Easy E-commerce using XML data source Without External database

Sell goods easier than before

Z. Mahesh Kumar
School of Computer Science & Engineering
VIT University
Vellore, India
mahesh.cse349@gmail.com

R. Manjula
School of Computer Science & Engineering
VIT University
Vellore, India
rmanjula@vit.ac.in

Abstract— **Usually customers buy the goods of their desire from the store using internet. Administrator sell the goods with the help of external databases like oracle, mysql etc at administrator side. But in our proposed system, end user is a departmental store where the application is hosted on the web and the administrator maintains the XML only as data source using which all transactions take place like external database. In this system, the application deployed at the client side, the details of the items are brought forward to the customer from the XML data source. Based on the selection query XML file is updated at the end of each transaction. Also the Small scale vendors must purchase a licensed version of database to maintain data. It is very costly and risky. Because of this reason we are looking to provide a quality service that helps vendor to maintain XML data source within the application. It uses XML pages for representing product details, customer details and transactional details.**

1. INTRODUCTION

1.1 Overview

The main aim of the application is providing service to client to do shopping using the internet without external database. Customers to buy the items and articles of their desire from the store where all data about these products maintained in a single XML file at administrator side. The end user of this system is a departmental store where the application is hosted on the web. The application uses XML page as data source. Once the authorized personnel feed the relevant data into the system, reports could be generated as per the security. The main objectives of this paper are explaining about identifying the requirements for web based online shopping system for purpose of consumer satisfaction.

1. Providing web based online shopping system without any external database purchasing.
2. Providing inbuilt data source maintenance which is useful for small scale vendors.

This inbuilt database is to provide flexibility in maintaining data source for small scale vendors, easy access, and easy interoperable with other organizations. A application main menu, which lists the available options to the user. A login screen where the user authenticates in order to get access to the rest of the system's functionality. A registration process is provided where new users enter their details before they can access the system using the above logon screen. A shopping cart provided by which customer adds items to the cart. Other than this editing functionality also provided to the customer to modify the choices that already selected. An ordering mechanism which creates purchasing of goods based on the user's selection and sends them to the appropriate supplier. An order completion mechanism which simulates supplier input and automatically updates pending orders.

2. OVERVIEW OF SYSTEM ARCHITECTURE

2.1 Web application

Web application is accessed via web over a network such as the intranet. Web applications are popular due to ubiquity of client. The ability to update and maintain web applications without distributing and installing software on potential thousands of client computers**.**

Based on the task category, the functional requirements from the customer view point are:

1. Browsing through site
2. Authentication process
3. Selection of items
4. Checking selected items
5. Adding items to the cart
6. Modifying added items
7. Buying items
8. Checking mail
9. Giving feedback to customers

Similarly the functional requirements from the administrator view point are:

1. Authentication process
2. Updating of product details
3. Checking Orders
4. Contact with suppliers
5. Send mail

Based on the above set of tasks, the visualization of functionalities and displaying the shopping items can be illustrated in the work flow model in figure1.

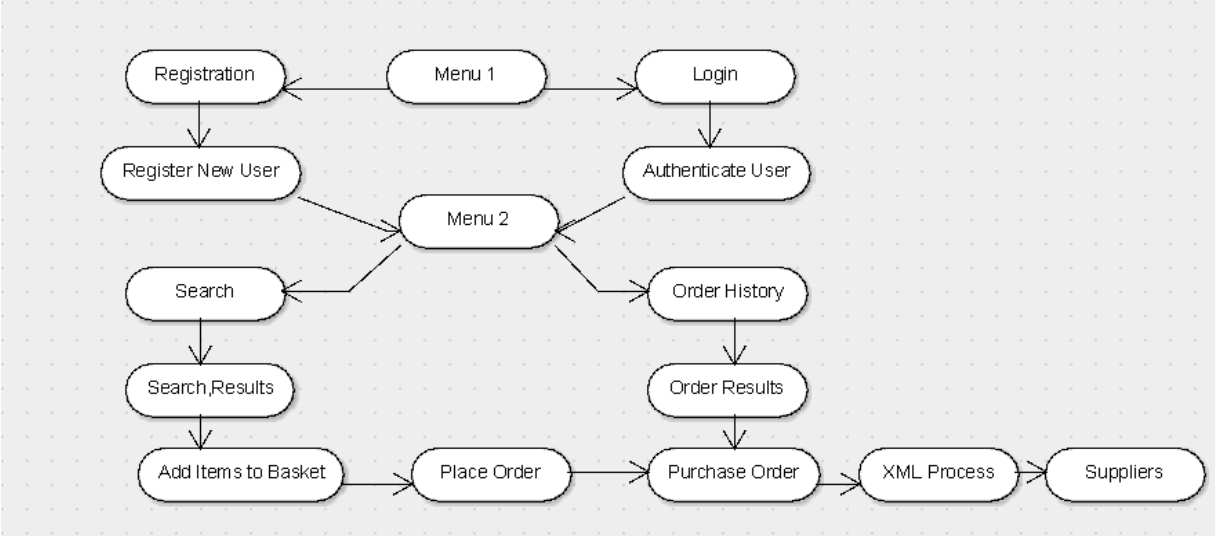

Figure 1: XML application Workflow model

User module performs the following steps.

1. Customer login
2. Choosing the products' catalogs to enter
3. After entering a product catalog, the customer goes through the products' list
4. Enter the product's page he prefers

5. Click on 'buy now' to buy this product
6. After order saved, he can go back to homepage to reselect products
7. Repeat steps 2-6
8. Click on 'take the bill' to check out his orders, and decide if he take them all, or delete something
9. Back to homepage, logout.

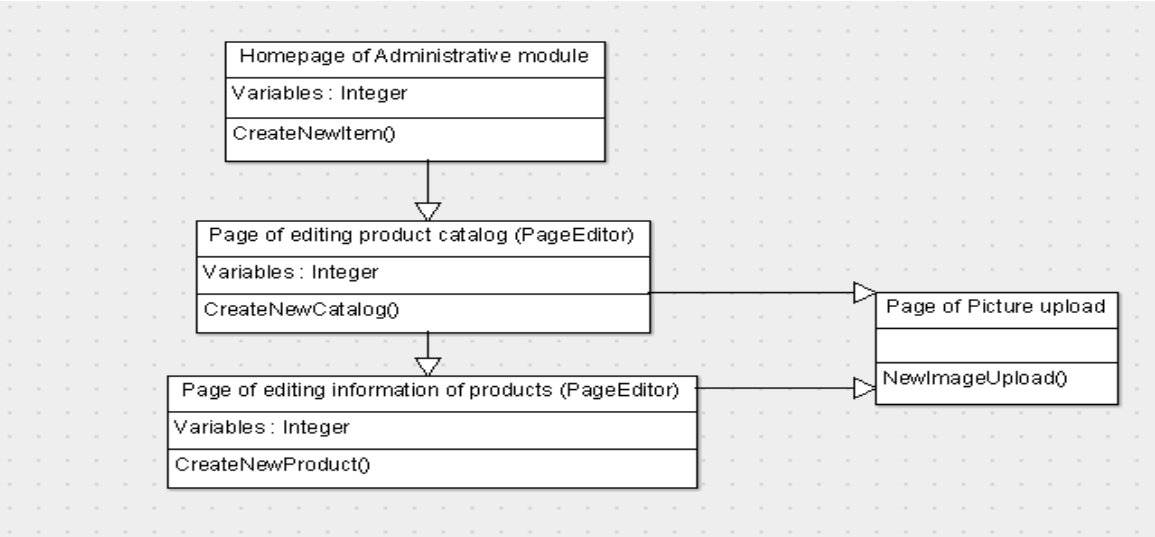

Figure 2 Prototype of Administrative module

Admin module performs the following steps:

1. Administrator login
2. Choose an item to edit, or create new items (Input description).
3. Click the item he wants to edit.
4. Choose product's catalogs to edit, or create new product's catalogs (Input description)
5. Choose a specific product to edit, or create new products (Input description)
6. Choose the type to enter of edit, for example the type of text description, graphical description and price information.
7. Click on the buttons to add, edit, or remove.
8. Finish product edit, back to step 3 to edit or add items, or back to step 4 to add or edit catalogs.
9. Repeat step 5-7 to edit or add new products.
10. Back to WAP SHOP EDIT homepage, log out.

2.2. Methodology:

2.2.1 Using JSP

JSP [2] is the universal programming language used in developing WOSS system, not only all the application files of Web server (WAP SHOP EDIT), but also the Customer Interface.

JSP overview

JSP stands for Java Server pages which are developed by SUN Micro system. It is a technology based on Java language and enables the development of dynamic websites. JSP enables server side development, and it mixes Java code, HTML [1][3] with special tags, and other mark-up languages, such as WML to provide dynamic contents.

Using JavaScript and HTML

JavaScript and HTML are definitely not strange for a sophisticated Web developer. But some parts of its syntax and semantics might be not familiar to specific readers, so in this section I give the description of major JavaScript and HTML syntax [4] used in WOSS system. Similarly, some source code will be quoted from WOSS system for readers well understanding the application JSP files.

JavaScript is a compact, object-based scripting language for programming client and server Internet applications. Most of the Internet navigators presently are all supporting JavaScript [2]. Client-side JavaScript statements embedded in an HTML page can respond to user events, for instance, mouse clicks, form input, or dropdown list and page navigation.

Document Object Model (DOM):

The Document Object Model (DOM) [5] is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML [6][7][8] Documents. Objects in the DOM tree may be addressed and manipulated by using methods on the objects. The public interface of a DOM [4] is specified in its application programming interface (API).

3. SYSTEM DESIGN

3.1 Overview

Shopping System include three main parts, Customer Interface, Administrative Module and Database management system. And in single module, there should be more than one application file which is in responsible for a specific task.

3.2 Design of User Interface

User interface module is to simulate the customer login, present the online shopping web pages to customers and submit customer's shopping requests. This module is designed for both mobile users and PC users. And designing information of this module can be shown as in figure3.

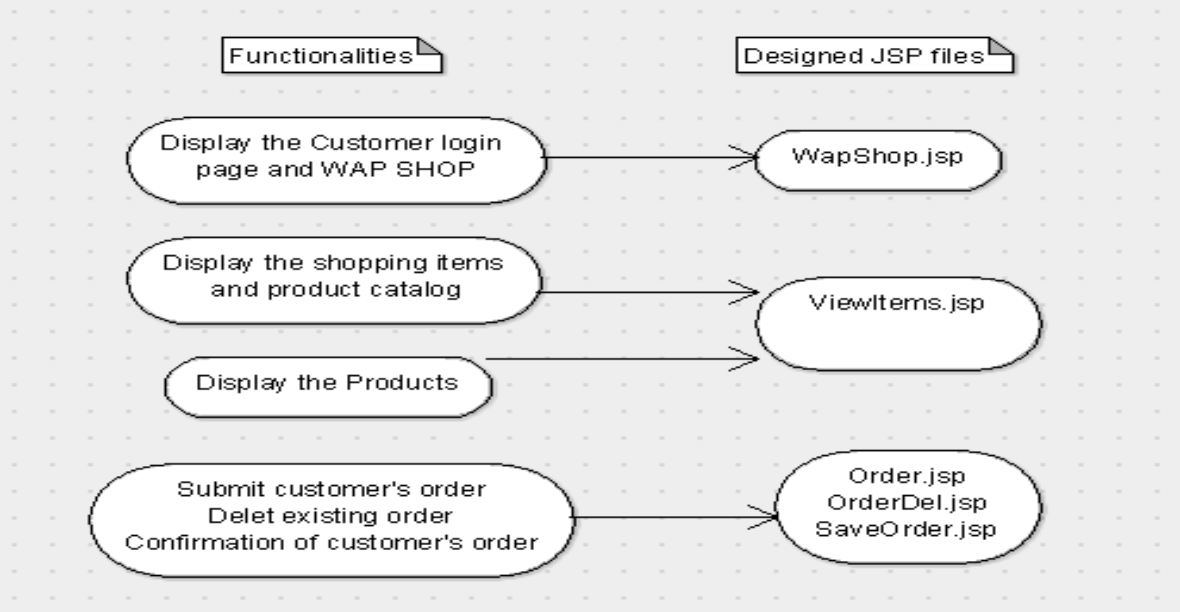

Figure 3 Arrangement of JSP files in Customer Interface

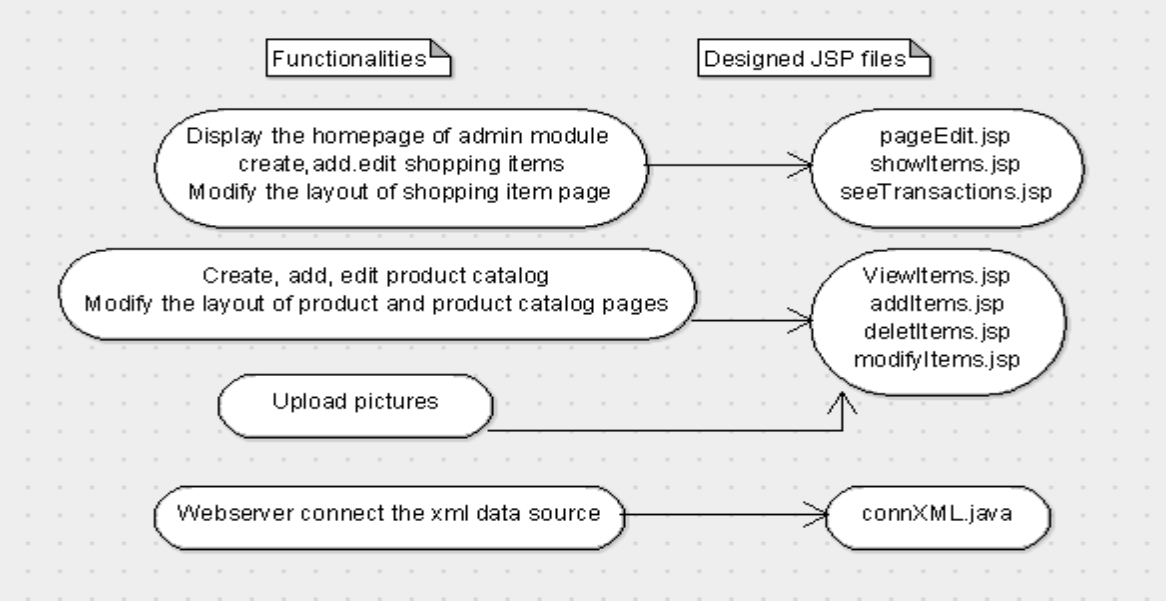

Figure 4: Arrangement of JSP files in Administrative interface

3.3 Design of Administrative Interface

Administrative module is to control the web server. It is designed to create, add, and edit shopping items; create, add and edit product catalogs; create, add, and edit product information; modify the layout of shopping pages and administrative interface; and send the data.

4. SYSTEM IMPLEMENTATION

Implementation of shopping basket and ordering system:

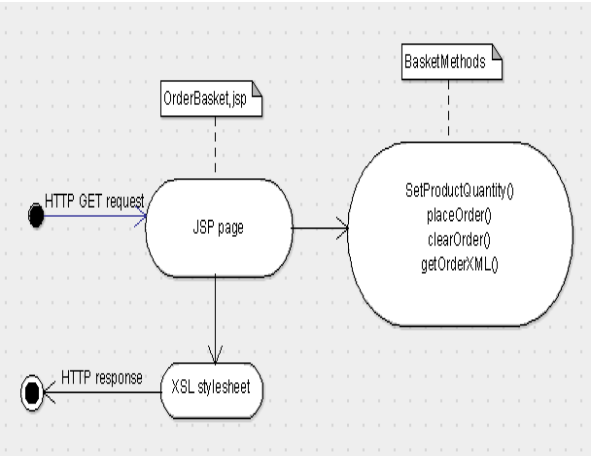

Figure5: Over view of architecture of order subsystem

When a user selects “Add to Order” on the search screen, the system needs to add the selected items into a “shopping basket”. The shopping basket is an object that must live in the *HttpSession* for the user so that it can remain between different pages. We choose to implement the shopping basket as a DOM Object [2].
This has two main benefits:
1. Displaying the current order involves simply applying a style-sheet to the current basket XML document [9].
2. When the customer places an order, the XML tree detailing the order already exists.

5. TESTING

5.1 Component Testing

5.1.1 Component test of User Module

When a customer wants to buy the products shown, the customer interface should show the price of products and feedback information of customer ordering action. All in all, after the test of Customer Interface, we concluded that all the *.jsp files related to Customer Interface are working correctly. Next step is to test the Administrative Module.

5.1.2 Component Test of Administrative Module

In the component test process of Administrative Module, all the visualization pages, by using Internet Explorer, were listed as the graphics indicated. *Image.jsp* is responsible for visualizing SHOP EDIT that is the homepage. For instance, the visualization of homepage SHOP EDIT. All in all, all the web pages of Administrative Module are displayed correctly. Finally, we finished the Component test of system.

5.2 Integration testing

After the component test of system, we can conclude that all the JSP files display correctly. Then we will do the integration test to find out whether these JSP files work together correctly or not. Integration test of system can be divided into three parts, e.g. integration test of Administrative editing process, integration test of Customer ordering process, and data saved in XML database

5.2.1 Integration test of editing functionalities

When an administrator logs on SHOP EDIT, it shows the existing items in that page. Administrator wants to add a new product catalog ‘test’ into this item. The integration test process of editing functionalities should be as following steps.

The last step is to input the price information for this picture, so customer chooses ‘Order’.

And in this customer shopping page, it display ‘buy now’ option. If Customer clicks on ‘buy now’ it will appear the respective price.

From the integration test process of editing functionality and Figures indicated above, I can conclude the integration between Administrative module and Customer Interface was correct. All the information edited or added in Page Editor can be correctly reflected in Customer Interface. All in all, the integration test between Customer Interface and Administrative module was successfully finished.

5.2.2 Integration test of ordering functionalities and data saved in XML database

From the above integration test, we can deduce the correctness of integration between Customer Interface and Administrative Module. However, we should still check the ordering functionalities; the data saved in the database [10] and whether the data are recorded correctly or not, for example, the customer orders are saved correctly or not. Because the correctness of displaying the shopping item, product catalogs and products, I can conclude the part of customer choosing products in the whole ordering functionalities is correct

Now to retrieve the customer order information, from XML database system [9] of system. All in all, from the graphics indicated above, we conclude that the customer order data are correctly saved in XML page. In other words, the correctness of integration between Customer Interface and page is proved.

6. RESULTS

Screenshots:

The screen shots of the admin system during the step by step process are given in the Figures 7a, 7b, 7c,7d,7e,7f and the user system in Figures 8a, 8b, 8c, 8d, 8e, 8f.

Screen shots of Administrator Module:

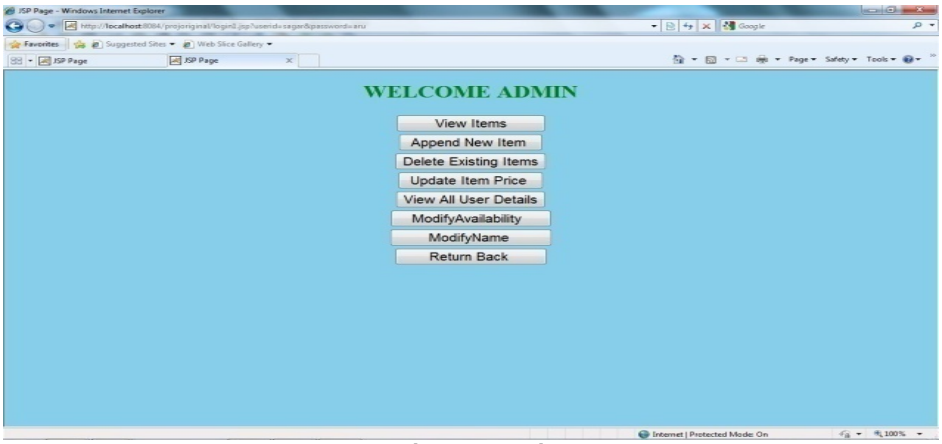

Figure 7a: Administrator welcome page

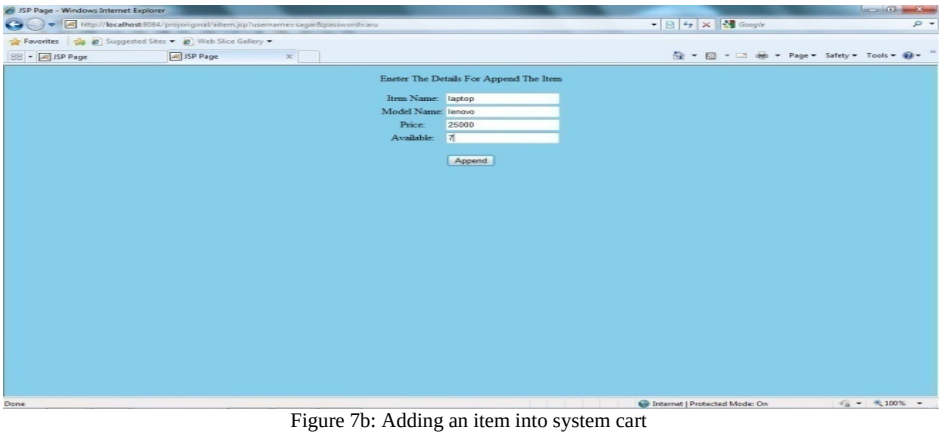

Figure 7b: Adding an item into system cart

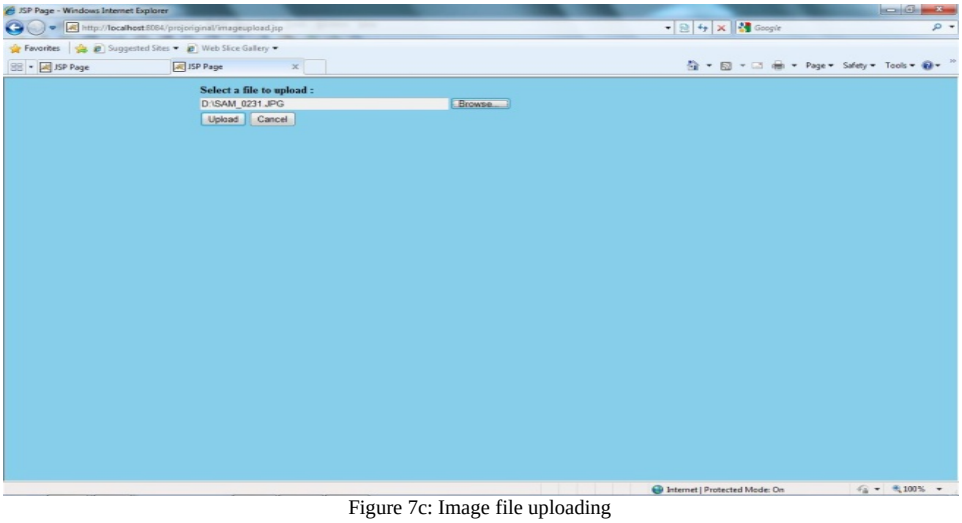

Figure 7c: Image file uploading

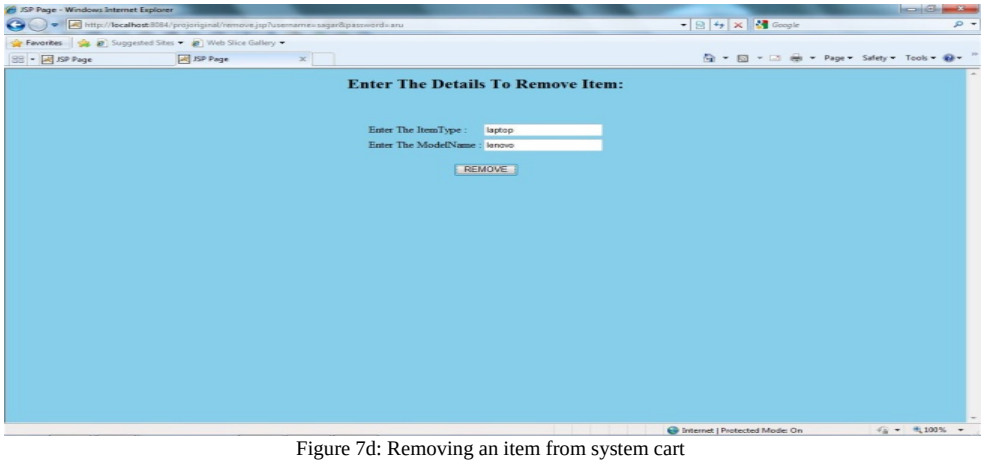

Figure 7d: Removing an item from system cart

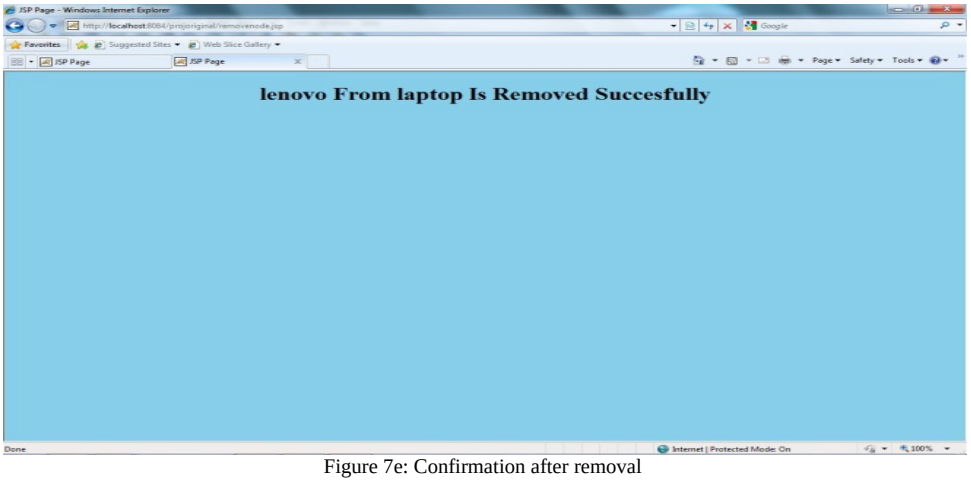

Figure 7e: Confirmation after removal

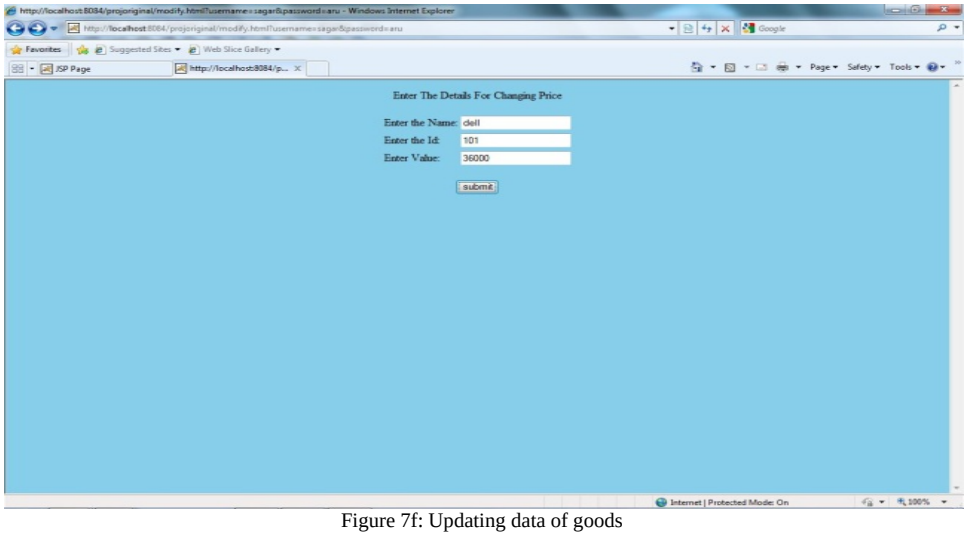

Figure 7f: Updating data of goods

Screen shots of User Module:

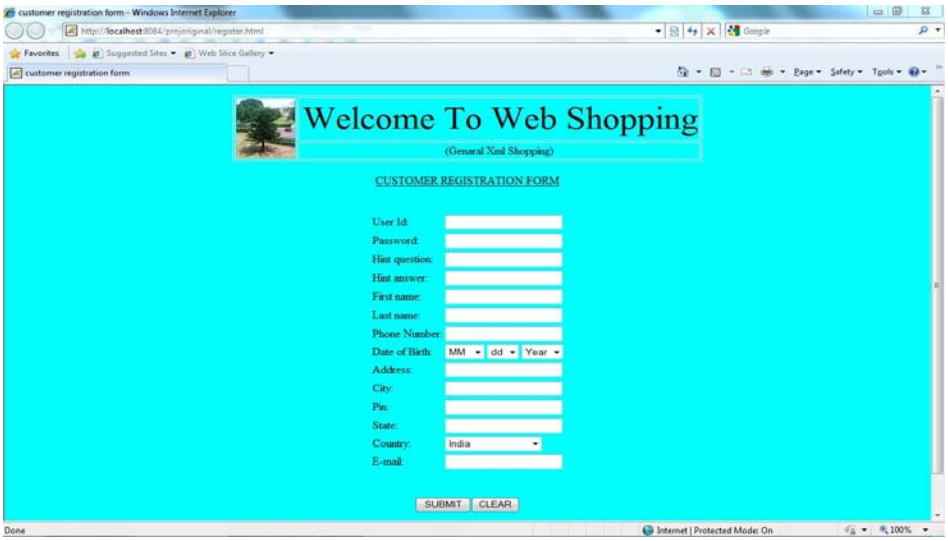

Figure 8a: New User Registration

Figure 8b: Login form (for registered users)

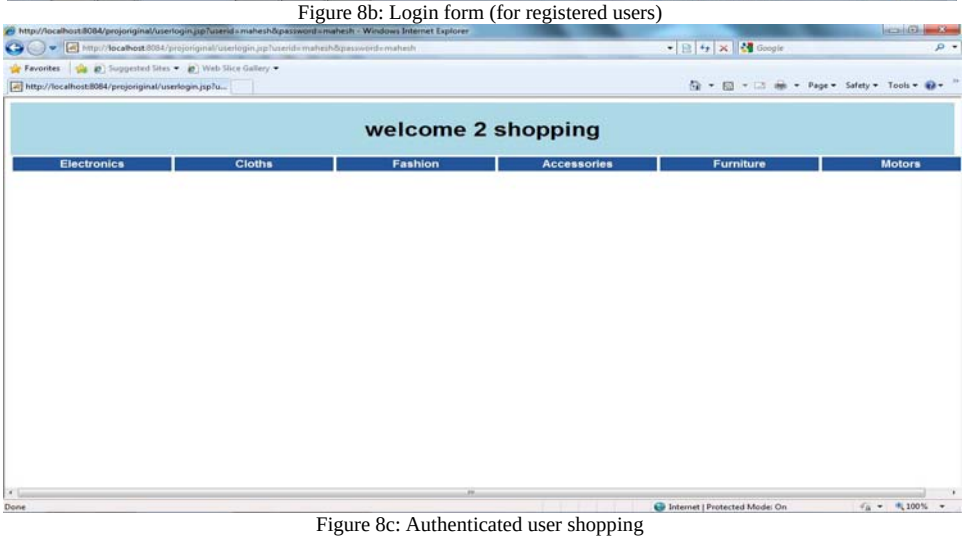

Figure 8c: Authenticated user shopping

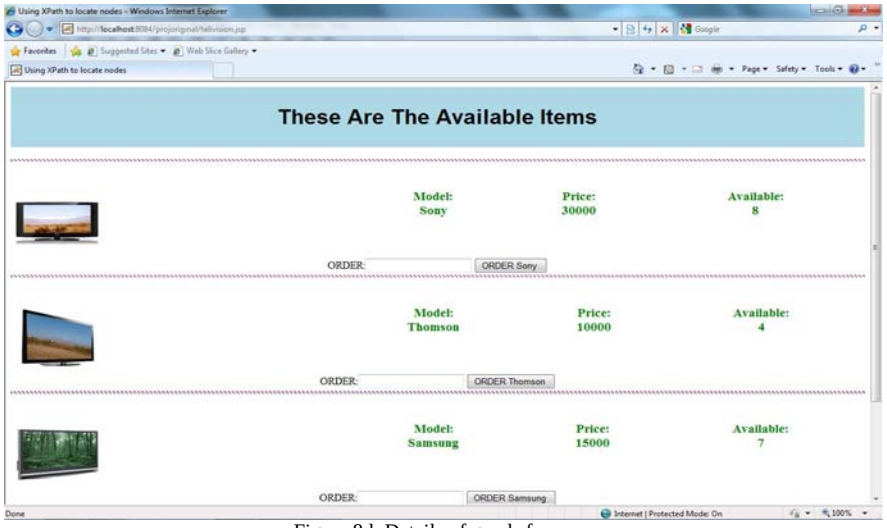

Figure 8d: Details of goods for user

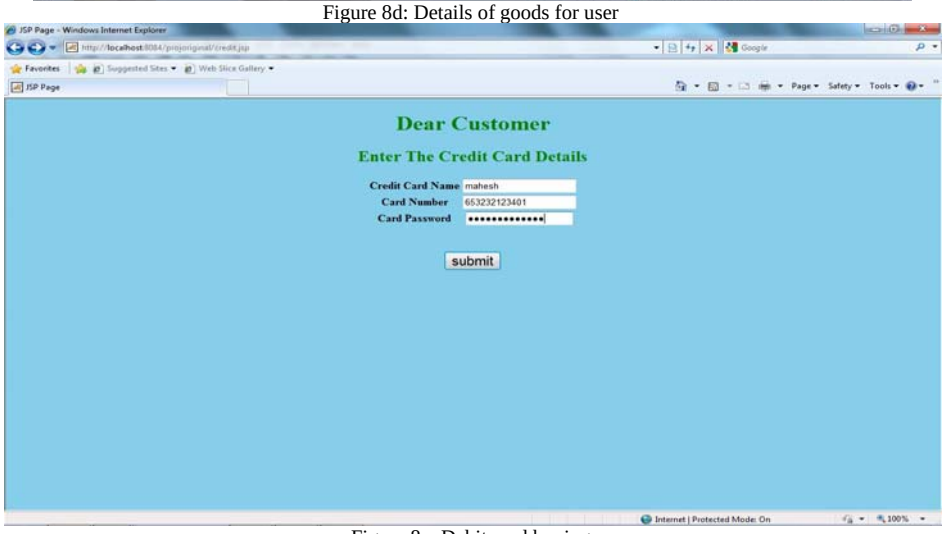

Figure 8e: Debit card buying

Figure 8f: Successful purchase of goods

6. Conclusion

We have successfully completed the online orientation for online shopping. The online shopping is the best and comfortable way for shopping and many people choose this as it saves their time and this is mainly helpful for small business vendors. This application envisages bridging the gap between the seller, the retailer and the customer. Online shopping is very advantageous as it is user friendly, quick to access and it is reliable software.

7. References

[1] "Document Object Model (DOM)". http://www.w3.org/: W3C. Retrieved 2012-01-12. "The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents."
[2] Flanagan, David (2006). *JavaScript: The Definitive Guide.* O'Reilly & Associates. pp. 312–313. ISBN 0596101996.
[3] Koch, Peter-Paul (May 14, 2001). "The Document Object Model: an Introduction". *Digital Web Magazine*. Retrieved January 10, 20094.
[4] Le Hégaret, Philippe (2002). "The W3C Document Object Model (DOM)". World Wide Web Consortium. Retrieved January 10, 2009.
[5] Guisset, Fabian. "What does each DOM Level bring?". *Mozilla Developer Center*. Mozilla Project. Retrieved January 10, 2009.
[6] Bourret, Ronald (20 June 2010). "XML Database Products". Retrieved 16 December 2011.
[7] Mustafa Atay and Shiyong Lu, "Storing and Querying XML: An Efficient Approach Using Relational Databases", ISBN 3-639-11581-3, VDM Verlag, 2009.
[8] O'Connell, S. *Advanced Databases Course Notes*, Southampton, University of Southampton, 2005, 9.2
[9] "Frequently Asked Questions About XML:DB". *The XML:DB Initiative*. Sourceforge. 2003. Retrieved 16 December 2011.
[10] "Extensible Markup Language (XML) 1.0 (Fourth Edition)". W3.org. Retrieved 2010-08-22.

Authors:

Mr. Z.Mahesh Kumar received his B.Tech degree in Computer Science and Engineering in 2011 from Acharya Nagarjuna University, Guntur, Andhra Pradesh and pursuing M.Tech degree in Computer Science and Engineering in 2011-13 from VIT university, Vellore, Tamilnadu. His areas of interest are Web Technologies, Database management systems. As part of this paper, he is working on developing new shopping system based on web which depend and work on XML data source, XML-Web Services and DOM Parsing.

Prof. R.Manjula received her B.E in Computer Science & Engineering from University of Vishwesvaraya and Engineering, Bangalore, Karnataka State, India in 1992 and M.E in Software Engineering from Anna University, Tamil Nadu, India in 2001. She is now working as Associate Professor and also as Ph.d Candidate affiliated with School of Computing Science and Engineering at Vellore Institute of Technology, Vellore, India. Her area of specialization includes Software Process modeling, Software Metrics, Software Metrics, Software Testing and Metrics, XML-Web Services and Service Oriented Architecture