CSCI 4448 Project Final Report

Name: Neil Nguyen

Project Name: Grocery Store Register

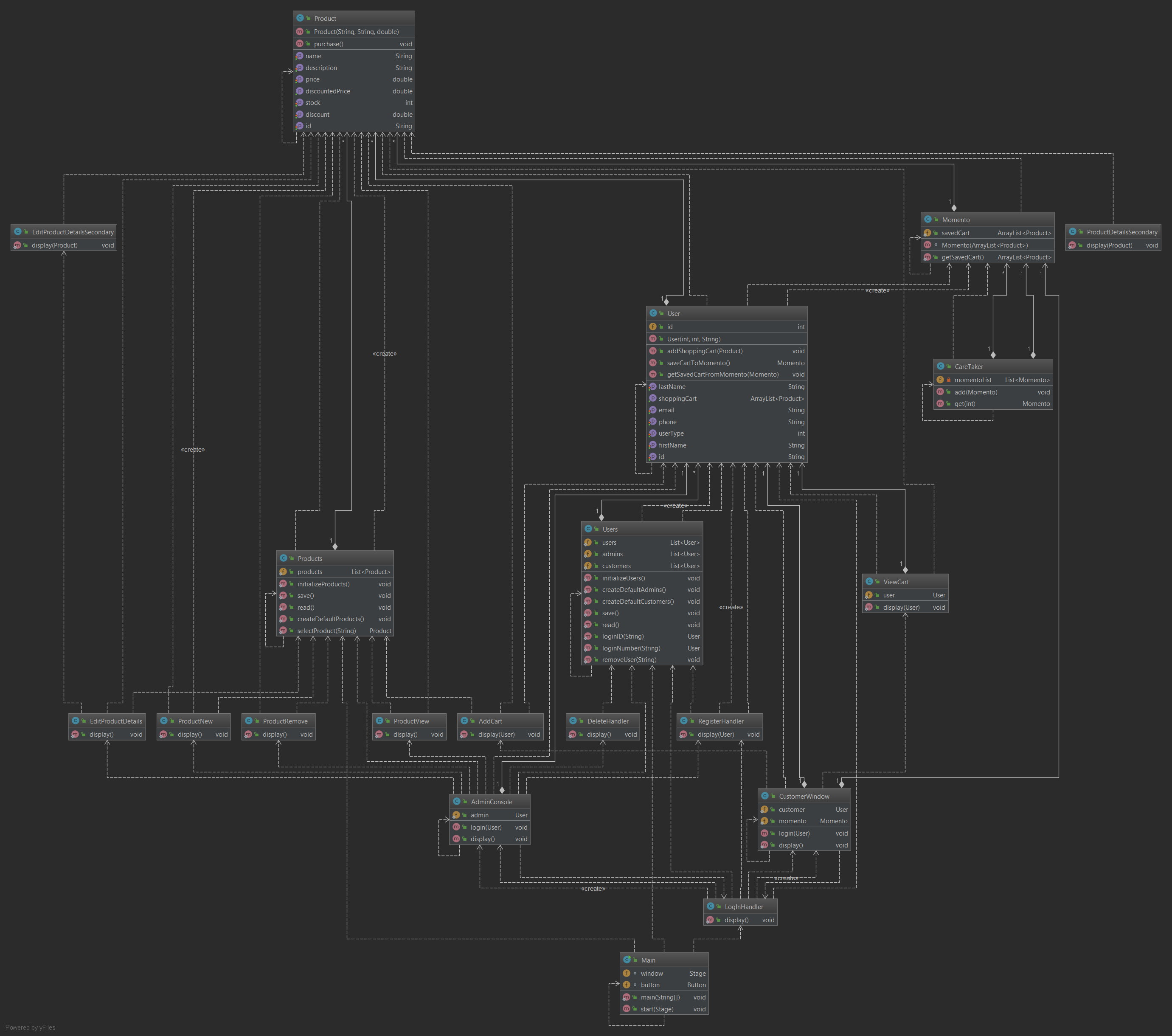
Features implemented:

|  |  |
| --- | --- |
| UR-001 | As a User, I want to be able to create a new User Profile |
| UR-002 | As an Admin, I want to be able to create a new customer profile |
| UR-003 | As an Admin, I want to be able to create a new admin profile |
| UR-004 | As an Admin, I want to be able to delete users |
| UR-005 | As an Admin, I want to be able to save the users even after the program is shut down |
| UR-006 | As an Admin, I want to be able to view products |
| UR-007 | As an Admin, I want to be able to edit products |
| UR-008 | As an Admin, I want to be able to save the products, even after program shut down |
| PR-001 | Products with discounts will have their final prices modified in accordance to the discount |
| UR-009 | As a User, I can log in with my user ID |
| UR-010 | As a User, I can log in with my phone number |
| UR-011 | As an Admin, I can add a new product |
| UR-012 | As an Admin, I can remove products |
| UR-013 | As a Customer, I want to be able to edit my profile |
| UR-014 | As a Customer, I want to be able to look up products and their details |
| UR-015 | As a Customer, I want to be able to add items to my shopping cart |
| UR-016 | As a Customer, I want to be able to view my shopping cart |
| UR-017 | As a Customer, I want to be able to view the price of my shopping cart |
| UR-018 | As a Customer, I want to be able to save my shopping cart for later |
| UR-019 | As a Customer, I want to be able to check out, which clears my shopping cart |
| UR-020 | As a User, I want to be able to log out and go back to the login screen |
| UR-021 | As a Customer, I want to be able to load a previously saved shopping cart |

Features I did not implement ☹

|  |  |
| --- | --- |
| UR-022 | As a Customer, I want to be able to easily remove single items from my shopping cart |
| UR-023 | As a Customer, I want to be able to add items to my shopping cart using Product Look-up |
| UR-024 | As a Customer, I want to be able to remove items from my shopping cart from the shopping cart view |
| UR-025 | As an Admin, I want to be able to manipulate products from the product list view |

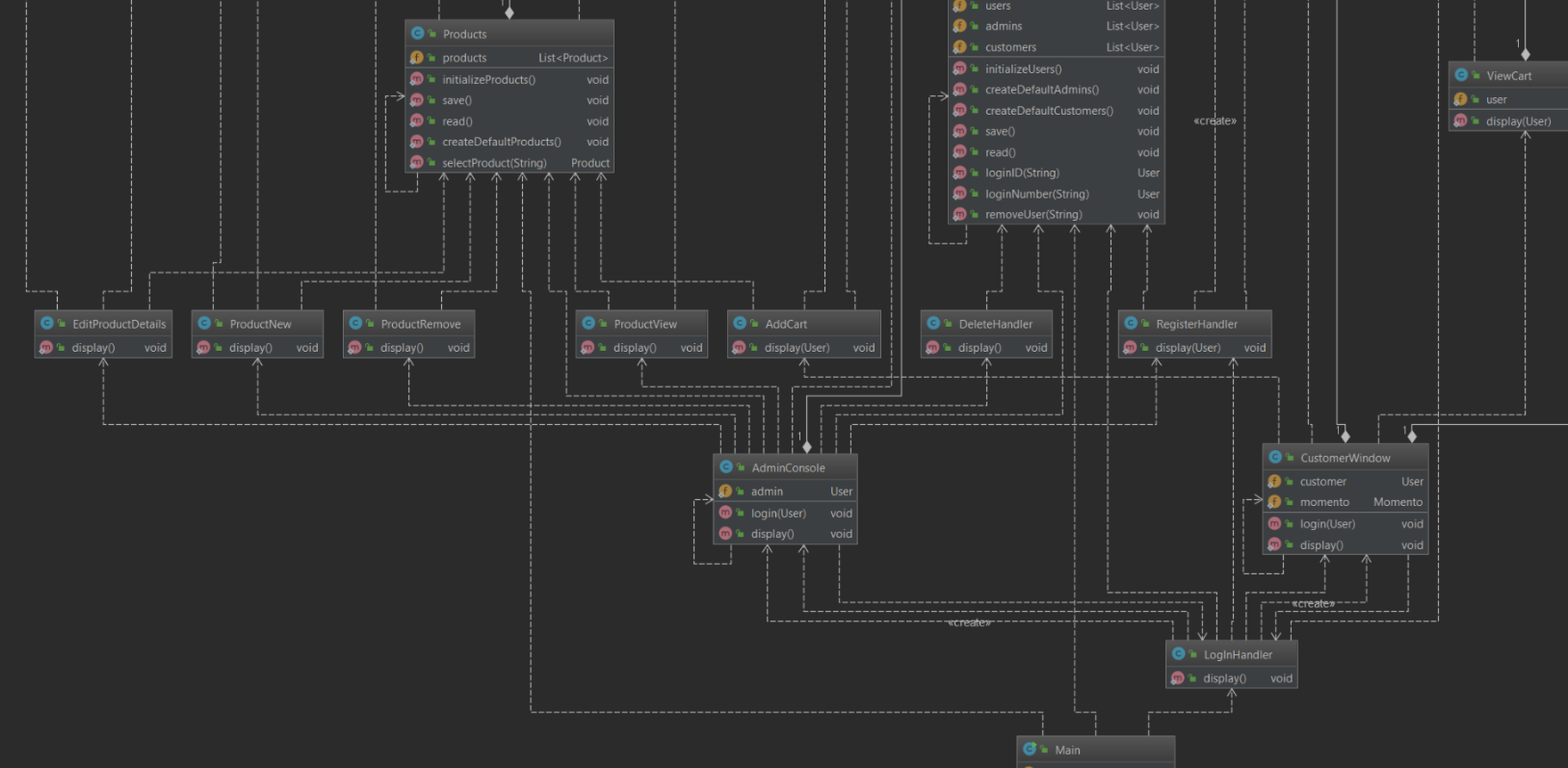
FINAL CLASS DIAGRAM



Everything changed because I didn’t even know JavaFX before starting this project, and I ended up learning it and using it to create this entire project.

I used a lot of classes to create the user interface and I ended up changing my classes for users, customers, and admins so that I could change their properties easier.

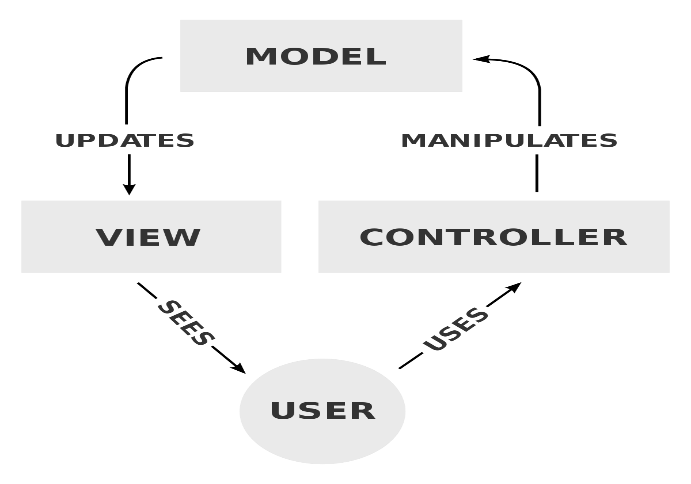
I ended up adding a lot more functionality than I was expecting at first, but I am happy with the results

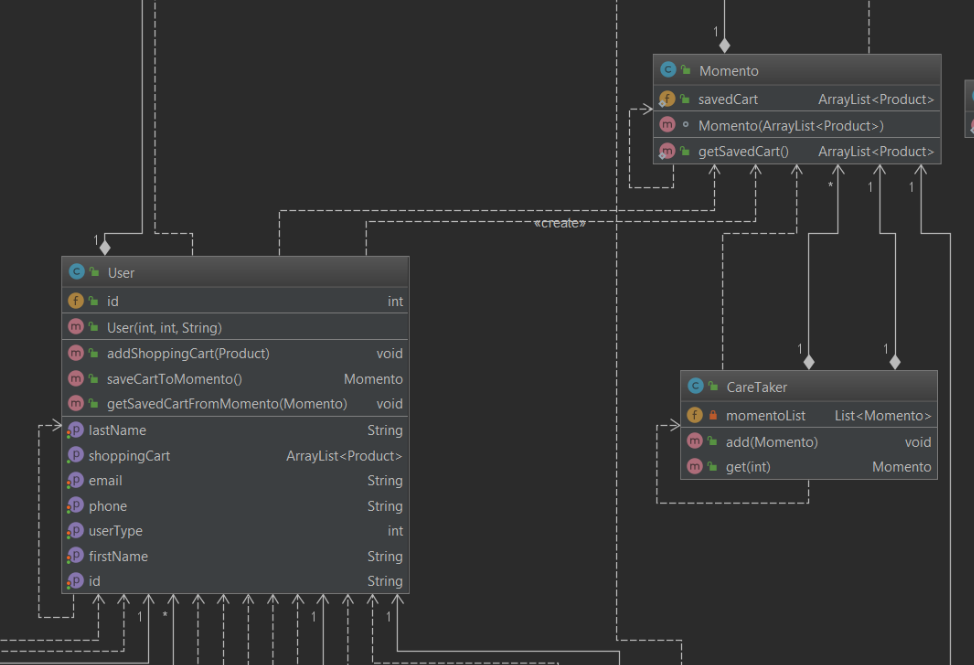
I used Model View Controller Design Pattern to control what the user can do within my program, allowing me to create a program that has few bugs, if any. The program is also easy to manage and easy to follow

All of the classes with display() methods create windows which allows the user to execute actions. Some methods require object parameters, which is to let the program know who is using the window, and whose properties are going to relevant for the use cases

Using JavaFX naturally implements the Model View Controller because the user’s functions depend on what buttons and actions are programmed into the UI. This allows the developer to control what the user can do.

Learning JavaFX and using it for this project was fun and I found JavaFX to be very useful for this application. I could look at what functions I wanted in my program, and I would create windows and buttons which perform those actions. Coding was straightforward and bugs were easy to find and easy to squish.

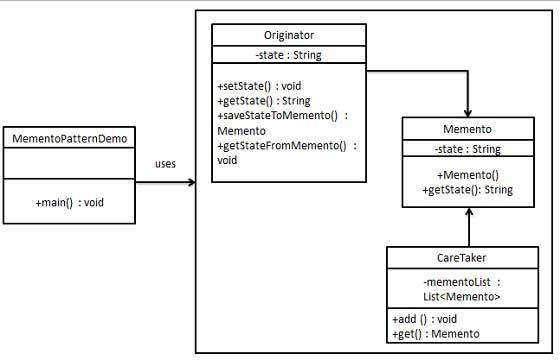




I used a Memento design pattern to save the user’s shopping cart, to be loaded up later. This pattern was a huge headache to implement but it was one of my required functions so I had to tough it out.

This pattern allows the User to save a snapshot of what their shopping cart ArrayList looks like, and allows it to be loaded up again later, even after other items have been added or removed from the cart.

Here is the memento class diagram



One of the biggest things I learned was how to code effectively in java, and how to use javaFX to create functional programs. I enjoyed using the UI to implement functionality into my program. I learned how to use Memento design pattern to save instances of objects, which can be loaded later on. Memento was giving me a lot of problems at first and debugging the issues gave me a lot of insight on how the Memento design pattern works. In doing this project, I became more comfortable with using Java, and how I could use classes, objects, and whatnot to design whatever I want.

I also learned how to use serialization to save lists of users and products, which could be loaded the next time the program is executed, even after the program is fully closed because it saves a permanent file of the lists of objects to the harddrive to be pulled up later. My system also checked if there was no file already in place for these objects and would create default ones so the system would function normally

Designing, creating, and implementing a system was daunting at first but the more I worked on it the more comfortable I was with it.

I learned that programming a system could be done in many different ways and that design patterns can solve a lot of issues that I may face while creating complex systems.

I am glad I took on this project and I am proud of what I managed to accomplish.