**Assignment 1**

The main objective of assignment is to implement marketplace with a MVC (Model View Controller) architecture through Remote Method Invocation (RMI) and to translate customer requirements into a Domain Model.

**Project Requirements**:

* **Registration and Login**: Only customer can register and customer and administrator will have different views. Upon login, controller decides which view is to be called. Customer and administrator view should be separated out.
* **Browsing Items**: this would be common for both customer and administrator as both of them have full access to browse items.
* **Updating Items**: can only be viewed in administrator view and only administrator has access to it.
* **Removing Items**: can only be viewed in administrator view and only administrator has access to it.
* **Purchasing Items**: can only be viewed in customer view and only customers has access to it.

**Remote Method Invocation**:

RMI provides interaction between 2 java virtual machines. We can remote invoke methods of those objects which are on another machine. We need to write an interface which declares all the methods which we want our client to remotely invoke. In this assignment “MarketplaceController.java” is the interface which has all the abstract methods that client can invoke when required. We need a class “MarketplaceControllerImpl.java” which implements this interface.

In our server we have create instance of this class and register (bind) the service with RMI registry.

*MarketplaceController controller =* ***new*** *MarketplaceControllerImpl();*

*Naming.rebind("//tesla.cs.iupui.edu/MarketPlace", controller);*

Our client should call the registry to obtain reference to the remote object. Client will receive reference to the interface.

MarketplaceController controller= (MarketplaceController) Naming.*lookup*("//tesla.cs.iupui.edu/MarketPlace");

**Domain Model**:

A domain model illustrates meaningful conceptual classes in problem domain. The following diagram illustrates the domain model for this assignment. It lists all the conceptual classes that will be used to build the framework for this assignment.

Domain Model

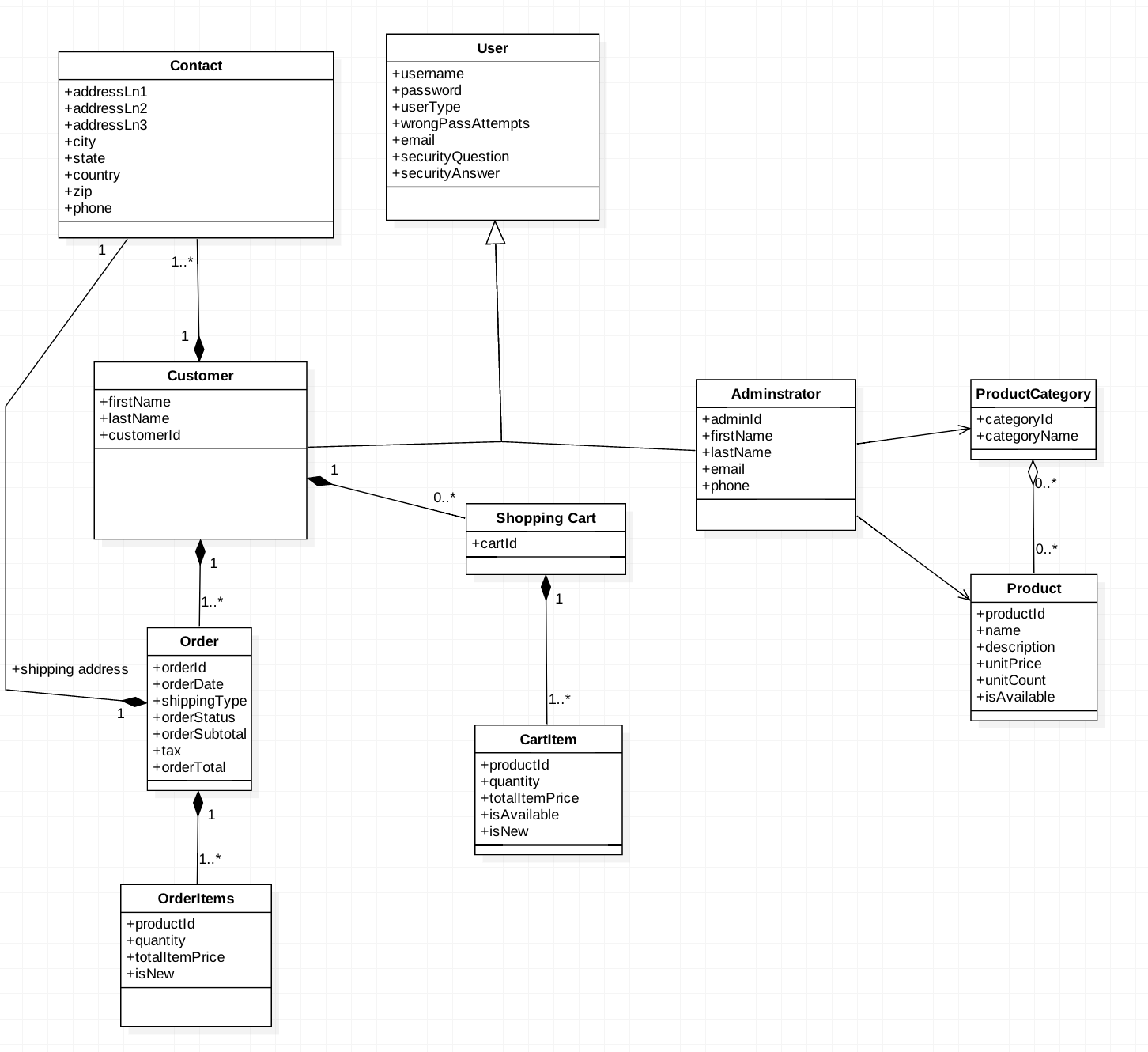


Fig 1: Domain Model

Model View Controller (MVC)

In the assignment Model and Controller sits on Server and view on Client. The Client and controller is separated out since code reusability is really high this way. We can change the view without worrying about the code part on the controller.

Server Side:  
Model – Account.java, Administrator.java, Customer.java, Item.java, Order.java, Product.java, ProductCategory.java, ShoppingCart.java

Controller- MarketplaceControllerImpl.java, MarketplaceServer.java

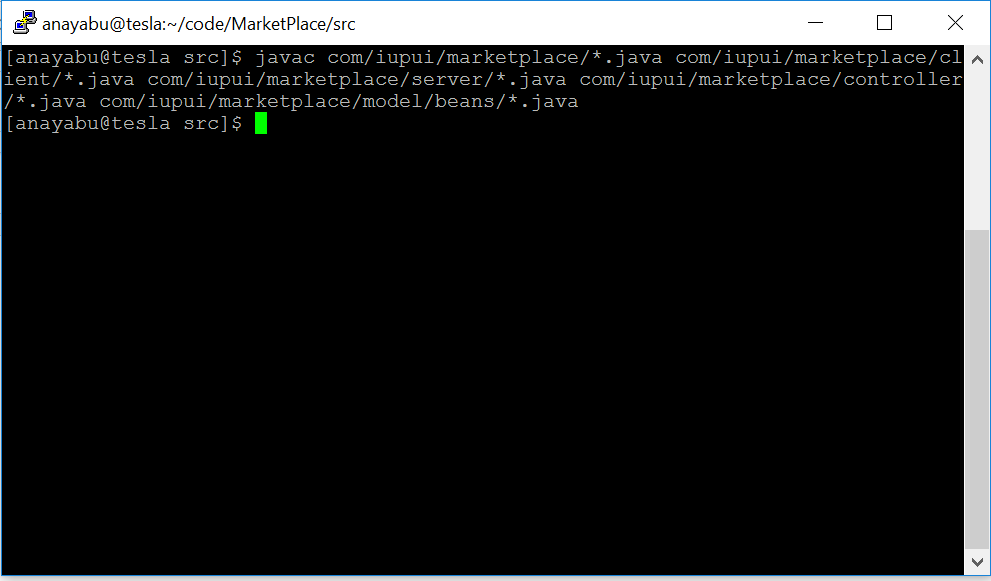
Client Side:

View: MarketplaceClient.java

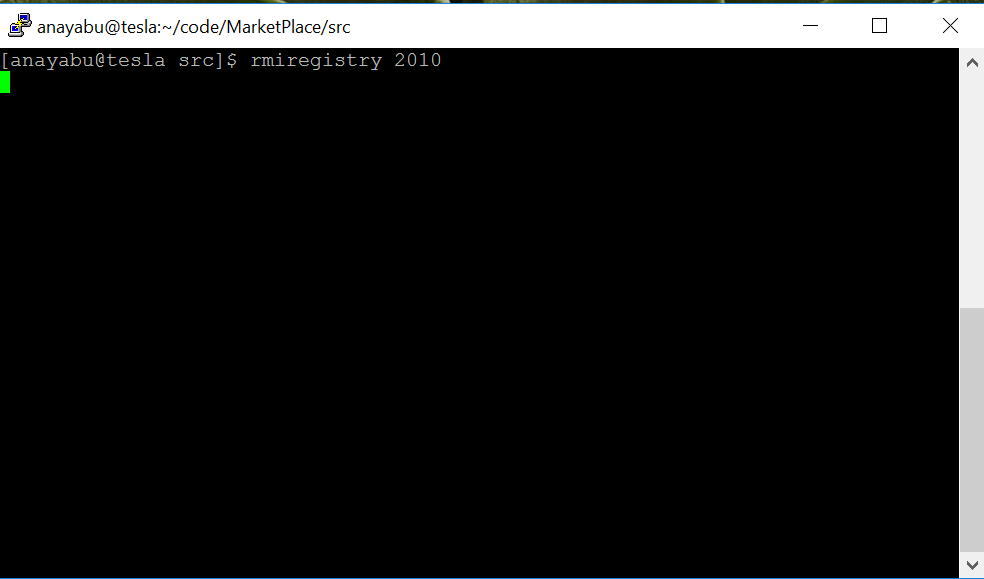
Interface:   
MarketplaceController.java

Sample runs:

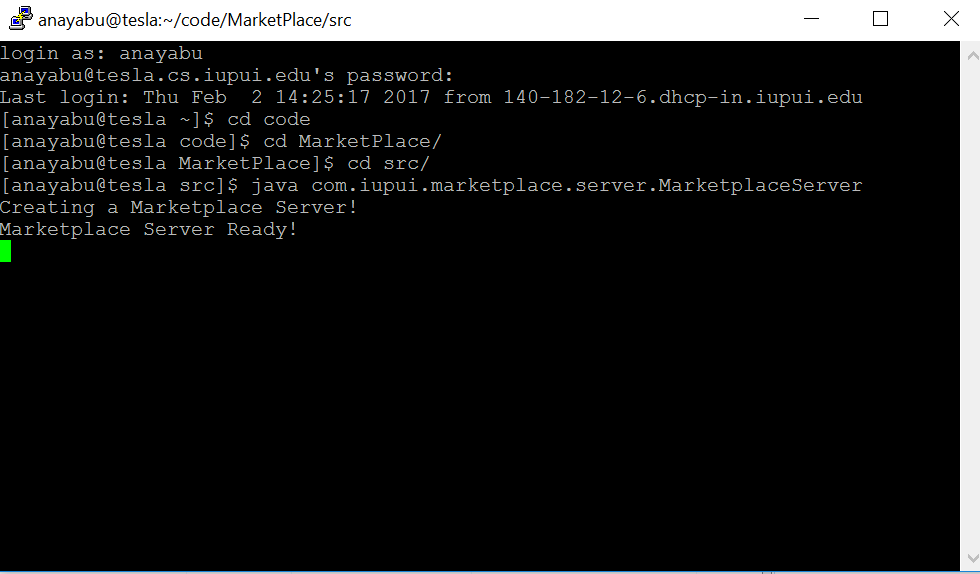
Compilation:

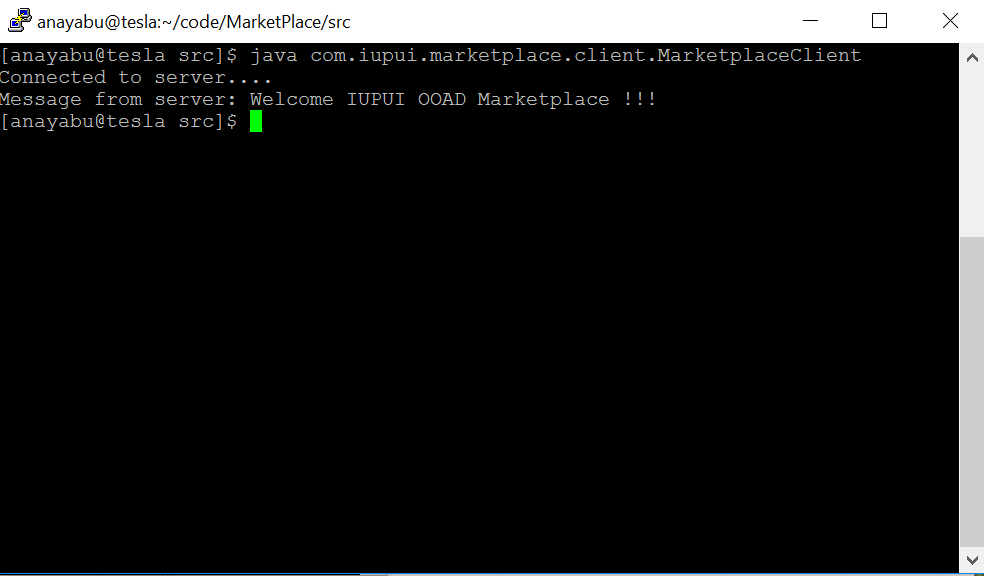


RMI :



Server:



Client:  


Summary:  
