Online Market Place: Pattern-based Design

CSCI 50700 – Object-Oriented Design and Programming

Assignment-1

Java RMI and MVC design pattern

Summary Report

Under the guidance of

Professor Ryan Rybarczyk

Computer & Information Science, IUPUI

By

Yashwanth Reddy Kuruganti

Computer & Information Science, IUPUI

Table of Contents

1. **Introduction**

This assignment aims at building a skeleton framework for Online Market place application, which provides a platform to customer to view and shop items and admin to maintain customers and items database. To build this distributed application, Java RMI is used to support communication between Client and Server. Along with this, to separate business logic and user interface MVC design pattern is used.

2. **Java RMI and MVC design pattern**
    2.1   **MVC Design Pattern**

Each software application will have both a front-end which act as a user interface and a back-end which acts as server for business logic and data storage. Instead of combining all these modules, maintaining them independently helps in lowering the coupling among components. As a result, even though the user interface changes as per client requirements, logic remains same in the background. This nature can be implemented in any software application involving user interface using Model-View-Controller software design pattern. In this application, user interface i.e., View provides interaction for admin to manage customers, other admins and items and customer can register, login, view products, purchase items and manage his/her profile. But all these actions or events when performed on the interface, this triggers the respective functions in Model and processes the data. Before this reaches the model, controller acts as intermediate to propagate two-way messages. This application framework is built on the following modules:

OnlineMarketView component provides methods that run when a customer or admin interacts with the application. Interface provides a means of presenting the model's data. Each time a user interacts with the application, view sends a request to controller, which then forwards this to the model. All the manipulations occur in model and changes as based on type of request send by view. OnlineMarketController acts as glue between market's View and Model. Though these look dependent on one another, all the three components can function independently.

2.2   **Java RMI**

To provide a reliable communication between client and server, there should be some means which allows remote objects to behave as local objects. This can be achieved through Remote Method Invocation(RMI) remote procedure calls, which allows easy accessing of multiple objects on random remote virtual machines. In this build, I have considered OnlineMarketController as server which implements the interface declared in OnlineMarket. Using this reference, OnlineMarketView acts as a client is defined. But as of now, only raw structure is defined. Many classes and their respective methods will be defined based on the upcoming requirements.

3. **Class Diagram**

Below is the class diagram depicting all the classes that are built as part of this first assignment. But this is not the final version. There are some other classes, methods, variables which will be added in the upcoming assignments. This domain model is a foundation which only shows very

few classes and methods that are required for RMI and MVC to work. Some methods which are in Controller or View or Model will also be added or modified as per requirements in the upcoming tasks. So this is not the final version.
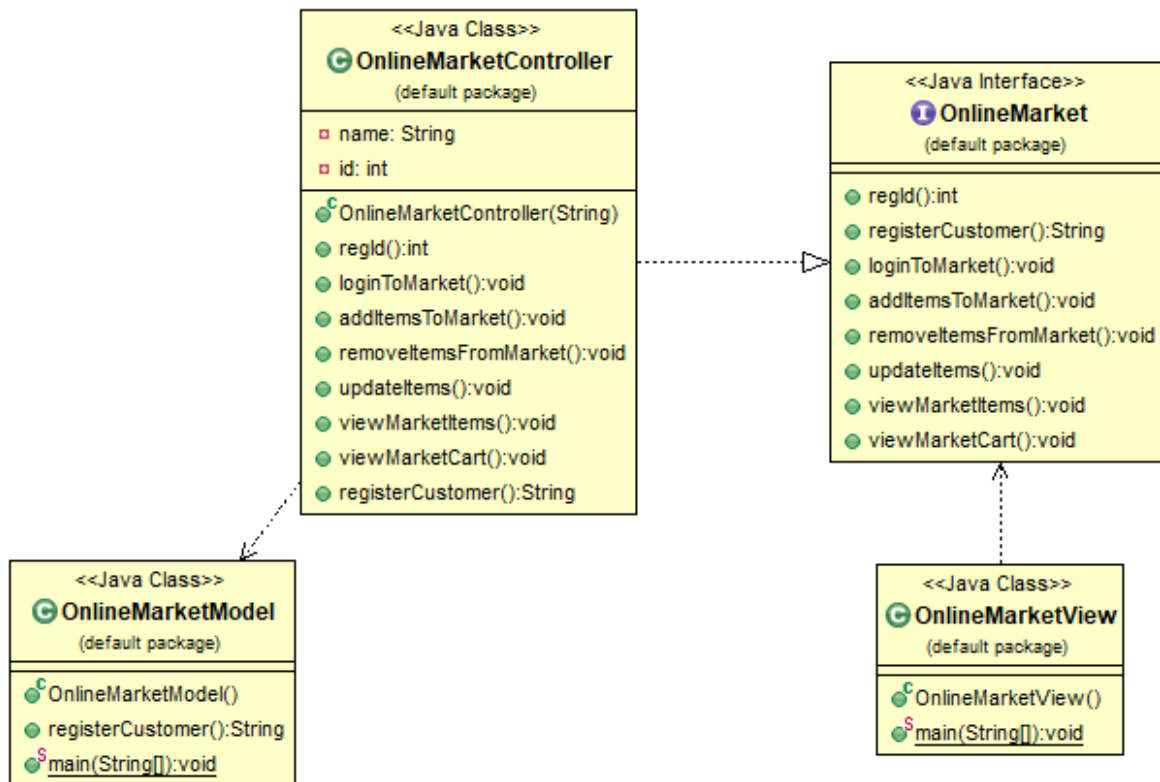

Fig 1. Class Diagram

4. Sample Run

Steps on running this software is given in Readme file. Below are few snapshots that show the working of built application.

- Uploaded all source files to Tesla

```
[yashkuru@tesla Assignment1]$ ls
makefileC.sh  OnlineMarketController.java  OnlineMarketModel.java  policy
makefileS.sh  OnlineMarket.java            OnlineMarketView.java   README.txt
[yashkuru@tesla Assignment1]$
```

- Compiling all java classes

```
[yashkuru@tesla Assignment1]$ javac *.java
[yashkuru@tesla Assignment1]$ ls
makefileC.sh                    OnlineMarketModel.class
makefileS.sh                    OnlineMarketModel.java
OnlineMarket.class              OnlineMarketView.class
OnlineMarketController.class    OnlineMarketView.java
OnlineMarketController.java     policy
OnlineMarket.java               README.txt
```

- Running RMI registry

```
[yashkuru@tesla Assignment1]$ rmiregistry 5432&
[1] 160744
```

- Running Server component

```
[yashkuru@tesla Assignment1]$ java -Djava.security.policy=policy OnlineMarketMod
el
You are now entering Online Market Place
Reaching server://tesla.cs.iupui.edu:5432/onlineMarketServer
Interface is Ready!You can register, login and shop
```

- Running client in another session

```
[yashkuru@tesla Assignment1]$ java -Djava.security.policy=policy OnlineMarketVie
w
Registration ID: 1
Registration Status: Registered
[yashkuru@tesla Assignment1]$
```

- Interaction between client and server. Connection established

```
[yashkuru@tesla Assignment1]$ java -Djava.security.polic   OnlineMarketController.java   policy
w                                                          OnlineMarket.java            README.txt
Registration ID: 1                                         [yashkuru@tesla Assignment1]$ rmiregistry 5432&
Registration Status: Registered                            [1] 160744
[yashkuru@tesla Assignment1]$ java -Djava.security.polic   [yashkuru@tesla Assignment1]$ java -Djava.security.policy=policy (
w                                                          el
Registration ID: 2                                         You are now entering Online Market Place
Registration Status: Registered                            Reaching server://tesla.cs.iupui.edu:5432/onlineMarketServer
[yashkuru@tesla Assignment1]$ java -Djava.security.polic   Interface is Ready!You can register, login and shop
w                                                          Registration page. Register here
Registration ID: 3                                         Registration page. Register here
Registration Status: Registered                            Registration page. Register here
[yashkuru@tesla Assignment1]$
```

- Running with the help of makefile(easy compiling and executing)

```
OnlineMarketController.java   policy        [yashkuru@tesla Assignment1]$ rmiregistry 5432&
OnlineMarket.java             README.txt    [1] 165786
[yashkuru@tesla Assignment1]$ sh makefileC.sh  [yashkuru@tesla Assignment1]$ sh makefileS.sh
Registration ID: 1                             You are now entering Online Market Place
Registration Status: Registered                Reaching server://tesla.cs.iupui.edu:5432/onlineMarketServer
[yashkuru@tesla Assignment1]$ sh makefileC.sh  Interface is Ready!You can register, login and shop
Registration ID: 2                             Registration page. Register here
Registration Status: Registered                Registration page. Register here
[yashkuru@tesla Assignment1]$
```

5. Conclusion

From this assignment, I have learnt and understood how Java RMI works and the way its implemented to provide reliable communication between a client and server. Also, MVC helped me to separate layers of code and user interface from server implementation. Integrating RMI along with MVC is a bit challenge, but this skeleton structure seems to help in modularizing tasks in future assignments to implement all the business logic and user interface.