
ONLINE MARKETPLACE APPLICATION

ASSIGNMENT-2

PROJECT REPORT

ABSTRACT FACTORY PATTERN, FRONT CONTROLLER, COMMAND

UNDER GUIDANCE OF

Dr. RYAN RYBARCZYK

BY

ADITHYA MORAMPUDI

Table of Contents

ASSIGNMENT 2 REPORT	3
Assignment#1 Feedback	3
Introduction	3
Application Flow	3
Software Design	4
Front Controller Pattern	4
Abstract Factory Pattern.....	4
Command Pattern	4
Class Diagram	5
Sample Runs	5
Conclusion	6
References	6

ASSIGNMENT 2 REPORT

Assignment#1 Feedback

- In Assignment#1, the first feedback was that my application wasn't running properly, which was rectified when the TA ran the program again
- Implementation of the MVC pattern, which I have done clearly this time by separating the Controller and View and Controller and Model.
- Domain Model – I am attaching it this time.

Introduction

In this Assignment, we were asked to partially implement the login functionality in the previously implemented client - server application, which was implemented using Java RMI as the framework and MVC pattern. In this assignment, the login functionality is implemented using three patterns, namely Abstract Factory, Command Pattern, Front Controller Pattern. These are implemented on the client side of the application in my design.

Application Flow

Entry point to the application is a class called Entry, which is displayed to the User at the Initial point, this is called by the Client Controller.

I have implemented three design patterns in this assignment, namely Abstract Factory, Command and Front Controller pattern, Front controller pattern is used here in order to make the authentication of the application and then display the required view to the user, once the Front controller gets the login values from the user they are then passed to the Client_Controller and then from Client_Controller, the values are passed to the Server_controller and from the Server_Controller, the values are passed to the model for verification, if the values are correct, then this model returns true to the Server Controller and then from there, the controller on the server side passes these values remotely to the Client side Controller, now the authentication is successful, now is the time to display the suitable view to the user, this displaying is done using the Dispatcher class of the Front Controller. This Dispatcher class passes the details to the Abstract Factory pattern using the Factory Producer class, This Factory Producer class creates the object of Concrete classes based upon the specific request from the dispatcher. I have implemented Command pattern for the Admin view only at this time, Admin View has options to Browse items, Update Items, Add Items and Remove Items. Once the Admin is logged into the system, he can then perform these tasks, The Admin view asks the user to enter a input based upon the shown output, upon user specific selection, the request is passed into the Dispatcher, then this Dispatcher sends the request to the Invoker of the Command pattern to execute the command, based upon the request type, specific concrete class is called and its execute method is executed. Once this method is executed, it calls the specific method to display the corresponding contents to the user, then the application exits.

Software Design

Front Controller Pattern

This is the place where Authentication and rendering of different views take place, The front controller pattern has a Front Controller and a Dispatcher, whenever the application has started, it should pass through the front controller, This is where the Authentication takes place, by which I mean, a remote call happens to check for the details from the model and then if the Model returns true, the front controller will call the dispatcher class, from where it renders the view of that particular user using the Abstract Factory pattern.

The classes present in my Front Controller pattern are FrontController.java, Dispatcher.java, and the interaction with the Model happens via the Client Controller from the Front Controller

Abstract Factory Pattern

As discussed above, after the login is successful, the dispatcher comes into action to render the required view by that particular user, for this to happen, I am creating a two Factories of Views, here the two View Factories I have are Admin Factory and User Factory, which will create the objects of the concrete users based on the request by the dispatcher. Here the Factory producer uses the Abstract Factory to generate the objects of Concrete Admin or Concrete User using the Admin Factory or User Factory.

The classes I have here are FactoryProducer.java which is contacted by the Dispatcher of the Front controller, The Factory producer uses the Abstract Factory, Admin Factory, User Factory, ConcreteAdmin, ConcreteUser

Command Pattern

After all the login is successful and user sees the respected view, This pattern comes into picture, once the user wants to browse any items, update items, Delete items these are taken as commands and then returned to the Dispatcher, based upon the request by the user, the Dispatcher calls the Invoker of the command pattern to execute the command requested by the user. The invoker calls the concrete classes of that particular method and then the view of this request gets displayed to the user.

The classes present in this command pattern are Command.java, invoker.java, Concrete Additems.java, Concrete UpdateItems.java, Concrete RemoveItems.java, Concrete BrowseItems.java, this also has a Error page, which gets displayed when an invalid request is being requested by the user.

Class Diagram

This is attached in a separate file named Class diagram.

Sample Runs

Screen Shot to show the running Instance of the Server, please note, the server should always be run on 10.234.136.57 only.

```
~ — ssh amorampu@10.234.136.57
[-bash-4.2$ sh makeServer.sh
Server Started
█
```

A running Instance of the client, which is being executing and sending requests and getting back the results

```
~ — ssh amorampu@10.234.136.57
[-bash-4.2$ sh makeClient.sh
welcome to RMI

  Enter 1 for Admin Login
  Enter 2 for Customer Login
Enter your choice:
1
Enter your userID:
[admin
Enter your password:
[12345
Welcome Admin

1.browse Items
2.Update items
3. remove items
4. add Items
Enter Choice
1
Browse Items will be updated next time
-bash-4.2$ █
```

I have only implemented a partial login this time, so to login into the application you need to use the hard coded values given in the ReadMe.MD file if not you won't be able to enter into the application.

Conclusion

By Doing this Assignment, I understood how to properly implement MVC architecture as this was kind of blurred in the first Assignment, in this Assignment, I had to brainstorm a lot to understand where exactly these patterns fall in the scope of our application, after implementing these patterns I understood where and how these can be applied in real world use cases.

References

- [1]. https://en.wikipedia.org/wiki/Front_controller
- [2]. https://www.tutorialspoint.com/design_pattern/front_controller_pattern.htm
- [3]. https://en.wikipedia.org/wiki/Abstract_factory_pattern
- [4]. https://www.tutorialspoint.com/design_pattern/abstract_factory_pattern.htm
- [5]. https://en.wikipedia.org/wiki/Command_pattern
- [6]. https://www.tutorialspoint.com/design_pattern/command_pattern.htm
- [7]. Class Slides and Materials
- [8]. Gang of Four Text