Online Store Application
Results Report

-Venkata Siva Prasad Kakumani

Requirements Description: The following is a description of the clients request for an Online Store. The client desires an online store where they can sell goods (and possibly services) to customers geographically dispersed around the world. Think Amazon but on a smaller scale and budget. Their desire is to have a system that is constructed in a portable language (Java) and makes use of their existing network. The system itself should present a view for the customer to interact with as well as a view for the employees or administrators of the company to interface with. For the customer there is a need for them to be able to browse available products – this should present the customer with the type, description and price of the item with the options to add to their shopping cart. If the customer attempts to add a quantity of the item more than the current supply the system should prevent the customer from adding these and prompt them with a message on the availability of the item. The customer should be able to also purchase their items from the shopping cart. The administrators should be able to update an item's description within the system, update its price, and update its quantity. The administrator should also be able to remove items from the system if so desired. Administrators should be able to add other administrators as well as add/remove customer accounts. A customer should be able to initially register for their account by themselves. The system should handle any faults or unexpected scenarios gracefully. It should be reliable and should allow for multiple customer requests during the course of execution.

Specific actions/events that should be handled as part of the system:

- Users/Administrators Login & Registration
- Browsing Items
- Updating Items
- Removing Items
- Adding Items
- Purchasing Items

Working Demonstration of Online Store :

Step 1: Initially I connected to the server side using 6080 RMI registry.

```
vkakuman@in-csci-rrpc01:~/src$ javac */*.java
vkakuman@in-csci-rrpc01:~/src$ java server.Server in-csci-rrpc01.cs.iupui.edu 6080
Server is ready to communicate with client at //in-csci-rrpc01.cs.iupui.edu:6080
```

Step 2: Then I connected the client side with same registry.

```
vkakuman@in-csci-rrpc02:~/src/client$ cd ..
vkakuman@in-csci-rrpc02:~/src$ java client.Client in-csci-rrpc01.cs.iupui.edu 6080
Server found for //in-csci-rrpc01.cs.iupui.edu:6080/RemoteUser
1. To register a new user
2. User login
3. Exit
```

Step 3: After connection had set up between both server and client below are the working of complete Online store model. Also, initially I am showing the Customer side functions. For that I registered myself as a new customer to the store.

Customer Side

```
1. To register a new user
2. User login
3. Exit
1
Hi! Are you a new user to this store
Please Sign UP by entering your details below
1. Name: Siva Kakumani
2: Email: kvsp1999@gmail.com
3. Phone: 3175838396
4: Address details: 804 Blake st
5: Password: 12345
6: ROLE CUSTOMER OR ADMIN: CUSTOMER
User Successfully Registered
1. To register a new user
2. User login
3. Exit
```

Now, I am logging as a customer as below:

```
2
Please login by entering details below
1. EmailID: kvsp1999@gmail.com
2. Password: 12345
User Details
Name :Siva Kakumani
Address :804 Blake st
Phone :3175838396
Email :kvsp1999@gmail.com
Role :CUSTOMER
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
```

Showing the functionalities of the customer:

1. Sowing al the available products in onlinestore.

```
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > Apple Smart Watch"
Description > Electronics
Price > 10.99
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Toys1
Name > Teddy bear (Giant)
Description > Gifts
Price > 12.99
Quantity > 10
-----------------------
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
```

2. Now, adding products to the shopping cart.

```
2
ProductID > Pens1
Quantity > 2
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
2
ProductID > Toys1
Quantity > 5
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
```

3. Now, I am showing all the products added into the cart.

```
3
SHOPPING CART-----------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 2
----------------------
SHOPPING CART-----------
ProductID > Toys1
Name > Teddy bear (Giant)
Description > Gifts
Price > 12.99
Quantity > 5
----------------------
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
```

4. Now I am removing a product from the shopping cart. Here You can also observe the removed product again added back to the online store.

```
4
ProductID > Toys1
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 8
----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > Apple Smart Watch"
Description > Electronics
Price > 10.99
Quantity > 10
----------------------
AVAILABLE--------------
ProductID > Toys1
Name > Teddy bear (Giant)
Description > Gifts
Price > 12.99
Quantity > 10
----------------------
```

5. In 5th function we can clear all the products in the shopping cart.
6. In 6th step we can buy all the available products in the cart. After buying them we cannot see the products in the store, also the shopping cart become empty. Here I am just using option 6 to purchase the items in the cart instead of payment system.

```
6
Pens1 Quanity 2
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 8
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > Apple Smart Watch"
Description > Electronics
Price > 10.99
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Toys1
Name > Teddy bear (Giant)
Description > Gifts
Price > 12.99
Quantity > 10
-----------------------
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
3
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a product to OnlineStore Shopping Cart
3. Show products in OnlineStore Shopping Cart
4. Remove a Product in OnlineStore Shopping Cart
5. Clear products in OnlineStore Shopping Cart
6. Buy products from OnlineStore Shopping Cart
7. Exit Product Actions
```

Admin Side

Now, I am demonstrating the functions of 'ADMIN'

- As a first step I am registering as a admin in the user registration.

```
1. To register a new user
2. User login
3. Exit
1
Hi! Are you a new user to this store
Please Sign UP by entering your details below
1. Name: Venkata Kakumani
2: Email: vkakuman@iu.edu
3. Phone: 3178838306
4: Address details: 804 Blake st
5: Password: 12345
6: ROLE CUSTOMER OR ADMIN: ADMIN
User Successfully Registered
```

- Login step for admin

```
2
Please login by entering details below
1. EmailID: vkakuman@iu.edu
2. Password: 12345
User Details
Name :Venkata Kakumani
Address :804 Blake st
Phone :3178838306
Email :vkakuman@iu.edu
Role :ADMIN
```

- You can also observe the functionalities of admin in the below window. If you observe the functionalities I added some more for admin to buy the products directly from his admin account, through this we don't need to register as customer again. But, still this and=min can register as customer for this store as wrote the code like that.

```
2
Please login by entering details below
1. EmailID: vkakuman@iu.edu
2. Password: 12345
User Details
Name :Venkata Kakumani
Address :804 Blake st
Phone :3178838306
Email :vkakuman@iu.edu
Role :ADMIN
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
4. Update a Product in OnlineStore
5. Update a Product Stock in OnlineStore
6. Get a Product Details in OnlineStore
7. Clean all products and stock and Initiaize OnlineStore
8. Add a product to OnlineStore Shopping Cart
9. Show products in OnlineStore Shopping Cart
10. Remove a Product in OnlineStore Shopping Cart
11. Clear products in OnlineStore Shopping Cart
12. Buy products from OnlineStore Shopping Cart
13. Exit Product Actions
```

1. The first option is like the customer sider.
2. In this step of choice we can add product to online store. Also, you can see the updated products to the online store by selecting the first option.

```
2
ProductID > Fruit1
Name > Apple
Description > Healthy
Price > 0.99
Add Quantity > 50
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
4. Update a Product in OnlineStore
5. Update a Product Stock in OnlineStore
6. Get a Product Details in OnlineStore
7. Clean all products and stock and Initiaize OnlineStore
8. Add a product to OnlineStore Shopping Cart
9. Show products in OnlineStore Shopping Cart
10. Remove a Product in OnlineStore Shopping Cart
11. Clear products in OnlineStore Shopping Cart
12. Buy products from OnlineStore Shopping Cart
13. Exit Product Actions
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 8
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > Apple Smart Watch"
Description > Electronics
Price > 10.99
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Fruit1
Name > Healthy
Description > Apple
Price > 0.99
Quantity > 50
-----------------------
AVAILABLE--------------
ProductID > Toys1
Name > Teddy bear (Giant)
Description > Gifts
Price > 12.99
Quantity > 10
-----------------------
```

3. N this step we can remove the product from the store as show below and you can observe the available products in the online store.

```
3
ProductID > Toys1
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
4. Update a Product in OnlineStore
5. Update a Product Stock in OnlineStore
6. Get a Product Details in OnlineStore
7. Clean all products and stock and Initiaize OnlineStore
8. Add a product to OnlineStore Shopping Cart
9. Show products in OnlineStore Shopping Cart
10. Remove a Product in OnlineStore Shopping Cart
11. Clear products in OnlineStore Shopping Cart
12. Buy products from OnlineStore Shopping Cart
13. Exit Product Actions
```

```
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 8
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > Apple Smart Watch"
Description > Electronics
Price > 10.99
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Fruit1
Name > Healthy
Description > Apple
Price > 0.99
Quantity > 50
-----------------------
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
```

4. In this choice we can update the details of the existing product. But, for this option there are some constraints I could not able to rectify al of them within the time. In the below windo you can see the updated details of ‘Gagets1’.

```
4
ProductID > Gadgets1
Name > SAMSUNG watch
Description > Price > 999
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
4. Update a Product in OnlineStore
5. Update a Product Stock in OnlineStore
6. Get a Product Details in OnlineStore
7. Clean all products and stock and Initiaize OnlineStore
8. Add a product to OnlineStore Shopping Cart
9. Show products in OnlineStore Shopping Cart
10. Remove a Product in OnlineStore Shopping Cart
11. Clear products in OnlineStore Shopping Cart
12. Buy products from OnlineStore Shopping Cart
13. Exit Product Actions
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 8
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > SAMSUNG
Description > watch
Price > 999
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Fruit1
Name > Healthy
Description > Apple
Price > 0.99
Quantity > 50
-----------------------
```

5. In this choice we can update the product quantity by selecting it ID. Here also a small sentence error is there as I could not identify them in many file I saw when I am checking on command prompt. Again it is time taking to close the server and again restart that is why I left like this. Sorry for the inconvenience.

```
5
ProductID > Pens1
To Increase Quanity + To Decrease Quantiy - To clear stock set 0 > 10
Please select any one of the Action given below by entering 1,2..
1. Show all products in OnlineStore
2. Add a Product to OnlineStore
3. Remove a Product from OnlineStore
4. Update a Product in OnlineStore
5. Update a Product Stock in OnlineStore
6. Get a Product Details in OnlineStore
7. Clean all products and stock and Initiaize OnlineStore
8. Add a product to OnlineStore Shopping Cart
9. Show products in OnlineStore Shopping Cart
10. Remove a Product in OnlineStore Shopping Cart
11. Clear products in OnlineStore Shopping Cart
12. Buy products from OnlineStore Shopping Cart
13. Exit Product Actions
1
AVAILABLE--------------
ProductID > Pens1
Name > Parker Black Ink
Description > Stationary
Price > 15.99
Quantity > 18
-----------------------
AVAILABLE--------------
ProductID > Gadgets1
Name > SAMSUNG
Description > watch
Price > 999
Quantity > 10
-----------------------
AVAILABLE--------------
ProductID > Fruit1
Name > Healthy
Description > Apple
Price > 0.99
Quantity > 50
```

Like this we can perform all other functions, here I did not give two functionalities like adding and removing of customers and admin as it is making illogical for online store.

Introduction

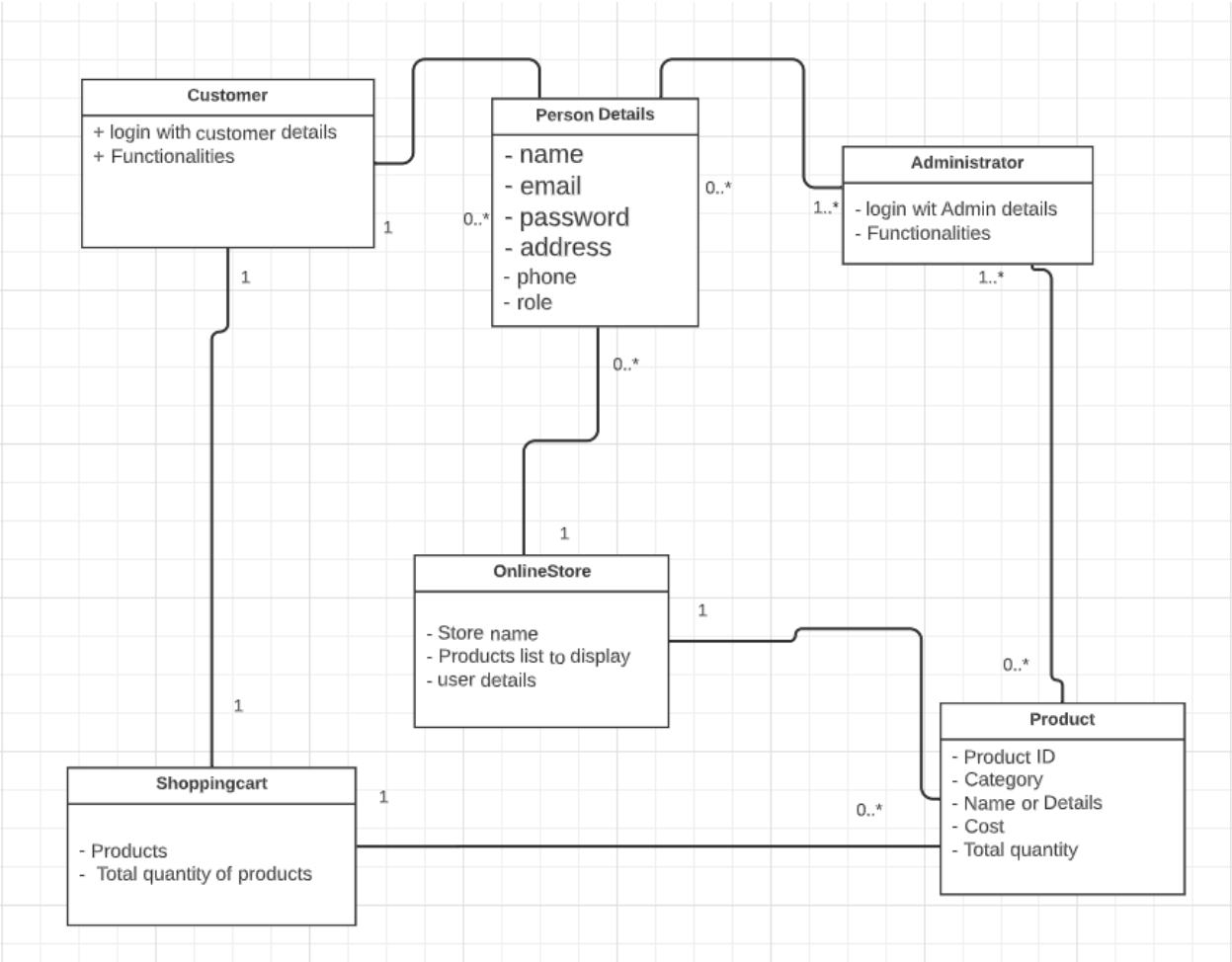
The aim of this research is to describe the creation and execution of a fundamental online store using Java RMI. The major task is establishing a framework for the application, recognizing domain-level classes and properties, and generating a Domain Model and Class Diagrams. We aimed to develop a functional application capable of performing basic functions such as User/Administrator Login & Registration, Viewing Items, Updating Items, Deleting Things, Adding Items, and Buying Goods. This report will provide an overview of our design choices, the Domain Model and Class Diagrams I created, and any other essential documentation required to explain our Java RMI implementation. Additionally, we will include screenshots of sample runs to showcase our expertise in Java RMI.

In order to complete the assignment, I followed a well-defined process that allowed me to design and implement a basic online store utilizing Java RMI. The first step in this process was to create a domain model, which involved identifying the relevant classes and properties that would be necessary to build the application. Next, I developed a class diagram based on these requirements, ensuring that the system could handle all the necessary events, including User/Administrator Login & Registration, Viewing Items, Updating Items, Deleting Things, Adding Items, and Buying Goods.

Once the class diagram was complete, I moved on to creating a use case diagram, which provided a detailed overview of how users would interact with the system. This helped me to identify any potential issues or areas for improvement in the system design before proceeding to the next step. Finally, I built the skeleton type using the Model-View-Controller (MVC) pattern on Java RMI. This involved creating the necessary classes, methods, and interfaces to ensure that the system could handle all the necessary events.

1. Domain Model

A domain model is a conceptual model that describes the many entities, properties, and interactions inside the domain of the application. It aids in identifying the main objects and their qualities that are required for the system to work properly.

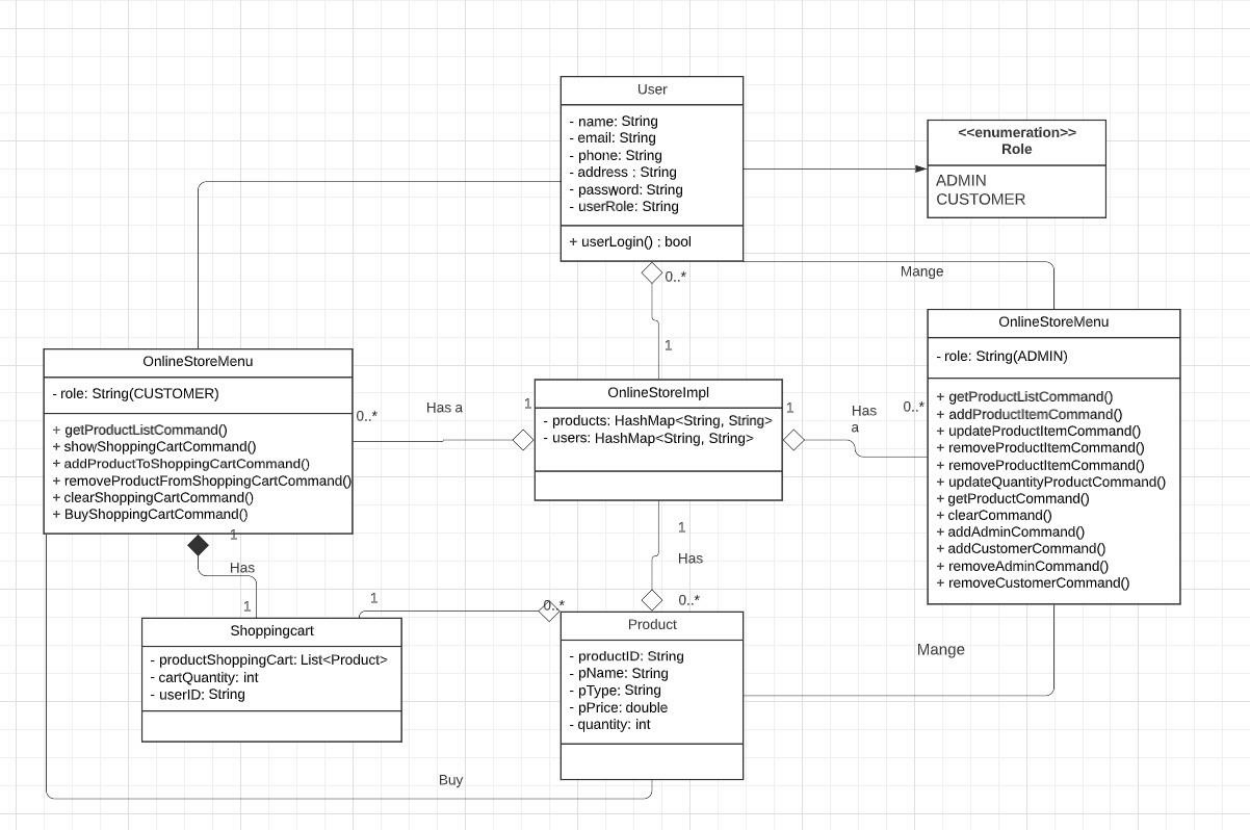


Here, by looking at domain model we can understood that the online store model is having user details which helps to login as customer or admin. also, customer and admin have their own functionalities Online store can have all the data of products and users.

2. Class Diagram

Updated class diagram from assignment 2. Compared to assignment 2 I updated some of the details in the class diagram as per the requirement and modification did in the code.

In the current class diagram, I updated by creating only one class for user details and used enumerations to access the user details based on the role of the user like 'ADMIN' and 'CUSTOMER". Then the functionality of the software can improve. Also, I removed the payment class from assignment 2 as there is no real time payment happening. Instead, I used the direct payment option for customer by selecting the payment choice. Also, I updated all the methods or functionalities of the customer and admin in the current class diagram.



Architecture

The Online Store project uses a client-server architecture, with the server managing product and user data, and the client providing a user interface for interacting with the store.

The server-side of the project is implemented using Java RMI, which allows remote method invocation between the client and server. The server includes two main classes: AuthenticateUserImpl.java and OnlineStoreImpl.java. AuthenticateUserImpl.java is responsible for authenticating user credentials, while OnlineStoreImpl.java is responsible for managing product data and implementing the Remote Online Store interface.

The client-side of the project includes several classes responsible for handling user input and displaying the store interface. The main client class is Client.java, which connects to the server using Java RMI and instantiates a FrontController.java class. The FrontController.java class applies the Front Controller pattern, which handles user requests and forwards them to the appropriate command object. The Dispatcher.java class handles the routing of requests to the appropriate command object.

The OnlineStoreMenu.java class provides a menu of available store functions to the user, while the commands folder contains classes that implement specific user commands, such as adding a product to the cart or removing a product. All the files in this folder follows the command control pattern and abstract factory pattern.

Data Models

The project includes several data models used to manage product and user data. The Product.java class represents a product in the store and includes information such as the product ID, name, and price. The

ProductItem.java class represents a specific item of a product and includes information such as the item ID, quantity, and status. The RemoteOnlineStore.java interface defines the methods available for managing products in the store, such as adding a product or updating its quantity.

The User.java class represents a user in the system and includes information such as the user's name, email, phone number, and address. The Role.java class is an enum that defines the available user roles, which are either ADMIN or CUSTOMER. The RemoteUser.java interface defines the methods available for managing users in the system, such as adding a user or modifying their account.

Overview: Using this Online store model a user can register his details as new user to this store. After that he can able to login into the store like amazon. As suggested in the assignment requirements a customer can access the online store and browse the products, add or remove products and can purchase the items. In the same an admin can do all his functionalities as per the requirements except adding or removing admin and customers.

Patterns:

The aim of this project is to enhance the RMI-based framework/skeleton of an online store from assignment 2. The focus is on implementing the Application Control Patterns using the Model-View-Control architecture and applying the Front Controller pattern, Authorization pattern, Command pattern, Factory/Abstract Factory design patterns, and Template design pattern to implement a complete login and authorization process for administrators and customers.

Application Control Patterns:

The Model-View-Control architecture divides the application into three components: Model, View, and Controller. The Model component represents the data and business logic, View component is responsible for the user interface, and the Controller component manages the interaction between the Model and View components.

Front Controller Pattern:

The Front Controller pattern is implemented in the FrontController class. The FrontController class is responsible for handling all the requests and delegating them to the appropriate controller. The FrontController class also applies the Template design pattern, where it defines a skeleton of an algorithm and delegates the implementation of the individual steps to the subclasses.

Authorization Pattern:

The Authorization pattern is implemented to manage the login and authorization process for the online store. Two roles are considered: administrator and customer. The LoginController class handles the login process, and the AuthorizationFilter class filters the requests based on the user role.

Command Pattern:

The Command pattern is implemented to handle the requests. All the requests are implemented as separate commands in the commands package. The commands are executed by the CommandExecutor class.

Factory/Abstract Factory Design Patterns:

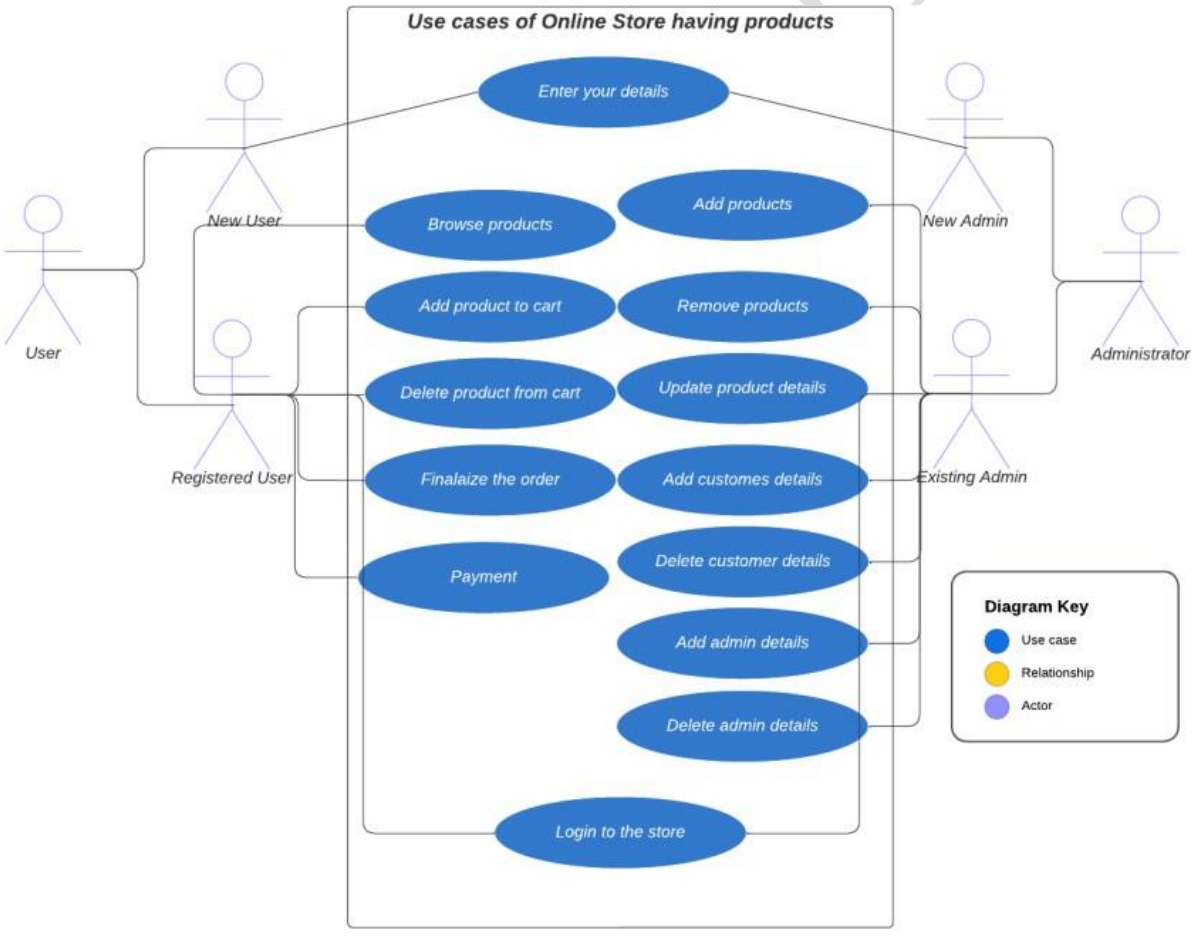
The Factory/Abstract Factory design patterns are implemented to create the instances of the objects. The ProductFactory class is implemented using the Factory pattern to create instances of Product objects. The ShoppingCartFactory class is implemented using the Abstract Factory pattern to create instances of ShoppingCart objects.

3. Use Case Diagram

Use Cases of customer or user:

The use case diagram illustrates the various interactions between the system and its users. In this case, the use case diagram for the online store with products depicts the possible interactions between the users and the online store system. The diagram contains two actors: new users and registered users. New users can interact with the system by providing their details through the "Enter your details" use case. Registered users, on the other hand, can log in using their account information. After logging in, both new and registered users can browse the products in the online store.

Once the users find a product they want to purchase, they can add it to their shopping cart through the "Add to cart" use case. The user also has the option to remove items from the cart using the "Delete item from cart" use case. After the user has finished adding and removing products from the cart, they can finalize their order using the "Finalize order" use case. Finally, the payment process is initiated through the "Payment" use case, where the user selects their payment method and enters the required payment details. Overall, this use case diagram provides a clear representation of the user's interaction with the system and the sequence of events involved in the online purchase process.



Server Connections:

Step 1:

```
C:\Users\VKAKUMAN>ssh -p 1294 vkakuman@in-csci-rrpc01.cs.iupui.edu
The authenticity of host '[in-csci-rrpc01.cs.iupui.edu]:1294 ([134.68.136.31]:1294)' can't be established.
ECDSA key fingerprint is SHA256:XFk9pka4AupNGJAlHmbwa/1NCLNTAKnpOd8cOjU3Tlc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[in-csci-rrpc01.cs.iupui.edu]:1294,[134.68.136.31]:1294' (ECDSA) to the list of known hosts.

vkakuman@in-csci-rrpc01.cs.iupui.edu's password:
```

C:\Users\VKAKUMAN>ssh -p 1294 vkakuman@in-csci-rrpc01.cs.iupui.edu
vkakuman@in-csci-rrpc01.cs.iupui.edu's password:

Step 2: Open the source folder

```
vkakuman@in-csci-rrpc01:~$ ls
src
vkakuman@in-csci-rrpc01:~$ cd src
vkakuman@in-csci-rrpc01:~/src$ ls
client  common  Makefile  README.txt  server
```

Step 3: compile all the files using the command make to access the make file.

```
vkakuman@in-csci-rrpc01:~/src$ make
javac -cp reference-rmi.jar */*.java */*/*.java
jar cfe onlinestore.jar apps.Launcher \
        common/*.class server/*.class client/*.class
```

Step 4: Now create a RMI registry and run the 'Server.java' file to start the server.

```
vkakuman@in-csci-rrpc01:~/src$ rmiregistry 6080 &
[8] 351249
```

```
vkakuman@in-csci-rrpc01:~/src$ java server.Server in-csci-rrpc01.cs.iupui.edu 6080
Server is ready to communicate with client at //in-csci-rrpc01.cs.iupui.edu:6080
```

Client Side Connection:

Step1: Connect to the server using the following command below

```
C:\Users\VKAKUMAN>ssh -p 1294 vkakuman@in-csci-rrpc02.cs.iupui.edu
vkakuman@in-csci-rrpc02.cs.iupui.edu's password:
```

Step 2: Now, run the client file from the client folder to make a connection with server to communicate.

```
vkakuman@in-csci-rrpc02:~/src$ java client.Client in-csci-rrpc01.cs.iupui.edu 6080
Server found for //in-csci-rrpc01.cs.iupui.edu:6080/RemoteUser
1. To register a new user
2. User login
3. Exit
```