Fall 12-1-2009

E-Classifieds

Srikanth Adabala
Dakota State University

E-CLASSIFIEDS

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

December, 2009

By

Srikanth Adabala

Project Committee:

Dr. Ronghua Shan

Dr. Mark Moran

Dr. Stephen Krebsbach

DAKOTA STATE UNIVERSITY

PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Srikanth Adabala

Master's Project Title: E-Classifieds

Faculty supervisor: Ronghua Shan Date: 12/17/2009

Committee member: Mark Moran Date: 12/17/09

Committee member: Date: 12/17/09

ACKNOWLEDGMENT

I would like to thank my committee members Dr. Ronghua Shan, Dr. Mark Moran and Dr. Stephen Krebsbach for giving me an opportunity to work on this project. I thank the teaching and non-teaching staff for their support and encouragement in successful completion of master's program.

I would like to express my deepest appreciation to the dean of graduate studies Dr. Omar El-Gayar, who introduced me to literature review when I was working under him as a graduate assistant and his valuable suggestions helped me in successful completion of my project.

I thank First Bank and Trust for giving me an opportunity to do an internship and use the various technologies at the bank in shaping my project.

Finally, I would like to thank my family members and friends for supporting me financially and emotionally since the commencement of my masters program.

ABSTRACT

E-classifieds is a new website developed by Business Solutions which explains the software development process of an E-Classifieds system. E-classifieds gives an opportunity to the customers to buy and sell products online. E-classifieds addresses different phases of software development process which includes use cases, class diagrams which helps in understanding the objects used in the development of E-classified system. E-classifieds replaces the conventional ways of buying and selling products. It helps consumers a virtual shopping experience using internet and allows the customers to buy and sell their goods from home.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

____

Srikanth Adabala

TABLE OF CONTENTS

Contents

E-CLASSIFIEDS .... **I**

PROJECT APPROVAL FORM .... **II**

ACKNOWLEDGMENT .... **III**

ABSTRACT .... **IV**

DECLARATION .... **V**

TABLE OF CONTENTS .... **VI**

LIST OF FIGURES .... **X**

CHAPTER 1 .... **1**

INTRODUCTION **1**

CHAPTER 2 .... **2**

LITERATURE REVIEW **2**

2.1 J2EE 2

2.2 MS SQL SERVER 2000 2

CHAPTER 3 .... **4**

PROJECT PROPOSAL **4**

3.1 PROJECT DESCRIPTION 4

3.1.1 Software Overview *4*

3.1.2 Software Justification *5*

3.1.3 Targeted Customers *5*

3.2 PROJECT TEAM 5

CHAPTER 4 .... **6**

DOMAIN ANALYSIS **6**

4.1 INTRODUCTION 6

4.2 GLOSSARY 6

4.3 GENERAL KNOWLEDGE ABOUT THE DOMAIN 8

4.4 CUSTOMERS AND USERS 8

4.5 ENVIRONMENT 8
4.6 TASKS AND PROCEDURES CURRENTLY PERFORMED 9
4.7 COMPETING SOFTWARE 9
4.8 SIMILARITIES TO OTHER DOMAIN 10
4.9 REFERENCES 10
CHAPTER 5 11
REQUIREMENTS ANALYSIS 11
5.1 INTRODUCTION 11
5.1.1 Purpose of the application 11
5.1.2 Scope of the application 11
5.2 FUNCTIONAL REQUIREMENTS 11
5.2.1 User Registration 11
5.2.2 Login/Logout 12
5.2.3 My Profile 12
5.2.4 Product Search 12
5.2.5 Upload a Image 12
5.2.6 Category 12
5.2.7 Service Fees 12
5.3 NON-FUNCTIONAL REQUIREMENTS 13
5.3.1 Efficiency 13
5.3.2 Reliability 13
5.3.3 Security 13
5.3.4 Response Time 13
5.3.5 Documentation and Training 13
5.4 TARGET ENVIRONMENT 13
5.5 GLOSSARY 14
5.6 CHANGE CONTROL 14
5.7 TERMS AND CONDITIONS 15
5.8 WARRANTY PROCEDURES 17
CHAPTER 6 19
PROJECT IMPLEMENTATION 19
6.1 IMPLEMENTATION DETAILS 19
6.1.1 Architecture 19
6.1.2 Class Diagrams 20
CHAPTER 7 23

USE CASE ANALYSIS 23

7.1 UC_01: VIEW SALES STATUS 23

1.0 Description 23

2.0 Interacting Actors 23

3.0 Related Use Cases 23

4.0 Triggering Event(s) 23

5.0 Preconditions/Assumptions 23

5.1 Main Flow of Events 23

5.1.1 Main Flow Description – View Status of Sales 23

6.0 Post conditions 24

6.1 Notes 24

7.0 Business Rules 24

8.0 Comments 24

9.0 Design Notes 24

7.2 UC_02: EDIT POSTED PRODUCT 25

1.0 Description 25

2.0 Interacting Actors 25

3.0 Related Use Cases 25

4.0 Triggering Event(s) 25

5.0 Preconditions/Assumptions 25

5.1 Main Flow of Events 25

5.1.1 Main Flow Description – Update the products details 25

5.1.2 Alternate Flow Description 26

6.0 Post conditions 27

6.1 Notes 27

7.0 Business Rules 27

8.0 Comments 27

9.0 Design Notes 27

7.3 SEQUENCE DIAGRAMS 29

CHAPTER 8 33

CLIENT ACCEPTANCE TEST 33

8.1 INTRODUCTION 33

8.1.1 Document Purpose 33

8.1.2 Document Scope 33

8.2 Issues Resolved 33

8.3 Test Details – Acceptance Test 34

8.4 Client Sign-offs *35*

8.4.1 Unconditional signoff *35*

8.4.2 Conditional Signoff *35*

8.4.3 Rejection *36*

CHAPTER 9 37

CONCLUSION 37

REFERENCES 38

APPENDICES 39

APPENDIX A: SOURCE CODE 39

JAVA SOURCE CODE 39

XML CODE 55

DATABASE SCRIPTS 58

LIST OF FIGURES

Figure 1: Getting Registered on E-Classifieds 29

Figure 2: Log-On Scenario for E-Classifieds 30

Figure 3: UC-02(i) Edit Posted Product 31

Figure 4: UC-02(ii) Edit Posted Product 32

CHAPTER 1

INTRODUCTION

E-classifieds is an online application which gives an opportunity for buyers and sellers to buy and sell their goods from home replacing the conventional method of buying and selling. E-classifieds targets agents, industries and common people who wish to sell their goods for a desirable price. The purpose of analysis and development of E-classifieds is to let consumers to operate businesses remotely without actually having a physical place. E-classifieds uses MS SQL Server 2000 as the back end database and J2EE as front end to interact with the database.

JBOSS Application Server is a free open source Java EE based application server. It's basically cross platform software that can be used on any Java supported operating system.

The combination of the JBOSS suite which is Pure Java J2EE application servers and the Microsoft SQL Server RDBMS offers you a J2EE development and deployment environment on the Windows platform with outstanding superior performance, good return on investment (ROI) and substantially reduced time to market (RTTM) compared to other J2EE application suites and RDBMS solutions on the Windows platform. E-classifieds is built using these technologies there by providing a potential website for the consumers to buy and sell goods online replacing conventional methods of buying and selling goods.

CHAPTER 2

LITERATURE REVIEW

2.1 J2EE

J2EE stands for Java 2 Platform, Enterprise Edition. Java is a programming language developed by Sun Microsystems. The java platform is like a virtual Machine which acts like mini processors that translates computerized instructions into functions. The number 2 in J2EE stands for version 2. Sun Microsystems developed different editions. J2EE is the traditional name. There is Java ME where ME stands for Micro Edition which is used in mobile phones and PDA's. Java SE is the middle edition which runs on mobile devices, laptops and desktops. Java EE is the combination of features of Micro Edition and Standard Edition.

J2EE platform takes the advantage of "Write Once, Run Anywhere" feature. A J2EE application model provides a simplified approach to developing highly scalable and highly available internet or intranet based applications. Encapsulation of the functionality layers is another distinctive feature of J2E platform. An Enterprise JavaBeans (EJB) component encapsulates the enterprise business logic. Reusability is another feature of the J2EE components adds competitive advantage for IT organizations and enterprise developers.

2.2 MS SQL Server 2000

MS SQL server 2000 is a relational database system which offers a variety of administrative tools to ease the database development, maintenance and administration. MS SQL Server 2000 was developed by Microsoft which supports the database language called the structured Query Language. MS SQL Server 2000 has various features such as XML support, Scalability, Data Transformation Services (DTS), Enterprise Manager.

XML Support

XML support is the key to Microsoft's push towards Web-enable SQL Server 2000. The SQL Server 2000 supports T-SQL extensions that let SQL Server produce a result set as an XML document. In addition, you can use XML documents to directly insert and update values and SQL Server tables in the database.

Data Transformation Services (DTS)

DTS is a new feature which provides an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

Enterprise Manager

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

CHAPTER 3

PROJECT PROPOSAL

3.1 Project Description

3.1.1 Software Overview

This software satisfies the day-to-day need of the consumers to buy and sell goods from their own home at the best bargain price. A user has to register first to use the site. A user is either a seller or buyer in this site depending on how the user is using the site.

Common functionality

- Home page
- Registration
- Logout
- Login
- My profile

Seller functionality

Any user of E-classifieds who has either logged in or registered as a new user can act both as a seller and a buyer. A seller can put new products for sale in the E-classifieds site. The seller can monitor the status of the products put for sale on a continuous basis.

- Put for sale

Buyer functionality

- Search product
- Buy the Product

Solution to existing method of Classifieds:

This software replaces the conventional way of searching for the products. Both Seller and Buyer can do their work from home by connecting to E-classifieds through web.

3.1.2 Software Justification

In the existing market if someone wishes to either sell/buy a product , everything is done manually where the seller has to find a physical place to sell his/her goods in order to get the best bargain price and so is the case with the seller . This software provides day-to-day need of the consumers to buy and sell goods from their home at the best bargain price. A seller can put products for sale in the E-classifieds site. The Buyer who wants to buy the product can contact the seller from the information provided online. This application is very useful and quick solution for all the sellers and buyers.

3.1.3 Targeted Customers

This includes Agents, Industries and Common people (who wish to sell their products at the best price).

3.2 Project Team

Since this application would make a major impact in the market, Business Solution's best team would be working on it which includes Srikanth Adabala and Avinash Tiwari.

CHAPTER 4

DOMAIN ANALYSIS

4.1 Introduction

The central concept of the application E-classifieds is to allow the customers to shop virtually using the internet and allow customers to either buy or sell the items and articles of their desire from home.

A user must first create a profile in order to use the application and its features. Once a user has created his/her profile, he/she can perform various actions. A user can either buy or sell a product using his/her profile by logging on to the application.

The motivation to develop E- Classifieds is to:

- To change the conventional way of buying/selling goods.
- Both Seller and Buyer can do their work from home by connecting to E-classifieds through web.

4.2 Glossary

Home

The home page welcomes the user of E-classifieds to the site. It gives the provision for the registration for a new user and login for the existing user.

Login

Existing users can log on to the application by authenticating their user id and password.

Logout

The user's profile stored in the session object will be removed when the user logouts out of the e-classifieds site by clicking the 'logout' link. All the links of the links related to the buyer and seller functionality will become inaccessible.

Registration

New customers need to register first to use the functionality of the website by clicking the registration link on the home page, the user is directed to the user registration page. It captures all the user related information and on clicking the submit button a new user is created in the system

My Profile

A user who has either logged in to the site or registered himself as a new user can edit his/her profile by accessing the 'My Profile' page. The page displays all the attributes of the user in edit mode so that one or more of those attributes can be modified

Seller

A seller can put new products for sale in the e-classifieds site. The seller can monitor the status of the products put for sale on a continuous basis.

Buyer

The user who wants to buy a product which is for sale in the e-classifieds site can use the 'Search Product' link.

Product Search

The user who wants to buy a product put for sale in the e-classifieds site can use the 'Search Product' link. The system displays a list of categories under which the user can search for products. Once a category is chosen, the list of products put under that category will be listed.

Put for Sale

On clicking the 'Put for Sale' link, the user is directed to a page that captures all the information of a product the user is intending to sell through e-classifieds including its pictures.

For Sale

This is the initial state the product assumes soon after it is posted. It means that the sale has not started yet.

Sold out

When the product is sold out to the buyer (transaction is complete), the product gets this status.

Category

This field explains the category to which the product belongs which is kept for sale. The Categories are pre-populated in the database.

Price

The associated price of the product kept for sale.

Upload Image

It specifies the path to the image of the product which can be uploaded. Images are stored in the file system

4.3 General Knowledge about the Domain

- Statistics about the price of the product for sale.
- Categorization of the various products.
- The terms and conditions that user has to go through before being registered.
- Information of the user that can be displayed.
- Procedure involved in selling/buying a product
- The quality of a product.

4.4 Customers and Users

E-classifieds is being created to make the online shopping the best way of shopping and give it a whole new meaning.

The customer's to E-classifieds would be all the agents in the market who want to provide maximum customer satisfaction by allowing the users to search for what they are looking for. The users for E-classifieds would be all people who wish to either buy or sell their products online.

4.5 Environment

Hardware Requirements:

- Intel® Pentium® processor at 230 MHz or faster
- Minimum 256 MB of RAM

Software Requirements

- Mac OS 9.0.4v2 / 9.2.2, 10.1.2 / 10.3.4 / 10.3.5 / 10.4./ 10.4.2 /Microsoft Windows 2000 / Microsoft Windows XP Professional Version 2002, SP 2 (Recommended).
- Browser: Internet Explorer 6.0(Recommended) / Netscape 9.0 /Mozilla sunbird 0.5/ Mozilla Firefox 2.0.0.7

4.6 Tasks and Procedures Currently Performed

The application is meant to change the conventional way of buying and selling the products in the market.

The E-classifieds has the following functionalities which would change the conventional way of buying/selling the products:

Registration

- A user can get registered and start using the application
- Once the user is registered, he/she can start buying/selling the products.

User Profile

- A user once registered can create his/her own profile so that the people who wish to buy the product kept for sale by the user can be viewed

Put for sale

- A user can keep the product for sale along with its related price based on the category

Search the Product

- A user can search for the product once he/she logs on to the application based on the category

Upload Image

- A user can upload a picture related to the product as a result of which the buyer can have a real look at the actual product.

4.7 Competing Software

After an extensive market research done by Business Solutions, the following websites were found which would be of competition.

http://www.e-classifieds.net/

http://www.nytimes.com/ref/classifieds/

http://www.usfreeads.com/

The major differences in the websites listed above with the E-Classifieds application are as follows:

- All the classifieds listed charge the customer/user a monthly fee for using the application. The price ranges from \$3.99/month to \$49.99/month. E-Classifieds would be a free application that would be open for all the people who wish use it.
- Not all classifieds listed above cover all the categories that a user might wish to search his product for. Business Solutions came to conclusion after an extensive research through the above websites.
- None of the websites give an option to the customer to save the searched product to buy it later. Using E-classifieds, when a user wishes to buy the product he/she can contact the seller and let him/her know about his interest.

4.8 Similarities to Other Domain

- Classification of the products based on the categories is done among each and every field. Various fields present in the market can make use of the E-Classifieds by doing some minor changes in the various categories listed.

4.9 References

The market research was done by an extensive search done on the search engine www.google.com

CHAPTER 5

REQUIREMENTS ANALYSIS

5.1 Introduction

This software satisfies the day-to-day need of the consumers to buy and sell goods from their own home at the best bargain price. A user has to register first to use the site. A user is either a seller or buyer in this site depending on how the user is using the site.

5.1.1 Purpose of the application

In the existing applications the user should pay nominal fees in order to use the application. E-Classifieds has been built in order to change the way the shopping has been so far on the web. A user can either sell or buy a product of his choice based on the various categories listed.

5.1.2 Scope of the application

The application being developed would help the users of the E- classifieds to sell/ buy the goods based on the categories specified. This application is meant to save the time of users in shopping the goods.

5.2 Functional Requirements

The following are the functional requirements of E-classifieds application.

5.2.1 User Registration

- The user must register before using the application.
- During the registration, the application should be able to capture the user information.

5.2.2 Login/Logout

- The user should be able to Login using his own profile.

- Once logged in, the user should be able to log out.

5.2.3 My Profile

- Once the user is logged in the user should be able to change his profile by logging on to the application and selecting MyProfile tab.

5.2.4 Product Search

- Once logged in, the user can search for the product by typing in the name of the product based on the various categories already listed. A list of various categories is already present in the E-classifieds website.

- Once the user hits the enter button the various products should be listed based on the search name.

5.2.5 Upload a Image

- Once the user logs in using his profile, the user should be able to upload an image corresponding to the product along with its price and the necessary description.

5.2.6 Category

- After the user logs in he/she should be able to sell the product based on the various categories already listed. He can select a category from a list of categories already listed while placing the product in the E-classifieds website.

- After the user logs in to E- Classifieds website he/she should be able to search for the product based on the listed categories by typing in the name of the product in the search field.

5.2.7 Service Fees

- The user must agree to 'Terms and Conditions' (of the user) before getting registered to the states that the 2% of the total price that a user pays to purchase a product has to be paid by the seller. The amount should be payable to the Business Solutions.

5.3 Non-Functional Requirements

5.3.1 Efficiency

- The application being developed should have an efficiency of 98%.

5.3.2 Reliability

- The application should be able to back up at least 99% of the data in case of system crash down.

5.3.3 Security

- The application should be able to encrypt the sensitive business data from an authorized user.
- The application should make sure that the user information is secured.

5.3.4 Response Time

- The web page response time should be in between 4 to 7 sec when subjected to user load of 50 concurrent users over the period of 30 minutes on a 2GB RAM, 500 MHz Intel® Pentium® processor. Typically a user spends two minutes in between the pages.

5.3.5 Documentation and Training

After the completion of the project, Business Solution will provide all the required documents that are needed to operate, update, maintain and secure the system. But during the continuity of the project we will be in touch with the progress report either in soft copies or hard copies. And at the mean time we are always ready to get the feedbacks from the customer. This documentation will include the requirements documentation, the design specification, the test plan, and the acceptance test plan.

5.4 Target Environment

Hardware requirements for deploying the software

- Intel® Pentium® processor at 500 MHz or faster.
- Minimum 512 MB of RAM. 1GB of RAM is recommended.

Software requirements for deploying the software

- Microsoft Windows XP Professional Version 2002, Service Pack 2.
- SUN certified J2EE application server such as IBM WebSphere, BEA Weblogic or JBOSS.
- Relational Database like Oracle 8i/9i/10g or SQL server 2000.
- JDBC Thin drivers for connecting to these databases.

5.5 Glossary

Please refer Domain Analysis 3.0

5.6 Change Control

Business Solution has the flexibility options for updating or altering the information in between the process of project design. But there are certain constraints to overcome to the problems that may occur changing the information or the process in between. So the constraints that the Business follows are as below:

- Users can ask for changing the information anytime in Requirements documentation Process.
- In the above case, if the changes crosses the idea of previous process or if it is found that the process is completely new than that was planned before, than the new ideas are treated as new project and charged accordingly.
- The certain attributes and entities are always changed in RD time.
- Once the RD is completed and gets approved by the customer, the users are not allowed for any changes including attributes or entities of the RD. But we are flexible to change them, if that may not affect later on the process of designing and coding.

5.7 Terms and Conditions

ACCEPTANCE OF TERMS

The services that Business Solutions., provides to you are subject to the following Terms and Conditions ("Terms"). “Business Solutions” reserves the right to update the TERMS at any time without notice to you.

DESCRIPTION OF SERVICES

Through its network of Web properties, Business Solutions provides you with access to a variety of resources, including software tools, and database of the orders made. The Services, including any enhancements, new features, and/or the addition of any new Web properties, are subject to the TERMS.

PERSONAL AND NON-COMMERCIAL USE LIMITATION

Unless otherwise specified, the Services are for your personal and non-commercial use. You may not modify copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell any information, software, products or services obtained from the Services.

NOTICE SPECIFIC TO SOFTWARE PRODUCED.

Any reproduction or redistribution of the Software not in accordance with the License Agreement is expressly prohibited by law, and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible.

WITHOUT LIMITING THE FOREGOING, COPYING OR REPRODUCTION OF THE SOFTWARE TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THE SOFTWARE IS WARRANTED, IF AT ALL, ONLY ACCORDING TO THE TERMS OF THE LICENSE AGREEMENT. EXCEPT AS WARRANTED IN THE

LICENSE AGREEMENT, BUSINESS SOLUTIONS HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE SOFTWARE, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANT ABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT

RESTRICTED RIGHTS LEGEND

Any Software which has been produced from the Services for or on behalf of the United States of America, its agencies and/or instrumentalities ("U.S. Government"), is provided with Restricted Rights. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable.

MEMBER ACCOUNT, PASSWORD, AND SECURITY

If any of the Services requires you to open an account, you must complete the registration process by providing us with current, complete and accurate information as prompted by the applicable registration form. You also will choose a password and a user name. You are entirely responsible for maintaining the confidentiality of your password and account. Furthermore, you are entirely responsible for any and all activities that occur under your account. You agree to notify Business Solutions immediately of any unauthorized use of your account or any other breach of security. Business Solutions will not be liable for any loss that you may incur as a result of someone else using your password or account, either with or without your knowledge. However, you could be held liable for losses incurred by Business Solutions or another party due to someone else using your account or password. You may not use anyone else's account at any time, without the permission of the account holder.

NO UNLAWFUL OR PROHIBITED USE

As a condition of your use of the Services, you may not use the Services in any manner that could damage, disable, overburden, or impair anybody, or interfere with any other party's use and enjoyment of any Services. You may not obtain or attempt to obtain any materials or information through any means not intentionally made available through the Services.

UNSOLICITED IDEA SUBMISSION POLICY

BUSINESS SOLUTIONS OR ANY OF ITS EMPLOYEES DO NOT ACCEPT OR CONSIDER UNSOLICITED IDEAS, INCLUDING IDEAS FOR NEW ADVERTISING CAMPAIGNS, NEW PROMOTIONS, NEW PRODUCTS OR TECHNOLOGIES, PROCESSES, MATERIALS, MARKETING PLANS OR NEW PRODUCT NAMES. PLEASE DO NOT SEND ANY ORIGINAL CREATIVE ARTWORK, SAMPLES, DEMOS, OR OTHER WORKS. THE SOLE PURPOSE OF THIS POLICY IS TO AVOID POTENTIAL MISUNDERSTANDINGS OR DISPUTES WHEN BUSINESS SOLUTIONS PRODUCTS OR MARKETING STRATEGIES MIGHT SEEM SIMILAR TO IDEAS SUBMITTED TO BUSINESS SOLUTIONS. SO, PLEASE DO NOT SEND YOUR UNSOLICITED IDEAS TO BUSINESS SOLUTIONS OR ANYONE AT BUSINESS SOLUTIONS. IF, DESPITE OUR REQUEST THAT YOU NOT SEND US YOUR IDEAS AND MATERIALS, YOU STILL SEND THEM, PLEASE UNDERSTAND THAT BUSINESS SOLUTIONS MAKES NO ASSURANCES THAT YOUR IDEAS AND MATERIALS WILL BE TREATED AS CONFIDENTIAL OR PROPRIETARY.

5.8 Warranty Procedures

COVERAGE PROVIDED

Business Solutions will, for a period of one year from the date of final "walk through" inspection or date of first occupancy (whichever comes first), make repairs/changes which are necessary because of defects in Software supplied by the Manufacturer. We will fix the software, at no cost to you. Because of design changes and improvements, we may substitute few modules of substantially equal quality. This warranty covers you, as the first retailer.

WHAT IS NOT COVERED

- Damage such as that caused by misuse, abuse, alterations, lack of proper maintenance, improper repairs, or neglect in notifying us about necessary repairs or neglect in taking reasonable precautions (such as virus getting into the system due to improper use of the internet).
- Replacement of the complete software due to Natural Hazards irrespective of the time from the date of "Final Walk Through".

HOW TO OBTAIN SERVICE

Step 1: Business Solutions will arrange modification of modules covered by this warranty, which are identified on the Warranty Registration.

Step 2: Whenever you are facing a problem in managing or using the software "E-classifieds" please contact Business Solutions directly rather than the employees.

BUSINESS SOLUTIONS SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES such as tampering with the software provided by Business Solutions and loss of the Software due to carelessness.

CHAPTER 6

PROJECT IMPLEMENTATION

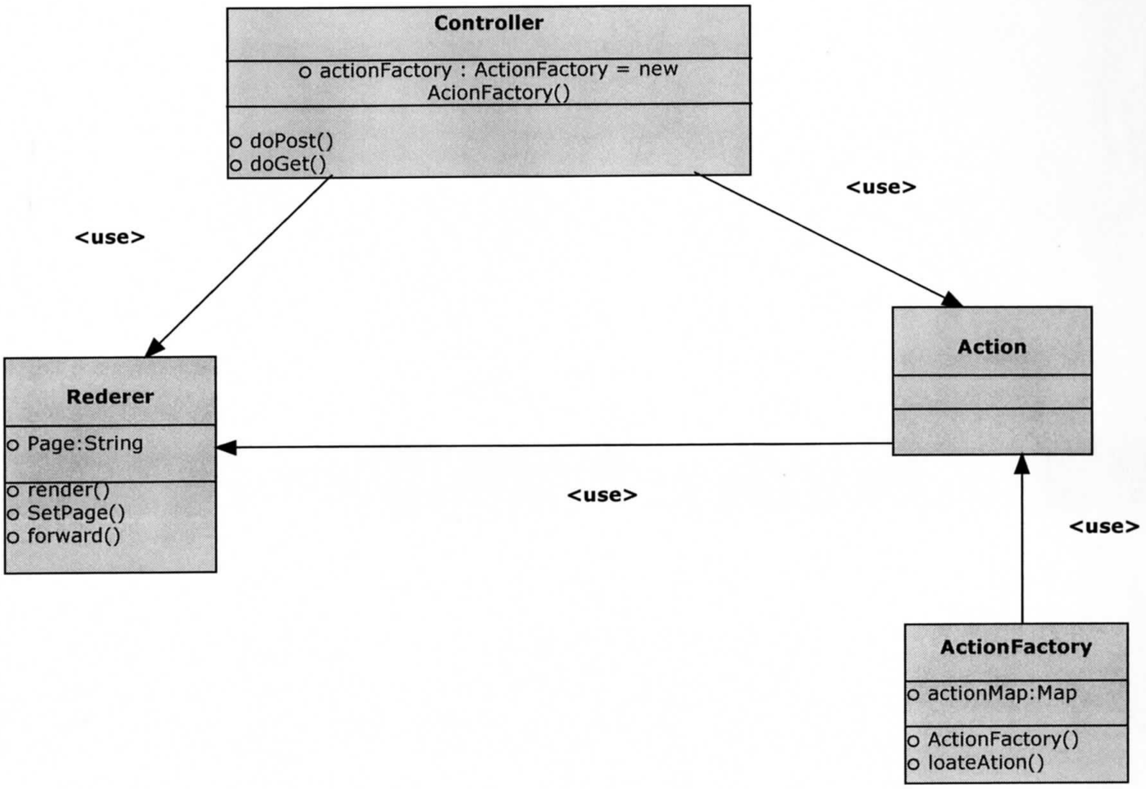
6.1 Implementation details

6.1.1 Architecture

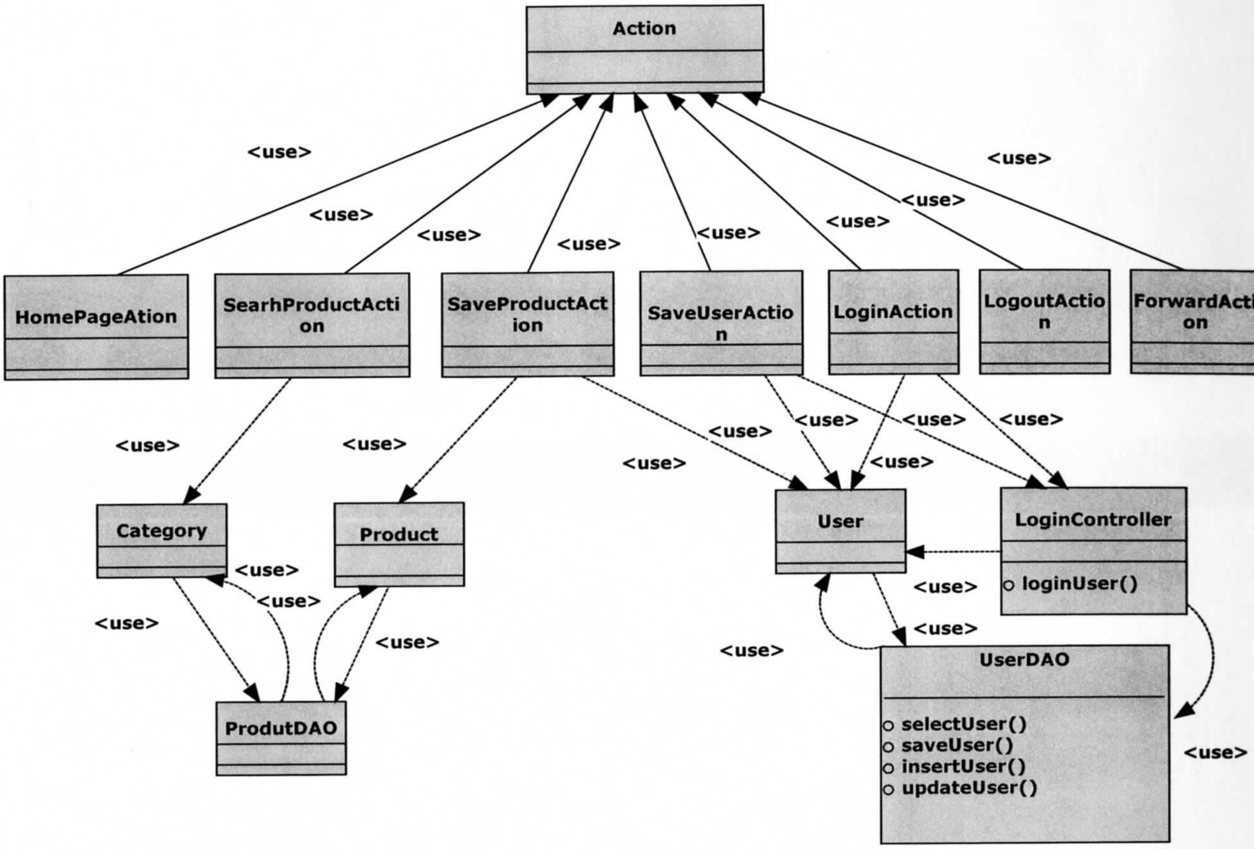
The model – view – control architecture is used to design the E-Classifieds site. The model part of it deals with the business functionalities and data manipulation. It is typically a set of business objects (java classes with methods performing business functions) and data access objects (java classes that have methods to perform database related functionalities). The controller part of the Architecture serves as an entry point for all the http requests from the site pages. The view part is concerned with the pages (served by the server if they are jsp pages) that are rendered by the browsers.

6.1.2 Class Diagrams

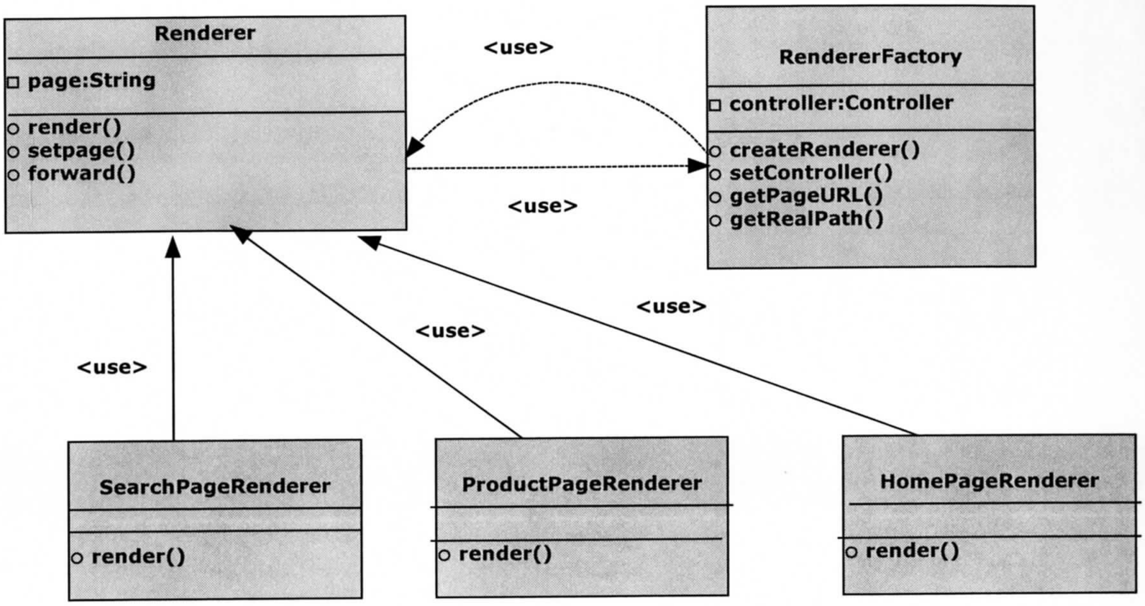
E-classifieds Control Layer



Action classes and Business Objects of E-classifieds



Renderer Objects of E-classifieds



CHAPTER 7

USE CASE ANALYSIS

7.1 UC_01: View Sales Status

1.0 Description

Describes the details of the user interaction with the system to view the status of all the products the user has placed to sell through E-classifieds.

2.0 Interacting Actors

- Seller

3.0 Related Use Cases

UC_01: Edit Posted Product

UC_03: View Product Details

4.0 Triggering Event(s)

None

5.0 Preconditions/Assumptions

- User must have been authenticated as a valid user of the system and must have been authorized as a seller to use this functionality sample usecase

5.1 Main Flow of Events

5.1.1 Main Flow Description – View Status of Sales

Step	Actor Action	System Response
1.	User selects the option to view the status of all products that were put	System displays the page where the name, description, the final price, Buyer

Step	Actor Action	System Response
	for sale	name, status and edit option of all the products the user had put for sale are listed. The products names appear as links such that the product details are displayed when the user clicks on it. The status of a product can be sold or settled in which case the buyer's name will not be applicable.

6.0 Post conditions

- None

6.1 Notes

- None

7.0 Business Rules

The products posted by the user are editable as long as it remains in the 'FOR_SALE' status.

8.0 Comments

- None

9.0 Design Notes

- The "View Sales Status" should appear as a link under the caption "seller".
- The details of all the products of the user should be displayed.
- The edit option should appear as a button across all editable product listings in a separate column.

7.2 UC_02: Edit Posted Product

1.0 Description

Describes the details of the user interaction with the system to edit the products details placed for sale in the E-classifieds site.

2.0 Interacting Actors

- Seller

3.0 Related Use Cases

- UC_01 View Sales Status

4.0 Triggering Event(s)

- None

5.0 Preconditions/Assumptions

- User must have been authenticated as a valid user of the system and must have been authorized as a seller to use this functionality.

5.1 Main Flow of Events

5.1.1 Main Flow Description – Update the products details

Step	Actor Action	System Response
1.	User chooses the edit option on an editable product in the list. (only editable products will have edit button)	System displays the page with form input controls to get the updated values name, short description, detailed description, category, Price and image for the selected product. The system pre-populates the existing values in the form input controls including the image.

Step	Actor Action	System Response
2.	User updates any/all of the seven fields displayed by the system and chooses the submit option.	The system validates whether at least a single field was modified on the client side using JavaScript. If not an error message is shown to the user. The system updates the details of the product edited and displays a success message.

5.1.2 Alternate Flow Description

Step	Actor Action	System Response
1.	User chooses the edit option on an editable product in the list. (only editable products will have edit button)	System displays the 'Edit Posted Product' page with form input controls to get the updated values name, short description, detailed description, category, Price and image for the selected product. The system pre-populates the existing values in the form input controls including the image.

Step	Actor Action	System Response
2.	User updates any/all/none of the seven fields displayed by the system and chooses the cancel option.	The system returns to the 'View Sales Status' page.

6.0 Post conditions

- None

6.1 Notes

- None

7.0 Business Rules

- None

8.0 Comments

- The products to be sold are represented in the system as a set of attributes namely the name, short description, detailed description, Category, Price, Product Image. All of these attributes are editable by the seller.

9.0 Design Notes

- The Edit option should appear as a button in across all the editable products when all the products posted by the user are listed in the 'View Sales Status' page.
- The 'Edit Posted Product' page should have **Submit** and **Cancel** options as buttons.
- The error message after the JavaScript validation should be a message box (generated at the client side) with an error message "Please edit a field or choose cancel". The success

message “The Product is successfully updated” should be displayed in the browser as a server side response.

7.3 Sequence Diagrams

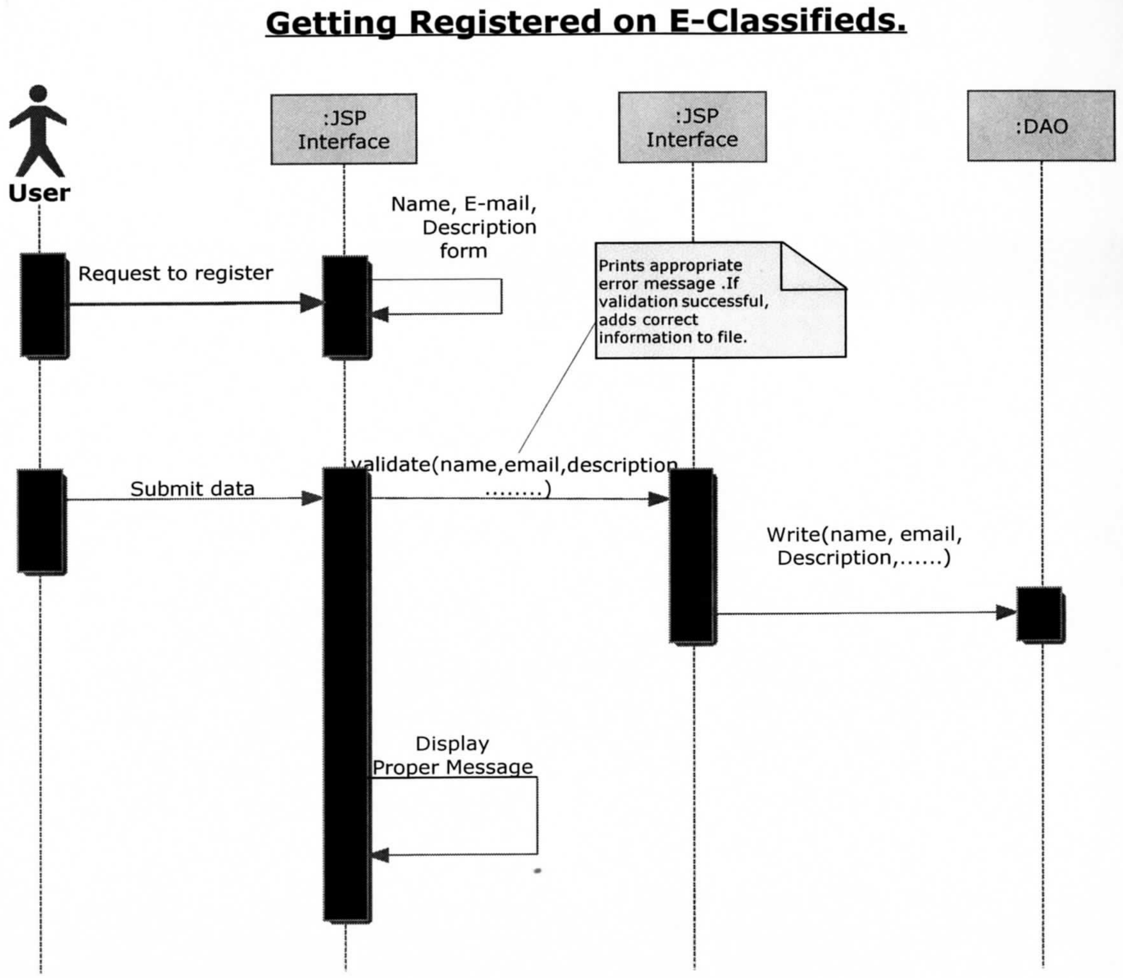

Getting Registered on E-Classifieds.
User
:JSP Interface
:JSP Interface
:DAO
Request to register
Name, E-mail, Description form
Prints appropriate error message .If validation successful, adds correct information to file.
Submit data
validate(name,email,description ....)
Write(name, email, Description,......)
Display Proper Message

Figure 1: Getting Registered on E-Classifieds

The above figure depicts the flow of logic when a user is trying to register on E-Classifieds website.

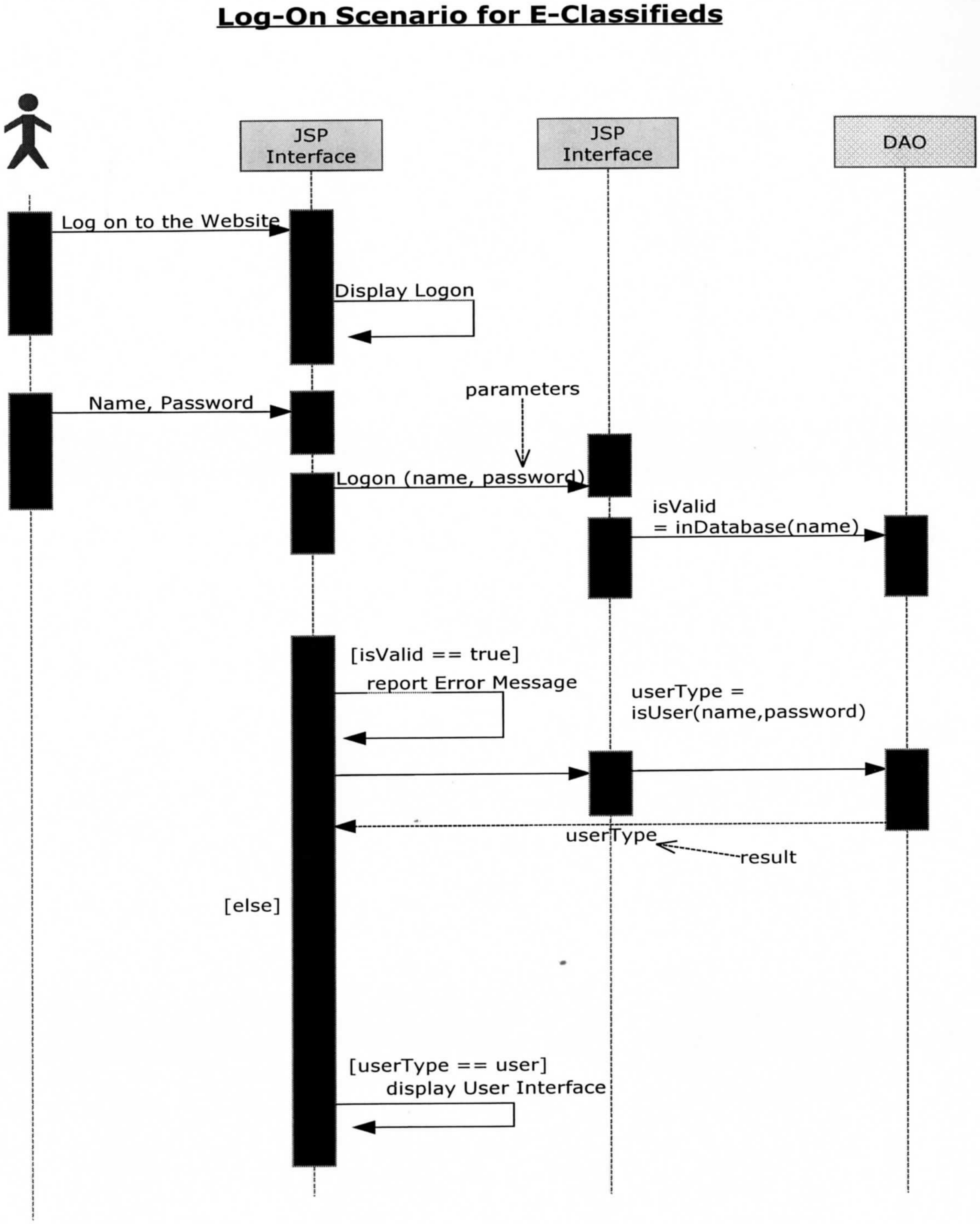


Figure 2: Log-On Scenario for E-Classifieds

The above figure depicts the flow of logic when a user tries to log on to the E-Classifieds website.

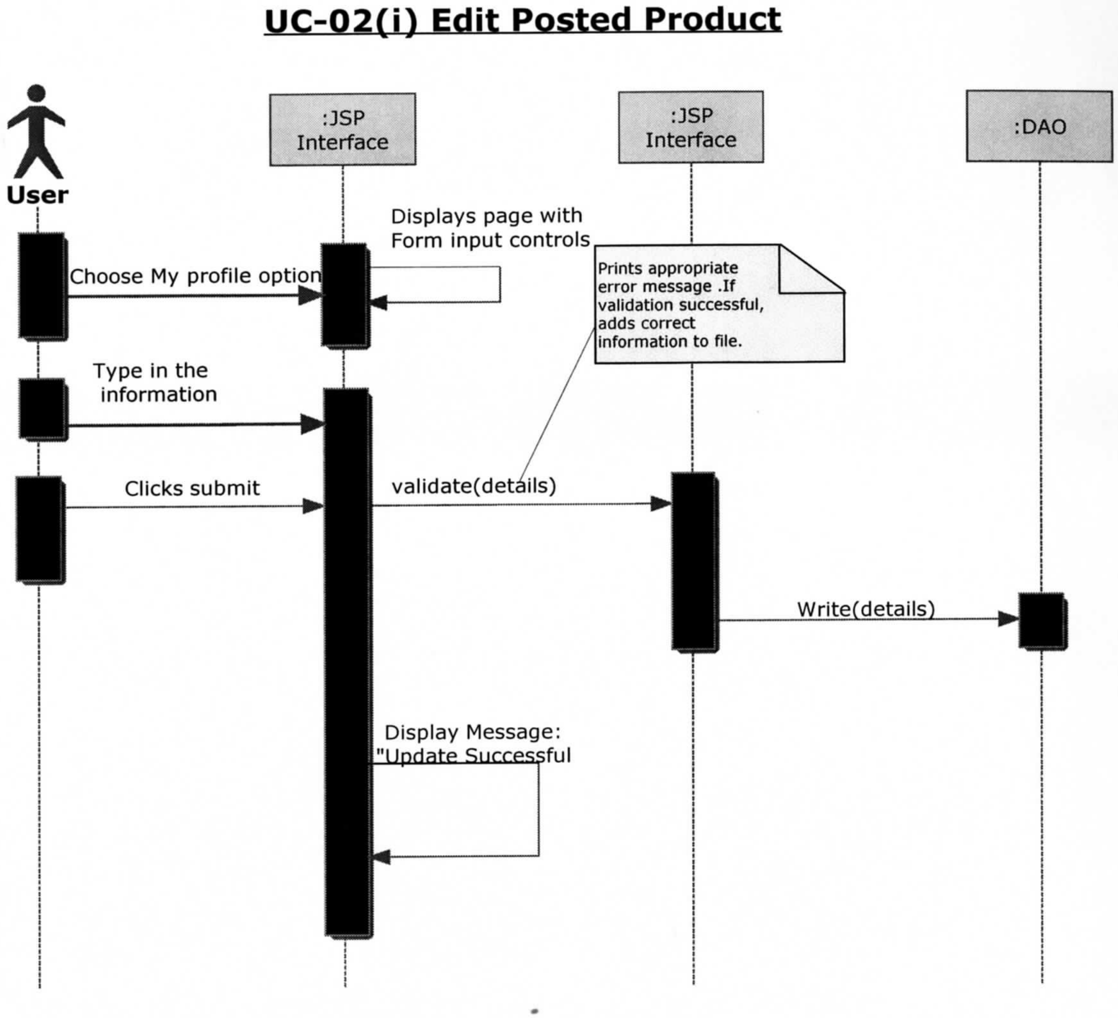

UC-02(i) Edit Posted Product
User
:JSP Interface
:JSP Interface
:DAO
Displays page with Form input controls
Choose My profile option
Prints appropriate error message .If validation successful, adds correct information to file.
Type in the information
Clicks submit
validate(details)
Write(details)
Display Message: "Update Successful

Figure 3: UC-02(i) Edit Posted Product

The figure above depicts the flow of logic when a user is updating the product description of the already posted product and at the end of it chooses to save the changes.

UC_02(ii) Edit Posted Product

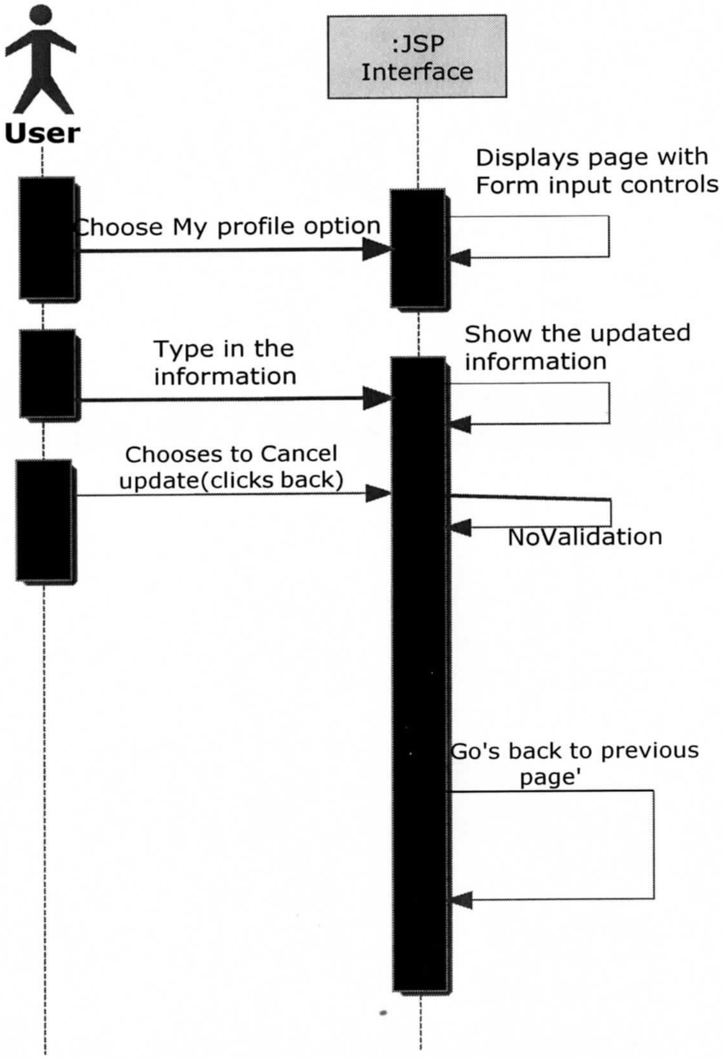

User
:JSP Interface
Displays page with Form input controls
Choose My profile option
Type in the information
Show the updated information
Chooses to Cancel update(clicks back)
NoValidation
Go's back to previous page'

Figure 4: UC-02(ii) Edit Posted Product

The above figure depicts the flow of logic when a user is updating the product description of the already posted product and at the end of it chooses not to save the changes by selecting cancel button.

CHAPTER 8

CLIENT ACCEPTANCE TEST

8.1 Introduction

8.1.1 Document Purpose

The purpose of this document is to find whether 'E-Classifieds' matches the user stories that have been stipulated by the client. Confirmation that these requirements have been fulfilled will occur through the Extreme Programming methodology of Acceptance Testing.

8.1.2 Document Scope

This document will state the client's requirements for 'E-Classifieds' and provide the client with all the User Stories that have been provided to date. The client will then be able to test the application against the information that the client has provided. This will ensure that all requirements stated in the User Stories by the client have been added to the system by the developers.

8.2 Issues Resolved

During the development of the E-Classifieds there were few issues. These issues are outlined below and the actions taken to solve the problems are also outlined. The client is required to look at each issue raised and then decide whether the issue has been corrected.

Issue to be Resolved	Seller details not associated with the product being sold		
Developer Notes	Associated the product with the user details		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

8.3 Test Details – Acceptance Test

The following is a series of user stories. These stories are the basis for the client acceptance test. The client is required to test each story and then sign off on the story or not.

User Story 1	User should be able to register in order to use E-Classifieds website		
Developer Notes	Completed		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

User Story 2	User should be able to Login and Logout of the E-classifieds website using his profile		
Developer Notes	Completed		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

User Story 3	User should be able to view / edit his profile once logged on to E-Classifieds website using MyProfile option		
Developer Notes	Completed		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

User Story 4	User should be able to search for a product which is put for sale by selecting category already listed		
Developer Notes	Completed		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

User Story 5	User should be able find the associated price of the product after perform a search		
Developer Notes	Completed		
Client Notes			
Client Acceptance	Agree	Disagree	Not Applicable

8.4 Client Sign-offs

8.4.1 Unconditional signoff

Upon completion of test, the Client is to indicate if he/she accepts that:

The product delivers that basic functionality as specified in the project definition

The product is fit for his/her use and is acceptable.

I, ________________________________, Client, hereby indicate by way of my signature below that the Simulator IT Project delivers the basic functionality described in the User Stories and is acceptable and fit for use.

Signed ______________________ Dated_____

8.4.2 Conditional Signoff

Should the Client decide that there is more work to be done, he/she is to indicate if he/she will accept the product in its current state with the required changes listed below –

I, ___________________________________, Client, hereby indicate by way of my signature below that the Simulator IT Project is conditionally accepted on the provision that changes are made to the product as outlined in the Test Details – Acceptance Test section.

Signed ________________________ Dated_____

8.4.3 Rejection

Should the client decide that the product is unacceptable and requires significant re-work, he/she is to list the major areas of concern below:

I, __________________________, Client, hereby indicate by way of my signature below that the Simulator IT Project is unacceptable and requires the concerns listed above in the Test Details – Acceptance Test section to be addressed and a follow-up user acceptance test to be arranged for no later than _______

Signed ________________________ Dated_____

CHAPTER 9

CONCLUSION

As per the requirements elicitation and the requirements specified in the requirements analysis phase of the Software Development Life Cycle, E-Classifieds has been successfully developed and implemented addressing all the functionalities specified. The developed system replaces the traditional in-store buying and selling of the products with web based application where, sellers can display their products at their desired price and buyers can buy the desired product. This project is developed within the time lines as expected following the base line architecture. E-Classifieds will ease the buying and selling of products as it saves time involved in the establishment of a physical location for business and also saves costs involved in the establishment. This web based application is available 24 hours in a day exceptionally during any maintenance plans, upgrading the software and any enhancements proposed.

One of the major limitations which have been identified in this application is, it doesn't give the opportunity for the user to bid a price on the product. The user can only buy a product displayed but can't bid for a desired price. This has been considered to be a potential development enhancement for the application which will addressed in the next phase of software development. Another limitation for this application is upgrading the SQL Server 2000 database to SQL Server 2008 database as some of the linked servers doesn't support for distributed queries in the test environment. This has been addressed by running installation catalogue database scripts recommended by Microsoft on the SQL Server 2000 Server which basically drops and creates all the server objects there by updating the SQL Server Database. Future enhancements to the application would be adding more functionalities such **PriceBid** on the product and upgrading the SQL Server database.

REFERENCES

1. Joseph Valof, 1995, Licensing Software to the Government Under the Revised [1995] FAR/DFAR* Regulations, 1998-2006, http://www.nanosft.com/igc/govlicen.html
2. Sun Microsystems, Inc., 1994-2009, Java 2 Platform, Enterprise Edition (J2EE) Overview, Retrieved on 11/25/2008, http://java.sun.com/j2ee/appmodel.html
3. David White, 2003-2009, What is J2EE?, Retrieved on 11/25/2008, http://www.wisegeek.com/what-is-j2ee.html

APPENDICES

APPENDIX A: SOURCE CODE

Java Source Code

LoginAction:

```
package com.eclassifieds.core.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.eclassifieds.core.business.LoginController;
import com.eclassifieds.core.business.User;
import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;

/**
 * This Action class handles the user login
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class LoginAction extends Action{

        /**
         * Executes the action and returns the chosen renderer
         *
```

```
     * @param req Http Request
     * @param req Http Response
     * @return renderer that displays a home page
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
        throws Throwable {
        String userName = req.getParameter("userName");
        String password = req.getParameter("password");
        User user = LoginController.loginUser(userName, password);
        if (user != null) {
            HttpSession session = req.getSession(true);
            session.setAttribute("user", user);
        } else {
            req.setAttribute("loginFailed", Boolean.TRUE);
        }
        return createRenderer("home.renderer");
    }

}
```

LogoutAction:

```
package com.eclassifieds.core.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;

/**
 * This Action class performs the user logout
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class LogoutAction extends Action{

    /**
     * Removes the user bean from the session.
     *
     * @param req Http Request
     * @param req Http Response
     * @return renderer that displays a page after logout
     * @throws Throwable if exception occurs during logout
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
                throws Throwable {
        HttpSession session = req.getSession(true);
        session.removeAttribute("user");
        return createRenderer("home.renderer");
    }

}
```

SaveProductAction:

```
package com.eclassifieds.core.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.eclassifieds.core.business.Product;
import com.eclassifieds.core.business.User;
import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;
import com.eclassifieds.core.controller.RendererFactory;



/**
 * This Action class saves the product
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class SaveProductAction extends Action{

    /**
     * Executes the action and returns the chosen renderer
     *
     * @param req Http Request
     * @param req Http Response
     * @return renderer based on the outcome of saving a product
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
            throws Throwable {
        HttpSession session = req.getSession(true);

        User user = (User) session.getAttribute("user");

        Product product = new Product();
        product.setProductName(getString(req, "productName"));
        product.setShortDesc(getString(req, "shortDesc"));
```

```
                product.setLongDesc(getString(req, "longDesc"));
                product.setStartPrice(getInt(req, "startPrice"));
                product.setCurrentPrice(getInt(req, "startPrice"));
                product.setStatus("FOR_SALE");
                product.setNewImagePath(getString(req, "imageFile"));

                System.out.println("SaveProductAction::execute::imagePath = "
                        + product.getImagePath());

                product.setSellerId(user.getUserId());
                product.setImageStream(getStream(req, "imageFile"));
                product.setBasePath(RendererFactory.getRealPath("/images"));
                product.save();

                return createRenderer("home.renderer");
        }

}
```

SaveUserAction:

```
package com.eclassifieds.core.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.eclassifieds.core.business.User;
import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;

import com.eclassifieds.core.business.LoginController;
```

```
/**
 * This Action class saves the user.
 * Used in registration flow and save my profile flow
 *
 * @see LoginAction
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class SaveUserAction extends Action{

    /**
     * Executes the action and returns the chosen renderer
     *
     * @param req Http Request
     * @param req Http Response
     * @return renderer chosen based on the result of saving the user
     * @throws Throwable if exception occurs during user save
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
                throws Throwable {
        HttpSession session = req.getSession(true);

        User user = (User) session.getAttribute("user");
        if (user == null) {
            user = new User();

        }
        user.setUserName(getString(req, "userName"));
        user.setFirstName(getString(req, "firstName"));
        user.setLastName(getString(req, "lastName"));
        user.setPassword(getString(req, "password"));
        user.setStreet(getString(req, "street"));
        user.setCity(getString(req, "city"));
        user.setState(getString(req, "state"));
        user.setPin(getInt(req, "pin"));
        user.setEmail(getString(req, "email"));
        user.setPhone(getString(req, "phone"));
        user.setPaypalAccount(getString(req, "paypalAccount"));
        user.save();
        User savedUser = LoginController
            .loginUser(user.getUserName(), user.getPassword());
        if (savedUser != null) {
            session.setAttribute("user", savedUser);
        }

        return createRenderer("home.renderer");
    }
}
```

SearchProductAction:

```
package com.eclassifieds.core.action;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.eclassifieds.core.business.Category;
import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;

/**
 * This Action class is used to search the product
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class SearchProductAction extends Action{

    /**
     * Executes the action and returns the chosen renderer
     *
     * @param req Http Request
     * @param req Http Response
     * @return renderer chosen based on the result of saving a product
     * @throws Throwable if exception occurs while saving the product
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
                throws Throwable {
        int categoryId = getInt(req, "categoryId");
        List productList = Category.getProducts(categoryId);
        req.setAttribute("productList", productList);
        return createRenderer("show_search.renderer");
    }
}
```

ForwardAction:

```
package com.eclassifieds.core.action;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.eclassifieds.core.controller.Action;
import com.eclassifieds.core.controller.Renderer;

/**
 * This Action class defines the method to return an
 * appropriate renderer for a given action
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class ForwardAction extends Action{

    /**
     * Executes the action and returns the chosen renderer
     *
     * @param req Http Request
     * @param req Http Response
     * @throws throwable when exception occurs or is propagated
     */
    public Renderer execute(HttpServletRequest req, HttpServletResponse res)
            throws Throwable {
        String actionName = (String) req.getAttribute("actionName");
        String rendererName = actionName.replaceAll(".action", ".renderer");
        System.out.println("ForwardAction::execute::rendererName = "
                + rendererName);
        return createRenderer(rendererName);
    }
}
```

User:

```
package com.eclassifieds.core.business;

import com.eclassifieds.core.data.UserDAO;

/**
 * This class defines the user object
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class User {

        /**
         * The user id
         */
         private int userId;

        /**
         * The user name
         */
         private String userName = "";

        /**
         * The first name
         */
         private String firstName = "";

        /**
         * The last name
         */
         private String lastName = "";

        /**
         * The login password
         */
         private String password = "";

        /**
         * The street address
         */
```

```
private String street = "";

/**
 * The city
 */
private String city = "";

/**
 * The state
 */
private String state = "";

/**
 * The pin code
 */
private int pin;

/**
 * The email address
 */
private String email = "";

/**
 * The phone number
 */
private String phone = "";

/**
 * The pay pal account
 */
private String paypalAccount = "";

/**
 * Gets the userId
 *
 * @return userId
 */
public int getUserId() {
        return userId;
}

/**
 * Sets the userId
 *
 * @param userId
 */
```

```
public void setUserId(int userId) {
        this.userId = userId;
}

/**
 * Gets the userName
 *
 * @return userName
 */
public String getUserName() {
        return userName;
}

/**
 * Sets the userName
 *
 * @param userName
 */
public void setUserName(String userName) {
        this.userName = userName;
}

/**
 * Gets the firstName
 *
 * @return firstName
 */
public String getFirstName() {
        return firstName;
}

/**
 * Sets the firstName
 *
 * @param firstName
 */
public void setFirstName(String firstName) {
        this.firstName = firstName;
}

/**
 * Gets the lastName
 *
 * @return lastName
 */
public String getLastName() {
        return lastName;
}

/**
 * Sets the lastName
 *
 * @param lastName
 */
public void setLastName(String lastName) {
        this.lastName = lastName;
}

/**
 * Gets the password
 *
 * @return password
 */
public String getPassword() {
        return password;
}

/**
 * Sets the password
 *
 * @param password
 */
public void setPassword(String password) {
        this.password = password;
}

/**
 * Gets the street
 *
 * @return street
 */
public String getStreet() {
        return street;
}

/**
 * Sets the street
 *
 * @param street
 */
public void setStreet(String street) {
        this.street = street;
```

```
}

/**
 * Gets the city
 *
 * @return city
 */
public String getCity() {
        return city;
}

/**
 * Sets the city
 *
 * @param city
 */
public void setCity(String city) {
        this.city = city;
}

/**
 * Gets the state
 *
 * @return state
 */
public String getState() {
        return state;
}

/**
 * Sets the state
 *
 * @param state
 */
public void setState(String state) {
        this.state = state;
}

/**
 * Gets the pin
 *
 * @return pin
 */
public int getPin() {
        return pin;
}
```

```
/**
 * Sets the pin
 *
 * @param pin
 */
public void setPin(int pin) {
        this.pin = pin;
}

/**
 * Gets the email
 *
 * @return email
 */
public String getEmail() {
        return email;
}

/**
 * Sets the email
 *
 * @param email
 */
public void setEmail(String email) {
        this.email = email;
}

/**
 * Gets the phone
 *
 * @return phone
 */
public String getPhone() {
        return phone;
}

/**
 * Sets the phone
 *
 * @param phone
 */
public void setPhone(String phone) {
        this.phone = phone;
}
```

```
    /**
     * Gets the paypalAccount
     *
     * @return paypalAccount
     */
    public String getPaypalAccount() {
        return paypalAccount;
    }

    /**
     * Sets the paypalAccount
     *
     * @param paypalAccount
     */
    public void setPaypalAccount(String paypalAccount) {
        this.paypalAccount = paypalAccount;
    }

    /**
     * Saves this User object
     *
     * @throws Throwable if exception occurs while saving the user
     */
    public void save() throws Throwable {
        System.out.println("User::save::begin");
        if (userId == 0) {
            UserDAO.insertUser(this);

        } else {
            UserDAO.updateUser(this);
        }
        System.out.println("User::save::end");
    }

}
```

LoginController:

```
package com.eclassifieds.core.business;

import com.eclassifieds.core.data.UserDAO;

/**
 * This class defines the business object that
 * takes care of Login
 *
 * @author Business Solutions
 * @contact Business Solutions
 * @version 1.0
 */
public class LoginController {

        /**
         * Selects the user object with the given credentials and
         * returns null if no such user is present
         *
         * @param userName The unique name of the user
         * @param password The password of the user
         * @return The User object with the user details
         * @throws Throwable if exception occurs while retrieving the user
         */
        public static User loginUser(String userName, String password)
                        throws Throwable {
                return UserDAO.selectUser(userName, password);
        }


}
```

XML code

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>

  <!DOCTYPE web-app (View Source for full doctype...)>

- <web-app>

  <display-name>e-classifieds</display-name>

  <description />

- <servlet>

  <servlet-name>Controller</servlet-name>

  <servlet-class>com.eclassifieds.core.controller.Controller</servlet-class>

- <!--  Home page parameters

  -->

- <init-param>

  <param-name>home.action</param-name>

  <param-value>com.eclassifieds.core.action.HomePageAction</param-value>

  </init-param>
```

```
- <init-param>
  <param-name>home.renderer</param-name>
  <param-value>com.eclassifieds.core.renderer.HomePageRenderer</param-value>
  </init-param>
- <init-param>
  <param-name>home.page</param-name>
  <param-value>/home.jsp</param-value>
  </init-param>
- <!--  Login parameters
  -->
- <init-param>
  <param-name>login.action</param-name>
  <param-value>com.eclassifieds.core.action.LoginAction</param-value>
  </init-param>
- <!--  Logout parameters
  -->
- <init-param>
  <param-name>logout.action</param-name>
  <param-value>com.eclassifieds.core.action.LogoutAction</param-value>
  </init-param>
- <!--  Profile parameters
  -->
- <init-param>
  <param-name>save_user.action</param-name>
  <param-value>com.eclassifieds.core.action.SaveUserAction</param-value>
  </init-param>
- <!--  Product page parameters
  -->
- <init-param>
  <param-name>product.action</param-name>
  <param-value>com.eclassifieds.core.action.ForwardAction</param-value>
  </init-param>
- <init-param>
  <param-name>save_product.action</param-name>
  <param-value>com.eclassifieds.core.action.SaveProductAction</param-value>
  </init-param>
- <init-param>
  <param-name>product.renderer</param-name>
  <param-value>com.eclassifieds.core.renderer.ProductPageRenderer</param-value>
  </init-param>
- <init-param>
  <param-name>create_product.page</param-name>
  <param-value>/create_product.jsp</param-value>
  </init-param>
- <!--  Search page parameters
  -->
- <init-param>
  <param-name>show_search.action</param-name>
  <param-value>com.eclassifieds.core.action.ForwardAction</param-value>
  </init-param>
- <init-param>
  <param-name>search_product.action</param-name>
  <param-value>com.eclassifieds.core.action.SearchProductAction</param-value>
  </init-param>
- <init-param>
  <param-name>show_search.renderer</param-name>
  <param-value>com.eclassifieds.core.renderer.SearchPageRenderer</param-value>
  </init-param>
- <init-param>
  <param-name>search_product.page</param-name>
  <param-value>/search_product.jsp</param-value>
  </init-param>
  </servlet>
- <servlet-mapping>
  <servlet-name>Controller</servlet-name>
  <url-pattern>*.action</url-pattern>
  </servlet-mapping>
- <welcome-file-list>
  <welcome-file>home.jsp</welcome-file>
  </welcome-file-list>
  </web-app>
```

Database Scripts

```
CREATE TABLE CATEGORY (
        CATG_ID INTEGER,
        CATG_NAME VARCHAR(30)
);

INSERT INTO CATEGORY VALUES (1, 'Books');
INSERT INTO CATEGORY VALUES (2, 'Electronics');
INSERT INTO CATEGORY VALUES (1, 'Music');
INSERT INTO CATEGORY VALUES (1, 'Automobiles');
INSERT INTO CATEGORY VALUES (1, 'Cosmetics');
INSERT INTO CATEGORY VALUES (1, 'Garments');
INSERT INTO CATEGORY VALUES (1, 'Furnitures');
INSERT INTO CATEGORY VALUES (1, 'Sports');
INSERT INTO CATEGORY VALUES (1, 'Stationary');
INSERT INTO CATEGORY VALUES (1, 'Computers');


CREATE TABLE PRODUCT (
        PROD_ID INTEGER identity,
        CATG_ID INTEGER,
        PROD_NAME VARCHAR(30),
        SHORT_DESC VARCHAR(30),
        LONG_DESC VARCHAR(30),
        START_PRICE INTEGER,
        CURRENT_PRICE INTEGER,
        STATUS VARCHAR(30),
        IMG_PATH VARCHAR(60),
        SELLER_ID INTEGER,
        BUYER_ID INTEGER,
        FINAL_PRICE INTEGER
);

CREATE TABLE ECLASSIFIEDS_USER (
        ECLASSIFIEDS_USER_ID INTEGER,
        ECLASSIFIEDS_USER_NAME VARCHAR(30),
        FIRST_NAME VARCHAR(30),
        LAST_NAME VARCHAR(30),
        USER_PASSWORD VARCHAR(30),
        BUILDING_STREET VARCHAR(30),
        CITY VARCHAR(30),
        STATE VARCHAR(30),
        PIN INTEGER,
        PHONE VARCHAR(30),
        EMAIL VARCHAR(30),
        PAYPAL_ACOUNT VARCHAR(30)
);
```