

**CSCI507**  
**Object Oriented Design and Programming**

**Technical Project Report on**

# **Market Place** **Application**

**Dept. of Computer & Information Science,  
IUPUI**

**Name: Keyur Kirti Mehta**  
under the guidance of Prof. Ryan Rybarczyk

## Contents

<b>1. Introduction</b>	<b>5</b>
<b>I. Application Functionality</b>	<b>5</b>
<b>2. Domain Model</b>	<b>5</b>
<b>3. Technical Architecture</b>	<b>6</b>
<b>I. Java RMI</b>	<b>6</b>
<b>II. MVC Architecture</b>	<b>6</b>
<b>III. Concurrency in Java RMI</b>	<b>8</b>
<b>IV. Synchronization</b>	<b>9</b>
<b>4. Designing Patterns</b>	<b>10</b>
<b>I. Front Controller Pattern</b>	<b>10</b>
<b>II. Command Pattern</b>	<b>10</b>
<b>III. Abstract Factory Pattern</b>	<b>11</b>
<b>IV. Authorization Pattern</b>	<b>12</b>
<b>V. Proxy Pattern</b>	<b>13</b>
<b>VI. Reflection Pattern</b>	<b>13</b>
<b>VII. Monitor Object Pattern</b>	<b>13</b>
<b>VIII. Future Pattern</b>	<b>14</b>
<b>IX. Guarded Suspension Pattern</b>	<b>14</b>
<b>X. Scoped Locking</b>	<b>15</b>
<b>XI. Thread Safe Interface</b>	<b>15</b>
<b>XII. Singleton Pattern</b>	<b>16</b>
<b>XIII. Database Access Layer Pattern</b>	<b>16</b>
<b>5. UML Diagrams</b>	<b>17</b>
<b>I. Class Diagram and working</b>	<b>17</b>
<b>II. Sequence Diagram</b>	<b>20</b>
1. Login Activity Sequence	20
2. Admin User Sequence	20
3. Customer user Sequence	21
<b>6. Database Schema</b>	<b>21</b>
<b>7. Assignments Transition/ Overview</b>	<b>22</b>

I. Assignment 1 .....	22
II. Assignment 1 to Assignment 2.....	22
III. Assignment 2 to Assignment 3.....	23
IV. Assignment 3 to Assignment 4 .....	23
V. Assignment 4 to Assignment 5.....	23
VI. Assignment 5 to Assignment 6 .....	24
8. Sample Runs .....	24
I. Server start and Login view.....	24
II. Admin Login.....	24
III. Admin - browse Product .....	25
IV. Admin – Add product .....	26
V. Admin – Update product.....	27
VI. Admin – Delete Product.....	27
VII. Admin – Add Administrator.....	28
VIII. Admin – Add Customer.....	29
IX. Admin – Remove Customer .....	29
X. Customer Login.....	30
XI. Customer – Browse product .....	30
XII. Customer – Add product to Shopping cart.....	31
XIII. Customer – View Shopping Cart .....	32
XIV. Synchronization – Product deleted Removed from Cart.....	33
XV. Role Based Access Control.....	33
9. Conclusion .....	34
I. Final Conclusion.....	34
References .....	36

## Table of Figures

<i>Figure 1 Domain Model .....</i>	<i>6</i>
<i>Figure 2 MVC Architecture [3] .....</i>	<i>7</i>
<i>Figure 3 Concurrency .....</i>	<i>8</i>
<i>Figure 4 Synchronization.....</i>	<i>9</i>
<i>Figure 5 Command Pattern [4] .....</i>	<i>11</i>
<i>Figure 6 Abstract Factory Pattern [5] .....</i>	<i>11</i>
<i>Figure 7 Authorization Pattern [6].....</i>	<i>12</i>
<i>Figure 8 Monitor Object Pattern.....</i>	<i>13</i>
<i>Figure 9 Class Diagram .....</i>	<i>19</i>
<i>Figure 10 Login Sequence Diagram .....</i>	<i>20</i>
<i>Figure 11 Admin user Sequence Diagram .....</i>	<i>20</i>
<i>Figure 12 Customer user Sequence Diagram .....</i>	<i>21</i>
<i>Figure 13 Database Schema .....</i>	<i>22</i>

## 1. Introduction

Marketplace Application is an e-commerce platform to purchase goods and get it delivered at the door step. This application is hosted on server which makes it accessible to all its user online from anywhere. This application gives user to browse different products based on different categories and will be available for purchase. Also administrator of this application will have authority to control the details of the products.

This document will give detailed technical and general information about the application. Also it will give details about its architecture and its usage. This report will also have screenshot of functionality which will give idea to user about its different modules.

### I. Application Functionality

Market Place application will have 2 types of User.

- a. **Administrator User:** One administrator user will be created by default. And these type of users will have following functionalities:
  - i. Add other Admin Users
  - ii. Add/ Remove Customer users
  - iii. Add New products
  - iv. Update product detail
  - v. Delete a product
  - vi. Browse the products
- b. **Customer User:** This is other type of the user who can purchase the products from the marketplace Application. These kind of user will have following functionalities:
  - i. Register for application
  - ii. Browse the products
  - iii. Add product to the shopping cart
  - iv. View Shopping Cart
  - v. Purchase the products

## 2. Domain Model

Domain model of the application is as per the figure 1. It has one entity as User. User needs to log in to the application to access it. There are 2 types of users in this application. One is Administrator and Customers. Another entity is Product which has all the details about the products like its type, price, description etc. Administrator can add/ update and delete these products. Customers can purchase these products if the desired quantity is available in the stock. Every customer will have its own Shopping Cart where he can add the products which later on can be purchased.

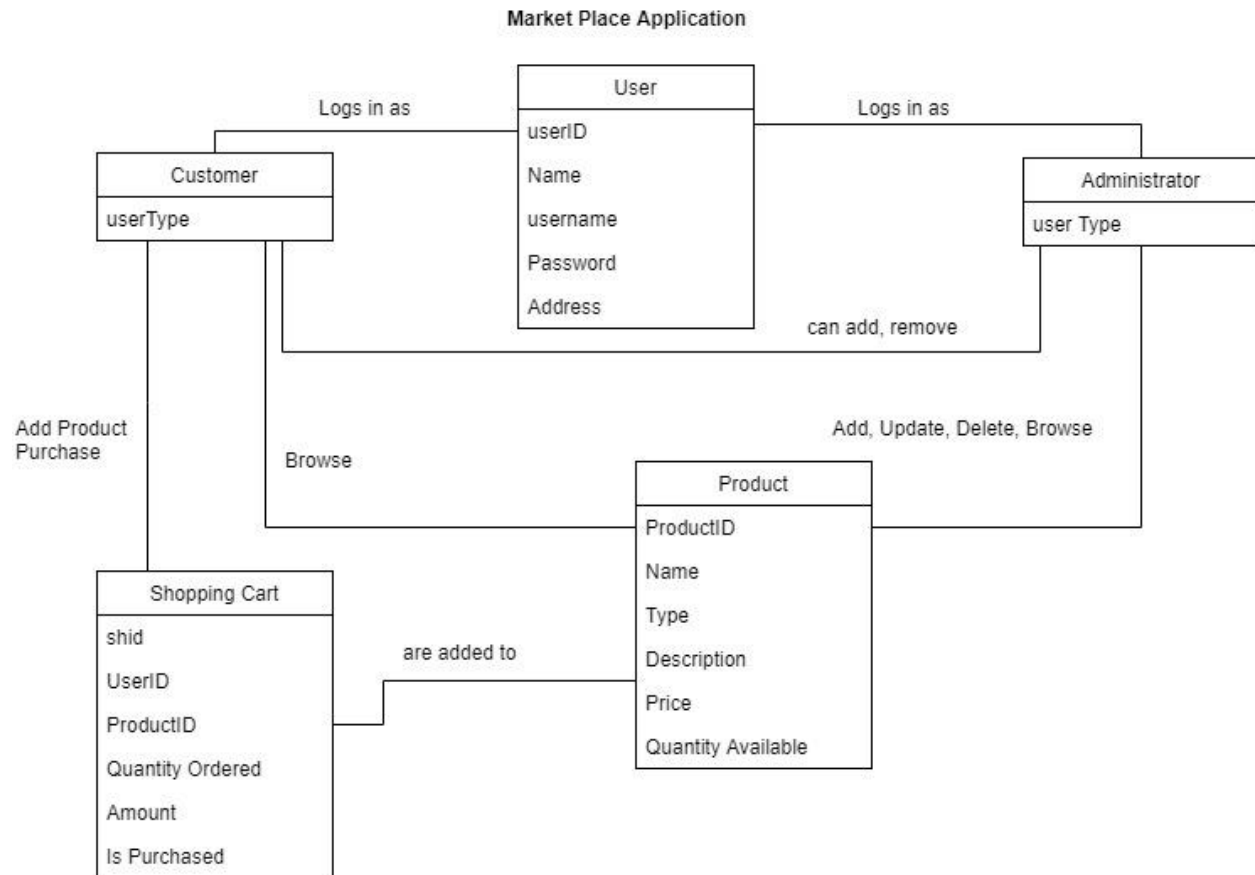


Figure 1 Domain Model

### 3. Technical Architecture

#### I. Java RMI

Java RMI is API which is used to communicate between client and server. It uses stub and skeleton framework for the communication. So, basically stub is an object on the client side which initiates the communication with the server. It lookup for the similar string in server as of in the client. Server will rebind with the clients. [1]

#### Assignment Discussion

In this project, views on the client side will lookup for the connection string on the server. Each view will have different connection string, which will then point to different interfaces on the server side. Server is rebind it with the connection to the client.

#### II. MVC Architecture

MVC is Model-View-Controller architecture. It is a design pattern used in software designing. This architecture separates the different modules with each other, so as changes in one will not or have little impact on the other. [2]

**Model** – This component holds the core business logic of the functionality. It processes all the requests received from the views.

**View** – This is the actual display which user can see. It will have the data along with the information. Views will be different for different type of the users. This can be GUI or the console where user can interact with it.

**Controller** – It acts as a mediator between the model and views. It takes request from views and send it to the model for processing. And similarly it response (processed request/ result) from server to the client.

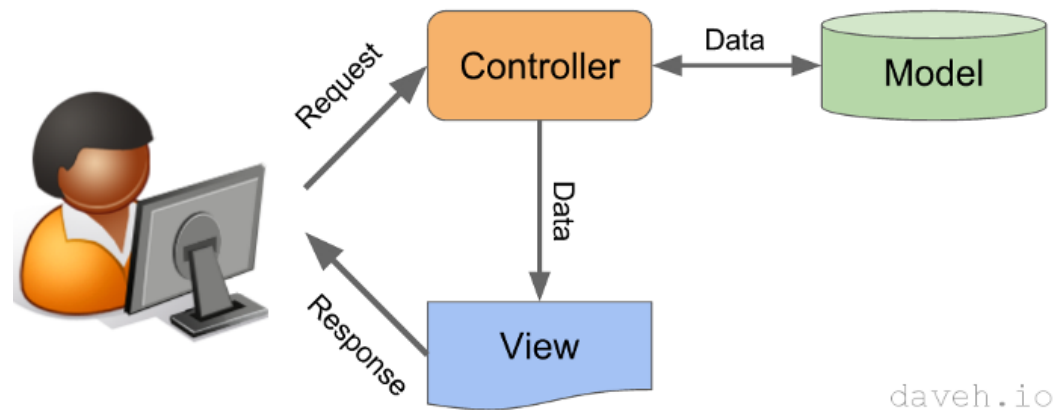


Figure 2 MVC Architecture [3]

### Assignment Discussion

In this project, we have MVC architecture in client server environment. The domain model designed is based on the MVC architecture where view is on client side and controller and the model are on server side. There can be multiple clients. So, if controller is on client side and is coupled with it, changes in views will result in controller update. So, because of this, the controller is kept on the server side.

**Views:** Once client starts, user will be displayed with general user view. And there are 2 additional views created which will be displayed based on the type of the customer logged into the application. These are Customer and Administrator views. Any requests from these clients will go to the server through RMI. Views will pass the data (request) through the objects of interfaces. Interfaces will hide the actual information from the client and will provide abstraction layer to the application.

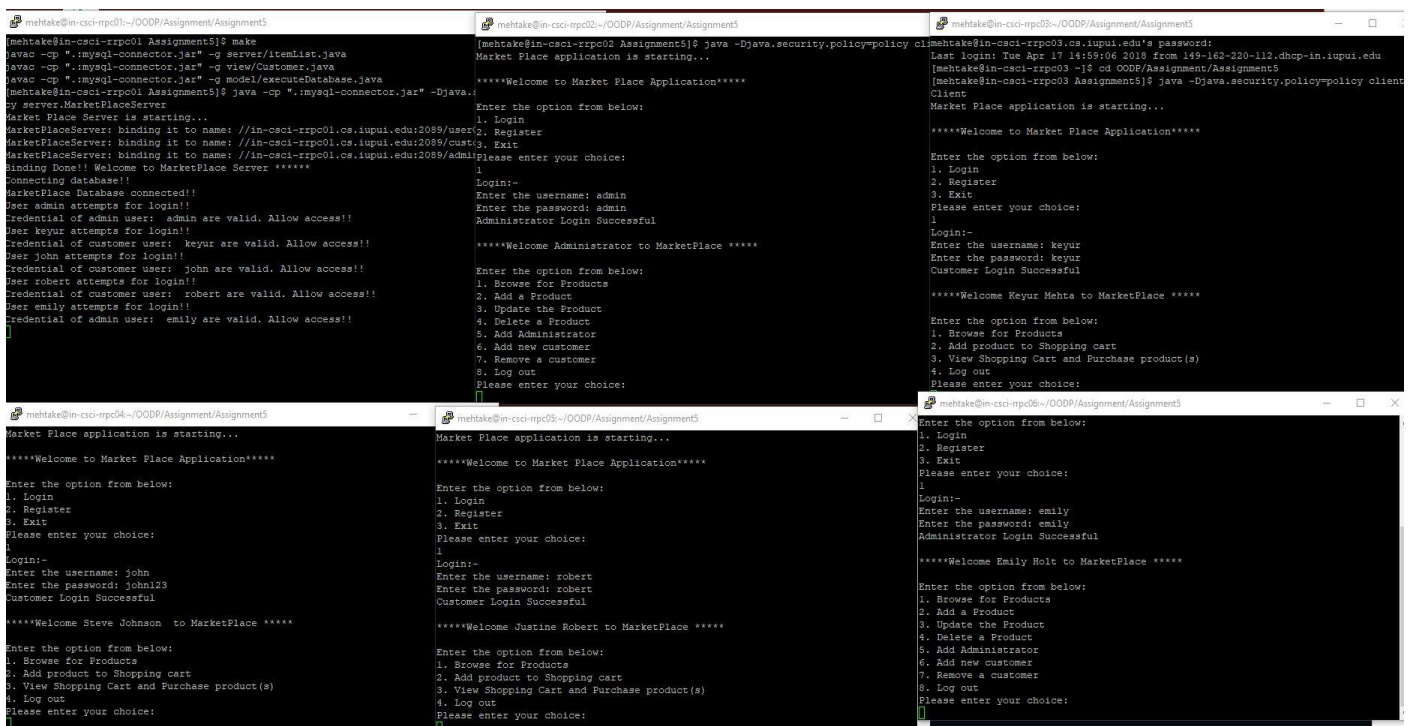
**Controller:** Controller are designed on the both client and server side. Interfaces for the controllers are implemented on the server side. Any requests (RMI) from views will come to front controller and then forwarded to server controllers where interface methods are implemented from server class where RMI connection is bind. Controller than will forward the request to the models through the objects of models. Each view has its own controller that will handle its request.

**Model:** Similar to controller model are designed on the server side. And for controller have its own model which will process the request forwarded by the controller. Here request will be processed and the result will be sent back to views through controller. And views will display / take necessary action based on the response.

In this designed all the components are loosely coupled with each other. So, the changes in one will have no or littler effect on the other component. i.e. changes in any 1 particular model will have no effect on other models or the controller or views. Similarly for changes in view will not make model to update its own logic.

### III. Concurrency in Java RMI

Java RMI provides a mechanism to create a distributed application. The RMI object allows to invoke a method which is running on an object of another machine. It provides communication mechanism between 2 objects of stub and skeleton. Whenever client invokes any method of the remote objects, it Marshalls the parameters together. And if the parameters are object then this objects are serialized. On the server side parameters are unmarshall or de-serialized.



### Figure 3 Concurrency

Java RMI supports the concurrency. So, there are multiple skeletons which can send the requests to the stub. Java RMI handles all these requests and make application concurrent. Multiple requests can be send to the server simultaneously. This functionality is tested properly during the implementation of this part of the assignment. Multiple clients are started on different machines. And these clients hits the server with their requests using RMI. So, RMI is multi thread mechanism.



All the requests are reached the server. Figure 3 shows the working of multiple clients and a server on different machines as an example of multithread.

#### IV. Synchronization

If two different clients sends the same request i.e. performing same functionality on the server, then also these requests came to server. In my test, if 2 clients wants to purchase an item from the product list at the same time, then this operation is performed for both of the client in simultaneous manner. For this I have used synchronized method. So, if 2 clients wants to purchase a product then the only one request will be executed first and quantity of the product is updated then. Then second request will get executed, which will have updated quantity.

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
Product successfully added to the customer's shopping cart!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 5 with quantity: 10 to his cart.
Product can NOT be added to the cart as ordered quantity is greater than the available quantity!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 3 to his cart.
Product can be added to the cart as ordered quantity is less than the available quantity!!
Product successfully added to the customer's shopping cart!!
Customer wants to view shopping cart!!
Customer wants to view shopping cart!!
Customer wants to Purchase the product(s) from his shopping cart!!
Product: 1 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 1 with quantity: 3
Product: 2 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 2 with quantity: 1
Customer wants to Purchase the product(s) from his shopping cart!!
Product id: 1 can NOT be purchased as ordered quantity is greater than the available quantity!!
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 2 to his cart.
Product quantity can be updated in the cart as its already present in cart!!
Product quantity is updated successfully in the shopping cart!!
Customer wants to view shopping cart!!
Customer wants to Purchase the product(s) from his shopping cart!!
Product: 1 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 1 with quantity: 2
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!!
Admin wants to browse the products!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
3
*****The Product List*****
ProductID  Name      Quantity Ordered  Amount
-----
1          iPad       2                998.00
2          Watch       1                100.00

Kindly confirm, Do you want to purchase all products in shopping cart (Y/N)?
Y

*****Order Detail*****
Partial Order is placed!!

Order placed for Product id: 1 with quantity: 2
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!! Kindly recheck the quantity.

*****Welcome Keyur Mehta to MarketPlace *****
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:

mehtake@in-csci-rpc04:~/OODP/Assignment/Assignment5
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
3
*****The Product List*****
ProductID  Name      Quantity Ordered  Amount
-----
1          iPad       3                450.00
2          Watch       1                100.00

Kindly confirm, Do you want to purchase all products in shopping cart (Y/N)?
Y

*****Order Detail*****
Complete Order is placed and will be delivered to your door step!!
Order placed for Product id: 1 with quantity: 3
Order placed for Product id: 2 with quantity: 1

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
1
*****The Product List*****
ItemID  Name      Description      Type      Price  Quantity available
-----
1       iPad      Tablet          Electronic 499.00  0
2       Watch     Watch           Accessories 100.00  0
3       MacBook   Laptop          Electronic 950.00  0
4       iWatch    Smart Watch     Accessories 300.00  0
5       T Shirt   Men's T shirt   Clothes    55.00  7

*****Welcome Administrator to MarketPlace *****
Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator

```

Figure 4 Synchronization

Figure 4 shows the synchronized example implemented using Java RMI and synchronized. Where initially there are 1 quantities of product ID 2. And 2 clients requests for the same product with 1 quantity each. Product is added to the cart of these 2 users as product is still not purchased (1 & 2). But 1 client places the order (purchase) which is executed successfully (3) and can see the reduced product quantity in (4). Another client then gets the messages of insufficient quantity (4).

## 4. Designing Patterns

### I. Front Controller Pattern

This pattern have 3 component in it whose working is mentioned as below:

**Controller:** The front controller is the point of contact for handling all the requests. This controller will perform basic authentication and then the request will move forward to the server. This request then processed at the server end. And send back the result the front controller.

**Dispatcher:** Once the request is received from the server, it will be navigated to the dispatcher. Dispatcher is responsible for navigating the response and display of view. Based on the response received, the dispatcher will render the view and display the corresponding view to the user.

**View:** Views are the displayed to the user. It will contain all the text and takes the input from users which then will be transferred to the front controller to process and based on the received response displays the new information.

#### **Assignment Discussion**

We have implemented the front controller pattern in market place application, where once the application starts user will see the user view. And this view will take username and password and send it to the front controller. Front controller will perform basic validation on it and send it to model via RMI for authentication. And once the response is received, it will display the corresponding view based on the role (Admin or Customer) with the help of dispatcher.

### II. Command Pattern

In command pattern, command is passed from client which will perform the desired action. Here the action is performed by other than the client. A request or command is enclosed in an object and then pass to the command Invoker. This command invoker will perform the desired action (method) based on the command received to him. It allows parameterization of methods with different requests. Also can be used to follow the sequence in which commands needs to be executed.

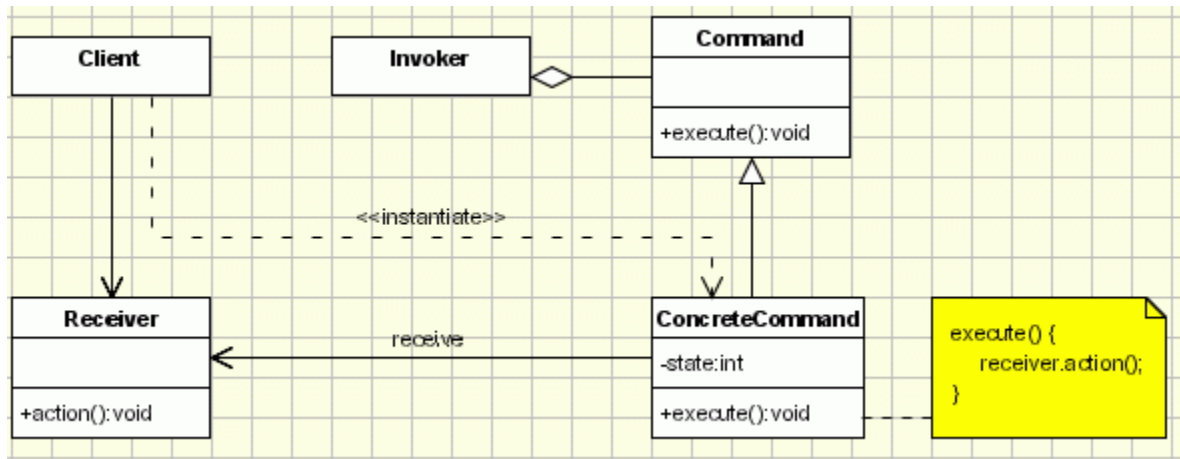


Figure 5 Command Pattern [4]

### Assignment Discussion

In this application, I have implemented command pattern on Administrator and Customer View. Where command performed by the admin, customers is sent to command invoker and then based on the received command, invoker will create an object of the class where interface is implemented. One interface is implemented to many different classes which individually performs the operation. I.e. sent the request to server via RMI.

### III. Abstract Factory Pattern

Abstract factory pattern will work on factories of factory method. An interface will create a factory of similar related objects. Factory creates the objects for the clients and then the implemented classes will be executed using created objects. Abstract factory will be interface which will create abstract products. And the concrete factories implements the interface.

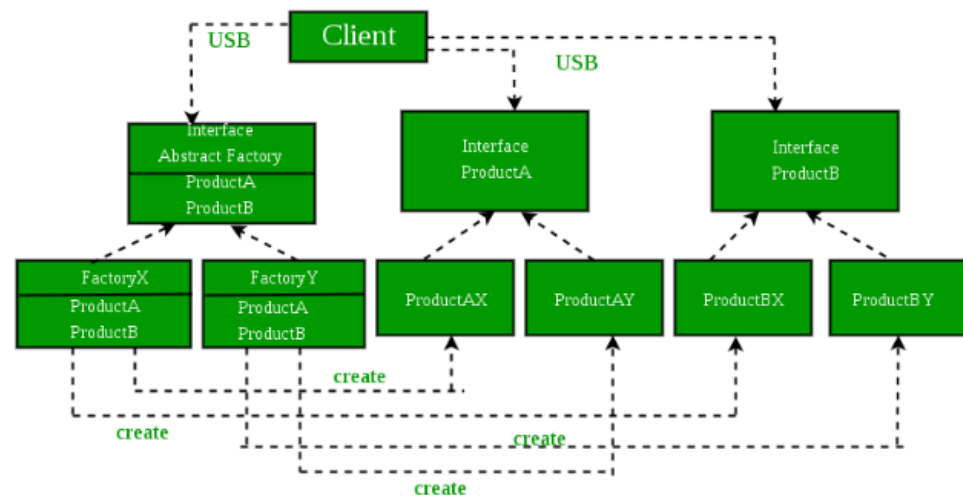


Figure 6 Abstract Factory Pattern [5]

### Assignment Discussion

I created 1 abstract factory which has abstract method `createView`. And this method is override in 2 concrete factories, one for Admin and other for customer. Also one more abstract factory created to display the view. This method is again override in 2 views which will have different menus based on the user roles. Each of the Concrete factory will call respective view.

## IV. Authorization Pattern

Authorization is necessary so as only legitimate user can have the access to the application or specific functionality. Once user is authenticated then this user should be able to access the functionality of the application. But implementation of authentication does not provide good security feature to the functionalities. In order to achieve that, we use **Role Based Access Control (RBAC)**.

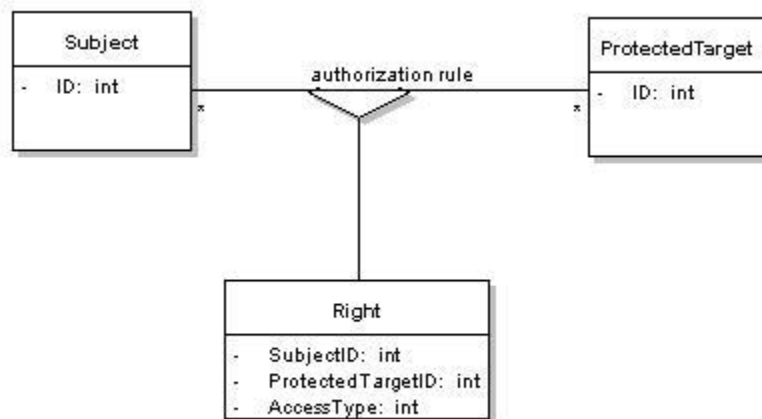


Figure 7 Authorization Pattern [6]

### Assignment Discussion

#### Role Based Access Control (RBAC) through Java Annotation

`@RequiresRole("ADMIN")`

Each functionality of market place application should not be accessible to the authenticated user. Certain functionality should be available only to certain type of users. This is implemented using Java annotations. We have created custom Java annotation which will define the role type required in order to access the functionality. So, whenever the user object invokes the method, it first checks the user role associated with the object through annotation. If role is matched then only the object access the method.

## V. Proxy Pattern

In proxy pattern, one class represent the functionality of the other class. An object of the class is instantiate when the object of the original class is invoked. The proxy class will handles the functionality of other class. Instead of calling the method of the original class, first proxy class method will be called and executed.

### Assignment Discussion

In our assignment, Role Based Access Control is implemented using Java annotation and proxy pattern. When client sends the request to controller to access the method, it will redirect it to proxy class which will check the role assigned to the method (annotation). If user role matches with the role required in annotation, then the corresponding method is called.

## VI. Reflection Pattern

Sometimes we must issue a request to an object without knowing much about that object. In this type of scenario, Reflection Pattern comes to the picture. In Reflection pattern, we create a reflection image of the class and we can then using the object, we access the information of that class.

### Assignment Discussion

In our assignment, Reflection pattern is used in the implementation of the proxy pattern where invocationHandler where it gets the details of the annotation used in class.

## VII. Monitor Object Pattern

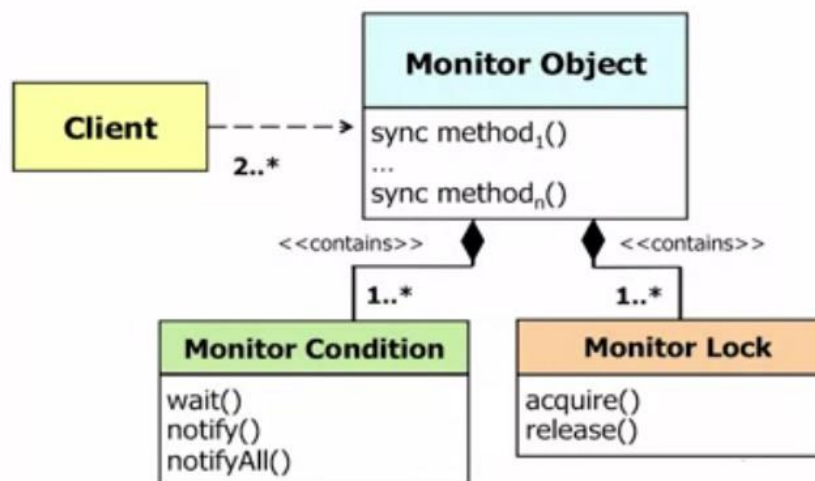


Figure 8 Monitor Object Pattern

The monitor object pattern is one of the pattern in achieving the synchronization in multi-thread environment. It synchronize the execution of the objects which are executed or called by different threads. It ensures that only 1 method runs at a time accessed by different objects. So, when

multiple objects or thread tries to execute the same method, only 1 object get the access to that method and other will go on wait state which is similar to the queue implementation. Once the first object uses finish executing the method, next in the wait will get the access to use the same method. This is similar to the acquiring the lock when object needs to use the resource and then releasing it when the execution is finished. When a thread acquires a lock, it is said to have entered the monitor. All other threads attempting to enter the locked monitor will be suspended until the first thread keep acquiring the monitor. Once the lock is released, it will notify all other thread about the same and then next in the queue will get chance to use the resource.

### **Assignment Discussion**

In this assignment, we have implemented the monitor object pattern using the synchronized keyword at the block level. To minimize the impact of the reserving the resource and reduce the wait time, we have identified the critical section in the code, which requires to be used as monitor object. And then enclosed the block in critical section. So, as only that part of the code will be locked by the executing thread and other thread if require can use the remaining section of the method.

## **VIII. Future Pattern**

Future pattern is used to reduce the computational time. There are some task while executing the method, which takes more time for execution like database queries. SO, in synchronized environment processing the execution halts until the result is obtained from the previous task. (Which takes more computational time) In order to avoid this, future pattern creates another virtual object which first stores the state of the computing task and then start with the processing of next code. Once the original execution task is completed then it will return the result to this object. So, in this case computation doesn't halt due to the processing delay of some task.

### **Assignment Discussion**

This pattern is implemented using the synchronized blocks in the java method. But this is not effectively implemented as I have used synchronized block at very low level i.e. only those part of the code which will effect when multiple threads tries to access the same. I have put synchronized block only when database operations are performed. No other task has been kept in the block. But yes, java internally manages the future pattern in that piece of code if required.

## **IX. Guarded Suspension Pattern**

This pattern manages the operation which require a precondition to be satisfied before the operation can be executed successfully. Whenever method wants to perform any task or operation it checks for certain pre-condition. If the condition is satisfied, then only perform the operation. Until that halt the thread to perform and do not suspend it. This pattern is used when user knows that suspension is only for the short duration of time. If the process is suspended for longer duration, then it will slow the overall execution of the entire application. Once the condition is met, the process will start performing the desired task.

### **Assignment Discussion**

In the assignment, Java synchronized keyword internally implement the Guarded Suspension pattern. I have put synchronized for part of code which performs database operation. So, whenever some other thread is working on same database resource, it will halt the execution for this thread. And put it in the queue as monitor object. Once the resource is free, the condition is met and thread will execute the code. For database operation, only 1 connection is available. If any other thread is using that connection on the same object, then second thread cannot use the connection to perform the database operation. So, it will go on halt and wait till it is released. I have used synchronized block for all DML operation and only those select operation which have insert or update effect. For e.g. selecting product details which might change if insert or delete product occurs. So, I have used synchronized block. But for user authentication, insert of new user will not affect retrieving of other user, so in this case synchronized is not used.

## **X. Scoped Locking**

Critical section which are identified in the code needs to be executed sequentially in order to achieve concurrent environment. So as no more than 1 thread uses that part of the code which results in some unexpected situation. Scoped Locking provide the mechanism to ensure that the only the tread which is currently using the section of the code have the access of it and no other threads can use that section of the code. So, whenever a thread needs to use any block of code, it will acquire a lock on it and once the execution is completed, the lock is released so as other thread can acquire it. This is called Scoped Locking.

### **Assignment Discussion**

Java synchronized provide the Scoped Locking mechanism. So, the critical section which are identified and enclosed between synchronized block will provide the scope locking. The object (thread) which is executing the critical section will acquire the lock using monitor object. And when the execution is finished, it will release the lock and allow other object to acquire the same. Now, the method calls which are in the synchronized block will automatically get the locks. Synchronized will handle the subsequent calling of methods. Whenever monitor will acquire the lock it will enter into monitorenter state and at the time of releasing it monitorexit will occur.

## **XI. Thread Safe Interface**

All the components in the multi thread application must be thread safe. When lock is acquired on method, and if they are recursive, then there are chances of having a dead lock. To avoid this divide the components in private and public methods in same components. And implement the private methods using the interface. So, when public method tries to access the private methods, it will acquire lock on the interfaces and the implemented private method will be free. This will avoid the chances of getting dead locks.

**Assignment Discussion**

I have implemented database operations using thread safe pattern. Where I divided the db operation components in public and private methods. Public methods then will call the private methods. I have not implemented through interfaces but critical section will acquire the lock on public methods and then those method will call the private methods. And again this is handled by the synchronized keyword in Java.

**XII. Singleton Pattern**

This pattern restrict the instantiation of a class to one object. This pattern is used when only one object should be created and used by all the methods and instances. The class is responsible to create only 1 object of the class and used by all other. Object will be shared among all other. This pattern is useful in few application like database connection.

**Assignment Discussion**

This assignment implements the singleton pattern in creating the database connection. When server starts the connection with the database is created and this only connection is used for every database operations. This is achieved by creating static method in the class and this static method will check the object. If the object is null then will create a new connection. But if the object contains the connection, then it will use the same to perform the operation.

**XIII. Database Access Layer Pattern**

Database Access layer pattern is a type of an infrastructural pattern. Basic purpose of this pattern is to separate the application logic from the persistent storage. This will be implemented using the separate class for database operation which will communicate with the model class of the application. The application logic will send the request to DB access layer. DB access layer will then perform the operation (DML) and fetch the data from the database. The result obtained from the database be it result set or error message will then returned back to the model class. Database access layer returns an object which holds the query result and not the actual row or the fields from the database tables. The advantage of implementing this pattern is abstraction. It hides the actual database operation from the application logic.

**Assignment Discussion**

In the assignment, we have created new class 'executeDatabase.java' which will create the queries based on the request from the model and then execute the same via different method. Different methods of this class will create the queries and send it to the one common private method which will execute the queries. All the request coming to the server, will use only 1 object of this class as singleton pattern is applied on database connection. This will ensure no dirty read/write operation performed any time via model class.



## 5. UML Diagrams

### I. Class Diagram and working

Figure 9 shows the class diagram created for the market place application. It has one client and server classes which are the entry point of the application on client and server respectively. There are 3 views User, Administrator and Customer view. Based on the roles these views are displayed. Initial request will be sent to front controller from User view and once it is processed, and response will go back to dispatcher class which will display the corresponding view. On a client side command pattern is implemented where classes have implemented the command Interface and that will be called by command Invoker class. Similarly abstract factory class is implemented to 2 concrete factories. Communication between client and server is handled by RMI client and RMI server classes.

On a server side, different controllers are designed for different functionality. I.e. controller for each User, Admin and customer functionality. These controller classes are then pass the requests to the concern model classes which have the actual business logic of the application. Once they perform the business logic return the response back to the views using similar path.

Views are designed using Abstract Factory pattern where dispatcher will create the object of the concrete factories (Admin or customer) based on the role received to him from server. And then subsequent admin or customer view will be displayed to the user.

When user is authenticated, a new session object is created for him which contains the user role. This object is pass through each transaction carried out by the user. Role based access control is provided in this application using the user role. Whenever admin or customer user asks for any operation, before executing the concern request, AuthorizationInvocationHandler class will check the user role is matching with the access required to perform that operation. For this we have used custom annotation (RequiresRole). This checking is carried out using proxy object on the server side.

As a part of assignment # 5, I have implemented the persistent storage of MySql database. So to create a connection with the database, new class is created name 'DatabaseConnection'. Method of this class will be called once when the server starts to make a database connection. Also I have created another class 'executeDatabase' which handles all the operation related to the database. Method of this class will use the connection made earlier to perform the database operation. Different model class will request for the database operation methods which are in 'executeDatabase' class.

Once customer user login to the application, customer view will display 3 option. Browse product which will display all the available products which he can purchase to the customer user along with its few details. (All the products whose quantity is not 0). Another option is to add the product to the shopping cart with the desired quantity. Third option is to view the shopping cart

and purchase the products added in the cart. Customer user will see all the products added in his cart, if customer user confirms to purchase, product will be purchased in mentioned quantity (if available) and will be delivered.

Admin user will get option to browse the products and add new product to the product list. Once he login to the application. Admin user will get all the products along with its details irrespective of its quantity. When the admin user select the option to add new product, he needs to enter all the details about the products and on confirmation the product will be added to the product list which then will be available for customer to purchase. Admin user can also update the product details, he needs to mention the product id which needs to be updated. Along with this admin needs to select which details he wants to update like description, price or quantity and enter new corresponding detail. Admin can remove the product by entering the product id. Admin will also have the option to add new administrator. For this he needs to enter all the basic details, if username is available then new admin will be created. Similarly, he can add new customer. Admin can also remove a customer from the user list. By entering the username of the customer he needs to remove.



## II. Sequence Diagram

### 1. Login Activity Sequence

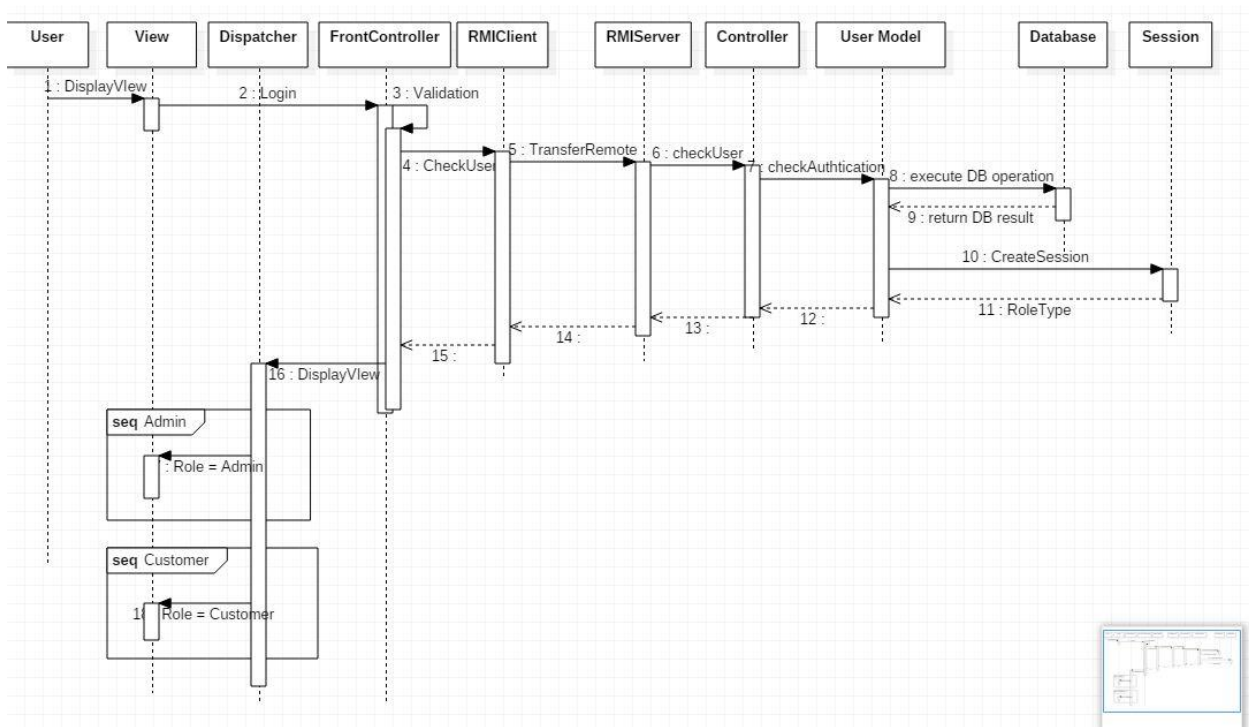


Figure 10 Login Sequence Diagram

Whenever user wants to log in to the application, his credentials are will go to front controller and then sent to server through RMI. On server, User model will validate the credentials by querying to the database and if authenticated then creates a new session for the user along with its user role. This session object is then passed back to the front controller on client and based on the user role dispatcher will show the respective view to the user.

### 2. Admin User Sequence

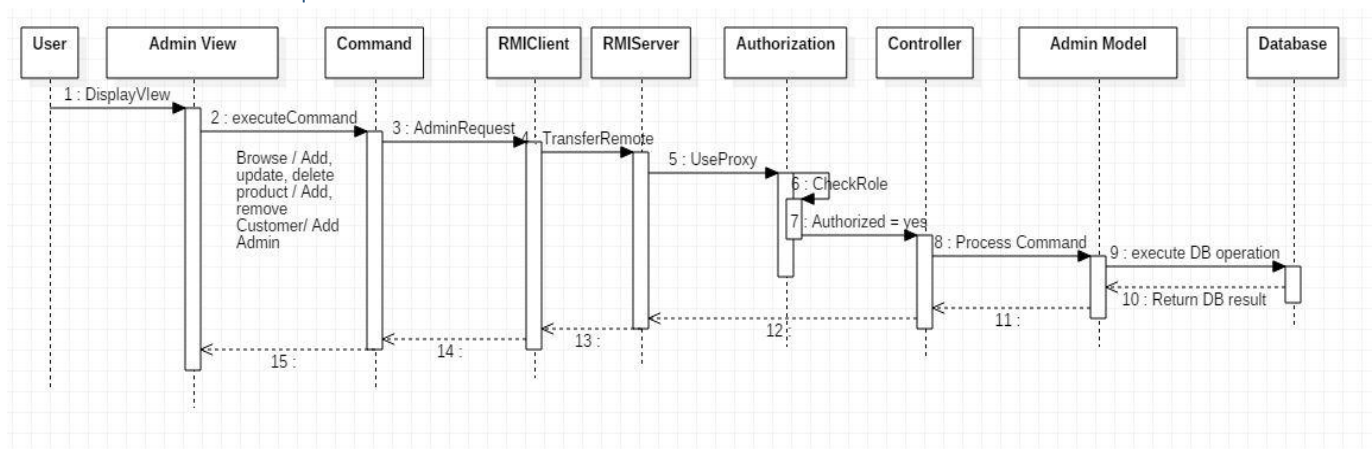


Figure 11 Admin user Sequence Diagram

This sequence diagram shows the sequence flow for Admin's any request operation. When Admin view request for any operation, then it is send as command to invoker and based on the command, it will call the desired method and sent the request to the server through RMI. Then on server side, whether the user has desired role to perform the requested operation is checked through proxy pattern. If he is authorized then the respective method is call and response is sent back to the client view. It will make a database connection to get the desired result and model will validate the result before sending it to the client.

### 3. Customer user Sequence

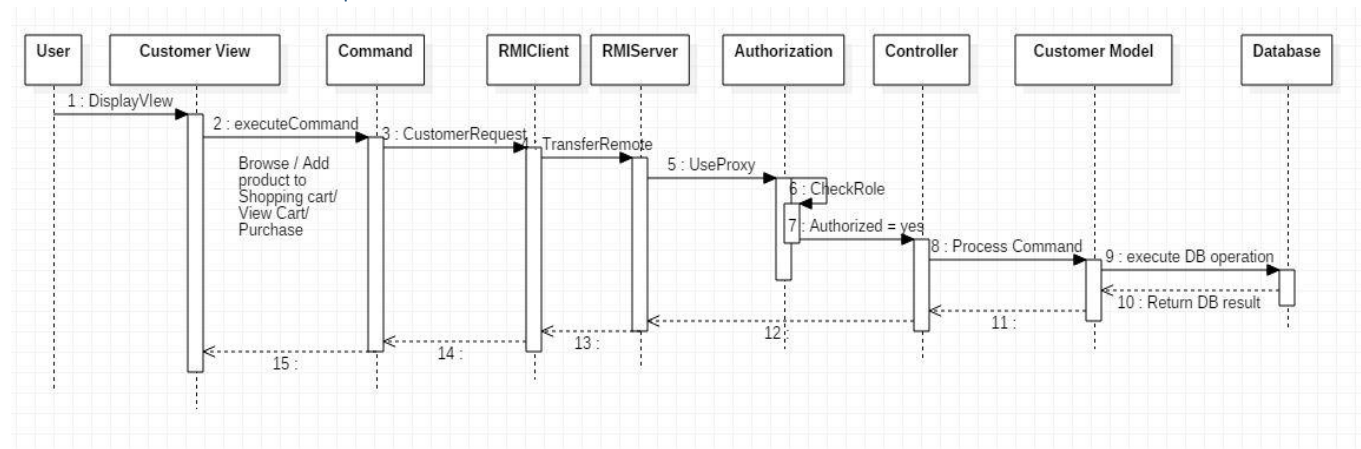


Figure 12 Customer user Sequence Diagram

This sequence diagram shows the sequence flow for customer's request operation. When customer view request for any operation, then it is send as command to invoker and based on the command, it will call the desired method and sent the request to the server through RMI. Then on server side, whether the user has desired role to perform the requested operation is checked through proxy pattern. If he is authorized then the respective method is call and response is sent back to the client view. It will make a database connection to get the desired result and model will validate the result before sending it to the client.

## 6. Database Schema

The database schema designed for market place application is a relational database. It is created on mySql. It contains 4 tables which are connected to each other. Figure 13 shows the schema design. **'User'** table contains the details of all the users along with their username and passwords. UserID is the primary key of the table. This table contains one for foreign key to **'usertype'** table which have type of users stored in it. (0-customer, 1-admin). **'product'** table contains the details of the product. Primary key of the table is productid. It also have other fields as name, description, type, price and quantity. Then 4<sup>th</sup> table is **'shoppingcart'** table which stores the user's shopping cart information. Shid is the primary key. And it stores which user has added which product in his cart in what quantity. It will also store the total amount based on price and quantity ordered. This table has 1 field **'isPurchased'** when user purchase the product this field will be set to 1 or by default it is 0. Two foreign key constraints are

applied on this table for userid and productid with on delete cascade effect. Once user is deleted or product is removed then all the data related to them will be removed from shoppingcart table.

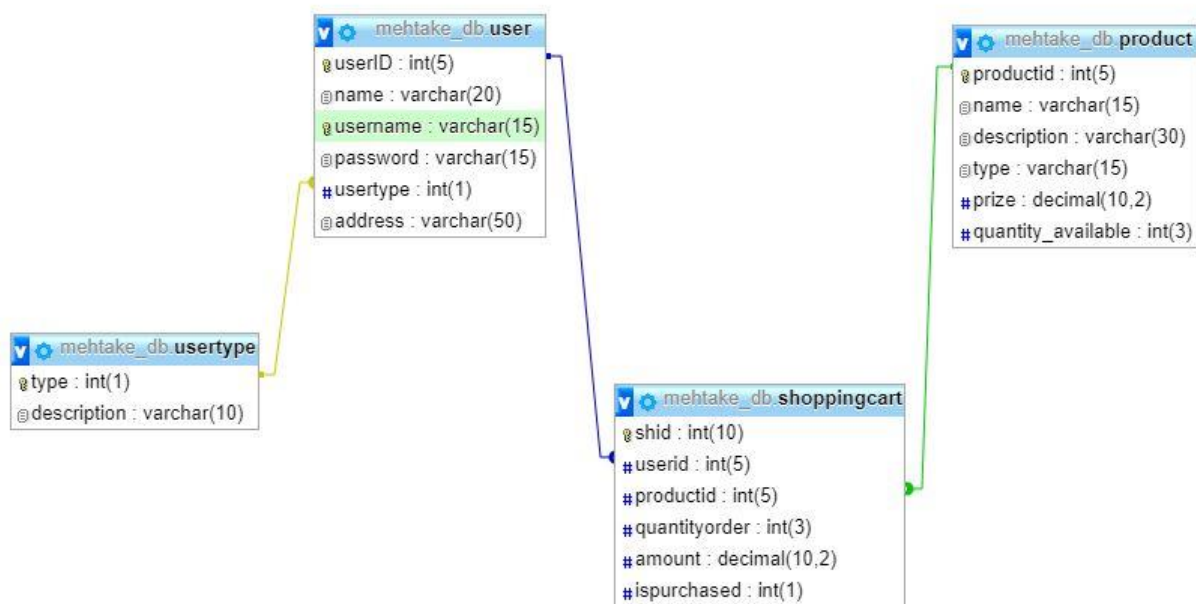


Figure 13 Database Schema

## 7. Assignments Transition/ Overview

### I. Assignment 1

To develop a market place application, I have initially started with the designing domain model where first I have identified different entities and their attributes. And then divided this domain model in Model –View –Controller Architecture framework. I kept view on my client side and controller and model on the server side. Client and server communicate through RMI mechanism implemented through Java. The purpose of this assignment is to create a framework on which we can implement the entire functionality of the system.

### II. Assignment 1 to Assignment 2

Initially, I worked on the comments received in assignment # 1. I need to separate the framework (RMI communication) from the view. Views are displayed to the user, so those cannot have the internal framework of how communication take place between client and server. So, I created new class which will act as RMI controller on both side and will responsible only for communication. And all the requests from views are now forwarded to RMI client through front controller and command pattern. On a server side, controller will receive the request through RMI server and then sent it forward.

Also in this assignment 2, three patterns front controller, command and abstract factory patterns are implemented whose details are mentioned in above sections.



### III. Assignment 2 to Assignment 3

As I received the comment on abstract factory pattern, I redesigned and implemented the Abstract Factory Pattern where each of the concrete factory will call the respective view based on the user role. And concrete factories will override the method of abstract factory. The detail explanation is mentioned in the section 4.3

As a part of this assignment, I have implemented Authorization pattern with Role Based Access Control which is implemented using Java annotation. Proxy and Reflection pattern is also implemented in the assignment to check the user has desired role to access the particular functionality. Proxy object will invoke and check the user role whenever admin/ customers operation is called by their respective objects. Annotation name and whether it's present or not on those operation is obtained by using the reflection pattern. Sessions objects are created for each successful login of the user. This session objects will be pass through each function call done by that particular user. Session objects holds the username, its role and isAuthenticated information. This information is used to check the RBAC and display desired view. In future, session object will also be used to track the transaction carried out by that particular user during that login session.

### IV. Assignment 3 to Assignment 4

As a part of assignment 4, I have implemented the 3 functionalities in Market place application. And also made application concurrent. Sever resides on in-csci-rrpc01.cs.iupui.edu and other 5 machine can be used as client. All this machine can simultaneously access the market place application and use the implemented functionality. Now, admin user can browse all the product present in the product list with its details. I have used txt file to maintain the product list. So as it will be persistent for all the customers. This text file is read in the list of type itemList. This itemList is a class whose objects will hold the product details and this will be stored in the list. Admin can also add new product to the list along with the require details. Customer user can browse available products (whose quantity is greater than 0) and can add the product to the shopping cart. Once the product is added to the shopping cart, user can purchase the same if the quantity is available. This all implemented using synchronized method in Java.

### V. Assignment 4 to Assignment 5

First I addressed the comments received in assignment 4 feedback. I have changed the scope of member variable and instance variable of all classes to private or public as per requirement from default. The database schema is designed and created on MySQL and connection is established in Marketplace application. Then started implementing the functionalities as part of assignment 5. As a part of assignment 5, I added the admin functionality of update and delete the products. Add new administrator. Add and remove the customer user. For customer user, added the functionality of view the shopping cart and purchasing the product form the cart. Also provide the functionality to register the new customer to the application. All this functionalities are implemented keeping in mind the different synchronization patterns. These synchronization

patterns will help to concurrently access the system by different client. Also tried to handle different scenario and exception while entering the user inputs.

## VI. Assignment 5 to Assignment 6

I have already implemented the database Access Layer Pattern as a part of the assignment 5. So, I have not made many changes in the implementation code. I did add section giving information about the database access layer pattern in the report and added the final conclusion part.

## 8. Sample Runs

### I. Server start and Login view

The image shows two terminal windows side-by-side. The left window is titled 'mehtake@in-csci-rrpc01:~/OODP/Assignment/Assignment4' and shows the server startup process. It includes commands like 'rmiregistry 20126' and 'java -Djava.security.policy=policy server. MarketPlaceServer'. The output shows the server binding to various ports and displaying 'Binding Done!! Welcome to MarketPlace Server \*\*\*\*\*'. The right window is titled 'mehtake@in-csci-rrpc02:~/OODP/Assignment/Assignment4' and shows the client startup. It runs 'java -Djava.security.policy=policy client. Client' and displays 'Market Place application is starting...' followed by a welcome message and a menu with options: 1. Login, 2. Register, 3. Exit. It prompts the user to 'Please enter your choice:'.

Once the server start, it will bind with the view on the client. And server will display start message. And on the other console, client will display initial user view with option to login and register.

### II. Admin Login

The image shows two terminal windows side-by-side. The left window is titled 'mehtake@in-csci-rrpc01:~/OODP/Assignment/Assignment5' and shows the server startup process. It includes commands like 'make', 'javac', and 'java -cp ".:mysql-connector.jar" -Djava.. cy server.MarketPlaceServer'. The output shows the server binding to various ports and displaying 'Binding Done!! Welcome to MarketPlace Server \*\*\*\*\*'. It then shows the server connecting to the database and logging in as 'admin'. The right window is titled 'mehtake@in-csci-rrpc02:~/OODP/Assignment/Assignment5' and shows the client startup. It runs 'java -Djava.security.policy=policy client.Client' and displays 'Market Place application is starting...' followed by a welcome message and a menu with options: 1. Login, 2. Register, 3. Exit. It prompts the user to 'Please enter your choice:' and then shows the user selecting '1. Login'. The client then prompts for 'Enter the username: admin' and 'Enter the password: admin', and finally displays 'Administrator Login Successful' and '\*\*\*\*\*Welcome Administrator to MarketPlace \*\*\*\*\*'. It then shows the client displaying a menu with options: 1. Browse for Products, 2. Add a Product, 3. Update the Product, 4. Delete a Product, 5. Add Administrator, 6. Add new customer, 7. Remove a customer, 8. Log out. It prompts the user to 'Please enter your choice:'.



There are 2 kinds of login provided in this application. One is admin and other is Customer user. On entering valid credentials of admin, it displays the admin view. And on server side, will get message showing access. Admin view will display all the options of admin have in MarketPlace Application.

### III. Admin - browse Product

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
[mehtake@in-csci-rpc01 Assignment5]$ make
javac -cp ".:mysql-connector.jar" -g server/itemList.java
javac -cp ".:mysql-connector.jar" -g view/Customer.java
javac -cp ".:mysql-connector.jar" -g model/executeDatabase.java
[mehtake@in-csci-rpc01 Assignment5]$ java -cp ".:mysql-connector.jar" -Djava.
cy server.MarketPlaceServer
Market Place Server is starting...
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2089/user
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2089/cust
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2089/admin
Binding Done!! Welcome to MarketPlace Server *****
Connecting database!!
MarketPlace Database connected!!
User admin attempts for login!!
Credential of admin user: admin are valid. Allow access!!
User keyur attempts for login!!
Credential of customer user: keyur are valid. Allow access!!
User john attempts for login!!
Credential of customer user: john are valid. Allow access!!
User robert attempts for login!!
Credential of customer user: robert are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to browse the products!!

```

```

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
1
*****The Product List*****
ItemID Name Description Type Price Quantity available
-----
1 iPad Tablet Electronic 499.00 5
2 Watch Watch Accessories 100.00 1
3 MacBook Laptop Electronic 800.00 10
4 iWatch Smart Watch Accessories 300.00 0
5 T Shirt Men's T shirt Clothes 55.00 7
-----
****Welcome Administrator to MarketPlace ****
Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:

```

When admin selects browse option, he will be displayed with all the products present in the product list along with its details. Admin will also see the products whose quantity is 0.

## IV. Admin – Add product

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2088/user
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2089/cust
MarketPlaceServer: binding it to name: //in-csci-rpc01.cs.iupui.edu:2089/admin
Binding Done!! Welcome to MarketPlace Server *****
Connecting database!!
MarketPlace Database connected!!
User admin attempts for login!!
Credential of admin user: admin are valid. Allow access!!
User keyur attempts for login!!
Credential of customer user: keyur are valid. Allow access!!
User john attempts for login!!
Credential of customer user: john are valid. Allow access!!
User robert attempts for login!!
Credential of customer user: robert are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to browse the products!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to add the products!!
6,Jeans,Men's Jeans,Clothes,75.0,10
6 productId available!! New Product can be added.
New product with id: 6 added successfully!!
Admin wants to add the products!!
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
Enter name of the product:
Jeans
Enter Description of the product:
Men's Jeans
Enter Type of the product:
Clothes
Enter price of the product:
75
Enter quantity of the product:
10
Details of the new product:
6  Jeans      Men's Jeans      Clothes      75.00  10
Kindly confirm, all information is correct and ADD product to the productList (Y/N)?
Y
New Product is added to the Product List!!
*****Welcome Administrator to MarketPlace *****
Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:

mehtake@in-csci-rpc06:~/OODP/Assignment/Assignment5
8. Log out
Please enter your choice:
2
*****Add Product to Product List*****
Enter product ID:
6
Enter name of the product:
Cap
Enter Description of the product:
Accessories
Enter Type of the product:
Clothes
Enter price of the product:
25
Enter quantity of the product:
5
Details of the new product:
6  Cap      Accessories      Clothes      25.00  5
Kindly confirm, all information is correct and ADD product to the productList (Y/N)?
Y
New Product can NOT be added to the Product List as same Product ID is present in the list!!

```

Admin can add new product to the product list. When he selects the respective operation, he will be prompted to enter the product details. But if user enters the same product id which is already available in the list then product cannot be added and user will get respective message.

But the user enters new product ID along with all other details, then the new product is added to the list and will be displayed while browsing the list of the products.

## V. Admin – Update product

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
MarketPlace Database connected!!
User admin attempts for login!!
Credential of admin user: admin are valid. Allow access!!
User keyur attempts for login!!
Credential of customer user: keyur are valid. Allow access!!
User john attempts for login!!
Credential of customer user: john are valid. Allow access!!
User robert attempts for login!!
Credential of customer user: robert are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to browse the products!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to add the products!!
6,Jeans,Men's Jeans,Clothes,75.0,10
6 productId available!! New Product can be added.
New product with id: 6 added successfully!!
Admin wants to add the products!!
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
Kindly confirm, all information is correct and ADD product to the productList (Y/N)?
Y
New Product is added to the Product List!!

*****Welcome Administrator to MarketPlace *****

Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
3
*****Update Product Details from Product List*****
Enter product ID: 3

Enter which detail of the product: 3 wants to update:
1. Description  2. Price      3. Quantity
2
Enter new price of the product: 950
Update prize of the product id: 3 to 950
Kindly confirm, all information is correct and Update the mentioned product detail (Y/N)?
Y
Product detail is updated in the Product List!!

mehtake@in-csci-rpc06:~/OODP/Assignment/Assignment5
ItemID Name          Description          Type          Price  Quantity ava
lable
-----
1 iPad              Tablet              Electronic    499.00  5
2 Watch            Watch              Accessories   100.00  1
3 MacBook          Laptop              Electronic    950.00  10
4 iWatch           Smart Watch        Accessories   300.00  0
5 T Shirt          Men's T shirt      Clothes       55.00   7
6 Jeans            Men's Jeans        Clothes       75.00   10
-----
*****Welcome Emily Holt to MarketPlace *****

```

Admin needs to enter the product id whose detail needs to be updated. Once admin enters the id, he will be prompted to update which detail (Description, Price, and Quantity). Based on his selection, admin will be prompted to enter new detail. And once admin gives confirmation to update the detail, product details will be updated.

## VI. Admin – Delete Product

Admin needs to enter the product id which he needs to delete. Once the product id is entered, admin will be prompted with confirmation. Based on his confirmation product will be deleted.

If the product id is not present in the product list, then admin will get respective message.

Below screenshot also shows the synchronization applied. Two admin tries to delete the same product, then only 1 can delete the product and other will get no product id message.

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
User john attempts for login!!
Credential of customer user: john are valid. Allow access!!
User robert attempts for login!!
Credential of customer user: robert are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to browse the products!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
Admin wants to add the products!!
6,Jeans,Men's Jeans,Clothes,75.0,10
6 productId available!! New Product can be added.
New product with id: 6 added successfully!!
Admin wants to add the products!!
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!
Admin wants to delete the products!!
6 productId available!! Product can be deleted.
Product with id: 6 deleted successfully!!
Admin wants to delete the products!!
Product ID: 6 is unavailable!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
4
*****Delete Product from Product List*****
Enter product ID:
6
Kindly confirm, Do you want to permanently delete the product id: 6 from product list (Y/N)?
y
Product is deleted from the Product List!!

*****Welcome Administrator to MarketPlace *****

Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:

mehtake@in-csci-rpc06:~/OODP/Assignment/Assignment5
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
4
*****Delete Product from Product List*****
Enter product ID:
6
Kindly confirm, Do you want to permanently delete the product id: 6 from product list (Y/N)?
y
Product can NOT be deleted from the Product List as same Product ID is NOT present in the list!!

```

## VII. Admin – Add Administrator

```

mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
User emily attempts for login!!
Credential of admin user: emily are valid. Allow access!!
User john attempts for login!!
Credential of admin user: john are valid. Allow access!!
Admin wants to add the products!!
6,Jeans,Men's Jeans,Clothes,75.0,10
6 productId available!! New Product can be added.
New product with id: 6 added successfully!!
Admin wants to add the products!!
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!
Admin wants to delete the products!!
6 productId available!! Product can be deleted.
Product with id: 6 deleted successfully!!
Admin wants to delete the products!!
Product ID: 6 is unavailable!!
Admin wants to add new Administrator!!
John Burris,john,john,Chicago, Indianapolis
Username: john is unavailable!!
Admin wants to add new Administrator!!
Drew Matthew,drew,drew,texas
drew username available!! Registration process started.
New Admin Drew Matthew is registered successfully!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
5
*****Add New Admin*****
Enter the Name: John Burris
Enter the username: john
Enter the password: john
Enter the address: Chicago, Indianapolis
Username already present. Please use another username!!

*****Welcome Administrator to MarketPlace *****

Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:

mehtake@in-csci-rpc06:~/OODP/Assignment/Assignment5
8. Log out
Please enter your choice:
5
*****Add New Admin*****
Enter the Name: Drew Matthew
Enter the username: drew
Enter the password: drew
Enter the address: texas
Admin: Drew Matthew is registered/added successfully!!
Please share the credentials with admin to login.

*****Welcome Emily Holt to MarketPlace *****

```

Admin needs to fill all the details in order to add new administrator. If username entered by the admin is available (not used by any other user) then new administrator will be created or else admin will get message to change username.



## VIII. Admin – Add Customer

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
Admin wants to add the products!!
6,Jeans,Men's Jeans,Clothes,75.0,10
6 productId available!! New Product can be added.
New product with id: 6 added successfully!!
Admin wants to add the products!!
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!
Admin wants to delete the products!!
6 productId available!! Product can be deleted.
Product with id: 6 deleted successfully!!
Admin wants to delete the products!!
Product ID: 6 is unavailable!!
Admin wants to add new Administrator!!
John Burris,john,john,Chicago, Indianapolis
Username: john is unavailable!!
Admin wants to add new Administrator!!
Drew Matthew,drew,drew,texas
drew username available!! Registration process started.
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
6
*****Add New Customer*****
Enter the Name: Ron
Enter the username: ron
Enter the password: ron123
Enter the address: USA
Customer: Ron is registered/added successfully!!
Please share the credentials with customer to login.

****Welcome Administrator to MarketPlace ****

Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
```

Admin needs to fill all the details in order to add new customer. If username entered by the admin is available (not used by any other user) then new customer will be created or else admin will get message to change username.

## IX. Admin – Remove Customer

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
6,Cap,Accessories,Clothes,25.0,5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!
Admin wants to delete the products!!
6 productId available!! Product can be deleted.
Product with id: 6 deleted successfully!!
Admin wants to delete the products!!
Product ID: 6 is unavailable!!
Admin wants to add new Administrator!!
John Burris,john,john,Chicago, Indianapolis
Username: john is unavailable!!
Admin wants to add new Administrator!!
Drew Matthew,drew,drew,texas
drew username available!! Registration process started.
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!
Admin wants to delete the Customer user!!
Customer user: mattew is not present!!
Admin wants to delete the Customer user!!
ron username present!! Customer user can be deleted.
Customer user with username: ron deleted successfully!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
7
*****Delete Customer from User List*****
Enter username of the customer:
ron
Kindly confirm, Do you want to permanently delete the customer user: ron from user list (Y/N)?
Y
Customer user is deleted from the User List!!

****Welcome Administrator to MarketPlace ****

Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:

mehtake@in-csci-rpc06:~/OODP/Assignment/Assignment5
Please enter your choice:
7
*****Delete Customer from User List*****
Enter username of the customer:
mattew
Kindly confirm, Do you want to permanently delete the customer user: mattew from user list (Y/N)?
Y
Customer user can NOT be deleted from the User List as username is NOT present in the list!!

****Welcome Emily Holt to MarketPlace ****
```

In order to delete the customer from user list, admin needs to enter his username, if user name is valid then the customer account is deleted or else user will get message regarding invalid username.

## X. Customer Login

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!
Admin wants to delete the Customer user!!
Custoemr user: mattew is not present!!
Admin wants to delete the Customer user!!
ron username present!! Customer user can be deleted.
Customer user with username: ron deleted successfully!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than
Product successfully added to the customer's shopping cart!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than
Product successfully added to the customer's shopping cart!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 5 with quantity: 10 to his
Product can NOT be added to the cart as ordered quantity is greater
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 3 to his
Product can be added to the cart as ordered quantity is less than
Product successfully added to the customer's shopping cart!!
Customer wants to view shopping cart!!

mehtake@in-csci-rpc03:~/OODP/Assignment/Assignment5
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
2
*****Add Product to Shopping Cart*****
Enter product ID:
1
Enter quantity of product:
3
Product Added to the Shopping cart!!
*****Welcome Keyur Mehta to MarketPlace *****
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
3
*****The Product List*****
ProductID      Name              Quantity Ordered  Amount
-----
1              iPad              3                 1497.00
2              Watch             1                 100.00
-----
Kindly confirm, Do you want to purchase all prodcuts in shopping cart (Y/N)?
```

On entering valid credentials of customer, it displays the customer view. And on server side, will get message showing access. Customer view will display all the options customers have in MarketPlace Application.

## XI. Customer – Browse product

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
Product ID: 6 is unavailable!!
Admin wants to browse the products!!
Admin wants to update the products!!
3 productId available!! Product details can be Updated.
Detail of the Product id: 3 updated successfully!!
Admin wants to browse the products!!
Admin wants to delete the products!!
6 productId available!! Product can be deleted.
Product with id: 6 deleted successfully!!
Admin wants to delete the products!!
Product ID: 6 is unavailable!!
Admin wants to add new Administrator!!
John Burris,john,john,Chicago, Indianapolis
Username: john is unavailable!!
Admin wants to add new Administrator!!
Drew Matthew,drew,drew,texas
drew username available!! Registration process started.
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!
Admin wants to delete the Customer user!!
Custoemr user: mattew is not present!!
Admin wants to delete the Customer user!!
ron username present!! Customer user can be deleted.
Customer user with username: ron deleted successfully!!
Customer wants to browse the products!!

mehtake@in-csci-rpc03:~/OODP/Assignment/Assignment5
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
1
*****The Product List*****
ItemID  Name      Description      Type      Price  Quantity
-----
available
-----
1       iPad      Tablet          Electronic 499.00  5
2       Watch     Watch           Accessories 100.00  1
3       MacBook   Laptop          Electronic 950.00  10
5       T Shirt   Men's T shirt   Clothes    55.00  7
-----
*****Welcome Keyur Mehta to MarketPlace *****
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
```

When customer user select the browse product option, he will get all the details of **available** products. I.e. products whose quantity is greater than 0 will only be visible to the customer.



## XII. Customer – Add product to Shopping cart

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
drew username available!! Registration process started.
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!
Admin wants to delete the Customer user!!
Custoemr user: matthew is not present!!
Admin wants to delete the Customer user!!
ron username present!! Customer user can be deleted.
Customer user with username: ron deleted successfully!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than t1
Product successfully added to the customer's shopping cart!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than t2
Product successfully added to the customer's shopping cart!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 5 with quantity: 10 to his1
Product can NOT be added to the cart as ordered quantity is greater
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 3 to his
Product can be added to the cart as ordered quantity is less than t
Product successfully added to the customer's shopping cart!!

mehtake@in-csci-rpc03:~/OODP/Assignment/Assignment5
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
2
*****Add Product to Shopping Cart*****
Enter product ID:
2
Enter quantity of product:
1
Product Added to the Shopping cart!!
****Welcome Keyur Mehta to MarketPlace ****

mehtake@in-csci-rpc04:~/OODP/Assignment/Assignment5
*****Add Product to Shopping Cart*****
Enter product ID:
2
Enter quantity of product:
1
Product Added to the Shopping cart!!
****Welcome Steve Johnson to MarketPlace ****

Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
2
*****Add Product to Shopping Cart*****
Enter product ID:
5
Enter quantity of product:
10
Quantity ordered for product id: 5 is greater than the available Quantity!
Kindly recheck the quantity!!

mehtake@in-csci-rpc05:~/OODP/Assignment/Assignment5
4. Log out
Please enter your choice:
1
*****The Product List*****
ItemID Name Description Type Price Quantity
available
=====
1 iPad Tablet Electronic 499.00 5
2 Watch Watch Accessories 100.00 1
3 MacBook Laptop Electronic 950.00 10
5 T Shirt Men's T shirt Clothes 55.00 7
=====
****Welcome Justine Robert to MarketPlace ****

Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
```

Once user select the option to add product to shopping cart, he needs to enter the product ID and quantity needs to purchase. Then if the quantity greater than the available quantity, product will not be added to the cart and user will get appropriate message. User can add multiple products to his cart. If customer tries to enter same product twice or more times, then the quantity of the product will be updated. Product will be added only once in the cart. (If it is not purchased)

```
mehtake@in-csci-rpc03:~/OODP/Assignment/Assignment5
table quantity!! Kindly recheck the quantity.
=====
****Welcome Keyur Mehta to MarketPlace ****

Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
2
*****Add Product to Shopping Cart*****
Enter product ID:
1
Enter quantity of product:
2
Product's quantity is updated as product is already present in the customer's shopping cart!!
```

### XIII. Customer – View Shopping Cart

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
New Admin Drew Matthew is registered successfully!!
Admin wants to add new Customer!!
Ron,ron,ron123,USA
ron username available!! Registration process started.
New Customer Ron is registered successfully!!
Admin wants to delete the Customer user!!
Custoemr user: mawew is not present!!
Admin wants to delete the Customer user!!
ron username present!! Customer user can be deleted.
Customer user with username: ron deleted successfully!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than t
Product successfully added to the customer's shopping cart!!
Customer wants to browse the products!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 2 with quantity: 1 to his
Product can be added to the cart as ordered quantity is less than t
Product successfully added to the customer's shopping cart!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 5 with quantity: 10 to his
Product can NOT be added to the cart as ordered quantity is greater
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 3 to his
Product can be added to the cart as ordered quantity is less than t
Product successfully added to the customer's shopping cart!!
Customer wants to view shopping cart!!

mehtake@in-csci-rpc03:~/OODP/Assignment/Assignment5
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
2
*****Add Product to Shopping Cart*****
Enter product ID:
1
Enter quantity of product:
3
Product Added to the Shopping cart!!

****Welcome Keyur Mehta to MarketPlace ****
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
3
*****The Product List*****
ProductID      Name      Quantity Ordered      Amount
-----
1              iPad          3              1497.00
2              Watch         1              100.00
-----
Kindly confirm, Do you want to purchase all prodcuts in shopping cart (Y/N)?
```

Once customer select the option to view shopping cart, all the products added in his list will be displayed. And customer will get option to purchase the products. Based on his confirmation order will be placed.

```
mehtake@in-csci-rpc01:~/OODP/Assignment/Assignment5
Product successfully added to the customer's shopping cart!!
Customer wants to add the product to his shopping cart!!
CustomerID: 2 wants to add a product id: 5 with quantity: 10 to his cart.
Product can NOT be added to the cart as ordered quantity is greater than the available quantity!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 3 to his cart.
Product can be added to the cart as ordered quantity is less than the available quantity!!
Product successfully added to the customer's shopping cart!!
Customer wants to view shopping cart!!
Customer wants to view shopping cart!!
Customer wants to Purchase the product(s) from his shopping cart!!
Product: 1 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 1 with quantity: 3
Product: 2 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 2 with quantity: 1
Customer wants to Purchase the product(s) from his shopping cart!!
Product id: 1 can NOT be purchased as ordered quantity is greater than the available quantity!!
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!!
Customer wants to add the product to his shopping cart!!
CustomerID: 3 wants to add a product id: 1 with quantity: 2 to his cart.
Product quantity can be updated in the cart as its already present in cart!!
Product quantity is updated successfully in the shopping cart!!
Customer wants to view shopping cart!!
Customer wants to Purchase the product(s) from his shopping cart!!
Product: 1 can be purchased ordered quantity is less than the available quantity!!
Order placed for Product: 1 with quantity: 2
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!!
Admin wants to browse the products!!

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
3
*****The Product List*****
ProductID      Name      Quantity Ordered      Amount
-----
1              iPad          2              998.00
2              Watch         1              100.00
-----
Kindly confirm, Do you want to purchase all prodcuts in shopping cart (Y/N)?
y

*****Order Detail*****
Partial Order is placed!!
-----
Order placed for Product id: 1 with quantity: 2
Product id: 2 can NOT be purchased as ordered quantity is greater than the available quantity!! Kindly recheck the quantity.
-----
****Welcome Keyur Mehta to MarketPlace ****
Enter the option from below:
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:

mehtake@in-csci-rpc04:~/OODP/Assignment/Assignment5
1. Browse for Products
2. Add product to Shopping cart
3. View Shopping Cart and Purchase product(s)
4. Log out
Please enter your choice:
3
*****The Product List*****
ProductID      Name      Quantity Ordered      Amount
-----
1              iPad          3              450.00
2              Watch         1              100.00
-----
Kindly confirm, Do you want to purchase all prodcuts in shopping cart (Y/N)?
y

*****Order Detail*****
Complete Order is placed and will be delivered to your door step!!
-----
Order placed for Product id: 1 with quantity: 3
Order placed for Product id: 2 with quantity: 1
-----

mehtake@in-csci-rpc02:~/OODP/Assignment/Assignment5
5. Add Administrator
6. Add new customer
7. Remove a customer
8. Log out
Please enter your choice:
1
*****The Product List*****
ItemID Name      Description      Type      Price      Quantity available
-----
1      iPad      Tablet          Electronic 499.00      0
2      Watch     Watch           Accessories 100.00      0
3      MacBook   Laptop          Electronic 950.00      10
4      iWatch    Smart Watch     Accessories 300.00      0
5      T Shirt   Men's T shirt   Clothes    55.00      7
-----
****Welcome Administrator to MarketPlace ****
Enter the option from below:
1. Browse for Products
2. Add a Product
3. Update the Product
4. Delete a Product
5. Add Administrator
```

If same product is placed by multiple customers, then order of 1<sup>st</sup> customer will be placed successfully and 2<sup>nd</sup> customer will get message regarding the quantity of the product. Product will remain in his cart, until he updates the product quantity.





## 9. Conclusion

The agile methodology of the designing and developing the MarketPlace Application will result in good and flawless design. Also this first iteration created the basic structure of the application using MVC architecture and the communication between different models are setup through RMI.

As a part of assignment 2, I have implemented 3 design pattern which results in good design of the market place application. Also corrected the feedback received from the assignment 1. This assignments made the application independent of different modules creating more layers.

In assignment 3, we implemented authorization along with Role based Access Control, reflection and proxy pattern in our market place application. Also we created session objects which will be used in all transaction carried out by the user during his 1 login session.

In assignment 4, we achieved the concurrency using the Java RMI and synchronized method. We have also implemented the browse product functionality for both type of customers. Add product for admin customer and purchase product functionality for the customer user.

In assignment 5, apart from implanting all the remaining functionalities main focus was to achieve the synchronization for multi thread application. This is implemented using the Java synchronized feature. Also connecting the application with persistent storage was integral part of the assignment.

In assignment 6, I have added the section in the report giving information about the Database Access Layer Pattern.

### I. Final Conclusion

#### 1) What did you like about this assignment?

The agile/ incremental methodology used for the assignment, during entire semester where we are building the new content on the existing one and keep on improving the same. The design patterned which we learned during the course of the semester was more cleared as we have actually implemented it via this set of assignments.

#### 2) What did you dislike about this assignment?

If implementation of the database would have been introduced in assignment 4 then it would be better as we started implementing some of the functionalities from A4, which we need to change it once the database is incorporated.

#### 3) What would you change about your design if you could go back? Why?

There are couple of things which I would like to change as per design approach, as I found some difficulty at the later stages of the assignment.

1. During the A1, I researched on the net and found that there should be different controllers and models for different type of the operation/ user. So, accordingly I designed separate server controllers for admin and customers. But in A3, when we need to implement the RBAC, it was not failing in my case as all my customer request goes to customer controller and admin request goes to admin controller. So, I thought RBAC is not of much use. But then I tried to separate the browse functionality in different interface (and implement in separate 1 controller). The method will have both the role assigned to it. But later on, I came to know though multiple annotation is allowed only from Java 7, it is bit complicated to implement it. I tried for couple of days but couldn't figure it out. So, I am not sure which design choice is better having separate controller or only 1 but now I am having thought to change it.
2. Due to other class commitments and end of semester load, I was not able to give much time in A5 where we need to synchronize all our operations. I have just used 'synchronized' word which eventually handles all my concurrency and synchronization issue very well. But yes, if time would have permitted, I would like to implement few of the 5 patterns without using synchronized word using threads etc.
3. Another thing, which I might need to change is in case of layering structure. My server controller and command pattern classes are not doing much work other than passing the request further. In case of server controller needs to do some responsible work which I need to figure it out. And creating different command classes and passing those ahead is also not that significant. Yes, it provides the level of abstraction, but I think, if they could have been used for some other work like validations or processing then it would have been great.

## References

- [1] [Online]. Available: <https://www.javatpoint.com/RMI>.
- [2] [Online]. Available: [https://developer.chrome.com/apps/app\\_frameworks](https://developer.chrome.com/apps/app_frameworks).
- [3] [Online]. Available: <https://daveh.io/img/the-mvc-pattern.png>.
- [4] [Online]. Available: <http://www.oodesign.com/command-pattern.html>.
- [5] [Online]. Available: <https://www.geeksforgeeks.org/abstract-factory-pattern/>.
- [6] [Online]. Available: <http://www.cnblogs.com/callwangxiang/archive/2009/02/15/1391043.html>.