**Report-Assignment 2**

**Supriya Cirimoni**

**Introduction**

This report is to detail the design and implementation of a basic online store that meets the requirements specified by the client by using JavaRMI. The online store has browsing items, updating items, removing items, adding items, and purchasing items. Additionally, it supports user and administrator login and registration.

The system is constructed in Java, and it provides a customer-facing view as well as an administrator view.

The purpose of this assignment is to familiarize ourselves with translating customer requirements to a domain model and identifying the classes, class responsibilities, and operations.

**Domain Model and Class Diagram**

The domain model is a conceptual representation of real-world entities and their relationships within the system.

A picture containing text, whiteboard

Description automatically generated

The class diagram represents the classes, their attributes, and their relationships with one another.

Diagram

Description automatically generated

**Design**

The design of this online store system consists of three main components: the server, the client, and the common objects.

**Server**

The server component is responsible for maintaining the state of the system and providing the business logic for the online store. The server is implemented using Java RMI, which allowed the client to communicate with the server and make remote method invocations.

The "PersonFactoryImpl" and "ProductManagerImpl" classes implement the "PersonFactory" and "ProductManager" interfaces respectively, which are defined in the "common" package.

**Client**

The client communicates with a remote server that provides access to a list of products and allows customers to browse through the products, add them to a cart, remove them from the cart, and purchase the items in the cart. This has both customer and Admin interfaces to manage.

Here, I used the PersonFactory and ProductManager interfaces, which are defined in the common package, to interact with the remote server.

PersonFactory- is used to create instances of the Customer and Admin classes.

ProductManager- is used to access and manage the product list.

Here, in the end, I have also created a JAR file.

**Use Cases**

1. User/Administrator: The user and administrator both will be able to register for an account and log in to their account. However, Administrators will also be able to add and remove customer and administrator accounts.
2. Browsing: The customer will be able to browse items in the store.
3. Updating: Administrators will be able to change the description, price, and quantity of an item in the system.
4. Delete: Administrators can delete items from the system.
5. Adding: Administrators can add new items to the system.
6. Purchasing: Customers can purchase items from the shopping cart. The system will prevent the customer from buying more items than are currently available.

**Implementation**

It is tested by running the program on JavaRMI by making a connection between Server and the Client.

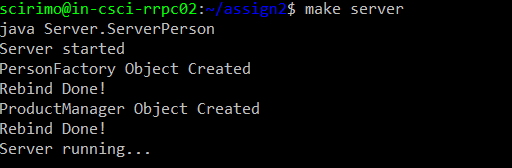
Once, the connection is established, Select the desired option, and continue shopping.

Happy Shopping!

**Results**

Please find the below-attached screenshots of the results after securing the Client-Server connection:

1. Starting the server



1. Testing on Client

Text

Description automatically generated

1. Test runs for all the options.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Conclusion:**

In conclusion, I designed and implemented a basic online store that meets the client's specifications. I identified the domain-level classes and attributes, created a domain model and class diagram, and developed the classes and use cases required to demonstrate a working Java RMI application. I've also included a client application that allows the user to interact with the system.