Assignment 1

Adithya Morampudi

In this assignment we are asked to implement the skeleton for the online MarketPlace application, The main aim of the assignment is to identify the classes required for the proper implementation of the project, to get the RMI and the MVC pattern up and running.

Software Design : The main part of this project is about the MVC pattern where I have implemented the client side part of the application to be the view, the server of the java RMI to be the model and a controller file which is responsible for maintaining the business logic of the application.

- Model file : Server.java, this is the Model of the application, where in right now I just have the main method of the application server and this class also binds the services to the naming server, which can be discovered by the client using java RMI naming server, which is the benefit of using any service like java RMI
- Controller file : MPlaceImplementation.java, this is the controller file where in it has all the required business logic for the application to run successfully, as of now I have the methods required for the project, which are loginAdmin() – the method which is responsible for making Admin logins, userLogin() – the method which is responsible for making userLogins, the reason for making them separate is for obvious reasons of the privileges which a admin might have. browsingAdmin() – the method for the admin where in he can use this method to find the required products, this method only takes care of browsing the items and so on, the reason for wiring them all in a single class is for simplicity as of now, if there is a requirement then I may change these methods into separate classes.
- View File : Client.java this is the file where the client can see what's going on, he can use all the functionality which is supported by the server, right now there is nothing in the client file, since this assignment doesn't focus much on the client side part of the project, in the second assignment I will implement a much better looking view file for the client.
- The client file has a lookup function where it looks for the services offered by the RMiRegistry and then it selects the required service, in my case the service which I am going to use is predefined and since I have only one server registered at the Registry, I will be using the only available service.

Interface File: During the implementation of RMI, we have to write an Interface File which is similar to RPC, In my Interface file I have all the abstract methods that the Controller will implement and eventually the Client(view) requests those methods using Remote calls.

This file has all the abstract methods required for the application.

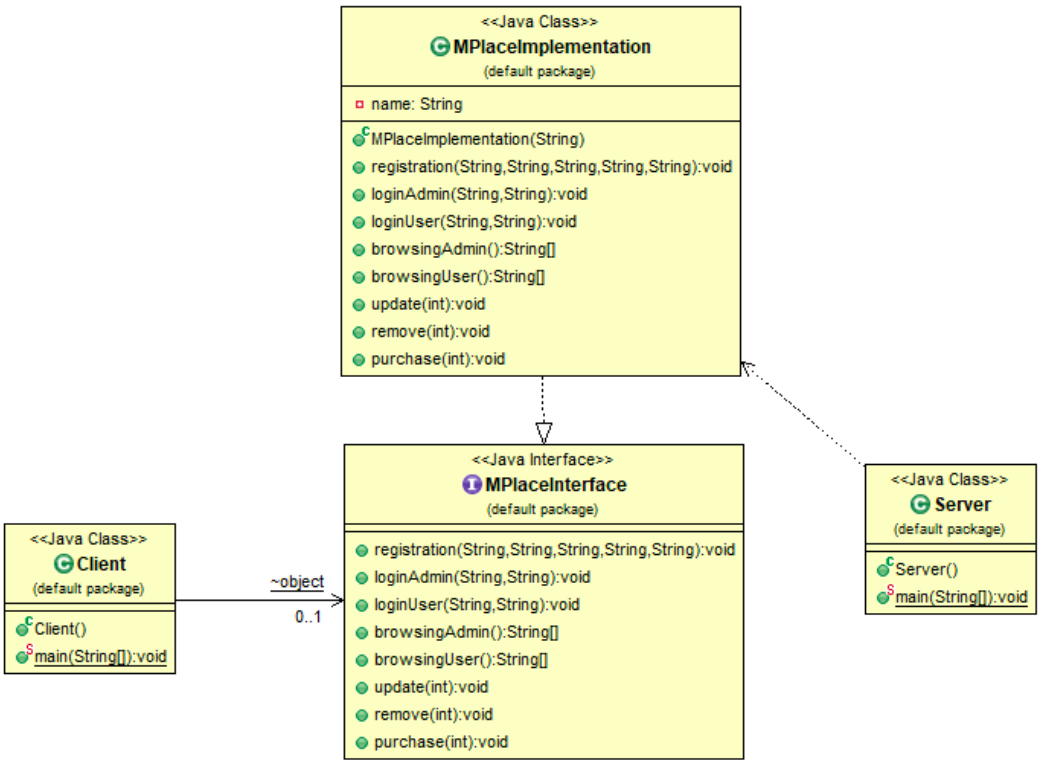
Execution of the Program

- Whenever a Remote procedure call program is Executed some steps are executed in the background, which the client doesn't have knowledge of. The client is unaware of the network and he thinks as if it is a local procedure call, which was the main moto behind creating RPC and from there came the RMI
- So once the System starts running, the Server waits for a client to connect or request a procedure, at this stage the server is idle and once it gets a request it immediately creates a thread and then the execution of the procedure requested is carried out
- The background process is carried out in the server stub and the server network interface.
- Once the procedure is called the server carries out the required operation and sends the data back to the client by performing required data conversions and serializations.

Client Side:

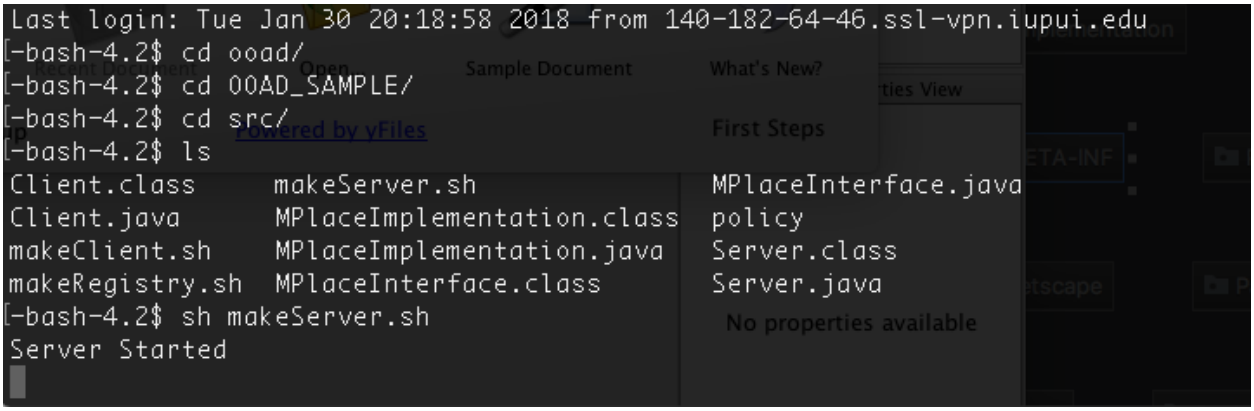
- As mentioned in the class that a third party RMIRegistry contains the information about the procedures available for execution at the server side, the client refers to the corresponding Registry and then requests a required operation or procedure which is requested by the client.
- Once the client requests a procedure, the client side program takes the required input(if any) from the user and passes it to the server
- Once the data is sent to the server the client program is blocked until it receives a response back from the server.
- The whole process of serializing and DE serializing of the data are performed on both the sides of the program i.e client side and server side.
- Once the response is received from the server it is displayed to the client and then the client can request another procedure.
- All these operations seem to be as close to local procedure calls.

Domain Model :




as seen in the above diagram, the Model of the application is the Server, The controller for the application is MPlaceImplementation and the view of the application is Client.

Sample Runs : I am attaching the sample run screenshots below




This is the screenshot for the server, this is run on the 10.234.136.57 machine, please note my server has to be run on this machine, otherwise the client wont connect to the server.


This is the client view, where it just echoes a message saying it is connected, The reason I haven't implemented any login is because, it doesn't really have any significance as of now.

Conclusion: By doing this assignment, I have got a good understanding of what Java RMI does and the role of MVC in any realtime development.