database Creation: A database must be created to store the information about customers, refrigerators, and items
- Security: Sensitive customer information must be protected from all possible malicious threats, both internal and external.
- The functionality must be accessible using a standard Internet browser on Windows, Mac, and mobile operating systems.
- The functionality must be available 24 hours a day, 7 days a week, with one hour per year downtime for system maintenance, as appropriate.

Intro:

I am taking a project concept from a graduate level course at Boston University called "Refrigerator Raider". This project's design and concept was born from a team of five, myself included, and progressed over the course of the 7-week term. Each team member was assigned one or more roles on the team, I was the Testing Engineer. I also contributed to the design through generating wire-frames, use-cases, user personas.

Refrigerator Raider proof of concepts were originally written in JavaScript by two of the team members. The application you will find here is written in Java by me utilizing our teams collaborative planning and design.

Data Structures:

I will be using source code for data structures from:
Michael T. Goodrich, Roberto Tamassia, and Michael T. Goldwasser (2014).
Data Structures and Algorithms in Java – 6th Edition.
John Wiley & Sons
Specifically, at this time I am using the LinkedPositionalList.java (see Appendix1)

Using Java, it is my intention to implement the data structures used for storing inventories and their respective items. The business logic of the application, as outlined by the functional

requirements, which will mostly pertain to accesses and updating inventory data (See Appendix 2). This effort will include error checking to ensure robustness and enable accurate user input.

Demo Video:

Please find a video demo of the application here

2. Use Cases

Use Case UC-01: Edit Refrigerator Contents – Add New Item		
ID	UC-01	
Brief description	This use case describes and validates the workflow of requirement(s) F-02. This use case walks through the workflow of adding a new item to the refrigerator.	
Primary actors	Refrigerator Owner, Business Manager, Food Pantry Manager	
Secondary actors	n/a	
Pre-conditions	1. Refrigerator Owner must be logged in 2. Refrigerator Owner must be currently on the refrigerator page in which they wish to add a new item to	
Main flow	**Actor:**	**System:**
	1. Refrigerator Owner clicks "Add Item"	2. System displays a form with fields for multiple refrigerator item characteristics. Including Item Category: Dairy, Meat, Produce, Misc.
	3. Refrigerator Owner selects the relevant Category inputs the relevant information and clicks "Save List"	4. System commits data to the database, displays a confirmation message, and redirects the user to the refrigerator contents page
Post-conditions	Refrigerator Owner is returned to the refrigerator contents page.	
Alternate flows	3.2 Refrigerator Owner clicks "Cancel"	4.2 System redirects the user to the refrigerator contents page and does not commit data to the database.
	3.3 Refrigerator Owner clicks "Reset"	4.3 System clears all fields

Use Case UC-02: Edit Refrigerator Contents – Decrement Item Quantity		
ID	UC-02	
Brief description	This use case describes and validates the workflow of requirement(s) F-03. This use case walks through the workflow of decrementing an existing items quantity in the refrigerator.	
Primary actors	Refrigerator Owner, Business Manager, Food Pantry Manager	
Secondary actors	n/a	
Pre-conditions	1. Refrigerator Owner must be logged in 2. Refrigerator Owner must be currently on the refrigerator page in which they wish to decrement an item quantity.	
Main flow	**Actor:**	**System:**
	1. Refrigerator Owner clicks the down arrow button	2. System displays decrements the item by 1 unit.
	3. Refrigerator Owner clicks "Save List"	4. System commits data to the database, displays a confirmation message, and redirects the user to the refrigerator contents page
Post-conditions	Refrigerator Owner is returned to the fridge contents view.	
Alternate flows	3.2 Refrigerator Owner hand keys a value into the quantity field and clicks "Save List"	4.2 System commits data to the database, displays a confirmation message, and redirects the user to the refrigerator contents page.
	3.3 Refrigerator Owner clicks "Reset"	4.3 System clears all fields

Use Case UC-03: View Refrigerator Contents		
ID	UC-03	
Brief description	This use case describes and validates the workflow of requirement(s) F-06. This use case walks through the workflow of viewing refrigerator contents that have been loaded from the database.	
Primary actors	Refrigerator Owner, Business Manager, Food Pantry Manager, Refrigerator Raider	
Secondary actors	n/a	
Pre-conditions	1. Refrigerator Owner must be logged in	
Main flow	**Actor:**	**System:**
	1. Refrigerator Owner selects the desired Refrigerator and clicks "View Fridge Contents"	2. System loads the data of the refrigerator and displays the contents.
	3. Refrigerator Owner clicks Refrigerator Raider logo.	4. System redirects the user to the home page
Post-conditions	Refrigerator Owner is returned to the home page.	

Use Case UC-04: Log in.		
ID	UC-04	
Brief description	This use case describes and validates the workflow of requirement(s) F-10. This use case walks through the workflow of a user logging into Refrigerator Raider at the appropriate level of security.	
Primary actors	Refrigerator Owner, Business Manager, Food Pantry Manager, Fridge Raider	
Secondary actors	n/a	
Pre-conditions	1. End user must already have an established account. 2. End user, must be currently on the RefrigeratorRaider Home page.	
Main flow	**Actor:**	**System:**
	1. Refrigerator Owner clicks “Log in”.	2. System displays a form with fields for username and password.
	3. Refrigerator Owner inputs the relevant information and clicks “Log in”.	4. System verifies the username exists and the password is correct.
	5.	6. If verified successfully System displays Refrigerator Raider Main Menu.
Post-conditions	Refrigerator Owner is returned to the refrigerator contents page.	
Alternate flows	3.2 Refrigerator Owner clicks “Cancel”	4.2 System redirects the user to the refrigerator contents page and does not commit data to the database.
	3.3 Refrigerator Owner clicks “Reset”	4.3 System clears all fields

Use Case UC-05: Items out of stock.		
ID	UC-05	
Brief description	This use case describes and validates the workflow of requirement(s) F-08. This use case walks through the workflow of the system reporting to the Refrigerator Owner which items are out of stock. (Quantity = 0).	
Primary actors	System	
Secondary actors	Refrigerator Owner, Business Manager, Food Pantry Manager	
Pre-conditions	1. At least one Refrigerator must have an Item with a quantity of 0.	
Main flow	**Actor:**	**System:**
	1. Refrigerator Owner follows Log In procedure (see UC-04)	2. System displays Refrigerator Raider Main Menu.
	3. Refrigerator Owner selects the desired Refrigerator and clicks "View Fridge Contents" (see UC-03)	4. System loads the data of the refrigerator and displays the contents. (See UC-03)
	5.	6. System checks inventory for Items with a Quantity of zero, and displays a report of each Item name and a sum total of items out of stock.
	7. Refrigerator Owner clicks "Okay"	8. System closes the report and returns the end user to the Refrigerator Contents page
Post-conditions	Refrigerator Owner is returned to the Refrigerator Contents page.	
Alternate flows	3.2 Refrigerator Owner clicks "Add to Shopping List"	4.2 System copies out of stock items over to shopping list with a default quantity of 1

3. Requirements

ID	Actor	Use Case	Requirement/User Story
F-01	Refrigerator Owner	Create Refrigerator	As a Refrigerator Owner, I want to create my refrigerator, so that I can add items and track my inventory.
F-02	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to add items to my refrigerator, so that I can account for items I've purchased.
F-03	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to decrement the quantity of an item, so that I can account for items consumed
F-04	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to remove an item from my refrigerator that I will not purchase again, so that I will not be alerted in the future.
F-05	Refrigerator Owner	Create Shopping List	As a Refrigerator Owner, I want to create a shopping list, so that I can know what to purchase when I go to the store.
F-06	Refrigerator Owner	Create Refrigerator Raider User	As a Refrigerator Owner, I want to create a Refrigerator Raider user, so that I can provide read only access to my fridge for children or employees.
F-07	Refrigerator Owner/ Refrigerator Raider	View Refrigerator Contents	As a Refrigerator Owner and Raider, I want to view the contents of my fridge, so that I can be aware of what I have in my fridge.
F-08	Refrigerator Owner	Create and edit nutrition information	As a Refrigerator Owner, I want to create and edit nutritional information for the food in my fridge, so that I can stock the fridge with healthy items.
F-09	Refrigerator Owner/ Refrigerator Raider	View nutritional information	As a Refrigerator Owner and Raider, I want to view the nutritional information about my fridge contents, so that I can enhance my health choices.
F-10	Business Manager	Edit Supplier Information	As a Business Manager, I want to create and edit supplier information, so that I can contact them quickly if I need to place an order.
F-11	Food Pantry Manager	Create Volunteers List	As a Food Pantry Manager, I want to create a list of food pantry volunteers, so that I may be able to contact them all at once.
F-12	Food Pantry Manager	Contact Volunteers with Needed Items	As a Food Pantry Manager, I want to create a list of needed items for volunteers, so that they can assist on obtaining these needed items for the food pantry.
F-13	Refrigerator Owner	Place Order	As a Refrigerator Owner, I want to place an order, so that I can replenish items running low.

F-14	Quantity	Notify User of Low Quantity Items	As a Refrigerator Owner, I want to be notified when items are running low, so that I can create a shopping list or place an order to replace them.
F-15	Time	Notify User when Item Expires	As a Refrigerator Owner, I want to be notified when items are expiring, so that I can create a shopping list or place an order to replace them.
F-16	Amazon Fresh API	Accept Order	As an external API, I want to connect with Refrigerator Raider over the internet, so that I may receive and confirm order information.
F-17	Amazon Fresh API	Accept Payment	As an external API, I want to securely connect with Refrigerator Raider, so that I can ensure privacy of information.
F-18	All Users	Account Log In	As a user, I want to securely log into my personal account, so I may edit my account and inventories as needed.

4. Packaging

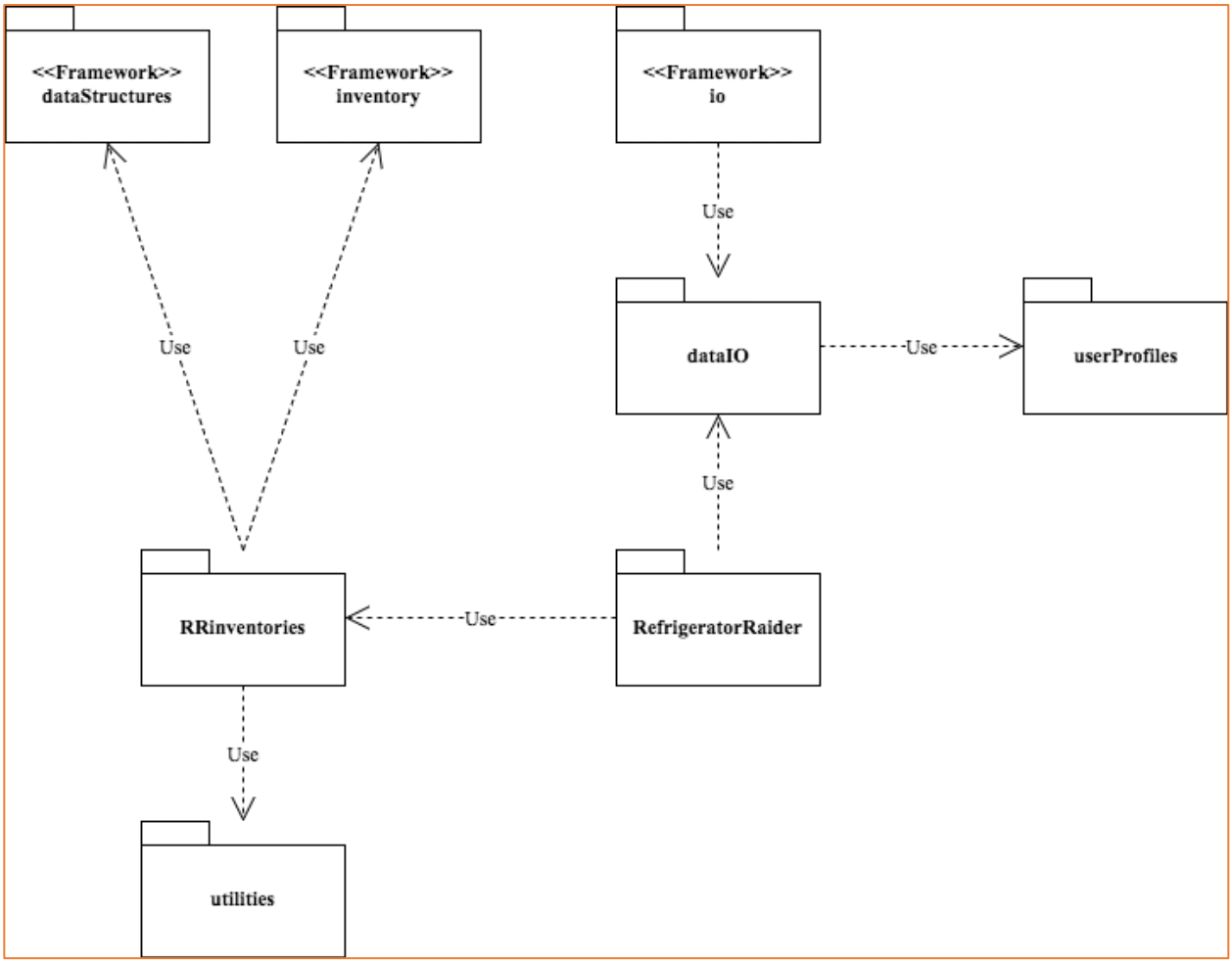

Figure 1: Package Diagram

5. Class Model

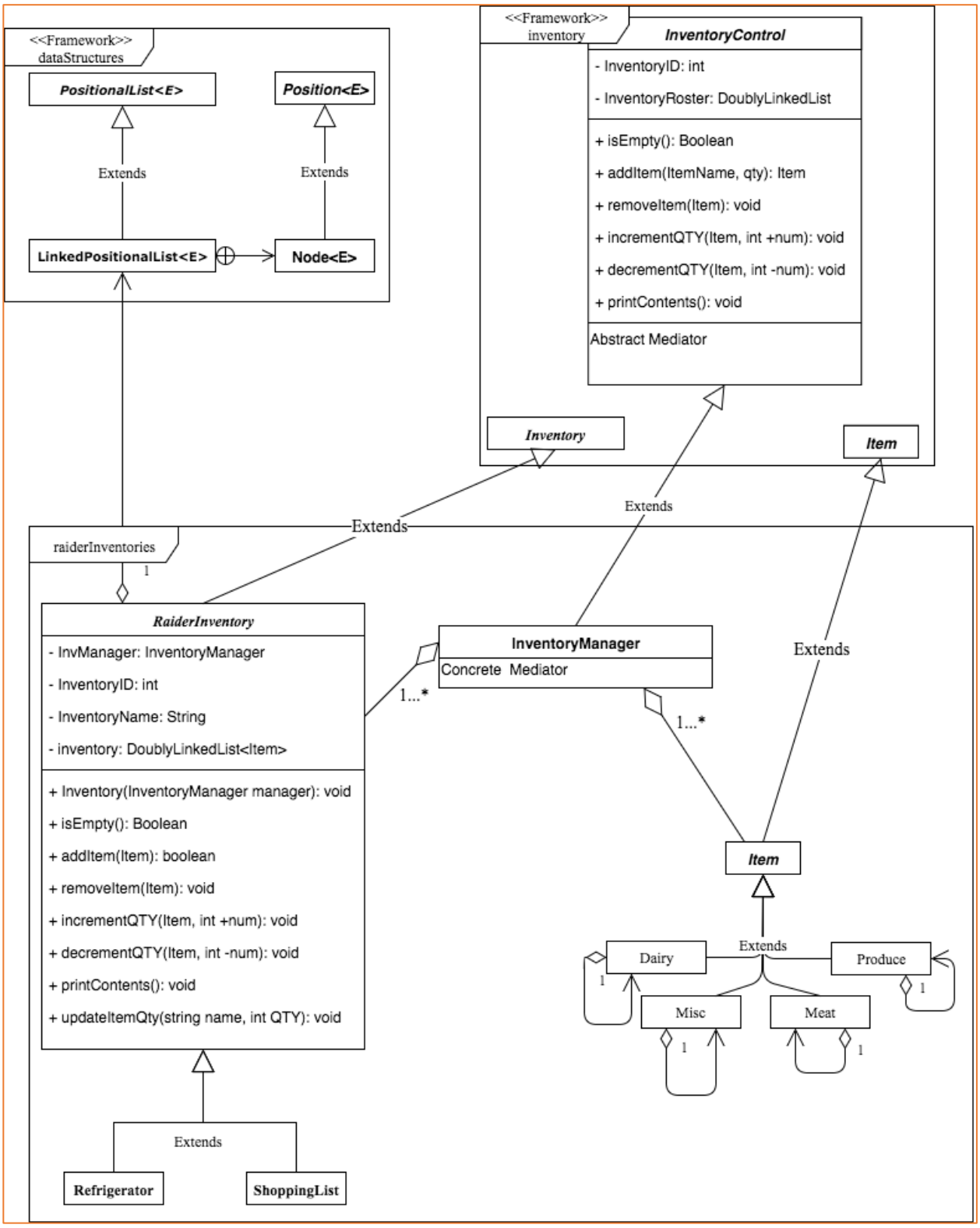

Figure 2: Class Diagram 1

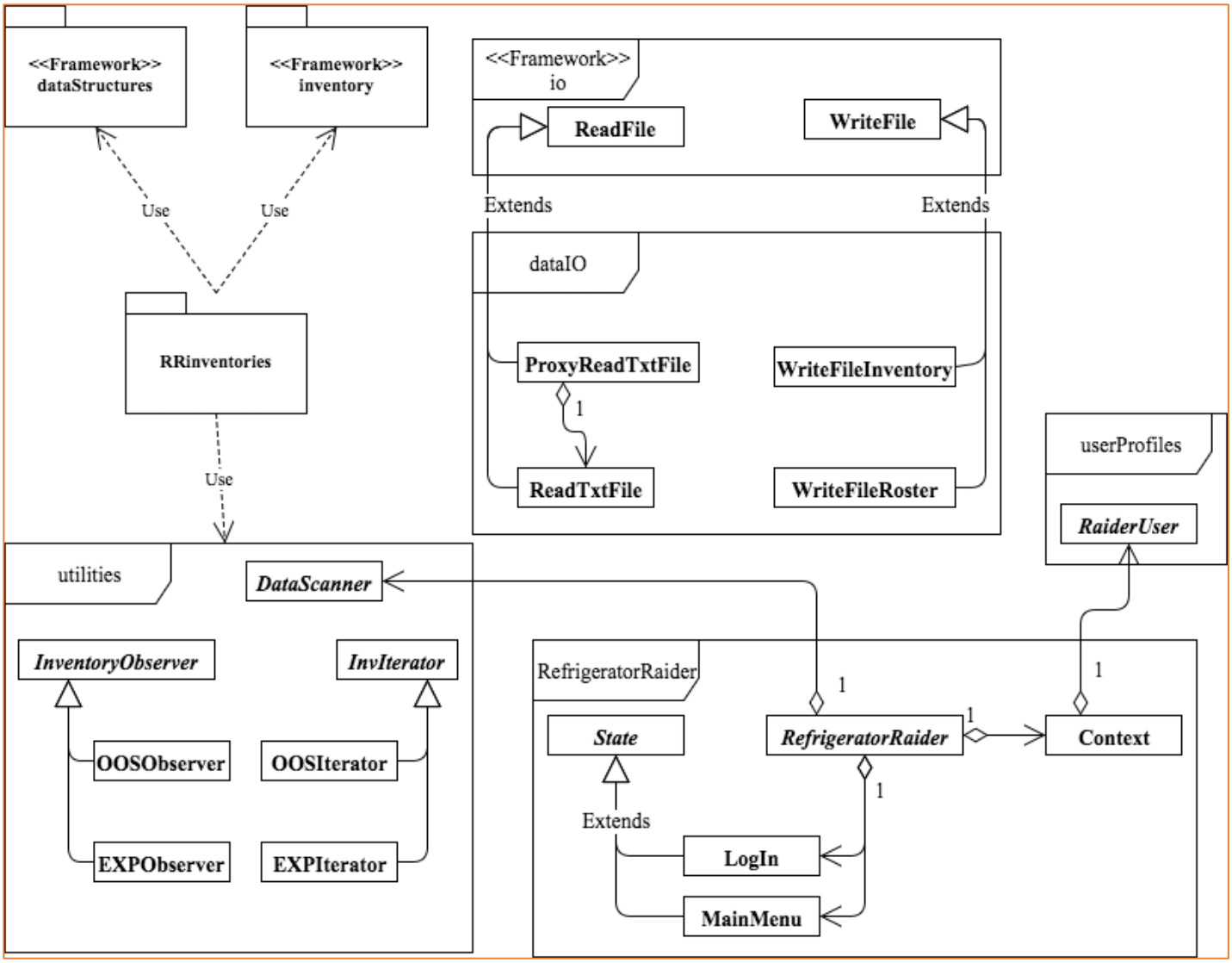

Figure 3: Class Diagram 2

6. State and/or Data Flow Model

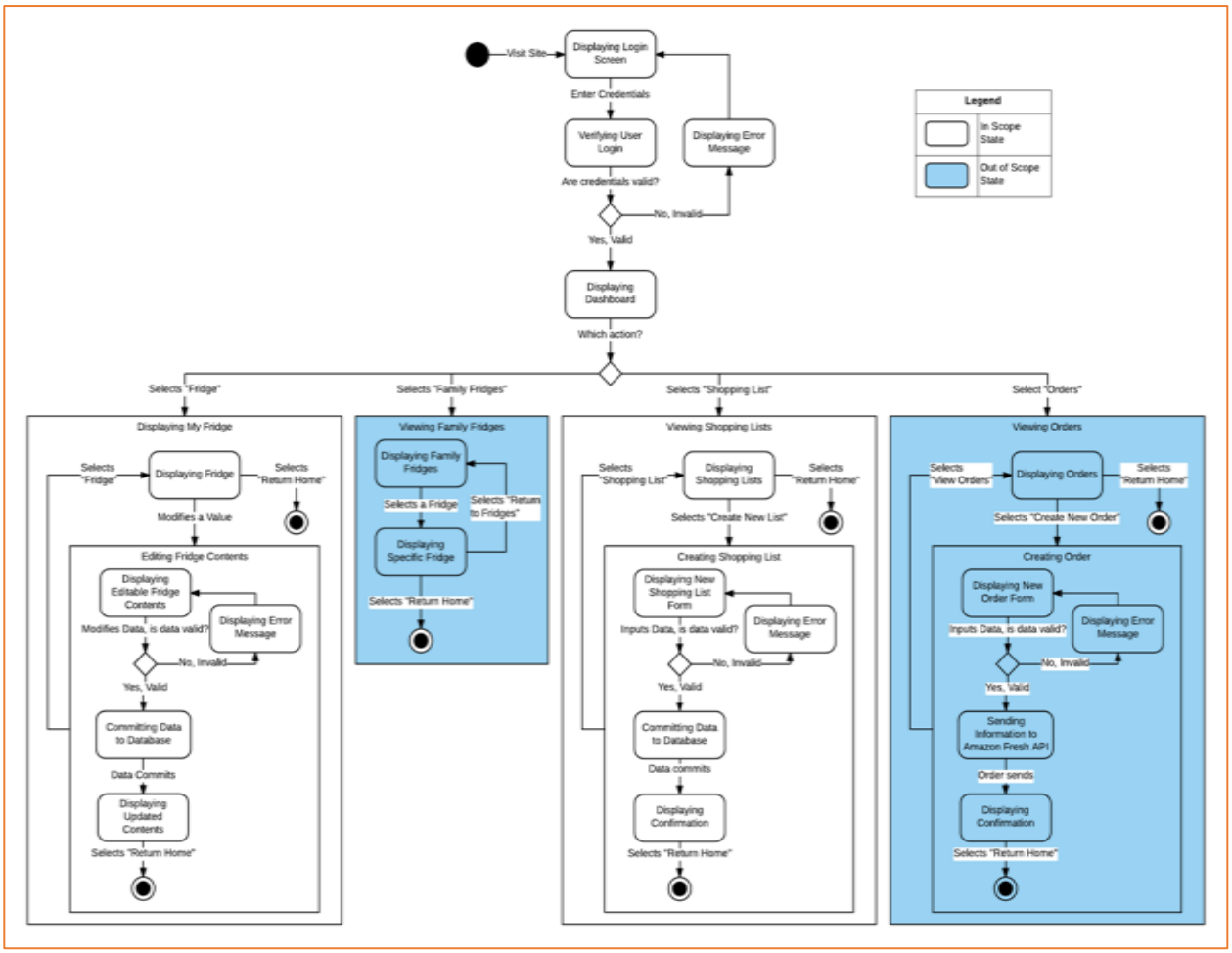

Figure 4: State Transition Diagram

7. Implementation: Source Code and Sample

Below is the main class of Refrigerator Raider. For the sake of space and easy viewing I have omitted Java docs and methods used purely for testing purposes. The main class of Refrigerator Raider runs the user interface and prompts the user to log in. Upon successful log in the users personal inventories are loaded from persistent storage. In this case, text files, then the system will wait for further instructions from the user.

```
public class RefrigeratorRaider extends Application {

	//private Scanner scan;
	private Context context;
	private State logIn;
	//private State mainMenu;

	public void start(Stage primaryStage) {
		context = new Context();
		logIn = new LogIn();
		//set initial state
		context.setState(logIn);
		try {
			runLogin(primaryStage);
		} catch(Exception e) {
			e.printStackTrace();
		}//end try catch
	}//end start


	public void runLogin(Stage primaryStage) {
		try {
			Parent root = FXMLLoader.load(getClass().getResource(
								"/RefrigeratorRaider/Login.fxml"));
			Scene scene = new Scene(root,400,600);
			scene.getStylesheets().add(getClass().getResource("application.css")
									.toExternalForm());
			primaryStage.setScene(scene);
			primaryStage.show();

		} catch(Exception e) {
			e.printStackTrace();
		}//end try catch
	}//end runLogin


	public void runMenu(Stage primaryStage) {

		try {
			Parent root = FXMLLoader.load(getClass()
					.getResource("/RefrigeratorRaider/menu.fxml"));
			Scene scene = new Scene(root,400,600);
			scene.getStylesheets().add(getClass().getResource("application.css")
									.toExternalForm());
			primaryStage.setScene(scene);
			primaryStage.show();

		} catch(Exception e) {
			e.printStackTrace();
		}//end try catch
	}//end runMenu


	public static void main(String[] args) throws FileNotFoundException, IOException {
		launch(args);
	}//end main
}//end RefrigeratorRaider
```

Appendices

Appendix 1: Data Structures & Algorithms Source Code

The source code referenced can be found at the following URL under:
Browse by Resource > Source Code

http://bcs.wiley.com/he-bcs/Books?action=index&itemId=1118771338&bcsId=8635

Appendix 2: Requirements

ID	Actor	Use Case	Requirement/User Story
F-01	Refrigerator Owner	Create Refrigerator	As a Refrigerator Owner, I want to create my refrigerator, so that I can add items and track my inventory.
F-02	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to add items to my refrigerator, so that I can account for items I've purchased.
F-03	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to decrement the quantity of an item, so that I can account for items consumed
F-04	Refrigerator Owner	Edit Refrigerator Contents	As a Refrigerator Owner, I want to remove an item from my refrigerator that I will not purchase again, so that I will not be alerted in the future.
F-05	Refrigerator Owner	Create Shopping List	As a Refrigerator Owner, I want to create a shopping list, so that I can know what to purchase when I go to the store.
F-06	Refrigerator Owner	Create Refrigerator Raider User	As a Refrigerator Owner, I want to create a Refrigerator Raider user, so that I can provide read only access to my fridge for children or employees.
F-07	Refrigerator Owner/ Refrigerator Raider	View Refrigerator Contents	As a Refrigerator Owner and Raider, I want to view the contents of my fridge, so that I can be aware of what I have in my fridge.
F-08	Refrigerator Owner	Create and edit nutrition information	As a Refrigerator Owner, I want to create and edit nutritional information for the food in my fridge, so that I can stock the fridge with healthy items.
F-09	Refrigerator Owner/ Refrigerator Raider	View nutritional information	As a Refrigerator Owner and Raider, I want to view the nutritional information about my fridge contents, so that I can enhance my health choices.
F-10	Business Manager	Edit Supplier Information	As a Business Manager, I want to create and edit supplier information, so that I can contact them quickly if I need to place an order.

F-11	Food Pantry Manager	Create Volunteers List	As a Food Pantry Manager, I want to create a list of food pantry volunteers, so that I may be able to contact them all at once.
F-12	Food Pantry Manager	Contact Volunteers with Needed Items	As a Food Pantry Manager, I want to create a list of needed items for volunteers, so that they can assist on obtaining these needed items for the food pantry.
F-13	Refrigerator Owner	Place Order	As a Refrigerator Owner, I want to place an order, so that I can replenish items running low.
F-14	Quantity	Notify User of Low Quantity Items	As a Refrigerator Owner, I want to be notified when items are running low, so that I can create a shopping list or place an order to replace them.
F-15	Time	Notify User when Item Expires	As a Refrigerator Owner, I want to be notified when items are expiring, so that I can create a shopping list or place an order to replace them.
F-16	Amazon Fresh API	Accept Order	As an external API, I want to connect with Refrigerator Raider over the internet, so that I may receive and confirm order information.
F-17	Amazon Fresh API	Accept Payment	As an external API, I want to securely connect with Refrigerator Raider, so that I can ensure privacy of information.
F-18	All Users	Account Log In	As a user, I want to securely log into my personal account, so I may edit my account and inventories as needed.

Appendix 3: Video Demo

https://youtu.be/YkyXC01Piwo