ONLINE SHOPPING CART APPLICATION

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Swati Gupta

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

June 2013

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Online Shopping Cart Application

By
Swati Gupta

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Kendall E. Nygard
Chair

Kenneth Magel

Brian Slator

Linda Langley

Approved by Department Chair:

7/1/2013	Dr. Brian Slator
Date	Signature

ABSTRACT

Electronic commerce, also known as ecommerce is a type of industry where buying and selling of a product is conducted over electronic systems such as the internet.

The purpose of this application is to bring knowledge to students about ecommerce and how an interactive ecommerce application can be designed from scratch using client-side languages, such as JavaScript and HTML, combined with the server-side Java language through Java Server Faces. The server side, mostly Java, contains all the implementation related to setting up the database, creating session models for joining different user-interface (UI) pages, calculating the shipping costs and sales tax, etc. It is responsible for taking information from the database and making it available to the UI by mapping the category or item ID to the respective IDs stored in the database. The client side is responsible for showing the entire user interface, containing the CSS, HTML, and JavaScript.

ACKNOWLEDGMENTS

I would like to express my deep sense of gratitude and convey thanks to everyone who helped me and supported during the completion of this project and my research paper. First, I would like to express a deep sense of gratitude to Dr. Kendall Nygard for helping, guiding, and supporting me throughout my master's degree and research completion. I also convey thanks to my all committee members for helping me from time to time and for being on my committee. I acknowledge my department for providing the courses and a great atmosphere that helped complete different chapters of this paper. I especially thank my supervisor, Mickey Arora, for supporting me and my concepts and for allowing me to do something the way I liked, as well as my company, Thomson Reuters, for helping me develop the skills necessary to design this application as part of my master's research. Last but not least, I would like to thank my family members for their constant and unrelenting support towards my education and for their impartial love for me. I would also like to thank my friends, without whom this project would have been impossible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

It is known globally that, in today's market, it is extremely difficult to start a new small-scale business and its sustenance with competition from the well-established and settled/brand owners. Most often, even if the quality of the product is really good, due to a lack of advertisement or business at the small scale, it just becomes another face in the sea, and the product does not reach a larger group of customers. In fast paced life of today when everyone is squeezed for time, the majority of people are finicky when it comes to doing physical shopping. Logistically, a consumer finds a product more interesting and attractive when they find it on the website of a retailer directly and are able to see item's details online.[4][5] The customers of today are not only attracted because online shopping is very convenient, but also because they have broader selections, highly competitive prices, better information about the product (including people's reviews) and extremely simplified navigation for searching regarding the product. Moreover, business owners often offer online shopping options at low rates because the overhead expenses in opening and running a physical store are higher. Further, with online shopping, their products have access to a worldwide market, which increases the number of customers from different ethnic groups, adds customer value, and overall sustainable in the marketing.[8]

Online web stores, such as Amazon and eBay, have gained huge popularity over the years because one can buy almost everything at these stores. These web stores also give an opportunity to a lot of small-scale companies and manufactures to reach the global market and to directly sell their products to people without involving different other companies or middlemen before their product can reach the shelves of a physical store. Further, instead of using the available platforms, manufacturers can bring a concept of designing their own web store to sell their products directly to the masses.

1.1. Motivation

The motivation for designing this shopping-cart application came because I love online shopping rather than spending lot of time at physical markets. Further, using the available stores to sell the products, there is also the possibility of designing one's own customized shopping-cart application from scratch because custom-designed platforms are expensive. Moreover, I value recent learning about the Java and JavaScript programming languages as well as seeing how powerful and dynamic they are when it comes to web designing and applications. Apart from helping computer science students understand the concepts of web-application designing, it would be very easy to incorporate the idea of using programming techniques from the available visuals to understand how a piece of code appears on a user interface. The languages used to build this application are JavaScript, HTML, and Java because I found them to be extremely useful while working on the technologies at my workplace, Thomson Reuters.

1.2. Aim of the Software

This software is developed to help computer science students learn about application designing using JavaScript and HTML from their basic capabilities. This application allows the student to understand the basics about the appearance of a first web page and how a complete working application can be built from scratch. It allows students to understand the concept of user-integrated graphics and how JavaScript can be embedded into HTML. Further, it gives insight about how the client-side language interacts with the server-side language, Java, and finally with the database. This shopping-cart application is designed, primarily, for computer science students to learn and understand the concept of application development, and can also be used to teach ecommerce and web-application topics. The application can be downloaded and installed on different machines, and students can view the source code for all the different parts shown on the UI to visually understand how a particular piece of code works. This shopping-cart application is very versatile and can be enhanced by adding more functions and modified graphics for use with commercial purposes.

1.3. Literature Review

The history of ecommerce shopping carts began immediately after the World Wide Web, or WWW, became a major medium to communicate information around the world. Ecommerce shopping-cart applications allow consumers to buy goods or services directly over the internet using a web browser. This online shopping evokes the business-to-consumer (B2C) process where a consumer buys directly from the business. The process where a business buys from another business is called a business-to-business (B2B) process. The best examples of shopping-cart applications using B2B process are eBay and Amazon, both of which were launched in 1995. At present, most users of these online shopping-cart applications are people who have higher levels of education, have exposure to technological advancements, and are in a better income group. Such users develop a positive attitude towards these convenient shopping techniques.[7] According to a study in December 2011, Equation Research surveyed 1,500 online shoppers and found that 87% of tablet owners made online transactions during the early Christmas shopping season. [6]

Building a new successful shopping cart is simple because of high competition in the market, and the designer of a shopping-cart application must consider the information load, complexity, and novelty.[9] Complexity refers to the number of features available on the shopping cart and the levels of marketing, whereas novelty involves the unexpected or unfamiliar aspects of the site. A designer must also consider the consumers' needs and expectations. A userfriendly design is very critical to the success of any shopping-cart application because, unlike physical stores, consumers at online stores come from all ages, genders, and cultures.[10] Logistics clearly says that, to have a successful and profitable online shopping application, businesses have to spend a significant amount of time and money for designing, developing, testing, and maintaining the application. Apart from the high-class design and user interface, a good practice needs to be done to provide quality customer service.[11]

A typical shopping cart should contain certain features such as adding items to the cart and checking out those items using the available payment methods. Most shopping-cart applications are implemented using HTTP cookies or query strings, and an HTML setup is required to install the shopping cart on the servers that ultimately hosts the site on the internet. Most of these server-based applications require data related to the items added in the shopping cart to be kept in a session object which can be accessed later and manipulated dynamically because the users can add or remove one or more items from the cart. Most simple shopping-cart applications do not allow checkout to be done before any items are added to the cart. Data are often stored in an external database or application-based databases which can be accessed in real time by the application administrator.[12]

There are many examples of online shopping applications developed in different languages. Choosing a development platform and language depends on policies set by the company for which the application is being designed. It also depends on several other factors which are very important when considering the platform to design an application, for example, how portable the application will be after being built or if the application is open sourced. Java is chosen for this application because various reasons: it is a simple, robust, and cross-platform language. Applications written in Java can be transported and run on any environment, be it

MAC or Linux, because Java programs are compiled into platform-independent byte codes.[11] Because of the robustness of Java, it is a very safe language, as they provide exception handling and a layer method to communicate with the database, which prevents the system from crashing easily. Another very important factor from the development point of view is that the Java language is object oriented, where everything is treated as an object and where class methods are implemented instead of functions or procedures, which makes it very simple to understand the code.

Several Java shopping-cart applications were examined, and implementation details were compared with the proposed design for this application to build an even simpler architecture was developed which is very easy to understand from the learning perspective. Some online shopping applications are as follows:

- SoftSlate Commerce[12]
- Commerce4j[13]
- Cs.Cart[14]
- Apache Ofbiz[15]

These applications are designed for industrial purposes to generate revenue by providing these applications to customers looking to launch a website for their respective businesses.[17] The application proposed in this paper is more focused on developing a simple, yet complete, application specifically designed for computer science students to learn the basics about application design and development. This application performs all the basic functions that the above-mentioned applications do, such as selecting an item and adding it to the shopping cart, user login or registering, checkout of the item, etc. Other functions that can be added to this application are proposed in the future work, and they would be necessary under a more complete and complex design. The final application is expected to teach students enough to start them on a course to implement more complex functionalities, as mentioned in the Future Work.

1.4. Paper Organization

The rest of the document is divided into three parts: Objectives, Implementation, and Testing. The Objectives chapter lists the need for building the system. It provides use cases to help the business and technical users with their understanding. It also gives a detailed explanation for each use case to help with design and implementation, and outlines the constraints regarding the software. The Implementation chapter contains the detailed design of the system, including the Class Diagram, Activity Diagram, and Component Diagram. This chapter also includes a detailed explanation for each component as well as the interaction of the class and its components with each other when carrying out certain tasks, besides software’s mock screen shots.

CHAPTER 2. OBJECTIVES

All the steps required in the software-analysis process related to this project (product function, user characteristics, functional and nonfunctional requirements, constraints, assumptions, and dependencies for the online shopping cart application) are described in the following sections.

2.1. Requirements Analysis

The requirements analysis and gathering processes are critical for the success of any software engineering project. Requirements analysis in software engineering is a process that determines the tasks that are required to determine the needs and conditions to design a new product or to make modifications in any existing product/application. This process considers all the stakeholders' conflicting requirements, and analyzes the documentation and validation of the system. The requirements should be actionable, measurable, testable, and related to the defined needs of the system design. From the software-engineering perspective, requirements analysis is a three-step process.

1. Requirements Elicitation: Elicitation of requirements, also known as requirements gathering, includes the task of identifying various requirement types from stakeholders or from project documentation.
2. Requirements Analysis: Analysis of requirements determines if the gathered requirements are clear, complete, and consistent. The analysis also handles any ambiguous requirements that do not clearly state what needs to be implemented, which could create a loss of resources and time if identified later in the development or testing phase.

   Requirement analysis requires identifying the stakeholders and taking their needs into account to help them understand the implications of designing the new system, along with what modules are worth implementing and which ones are more cost efficient, and then to create a software-requirement specification document. To clearly elicit the stakeholders' requirements, different processes, such as developing a scenario or user stories, and identifying the use case which is being used for the project, can be utilized.

Stakeholder analysis says that, to clearly gather the requirements of the project, analysts first need to identify the stakeholders. Stakeholders are people or organizations that have a valid interest or use in the system. The steps to identify the stakeholders are as follows:

- Anyone who operates the system.
- Anyone who benefits from the system
- Anyone who is directly or indirectly involved in purchasing the system
- People or organizations opposed to the system
- Organizations responsible for the system design
- Organizations that regulate the financial or safety aspects of the system

Once the stakeholders are successfully identified, interviews are conducted through different processes; the needs and requirements of the system are identified, and a requirements specification document is prepared. The document is then discussed with the major stakeholders to identify any ambiguity with the requirements and understanding of the system.

3. Requirements Documentation: This step involves documenting the requirements in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. A requirement specification document is categorized in different ways according to the stakeholders' need, helping to create a clear contract between development and business. The following sections include the different

categories of requirements specification document that are essential for designing this application: the functional requirements, constraints, system requirements, etc.

2.1.1. Product Perspective

The online shopping-cart application is a web-based system. It can be accessed using Internet Explorer 8.0 and above, Mozilla Firefox 2.0, and Google Chrome.

2.1.1.1. User Interface

The two interface types found in the online shopping-cart application are as follows:

1. **User Interface**: Users are able to view the home page of the shopping-cart application, browse the different categories, browse and add any number of items from any categories in the shopping cart, look for information about each product, delete the items in the shopping cart, save the cart for later viewing, check out or continue shopping after adding the item to the cart, and check out the items by completing the required information in the order form.
2. **Admin Interface**: The administrator is able to view the users' information that was entered during checkout in the database, can update the information, price, shipping costs of the items, add or remove items from the main display.

Both interfaces are described, in detail, in the External Interface Requirement section of Chapter 3.

2.1.1.2. Hardware Interface

The online shopping-cart application shall provide minimum hardware requirements. The following hardware configurations are required for a PC using the online shopping-cart application:

- Pentium processor

- 32 MB of free hard-drive space
- 128 MB of RAM

2.1.1.3. Software Interface

This section lists the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run the application, the driver for running Java web applications, the integrated development environment to develop the application, and the third-party tool used for editing purposes are as follows:

1. Operating System: Windows (Vista/Windows 7) or MAC OS
2. Web Brower: Internet Explorer (8.0 and above), Mozilla Firefox (3.0 and above), or Google Chrome
3. Drivers: Java Runtime Environment
4. Integrated Development Environment: Eclipse Juno or Apache Tomcat
5. Third-Party Tool: Microsoft Word

2.1.2. Product Function

The online shopping-cart application would have the following basic functions:

1. Display all the categories available for shopping on the system's main page.
2. Display all the items linked to each category listed on the main page.
3. Allow the administrator to add new items to the existing list of available items.
4. Allow users/administrator to remove items.
5. Allow the administrator to modify the price of each item.
6. Allow the administrator to update the description about each item.
7. Allow the administrator to view and edit information about each user that checkouts the items from the system.

2.1.3. User Characteristics

The users of the online shopping-cart application, based on their roles, are customers (users) and the administrator (owner). These users are identified based on their experience and technical expertise.

1. **Admin:** The administrator is the owner of this online shopping-cart application. One must have a basic understanding of computers and the internet as well as prior knowledge for operating the eclipse and Java programming languages. The administrator is responsible for maintaining all the training documents required for the system. The administrator can perform the following functions:
   - Assign or change the price of the items, update the items in the list, and delete the items.
   - Assign sales tax for different states at the time of checkout.
   - View the history of the customers who purchased the items.
2. **Users:** The users of this online shopping-cart application are all customers who would shop to test the application. These users are anyone with shopping experience and the know-how to browse through a shopping-cart application. They must have basic understandings about computers and the internet. The users should be able to perform the following functions using this system:
   - View, browse, and select a category on the home page.
   - View, add, and update items in the cart.
   - Delete items from the cart.
   - Check out the items from the application or continue shopping.
   - Sign-on/login using a username and password.
   - Place the order by completing the order form.

2.1.4. Constraints

1. Hardware Limitations: The minimum hardware requirement for the system is 128 MB of Ram and a 32-MB hard-disc drive.
2. Accessibility: Initially, the software should be available as a desktop application for a small set of users to test.
3. Others: The application should be built using Java and JavaScript inscribed in HTML, and it should, initially, be accessible through the eclipse IDE and later published on a server.

2.1.5. Assumptions and Dependencies

The assumptions and dependencies are as follows:

1. Users and the administrator are accustomed to the paper-based system and would require training to use the online shopping-cart application.
2. The system is dependent on the availability of an Apache Tomcat Server to run.
3. We assume that system users adhere to the system's minimum software and hardware requirements.
4. This system will use third-party software, and it is assumed that system users are familiar with the software.

2.1.6. Specific Requirements

This section contains details about all the software that is required for designers to create a system to satisfy the users' requirements and for testers to test the given requirements. This section contains the interface description of each GUI for the different system users. These sections also give descriptions about all the system inputs, all the functions performed by the system, and all the system output (responses).

2.1.6.1. Functional Requirements

This section contains the requirements for the online shopping-cart application. The functional requirements, as collected from the users, have been categorized as follows to support the types of user interactions that the system shall have.

1. **Educational Purpose:** The main purpose of this online shopping-cart application is to teach computer science students the basics of the Java, JavaScript, and HTML programming languages along with the concepts of web-application designing.
   - **FR01:** The students shall be able to view the source code for the entire application.
   - **FR02:** The students shall be able to, individually, view and understand the code for all pieces on the UI.
   - **FR03:** The students shall be able to debug the application's source code using Firebug, which is an online tool to inspect, edit, and monitor HTML, CSS, and JavaScript requests directly on the web page.
2. **User: View Categories and Items:** The users shall be able to see the home page of the online shopping-cart application when they first run the program. The users shall be able to view the different categories, select categories, browse through the items in each category, and add items to the shopping cart. The users shall be able to view the shopping cart and more information about each item.
   - **FR04:** The users shall be able to view the categories on the application's home page.
   - **FR05:** The users shall be able to view items in different categories.
   - **FR06:** The users shall be able add items to the cart.
   - **FR07:** The users shall be able to view more information about an item before adding it to the cart.

- **FR08:** The users shall be able to able view the shopping cart.
- **FR09:** The users shall be able to browse through the available items.

3. **User: View Shopping Cart:** After the first run of the application, the users shall be able to see their designated home page. After browsing through the items and adding items to the shopping cart, the users should be able to view the items in the shopping cart. The users shall be able to check out or continue shopping. The users shall be able to delete items from the cart.
   - **FR010:** The users shall be able to view the items added to the cart.
   - **FR011:** The users shall be able to check out with the current items in the cart.
   - **FR012:** The users shall be able to continue shopping.
   - **FR013:** The users shall be able to delete items from the cart.
4. **User: Checkout Items**
   - **FR014:** The users shall be able to check out items only when there are items in the shopping cart.
5. **Login/ User Authentication**
   - **FR015:** The users shall login or register using the user authentication form.
   - **FR016:** The users shall not login or register if the information is incomplete or invalid.
6. **User: Place Order**
   - **FR017:** The users shall place an order by completing the information in the order form.
   - **FR018:** The users shall not be able to place an order if the information in the order form is invalid or incomplete.
7. **Admin: View User Information**

- **FR019:** The administrator shall be able to view all the users' information that completes the order form and the checkout process.

8. **Admin: Add/Update/Delete Shopping Items**

- **FR020:** The administrator shall be able to add new items to the list of shopping items.
- **FR021:** The administrator shall be able to modify/update an item's price and description.
- **FR022:** The administrator shall be able to delete items from the main page of the shopping-cart application.

Additional Functional Requirements

- **FR023**: The administrator shall be able to view the entire history of the checked-out items.
- **FR024:** The administrator shall be able to view the entire history for the users who successfully complete the checkout process.

2.1.6.2. Performance Requirements

This section lists the performance requirements expected from the online shopping-cart application.

1. **PR01:** The users shall be able to add an item to the cart in fewer than 5 seconds.
2. **PR02:** The users shall be able to view information about an item in fewer than 5 seconds.
3. **PR03:** The users shall be able to check out the items in the shopping cart within 10 seconds after completing the order form.
4. **PR04:** The navigation between pages shall take fewer than 5 seconds.
5. **PR05:** The application shall be able to do a validation check on the information provided in the user-authentication form and the place-order form to avoid false or incomplete information.

2.1.7. Design Constraint

This section lists the design requirements for the online shopping-cart application.

1. **DC01:** The user interface (UI) must have specific fonts and font sizes. The system shall match the fonts and font sizes used for all the pages of the application.

2.1.8. Software System Quality Attribute

1. **Integrity**
    - **QA01:** The authorized user shall be allowed to access the online shopping-cart application.
    - **QA02:** Based on the user type, the online shopping cart application shall provide a user-specific interface.
2. **Correctness**
    - **QA03:** The assigned task should be received by the specified user.
3. **Availability**
    - **QA04:** The system shall be made available to the user/administrator year round.
4. **Robustness**
    - **QA05:** The system shall be able to save items to the shopping cart.

CHAPTER 3. IMPLEMENTATION

This chapter includes the detailed design used to build the online shopping-cart application. The system's design is used to create the functions and operations of the gathered requirements in detail, including screen layouts, business rules, process diagrams, and other documentation. The output of this chapter describes the new system which is defined as a collection of modules and subsystems. This design stage takes the initial input requirements that were identified in the approved requirements specification document. For each requirement, there is a set of one or more design elements that are produced using the different prototypes. These design elements describe the desired software features, in detail, including functional hierarchy diagrams, screen layouts, activity diagrams, and class diagrams. The intention of these diagrams is to describe the software in detail so that the system can develop the application with less additional design input. The system’s mock screen shots are shown later in this chapter.

3.1. Detailed Scope

This project is supposed to be delivered in three phases, with each phase being an add-on to the project that makes it more usable and acceptable.

1. In the first delivery, the application must be able to add an item to the shopping cart and case.

    - Browse categories on the home page
    - Select a category and browse through the items
    - View more information about an item.
    - Add an item to the shopping cart.
    - Continue shopping or go to checkout for the item.

2. The application must be able to check out the items in the cart.

- Check out the items.
- Continue shopping.
- Delete the items to update the shopping cart.

3. The application asks for user authentication before checking out.

- Add items to the cart.
- Check out the items
- Log in with a valid username and password.

4. The application must bring up the order form for the check out.

- Complete the information on the order form.
- Place the order.

3.2. Static Decomposition and Dependency Description

This section contains the system use-case diagram for the online shopping-cart application and also has a detailed explanation for each use case in the system.

3.2.1. High-Level Use Case Diagram

The system's use case shows the user a detailed view of the system and how the actors would interact with each other and with the system. The explanation for each use case is then provided below the system use case for the administrator (Figure 1) and the user (Figure 2), helping the user to understand who are the actors areas as well as giving the description for each use case along with its pre- and post-conditions that should be satisfied once the use case is implemented in the software.

Figure 1 demonstrates the use case of for an administrator where he or she has access to the application. The administrator can access the home page, select a category, or add/delete items to/from the cart.

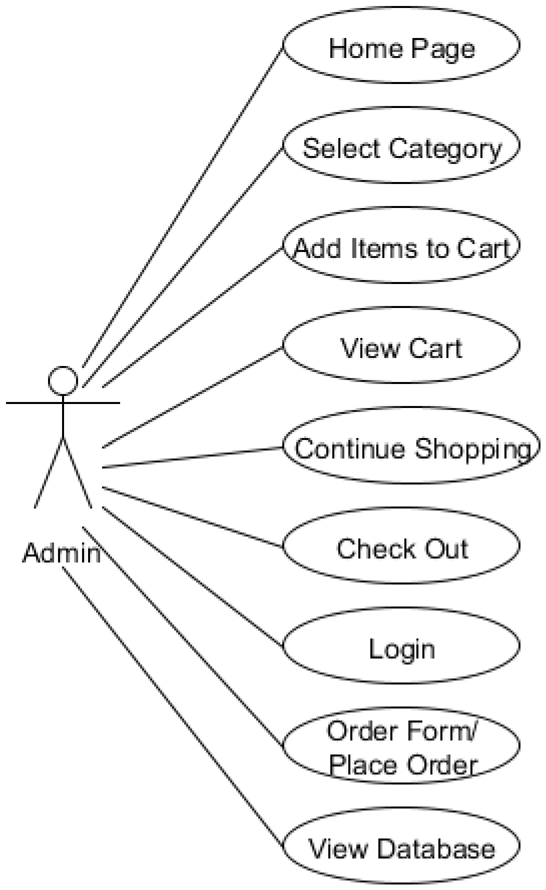


Fig. 1. Online Shopping Cart Application System Use-Case Diagram: Admin.

Figure 2 demonstrates the use case for users where they have access to the online shopping-cart application. They can access the home page, select a category, add/delete items to/from the cart, view the shopping cart, and decide to either continue shopping or check out. They are required to go through the user-authentication form (login) which would only allow them to place an order for the items they selected.

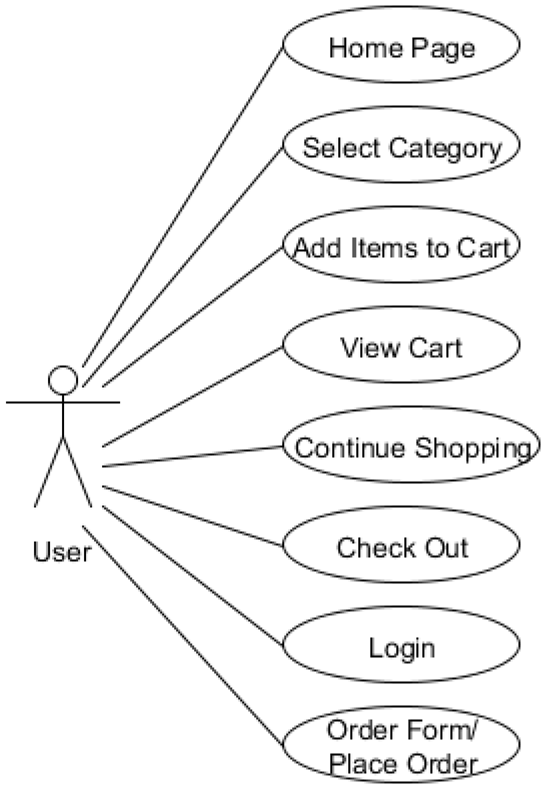


Fig. 2. Online Shopping-Cart Application System Use-Case Diagram: User.

Below are the different use cases in the system, the use case, and the actors associated with each use case. The description is used for a novice user to better understand the workings of the system and the pre-conditions that should be satisfied before invoking each use case.

1. **Use-Case Number:** US-001

   Application: Online shopping-cart application

   Use-Case Name: Home page

   Use-Case Description: This use case lets the user/administrator view the home page which has different categories when they first run the application.

Primary Actor: Admin/User

Precondition: Run the application.

Post-condition: The user successfully runs the application and is able to view the home page with the different categories.

Basic Flow:

- Run the application
- View the home page
- Browse the categories

2. **Use-case number:** US-002

Application: Online shopping-cart application

Use-case name: Select category

Use-case description: This use case details the category for selecting a process where the user can browse through the different categories and select one category to view items.

Primary actor: Admin/User

Precondition: The user/administrator successfully runs the application to view the home page and browse different categories.

Post-condition: The user successfully selects a category to view items in a particular category.

Basic Flow:

- Run the application

- View the home page
- Browse the categories
- Select a category
- Browse through the items

3. **Use-Case Number:** US-003

   Application: Online shopping-cart application

   Use-Case Name: Add Item

   Use-Case Description: This details the item-adding process for the system to access it. The user should be able to add items to the cart and view information about the item.

   Primary Actor: Admin/User

   Precondition: The items available for shopping are available for the user to browse.

   Post-condition: The user successfully adds items to the cart.

   Basic Flow:

   - View the home page
   - Browse the categories
   - Select a category
   - Browse through the items
   - Add item to the cart.

4. **Use-Case Number:** US-004

   Application: Online shopping-cart application

   Use-Case Name: View cart

   Use-Case Description: This use case helps the user to view the shopping cart in the application.

   Primary Actor: Admin/User

   Precondition: None. The cart can be viewed with or without adding any items.

   Post-condition: The user successfully adds the item to the shopping cart.

   Basic Flow:

   - View the home page
   - Browse the categories
   - Select a category
   - Browse through the items
   - Add item to the cart.
   - View cart

   Exceptional Flow:

   - Run the application
   - Select a category
   - Do not add items to the cart
   - View empty cart

5. **Use-Case Number:** US-005

   Application: Online shopping-cart application

   Use-Case Name: Continue shopping

   Use-Case Description: This use case helps the user to decide whether he or she would like to continue shopping after adding the item to the shopping cart or when he or she views the empty cart.

   Primary Actor: Admin/User

   Precondition: Continue shopping

   Post-condition: The user is successfully able to view the shopping cart with the option to continue shopping or to check out when there are items in the cart, or to continue shopping when there are no items in the cart.

   Basic Flow:

   - View the home page
   - Browse the categories
   - Select a category
   - Browse through the items
   - Add an item to the cart.
   - View cart
   - Continue shopping/check out

   Exceptional Flow:

   - Run the application

- Select a category
- Do not add items to the cart
- View empty cart
- Continue shopping

6. **Use-Case Number:** US-006

   Application: Online shopping-cart application

   Use Case Name: Checkout

   Use-Case Description: This use case helps the User/Admin to check out items from the shopping cart.

   Primary Actor: User/Admin

   Precondition: There is at least one item in the shopping cart to cause the checkout button to appear.

   Post-condition: The user is able to click on the checkout button when there are items in the cart.

   Basic Flow:

   - Run the application
   - Add items to the cart
   - Go to the view-cart page
   - Click the checkout button
   - Go to the order-form page

Exceptional Flow:

- Run the application
- Do not add any item to the cart.
- Go to the view-cart page
- The checkout button is not available to click.

7. **Use-Case Number:** US-007

Application: Online shopping-cart application

Use-Case Name: Login

Use-Case Description: This use case helps the User/Admin to check out items in the shopping cart by logging into the user-authentication form.

Primary Actor: User/Admin

Precondition: There is at least one item in the shopping cart to check out the items and to login to the user-authentication form.

Post-condition: The user is successfully able to log in.

Basic Flow:

- Run the application
- Go to the view-cart page
- Click the checkout button
- Enter the username and password.
- Login/Register.

Exceptional Flow:

- Run the application
- Go to the view-cart page
- Click the checkout button
- Enter an incorrect username and password.
- Login/Register fails.

8. **Use-Case Number:** US-008

Application: Online shopping-cart application

Use-Case Name: Place order

Use-Case Description: This use case helps the User/Admin to check out items in the shopping cart using the order form.

Primary Actor: User/Admin

Precondition: The user/administrator is able to login or register on the user-authentication form.

Post-condition: The user is successfully able to check out items using the order form.

Basic Flow:

- Run the application
- Go to the view-cart page
- Click the checkout button
- Enter the username and password.

- Login/Register.
- Enter the information in the Order Form
- Place order is success.

Exceptional Flow:

- Run the application
- Go to the view-cart page
- Click the checkout button
- Enter the username and password.
- Login/Register.
- Enter the information on the order form
- Invalid/Incomplete information
- Place order is not allowed

9. **Use-Case Number:** US-009

Application: Online shopping-cart application

Use-Case Name: View database

Use-Case Description: This use case helps the administrator to view the items, the user's information, product details, and the checkout details in the database.

Primary Actor: Admin

Precondition: The administrator is able to connect the database controller when he or she first runs the application.

Post-condition: The administrator is successfully able to view the database with the user's information as well as the product and checkout details.

Basic Flow:

- Run the application
- Connect to the database
- View the user's information.

3.2.2. Activity Diagram

This section lists the activity diagram and describes the flow of activities in the system. A detailed description is then given after the figure for each activity. Figure 3 provides the overview of the activity of online shopping cart application.

The figure below demonstrates the activity flow for this online shopping-cart application. The flow of the application is similar for both the user and administrator. The flow begins when the user first runs the application home screen online shopping-cart application that appears in the web browser. The user can browse through the available list of categories and can choose either to select a category or to directly view the cart. In the category, a user can select view more information for details about a particular item before deciding to add it to the shopping cart by clicking on the cart icon next to the item. The user can then decide to either continue shopping by clicking the continue shopping button or can check out by clicking on the checkout option. If there are no items in the cart, then the user does not have an option to click checkout. The user can check out after doing the user authentication by logging in with the username and password. Once the user successfully logins/registers, the order form, where the user can put the correct information to place the order appears. If the user includes incorrect or incomplete information, then placing the order is not allowed. After the user successfully inputs the correct information, placing order is successful, and the user can see the success message. The additional flow step for the administrators is that they can view the user's information, the user's checkout, and the product details by using the database after the user successfully places an order.

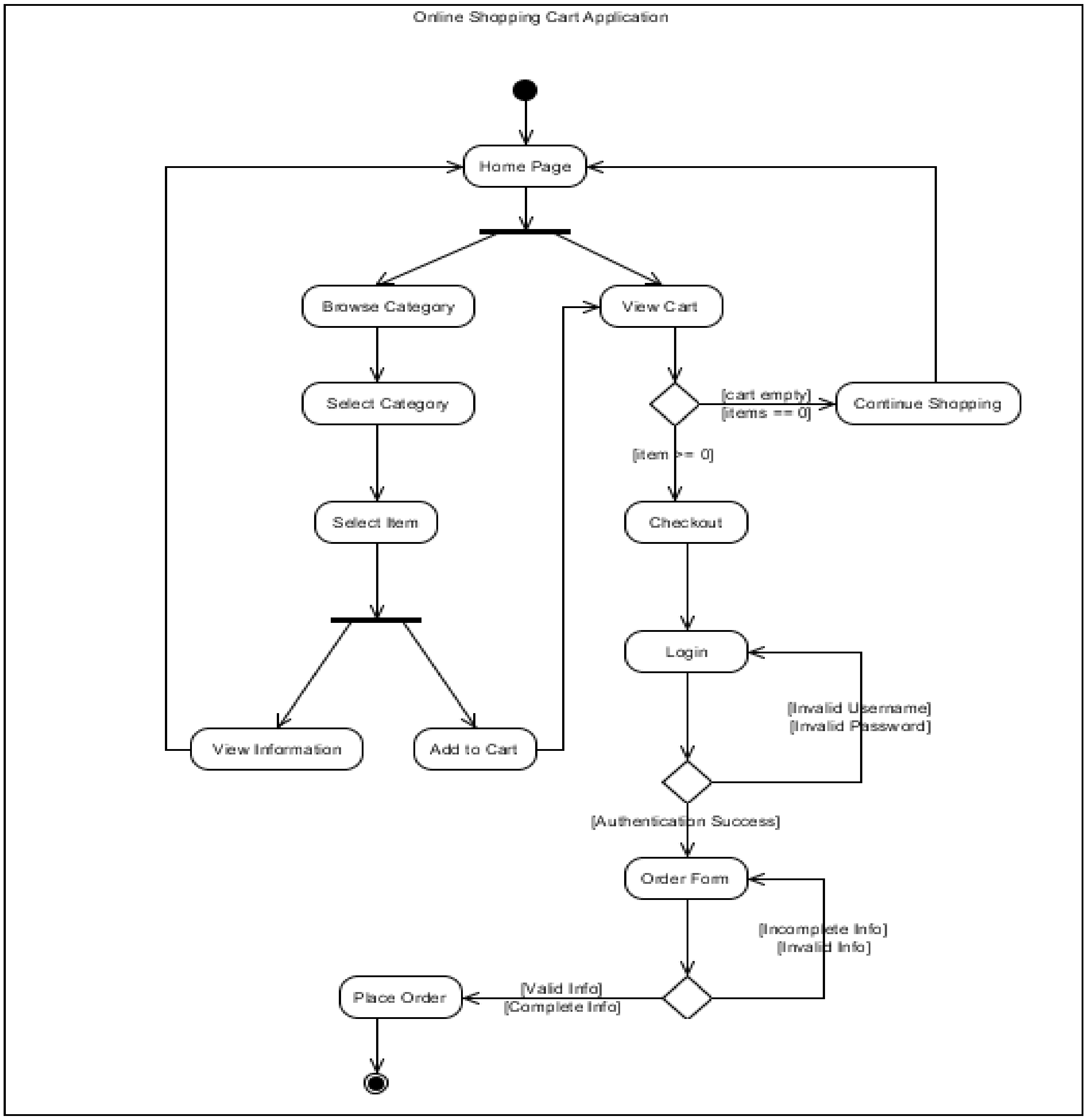


Fig. 3. Activity Diagram for Online Shopping-Cart Application.

3.2.3. Class Diagram

1. **User Authentication:** This class is utilized to get user information from the database and is for authenticating the users. The class diagram in Figure 4 shows the methods that are used in this class and the description of each class is listed below.

   Authenticate User: This message is used to authenticate a particular user who has provided the login credentials and wishes to login in the system. This method checks the credentials in the database.

   Check User Name: This method checks to see if the provided username already exists in the database. If there is an existing user with the same name, then the user is prompted to select another username to create an account.

   Register User: This method allows a new user to register for an online shopping-cart account by entering a valid username and password. If the username already exists in database, the user will be prompted to choose another name.

   Login User: This method allows the existing users to log in to the database with the credentials they used for first registering into the application.

2. **DB Controller:** This class is used for getting users and product information from the database, and it is also used to update the database with the information about new-user registration, product checkout, and user details.

   Initialize DB: This method allows the initialization of a database on the first run of the application.

   Retrieve Items List: This method fetches all items from the code/workspace into the database and allows the administrator to view information about the items.

Retrieve Category List: This method retrieves the list of all categories that are available for the shopping-cart application.

Retrieve User's Details: This role locates all the registered users in the database and also fetches any new user who registers by completing the user-authentication form.

Get Order Details: This role updates the row in the database with the details of the user who checks out the items and successfully places an order.

Get Ordered Product Details: This role inputs the details of the order once the items are checked out and the order is successfully placed. This role updates the row in the database to show the details of products that have been checked out.

3. **Place Order:** This class is used to process all information regarding the order of products.

   Get Checkout Information: This method provides information about items in the shopping cart once the checkout method is called. This will further invoke the user-authentication method.

   Get User Authentication: This method fetches the user-authentication information. The place-order class is invoked once all three methods are successfully called. If the user authentication fails, the place-order class is not executed.

   Retrieve Order Form Information: This method is executed once user authentication is a success. This method evaluates an order form on the UI for dynamic input from the user. If any of the information is incomplete or is invalid, then an error message is depicted, and the place-order class is not called.

4. **Cart:** This class invokes the shopping cart. This class can be called in cases when no items are added to the cart, when items are already there in the cart, or when the user adds an item to the cart.

Get Items: This method retrieves information for all the items that the user adds to the shopping cart.

5. **Checkout:** This class invokes the checkout button. The checkout class is only called when there are items in the shopping cart. In the case of an empty shopping cart, the checkout class is not called, thus no checkout option is available on user-interface screen.

   Retrieve Cart Info: This method retrieves the shopping cart's information if there are items in the cart, and then, the checkout class is called.

   Get Items: This method gets all the items in the shopping cart. If the count of items in shopping cart is greater than 0, then this function updates the card information, indicating that cart is not empty which then calls the checkout class.

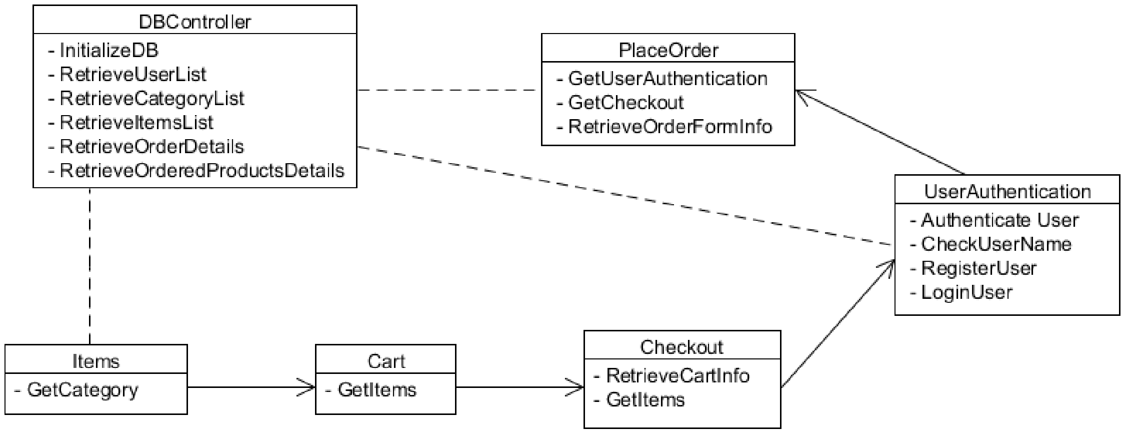
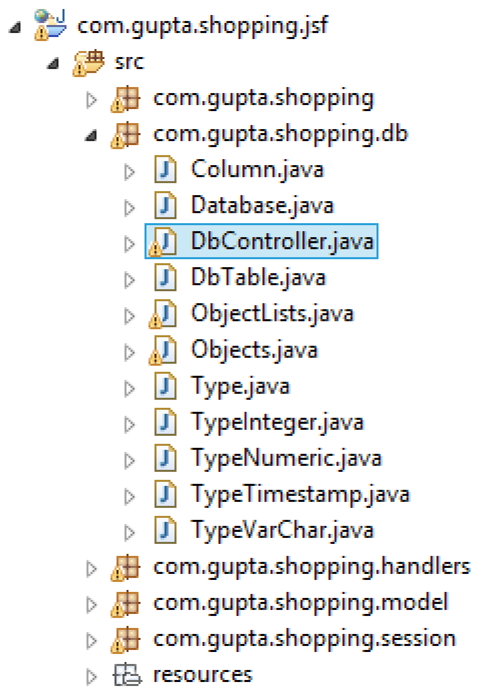


Fig. 4. Class Diagram.

3.3. The Shopping Cart Application Implementation

This section contains the implementation details for different packages and classes of the shopping-cart application. It also contains the coding snippets that help computer science students understand what a particular section of code represents. The following steps are required to run the application successfully:

1. Install the Eclipse IDE and Apache Tomcat
2. Configure the tomcat server, and import the project file into Eclipse IDE
3. In the package explorer, expand the database package (Figure 5), and run DbController.java as a Java application.



```
◢ com.gupta.shopping.jsf
  ◢ src
    ▷ com.gupta.shopping
    ◢ com.gupta.shopping.db
      ▷ Column.java
      ▷ Database.java
      ▷ DbController.java
      ▷ DbTable.java
      ▷ ObjectLists.java
      ▷ Objects.java
      ▷ Type.java
      ▷ TypeInteger.java
      ▷ TypeNumeric.java
      ▷ TypeTimestamp.java
      ▷ TypeVarChar.java
    ▷ com.gupta.shopping.handlers
    ▷ com.gupta.shopping.model
    ▷ com.gupta.shopping.session
    ▷ resources
```

Fig. 5. The Database Package.

4. Once the JDBC connection is established using the database, run the default.xhtml class from the web-content package that initiates the tomcat server and runs the application locally (Figure 6).

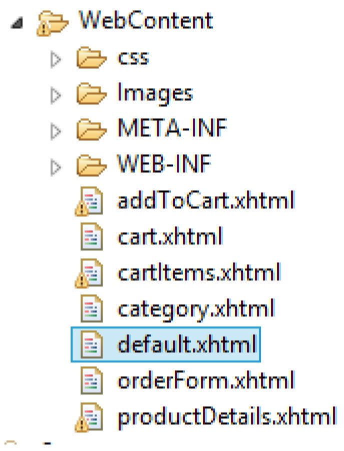


Fig. 6. Web-Content Package.

Default.xhtml is a client-side implementation and is responsible for displaying the application's default main page. The code is implemented in HTML and JavaScript which calls into the Master.XHTML which contains the template for the user-interface display. The coding snippet in Figure 7 shows the setup of the main default page that contains the application title and calls the display-category page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:f="http://java.sun.com/jsf/core">

<ui:composition template="/WEB-INF/templates/Master.xhtml">
        <ui:define name="Title">
                        Online Shopping Center
        </ui:define>
        <ui:define name="ContentPlaceHolder1">
                <script>
                        document.write("#{defaultPage.getCategoriesDisplay()}");
                </script>
        </ui:define>
</ui:composition>
</html>
```

Fig. 7. Implementation of the Default Page for the Shopping-Cart Application.

The shopping-session bean represents server-side implementation of this online shopping-cart application. This bean's client can add or remove an item or can retrieve information from the database. All session beans require a session model that maps the item ID with the one described in the database. The business interface for this online shopping-cart application is a fairly plain Java interface that defines all the business logic implemented in the session-model class. The primary purpose of this session model is to run business tasks for the stakeholders. The snippet in Figure 8 describes the session model used in the application code which is being mapped to the controller- and server-side functions for calculating the sales tax and validating the form information.

```
package com.gupta.shopping.session;

import static com.gupta.shopping.db.DbController.COL_CATEGORY_ID;
import static com.gupta.shopping.model.Model.MODEL;
import static com.gupta.shopping.model.SalesTaxMap.getSalesTaxRate;
import java.math.BigDecimal;
import java.math.MathContext;
import com.gupta.shopping.model.Product;
import com.gupta.shopping.model.User;

public final class SessionModel {

	private static final BigDecimal BD_ZERO = new BigDecimal("0.00");
	private static final BigDecimal BD_SHIPPING_CHARGES_DEFAULT = new
BigDecimal("6.99");
	private static final MathContext ROUNDING_POLICY = new MathContext(2);
	private final Products products = new Products();
	private String categoryId;
	private User user;
	private String zip;
	private BigDecimal orderTotal = BD_ZERO;
	private BigDecimal salesTax = BD_ZERO;
	private BigDecimal deliveryCharge = BD_SHIPPING_CHARGES_DEFAULT;
	private BigDecimal finalTotal = BD_ZERO;
	private boolean orderPlaced = false;
	private void calculate() {
```
Fig 8. Session Model Code Snippet

```
BigDecimal total = BD_ZERO;
for (final Product product: products) total = total.add(product.getPrice());
                orderTotal = total;
                final BigDecimal salesTaxRate = (zip == null) ? BD_ZERO :
getSalesTaxRate(zip);
                salesTax =
total.multiply(salesTaxRate).round(ROUNDING_POLICY).setScale(2);
                finalTotal = total.add(salesTax).add(deliveryCharge);}
        public final Products getProductsForCategory() {
                return MODEL.queryProducts(COL_CATEGORY_ID,categoryId);}
        public final void addProduct(final String id) {
                final Product product = MODEL.getProduct(id);
                if ((product == null) || products.contains(product)) return;
                products.add(product);
                calculate();
                orderInProgress();}
        public final void deleteProduct(final String id) {
                final Product product = products.removeById(id);
                if (product != null) calculate();
                orderInProgress();}
        public final User getUser() {return user;}
        public final boolean isLoggedIn() {return user != null;}
        public final User login(final String username, final String password) {
                user = MODEL.getUser(username,password); return user;}
        public final User register(final String username, final String password) {
                user = MODEL.registerUser(username,password); return user;}
        public final void setCategoryId(final String id) {
                categoryId = id;}
        public final String getCategoryName() {
return (categoryId == null) ? "Online Shopping Center" :
         MODEL.getCategories().getCategoryName(categoryId);}
        public final Products getProducts() {return products;}

        public final boolean hasProducts() {return !products.isEmpty();}
        public final boolean isLoggedInAndHasProducts() {
                return isLoggedIn() && hasProducts();}
        public final String getZip() {return zip;}
        public final void setZip(final String z) {zip = z; calculate();}
        public final BigDecimal getSalesTax() {return salesTax;}
        public final BigDecimal getDeliveryCharge() {return deliveryCharge;}
        public final void setDeliveryCharge(final BigDecimal charge) {
                deliveryCharge = charge; calculate();}
        public final BigDecimal getOrderTotal() {return orderTotal;}
        public final BigDecimal getFinalTotal() {return finalTotal;}
```

Fig 8. Session Model Code Snippet (continued)

```
	public void reinitialize() {
		products.clear();
		setZip(null);
		salesTax = BD_ZERO;
		deliveryCharge = BD_ZERO;
		finalTotal = BD_ZERO;
		setOrderPlaced(true);
	}

	public boolean isOrderPlaced() {
		return orderPlaced;
	}

	private void setOrderPlaced(final boolean op) {
		this.orderPlaced = op;
	}

	private void orderInProgress() {
		setOrderPlaced(false);
	}
}
```

Fig. 8. Session Model Code Snippet (continued)

3.4. The Shopping Cart Application Interface

This section describes the different interfaces for the online shopping-cart application. It contains a detailed description about each interface along with a screen shot of the interface.

1. **Home Page:** The home page of the application (Figure 9) is common to all the system users/administrators. This interface is available through the web application. At the time of logging into the home page, the page shows the categories available for shopping. By default, there are six categories:

    a. Electronics

    b. Clothing

c. Jewelry

d. Shoes

e. Furniture

f. Entertainment

Each category links to an individual page containing the items related to the category to which it is assigned.

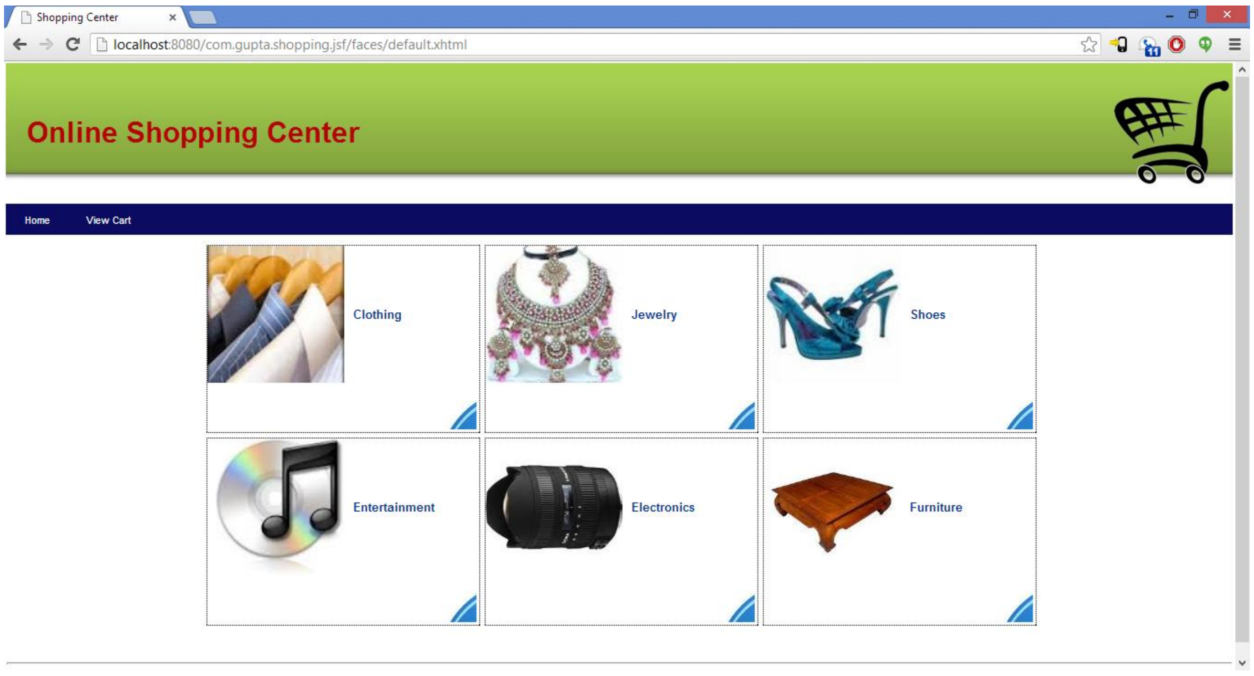
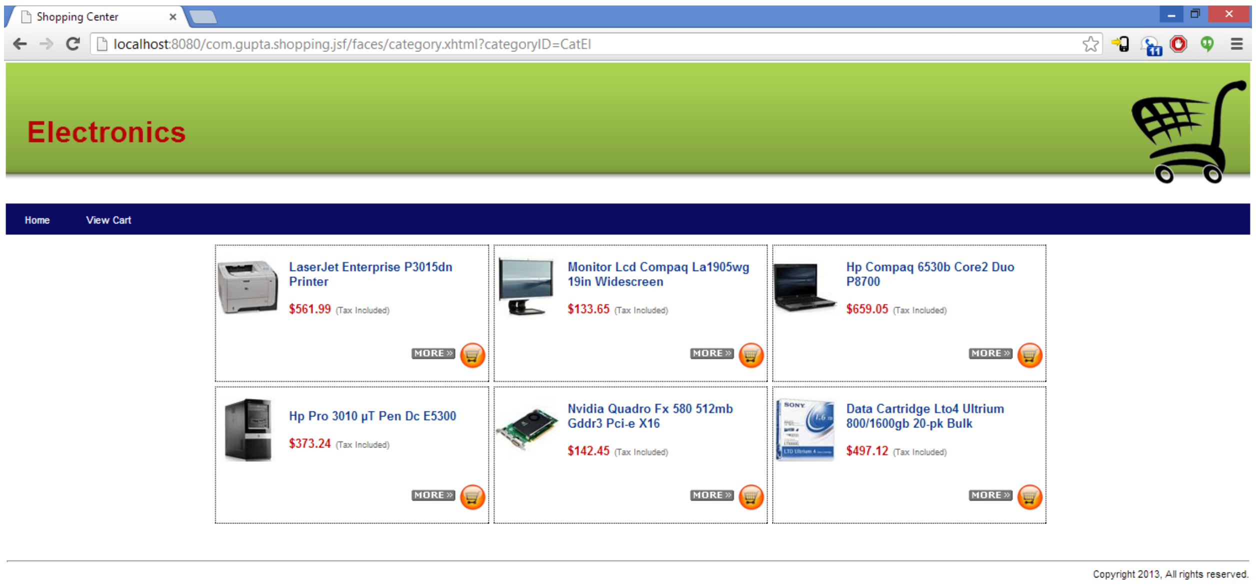


Fig. 9. Screenshot of the Home Page.

2. **Items Interface:** This interface links all categories to an individual page that contains items related to a particular category. Inside each category page, there are items which can be browsed and added to the shopping cart. There is a button/link back to the home page that would return the user to the application's main page, showing the categories, and a user can then choose to browse another category or directly view the cart to check out. There are six categories on the main page, and each category contains six items in the user interface that

are similar to the main page. A similar page structure is used for both the home page and the items page. There is a separate page for each category's items; the pages are joined using Java-server facets. The design and colors for the user interface are implemented by CSS and HTML tags embedded in JavaScript. The screenshots demonstrate items on each page of the category. The size of each image on the items page is a thumbnail (70 x 70 pixels). The size of an image on the description page is fuller (150 x 150 pixels).

3. **Electronics Category Item Interface:** The screenshot below shows items in the electronics category. There are six items listed individually with a price. There is also a "more" button that allows users to obtain additional information about the item. The little shopping cart icon next to "more" link adds the items to the shopping cart.


Shopping Center
localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatEl
Home
View Cart
MORE »

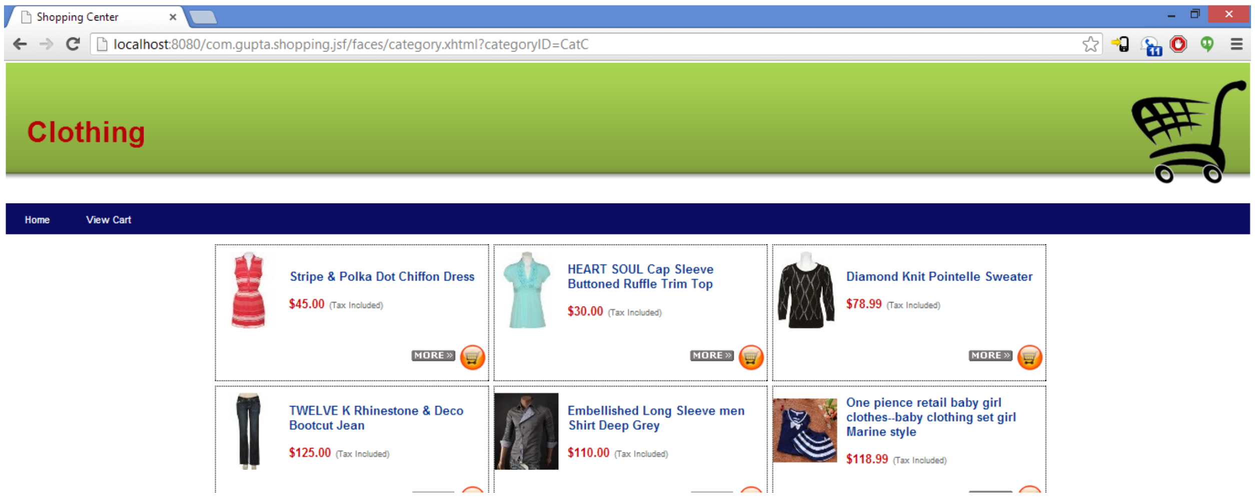
Electronics

LaserJet Enterprise P3015dn Printer $561.99 (Tax Included)	Monitor Lcd Compaq La1905wg 19in Widescreen $133.65 (Tax Included)	Hp Compaq 6530b Core2 Duo P8700 $659.05 (Tax Included)
Hp Pro 3010 µT Pen Dc E5300 $373.24 (Tax Included)	Nvidia Quadro Fx 580 512mb Gddr3 Pci-e X16 $142.45 (Tax Included)	Data Cartridge Lto4 Ultrium 800/1600gb 20-pk Bulk $497.12 (Tax Included)

Copyright 2013, All rights reserved.

Fig. 10. Screenshot of the Electronics Category.

a. **Clothing Category Item Interface:** Figure 11 is the screenshot of the clothing category that displays six clothing items.



localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatC

Clothing

Stripe & Polka Dot Chiffon Dress
$45.00 (Tax Included)

HEART SOUL Cap Sleeve Buttoned Ruffle Trim Top
$30.00 (Tax Included)

Diamond Knit Pointelle Sweater
$78.99 (Tax Included)

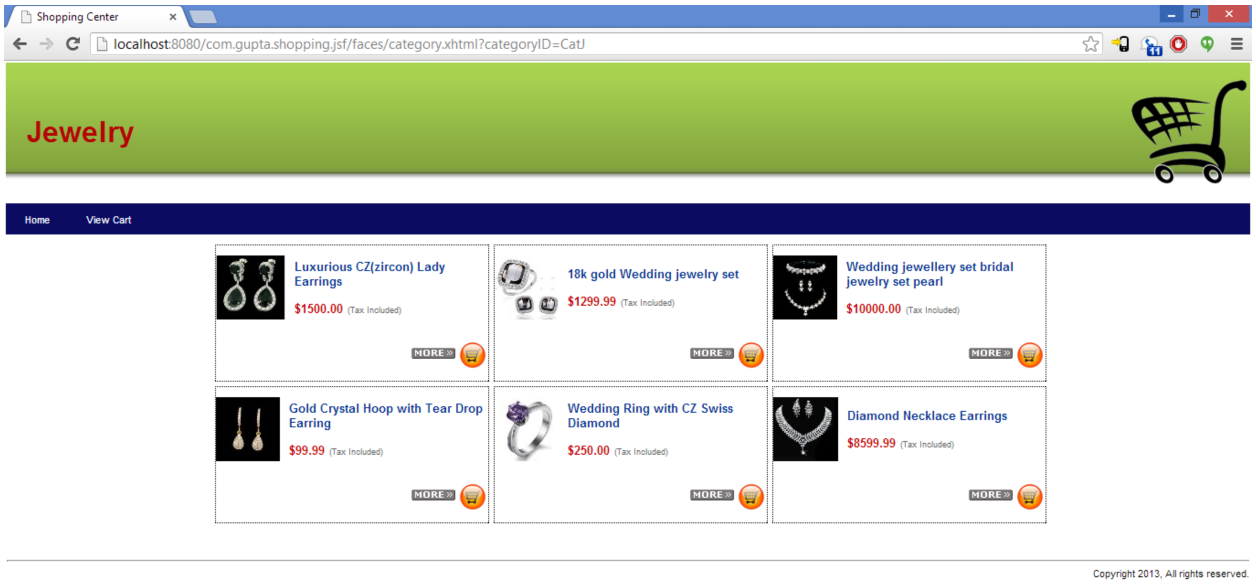
TWELVE K Rhinestone & Deco Bootcut Jean
$125.00 (Tax Included)

Embellished Long Sleeve men Shirt Deep Grey
$110.00 (Tax Included)

One pience retail baby girl clothes--baby clothing set girl Marine style
$118.99 (Tax Included)

Fig. 11. Screenshot of the Clothing Category.

b. **Jewelry Category Interface:** Figure 12 is the screenshot of the jewelry category that displays six jewelry items.



localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatJ

Jewelry

Luxurious CZ(zircon) Lady Earrings
$1500.00 (Tax Included)

18k gold Wedding jewelry set
$1299.99 (Tax Included)

Wedding jewellery set bridal jewelry set pearl
$10000.00 (Tax Included)

Gold Crystal Hoop with Tear Drop Earring
$99.99 (Tax Included)

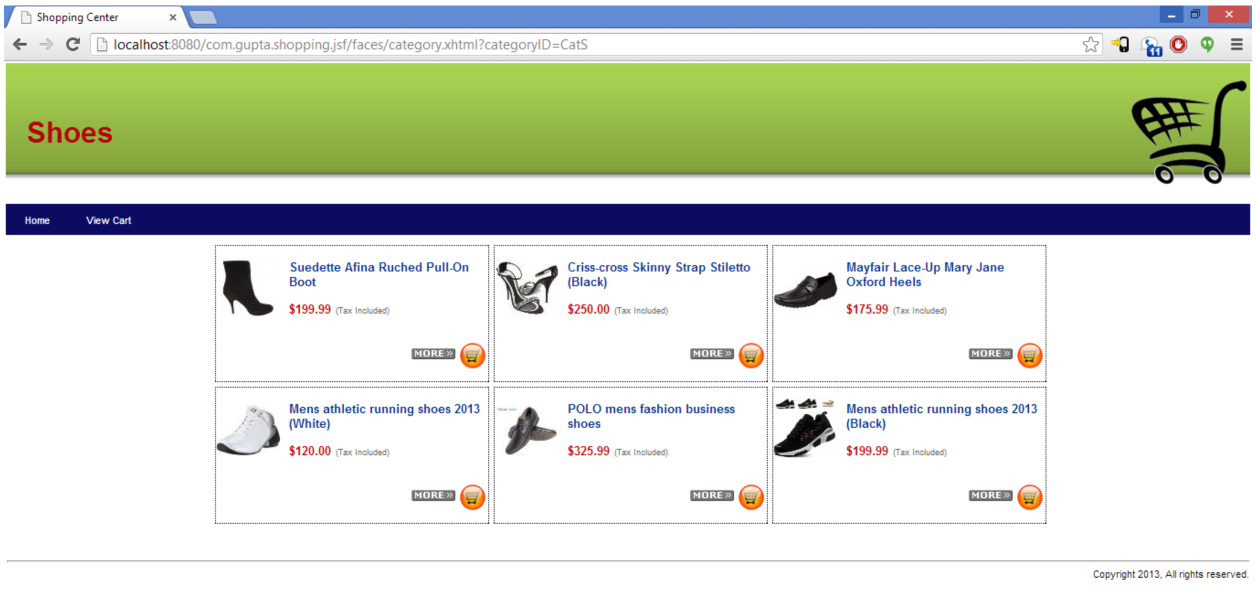
Wedding Ring with CZ Swiss Diamond
$250.00 (Tax Included)

Diamond Necklace Earrings
$8599.99 (Tax Included)

Copyright 2013, All rights reserved.

Fig. 12. Screenshot of the Jewelry Category.

c. **Shoes Category Interface:** Figure 13 is the screenshot of the shoes category that displays six shoe items.



localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatS

Shoes

Suedette Afina Ruched Pull-On Boot
$199.99 (Tax Included)

Criss-cross Skinny Strap Stiletto (Black)
$250.00 (Tax Included)

Mayfair Lace-Up Mary Jane Oxford Heels
$175.99 (Tax Included)

Mens athletic running shoes 2013 (White)
$120.00 (Tax Included)

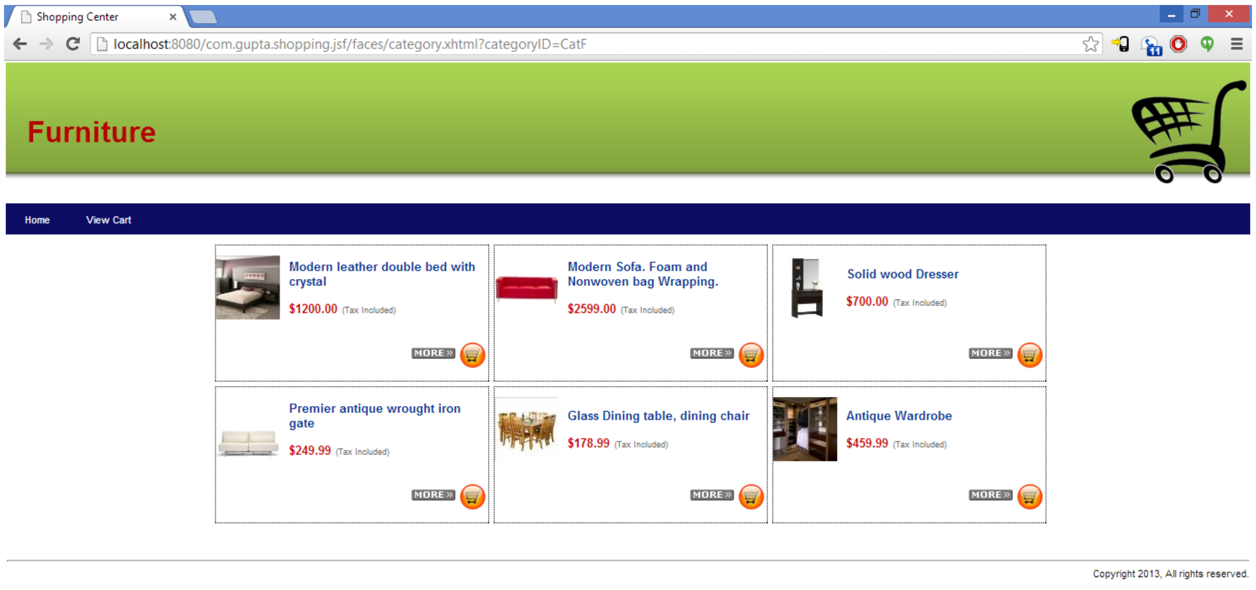
POLO mens fashion business shoes
$325.99 (Tax Included)

Mens athletic running shoes 2013 (Black)
$199.99 (Tax Included)

Copyright 2013, All rights reserved.

Fig. 13. Screenshot of the Shoes Category.

d. Furniture Category Interface: Figure 14 is the screenshot of the furniture category that displays six furniture items.



localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatF

Furniture

Modern leather double bed with crystal
$1200.00 (Tax Included)

Modern Sofa. Foam and Nonwoven bag Wrapping.
$2599.00 (Tax Included)

Solid wood Dresser
$700.00 (Tax Included)

Premier antique wrought iron gate
$249.99 (Tax Included)

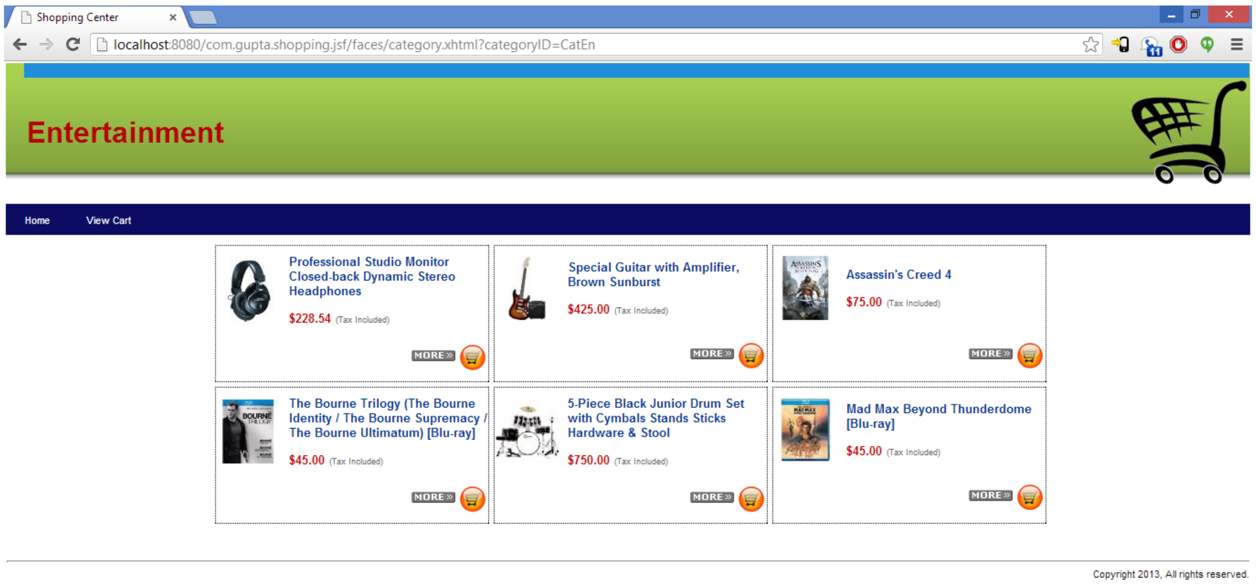
Glass Dining table, dining chair
$178.99 (Tax Included)

Antique Wardrobe
$459.99 (Tax Included)

Copyright 2013, All rights reserved.

Fig. 14. Screenshot of the Furniture Category.

e. **Entertainment Category Interface:** Figure 15 is the screenshot of the entertainment category and displays six entertainment items.


Shopping Center
localhost:8080/com.gupta.shopping.jsf/faces/category.xhtml?categoryID=CatEn
Entertainment
Home
View Cart
MORE »
Copyright 2013, All rights reserved.

Professional Studio Monitor Closed-back Dynamic Stereo Headphones
$228.54 (Tax Included)

Special Guitar with Amplifier, Brown Sunburst
$425.00 (Tax Included)

Assassin's Creed 4
$75.00 (Tax Included)

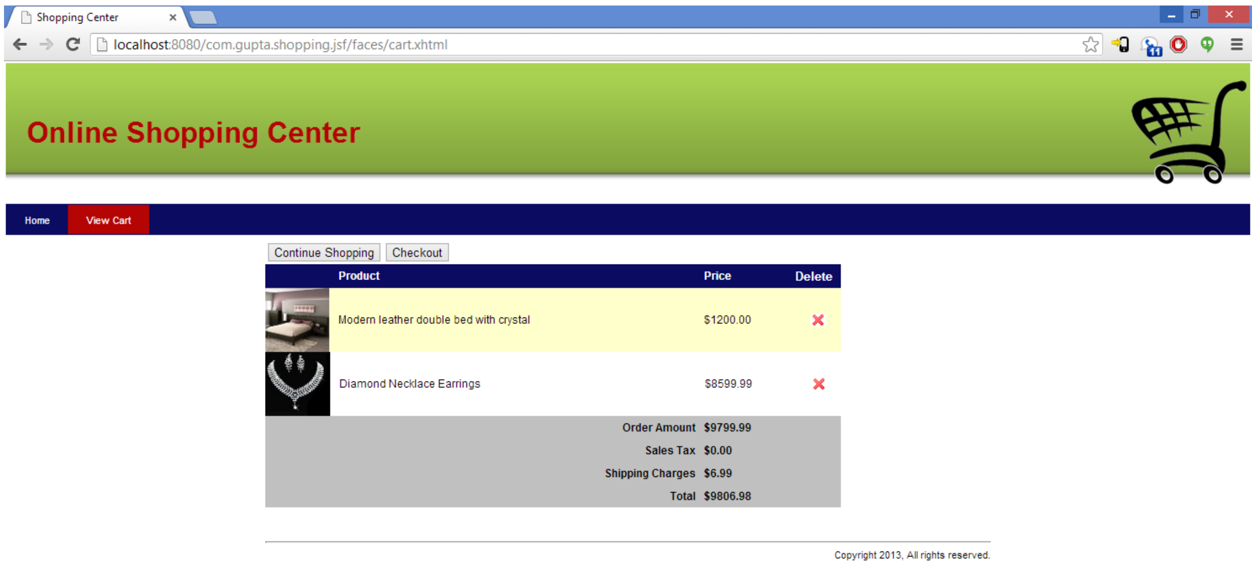
The Bourne Trilogy (The Bourne Identity / The Bourne Supremacy / The Bourne Ultimatum) [Blu-ray]
$45.00 (Tax Included)

5-Piece Black Junior Drum Set with Cymbals Stands Sticks Hardware & Stool
$750.00 (Tax Included)

Mad Max Beyond Thunderdome [Blu-ray]
$45.00 (Tax Included)

Fig. 15. Screenshot of the Entertainment Category.

4. **View Cart Interface:** In the view-cart interface, both users and the administrator can access items that were added to the cart (Figure 16). They can also access to view the empty cart. Inside the cart interface, the user/administrator has the option to update the cart by either deleting items from list of selected items or by adding items to the list by clicking the continue shopping button. Both the user and administrator also have the option to check out items in the cart by clicking the checkout button which takes them to the order-form page interface.

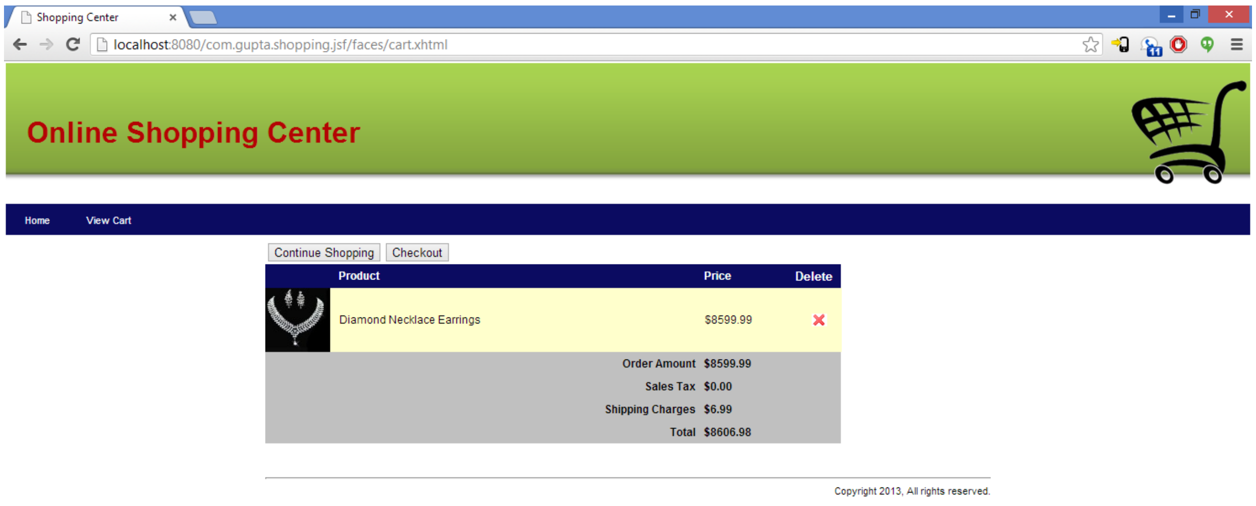


	Product	Price	Delete
	Modern leather double bed with crystal	$1200.00	✗
	Diamond Necklace Earrings	$8599.99	✗
	Order Amount	$9799.99	
	Sales Tax	$0.00	
	Shipping Charges	$6.99	
	Total	$9806.98	

Copyright 2013, All rights reserved.

Fig. 16. Screenshot of the View Cart.

Figure 17 displays the screenshot of the shopping cart after deleting the item from the cart. Once a user deletes an item from the cart, the cart updates/refreshes itself with the newer number of items in the view-cart interface.

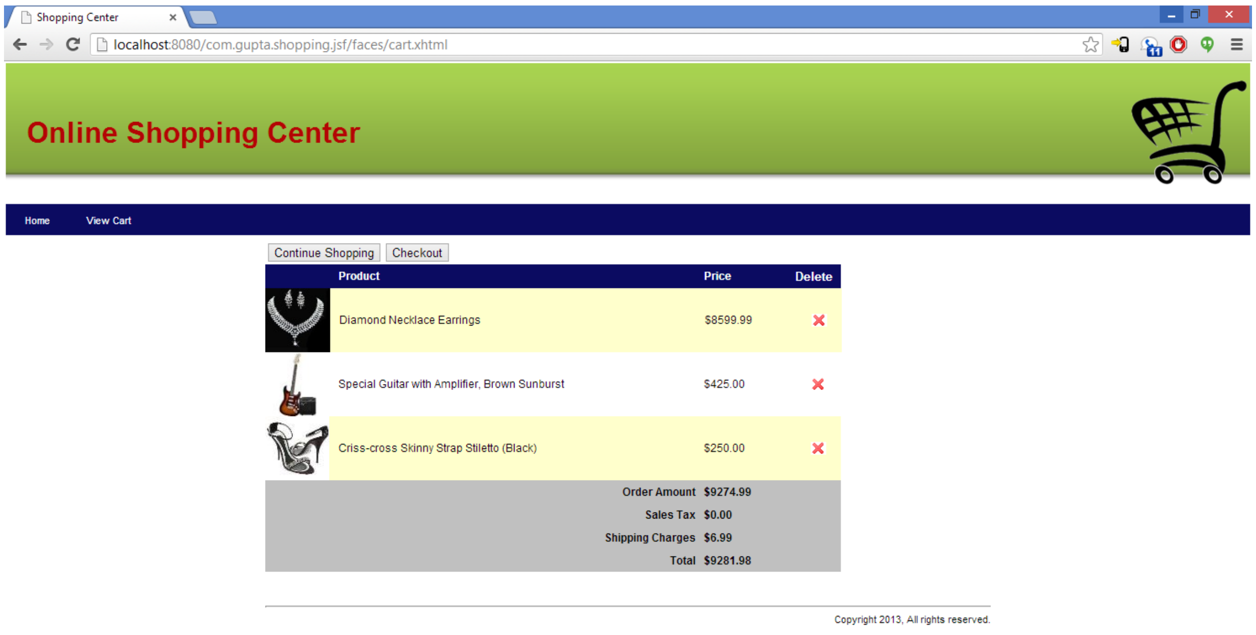


Product	Price	Delete
Diamond Necklace Earrings	$8599.99	✕
Order Amount	$8599.99	
Sales Tax	$0.00	
Shipping Charges	$6.99	
Total	$8606.98	

Copyright 2013, All rights reserved.

Fig. 17. Screenshot of the Updated Cart After Deleting an Item.

Figure 18 displays the screenshot of the shopping cart after adding the item. The user can view the updated shopping cart when any item is added to the cart by using the shopping-cart icon.



Product	Price	Delete
Diamond Necklace Earrings	$8599.99	✕
Special Guitar with Amplifier, Brown Sunburst	$425.00	✕
Criss-cross Skinny Strap Stiletto (Black)	$250.00	✕
Order Amount	$9274.99	
Sales Tax	$0.00	
Shipping Charges	$6.99	
Total	$9281.98	

Copyright 2013, All rights reserved.

Fig. 18. Screenshot of the Updated Cart After Adding an Item.

5. **Checkout Interface:** The checkout button is visible to the user only when there are items in the shopping cart. If there are no items in the shopping cart, the checkout button disappears (Figure 19). If there is at least one item in the shopping cart, then the checkout button is visible to the user, and the user can check out.

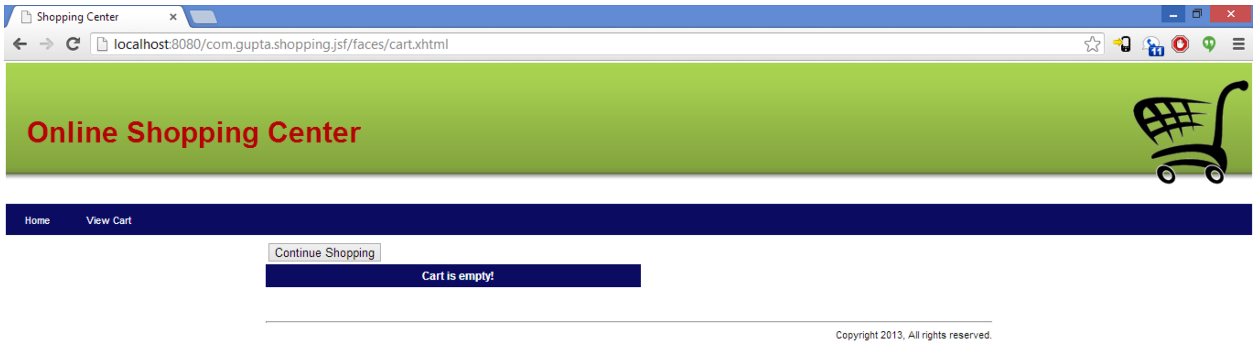
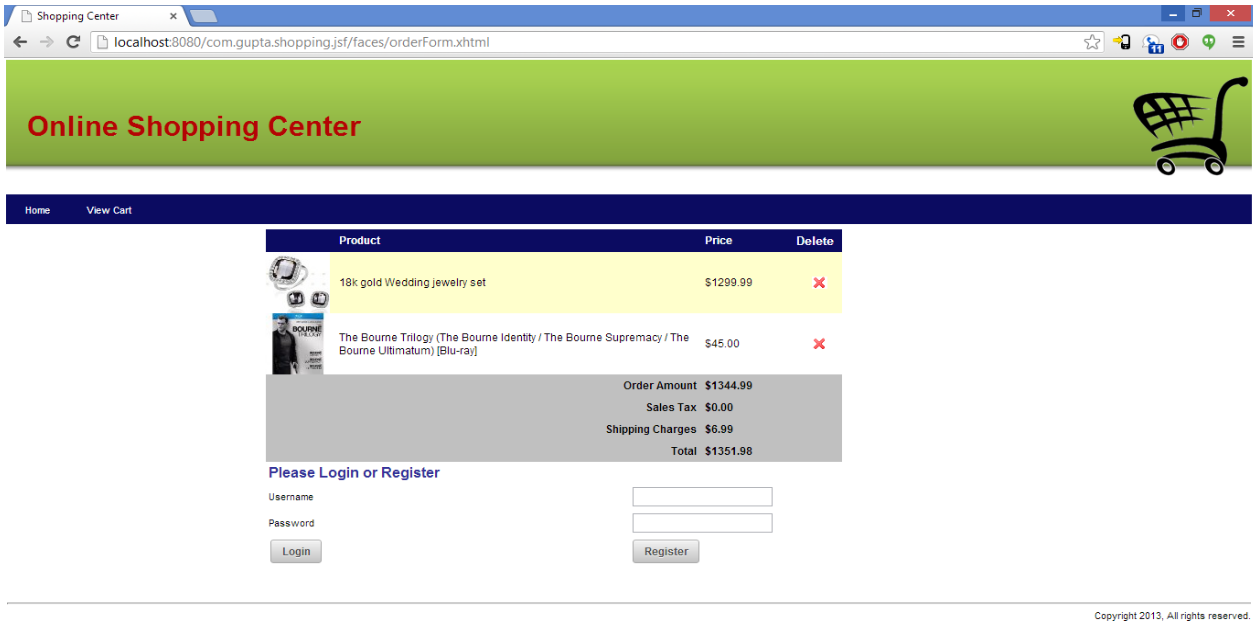

Shopping Center
localhost:8080/com.gupta.shopping.jsf/faces/cart.xhtml
Online Shopping Center
Home
View Cart
Continue Shopping
Cart is empty!
Copyright 2013, All rights reserved.

Fig. 19. Screenshot of the Empty Cart: Checkout Button Disappears.

6. **Login Interface:** The user has to login/register to the user-authentication form by utilizing the username and password once the checkout button is clicked to place an order. The login interface appears when the user clicks the checkout button (Figure 20).



	Product	Price	Delete
	18k gold Wedding jewelry set	$1299.99	✗
	The Bourne Trilogy (The Bourne Identity / The Bourne Supremacy / The Bourne Ultimatum) [Blu-ray]	$45.00	✗
	Order Amount	$1344.99	
	Sales Tax	$0.00	
	Shipping Charges	$6.99	
	Total	$1351.98	

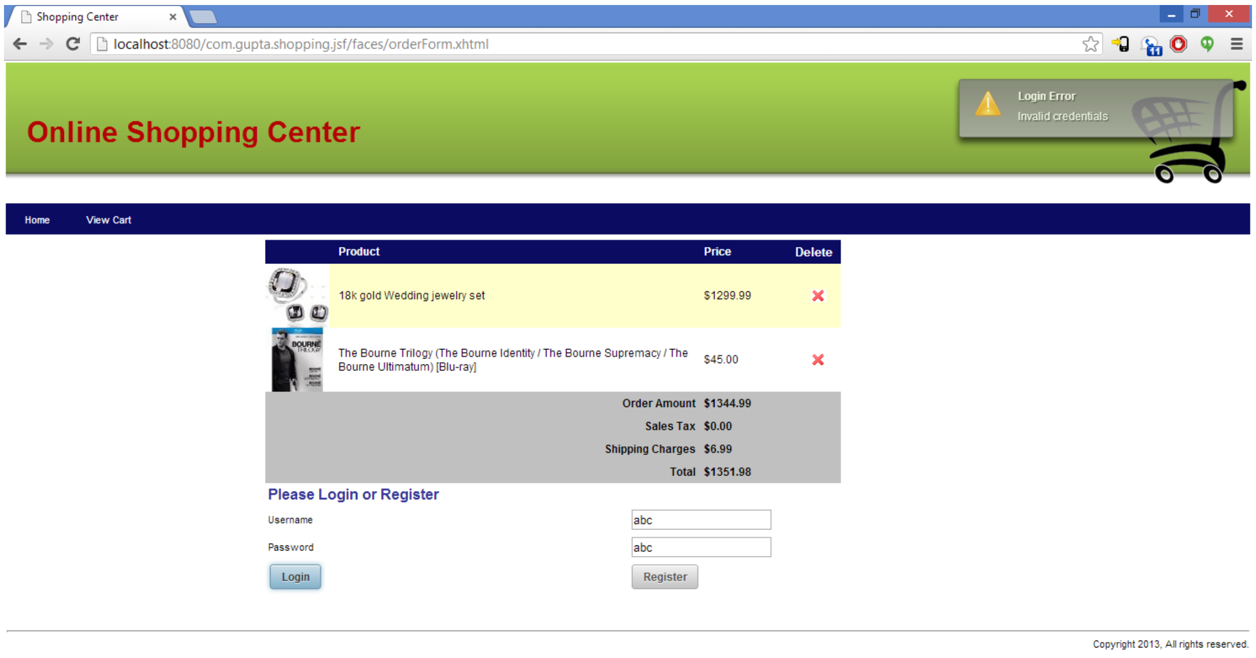
Please Login or Register

Username	
Password	
Login	Register

Copyright 2013, All rights reserved.

Fig. 20. Screenshot of the User-Authentication Form.

Figure 21 displays that, if the user enters invalid information in username and password fields, an error message appears on the top right of the screen, displaying invalid credentials. The error message in the gray box pops up when the user tries to login/register with invalid information and disappears after a few seconds of popping up the error message box.


Shopping Center
localhost:8080/com.gupta.shopping.jsf/faces/orderForm.xhtml
Online Shopping Center
Home
View Cart
Please Login or Register
Login
Register
Copyright 2013, All rights reserved.

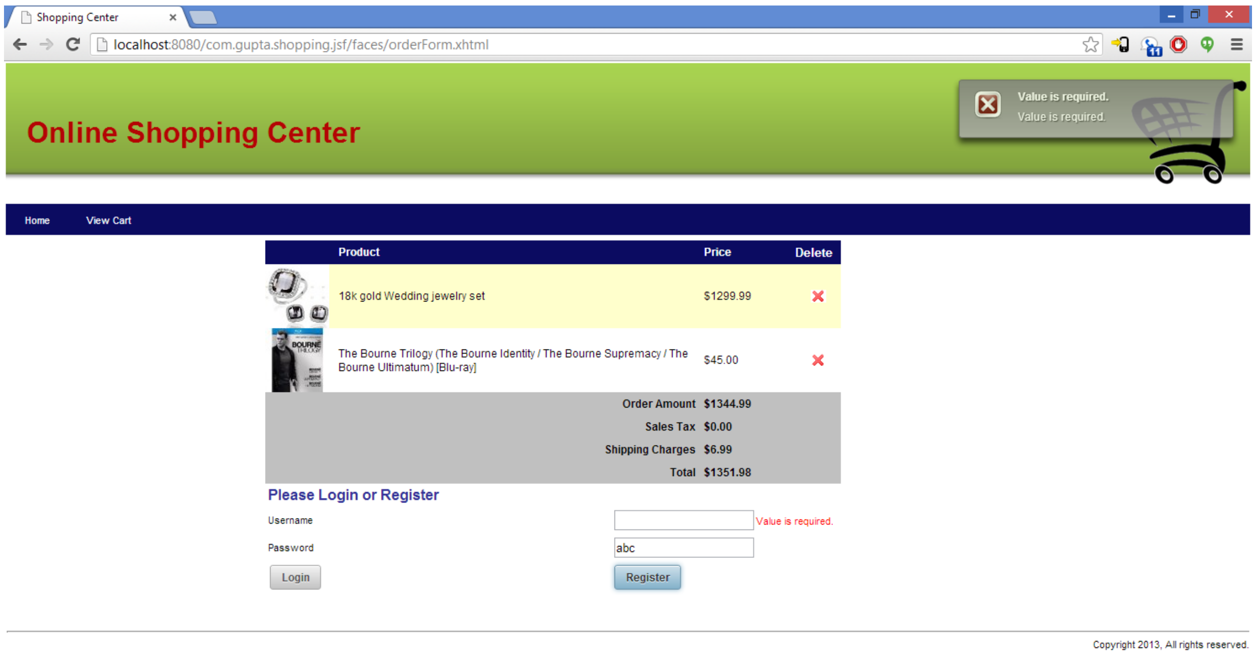
Login Error
Invalid credentials

Product	Price	Delete
18k gold Wedding jewelry set	$1299.99	✗
The Bourne Trilogy (The Bourne Identity / The Bourne Supremacy / The Bourne Ultimatum) [Blu-ray]	$45.00	✗
Order Amount	$1344.99	
Sales Tax	$0.00	
Shipping Charges	$6.99	
Total	$1351.98	

Username	abc
Password	abc

Fig. 21. Screenshot with Invalid Credentials

Figure 22 shows that, if the user enters incomplete information in the username and password fields, an error message appears next to the column, indicating that a value is required. The error message next to the column appears in the red text and stays there until the user does not provide the correct information.


Shopping Center

localhost:8080/com.gupta.shopping.jsf/faces/orderForm.xhtml

Online Shopping Center

Value is required.
Value is required.

	Product	Price	Delete
	18k gold Wedding jewelry set	$1299.99	✗
	The Bourne Trilogy (The Bourne Identity / The Bourne Supremacy / The Bourne Ultimatum) [Blu-ray]	$45.00	✗
	Order Amount	$1344.99	
	Sales Tax	$0.00	
	Shipping Charges	$6.99	
	Total	$1351.98	

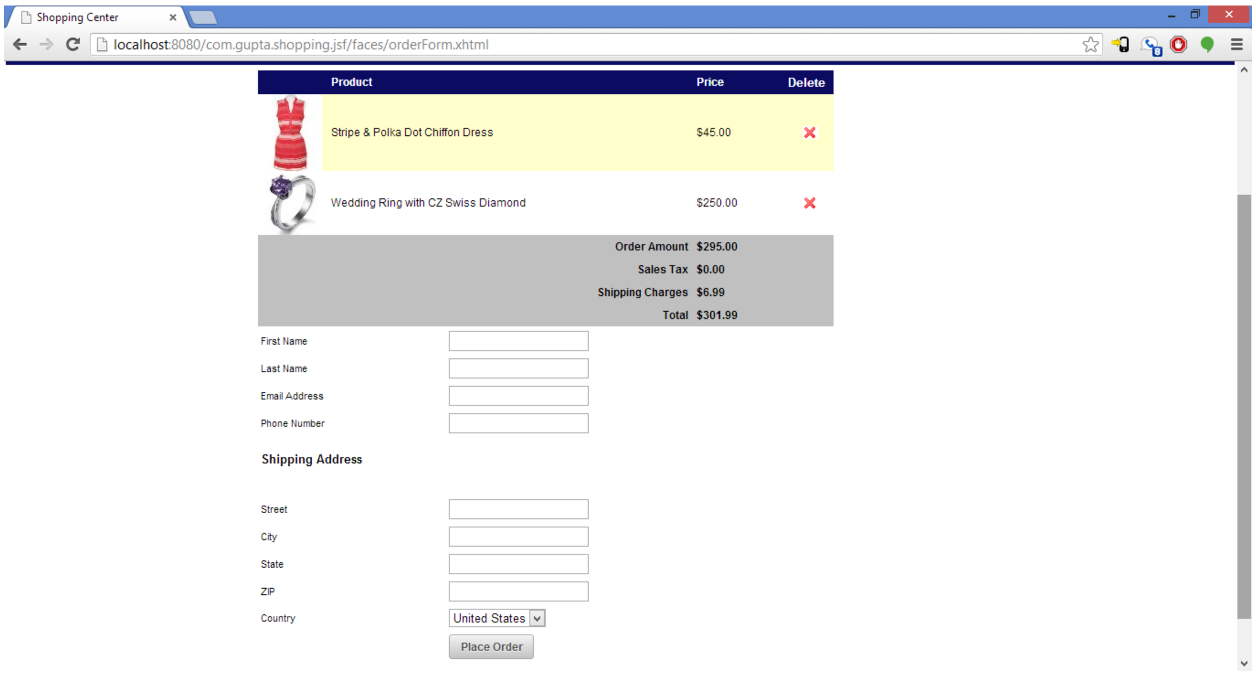
Please Login or Register

Username		Value is required.
Password	abc	

Copyright 2013, All rights reserved.

Fig. 22. Screenshot when the Information is Incomplete.

7. **Order Form Interface:** The order-form page (Figure 23) consists of a form where the user/administrator completes the personal information, such as name, email address, shipping address, and phone number, to check out. After filling in the information, the user places the order by clicking on the place button, bringing him to a final message page indicating that the order was successfully placed.



Product	Price	Delete
Stripe & Polka Dot Chiffon Dress	$45.00	✗
Wedding Ring with CZ Swiss Diamond	$250.00	✗
Order Amount	$295.00	
Sales Tax	$0.00	
Shipping Charges	$6.99	
Total	$301.99	

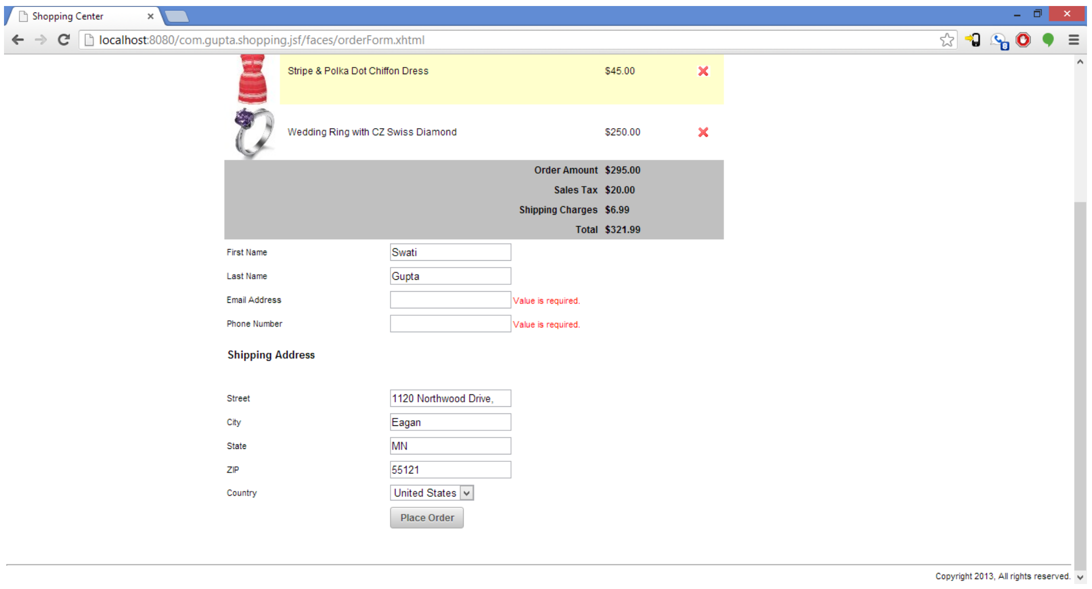
First Name	
Last Name	
Email Address	
Phone Number	

Shipping Address

Street	
City	
State	
ZIP	
Country	United States

Fig. 23. Screenshot of the Order-Form Interface.

Place Order: This interface notifies the user/administrator that the order was successfully placed after the user has submitted the shipping information. The user cannot place an order if the information is invalid or incomplete (Figure 24).


Shopping Center
localhost:8080/com.gupta.shopping.jsf/faces/orderForm.xhtml
Place Order
Copyright 2013, All rights reserved.

Stripe & Polka Dot Chiffon Dress	$45.00	✗
Wedding Ring with CZ Swiss Diamond	$250.00	✗
Order Amount	$295.00	
Sales Tax	$20.00	
Shipping Charges	$6.99	
Total	$321.99	

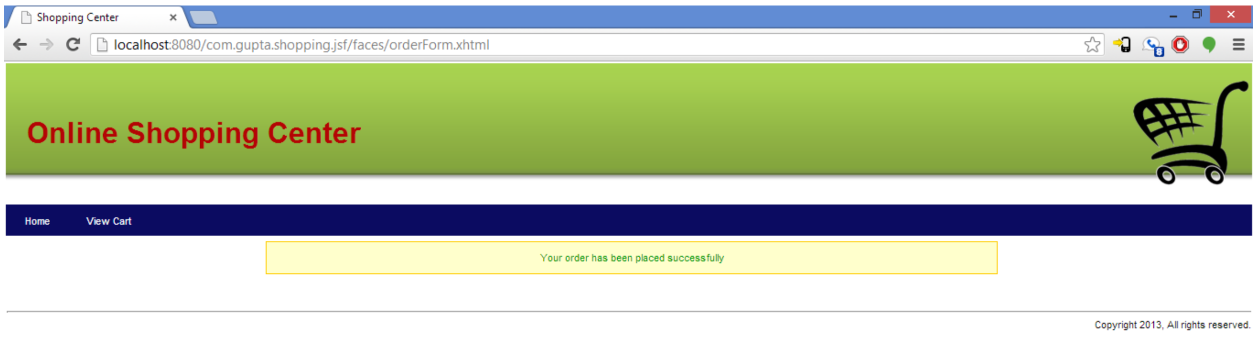
First Name	Swati	
Last Name	Gupta	
Email Address		Value is required.
Phone Number		Value is required.

Shipping Address

Street	1120 Northwood Drive,
City	Eagan
State	MN
ZIP	55121
Country	United States

Fig. 24. Screenshot of the Place-Order Page with Incomplete/Invalid Information.

Figure 25 displays the successful order-confirmation message that “your order has been placed successfully.”

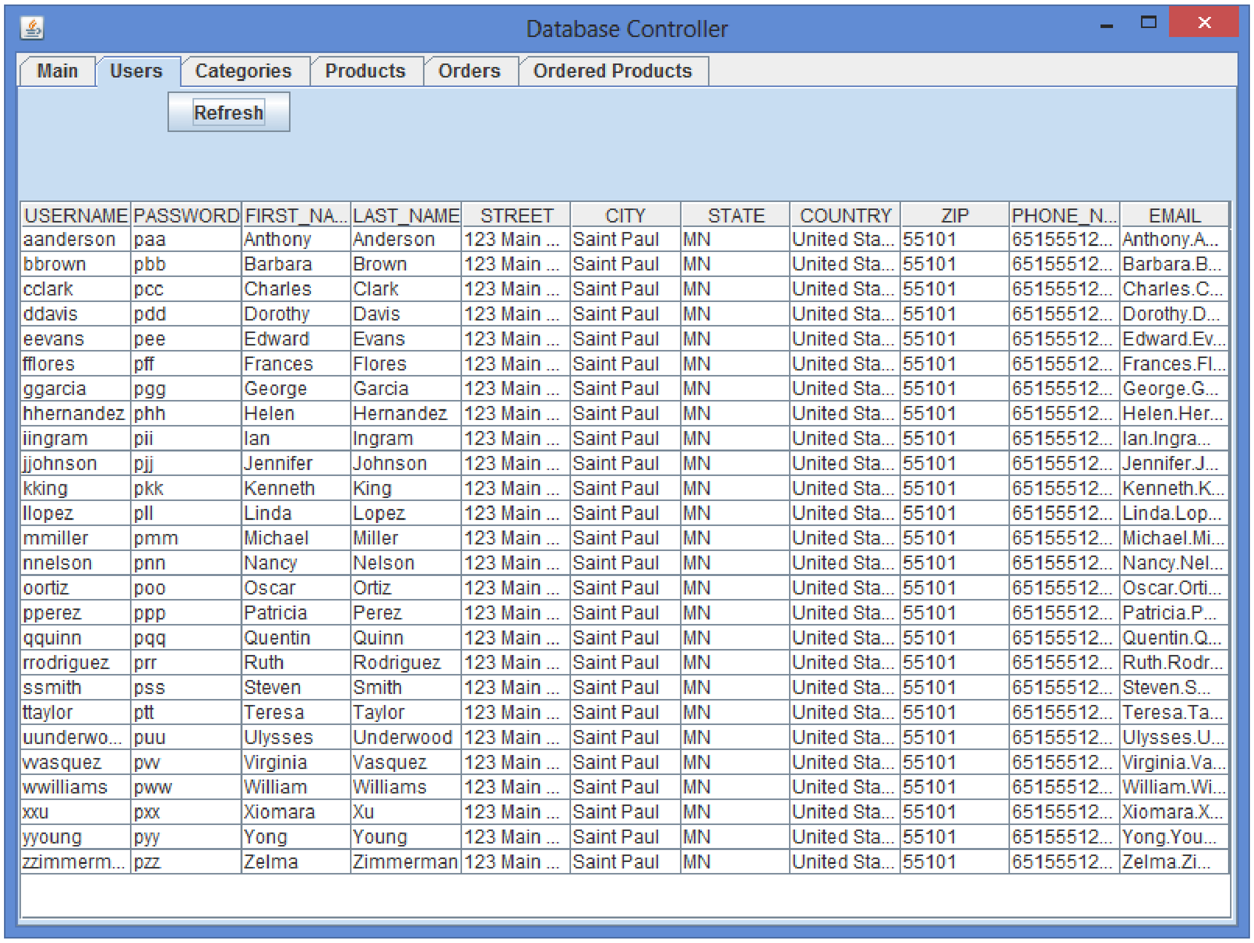


Your order has been placed successfully

Copyright 2013, All rights reserved.

Fig. 25. Screenshot of the Place-Order Success Interface.

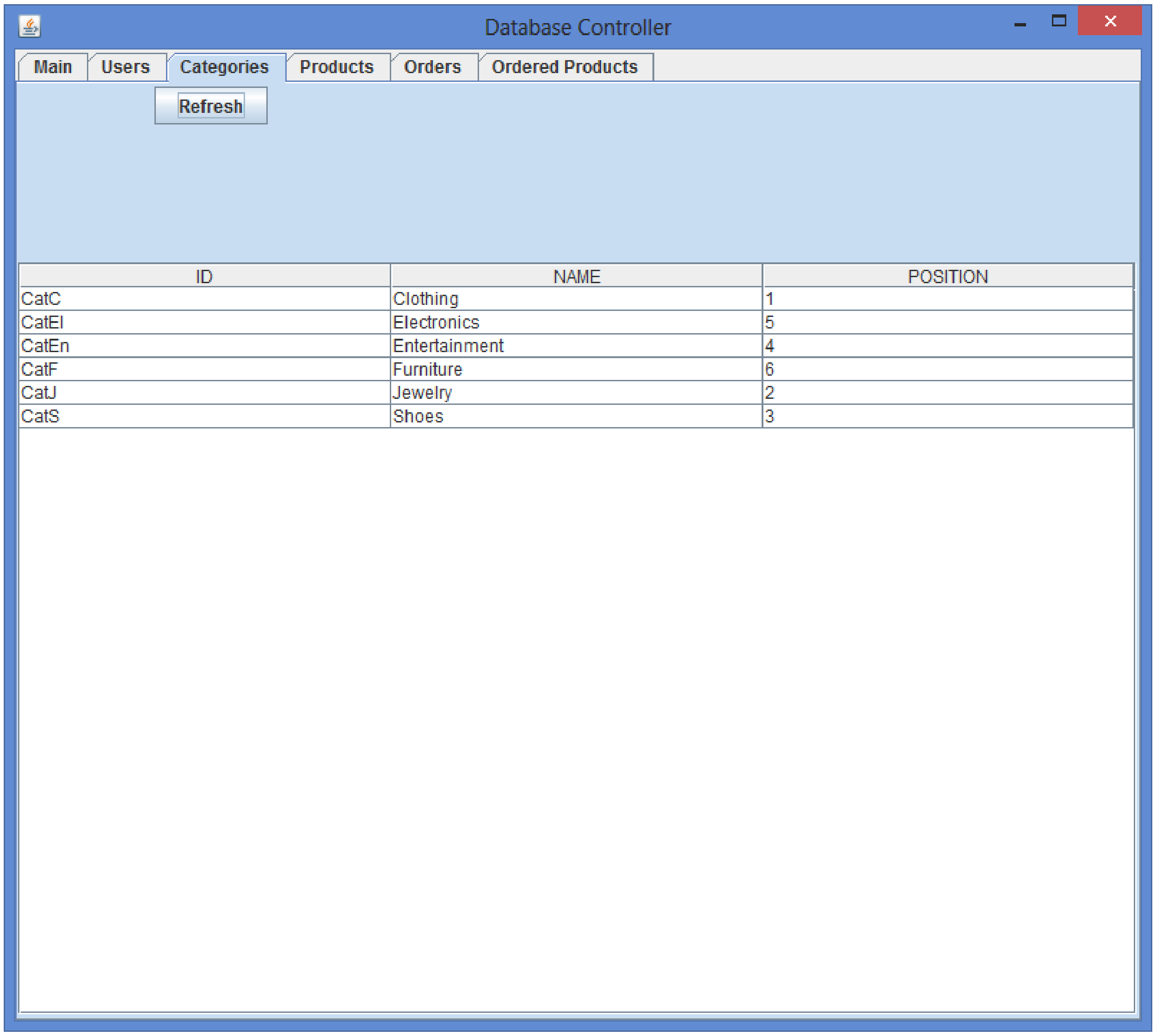
8. **Database Interface:** Using this, the administrator can access information for all users who have placed order of the items in access database table which is automatically updated when the place-order button is clicked. Figure 26 displays the database with the users’ tab that contains the information for all the users that authorizes their information using the user-authentication form. After clicking the refresh button on the users’ tab once a new user successfully registers, that user’s information is updated on the first row in the database table.



USERNAME	PASSWORD	FIRST_NA...	LAST_NAME	STREET	CITY	STATE	COUNTRY	ZIP	PHONE_N...	EMAIL
aanderson	paa	Anthony	Anderson	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Anthony.A...
bbrown	pbb	Barbara	Brown	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Barbara.B...
cclark	pcc	Charles	Clark	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Charles.C...
ddavis	pdd	Dorothy	Davis	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Dorothy.D...
eevans	pee	Edward	Evans	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Edward.Ev...
fflores	pff	Frances	Flores	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Frances.Fl...
ggarcia	pgg	George	Garcia	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	George.G...
hhernandez	phh	Helen	Hernandez	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Helen.Her...
iingram	pii	Ian	Ingram	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Ian.Ingra...
jjohnson	pjj	Jennifer	Johnson	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Jennifer.J...
kking	pkk	Kenneth	King	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Kenneth.K...
llopez	pll	Linda	Lopez	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Linda.Lop...
mmiller	pmm	Michael	Miller	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Michael.Mi...
nnelson	pnn	Nancy	Nelson	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Nancy.Nel...
oortiz	poo	Oscar	Ortiz	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Oscar.Orti...
pperez	ppp	Patricia	Perez	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Patricia.P...
qquinn	pqq	Quentin	Quinn	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Quentin.Q...
rrodriguez	prr	Ruth	Rodriguez	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Ruth.Rodr...
ssmith	pss	Steven	Smith	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Steven.S...
ttaylor	ptt	Teresa	Taylor	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Teresa.Ta...
uunderwo...	puu	Ulysses	Underwood	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Ulysses.U...
wasquez	pw	Virginia	Vasquez	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Virginia.Va...
wwilliams	pww	William	Williams	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	William.Wi...
xxu	pxx	Xiomara	Xu	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Xiomara.X...
yyoung	pyy	Yong	Young	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Yong.You...
zzimmerm...	pzz	Zelma	Zimmerman	123 Main ...	Saint Paul	MN	United Sta...	55101	65155512...	Zelma.Zi...

Fig. 26. Screenshot of the Database: Users.

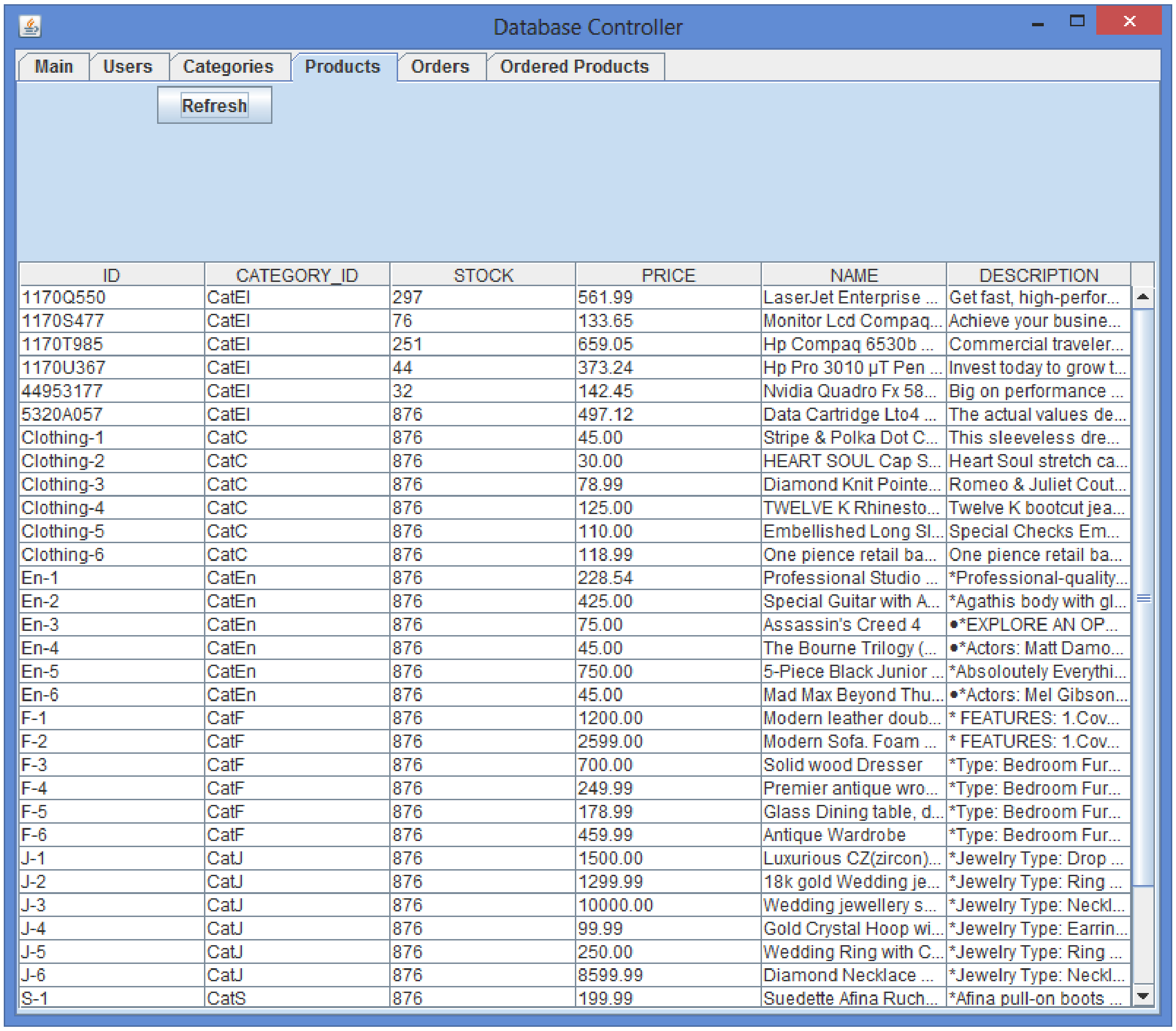
Figure 27 shows the categories tab which currently displays all six categories used in the shopping-cart application. If the administrator is to add/update categories for the application, then refreshing the page would also update the number of categories.



ID	NAME	POSITION
CatC	Clothing	1
CatEl	Electronics	5
CatEn	Entertainment	4
CatF	Furniture	6
CatJ	Jewelry	2
CatS	Shoes	3

Fig. 27. Screenshot of the Database: Categories.

Figure 28 displays all the products that are available in the shopping-cart application. Items from different categories can be identified by their ID and Category_ID. This product table also lists the total number of items in stock, the price, the name, and the description of each item.



ID	CATEGORY_ID	STOCK	PRICE	NAME	DESCRIPTION
1170Q550	CatEl	297	561.99	LaserJet Enterprise ...	Get fast, high-perfor...
1170S477	CatEl	76	133.65	Monitor Lcd Compaq...	Achieve your busine...
1170T985	CatEl	251	659.05	Hp Compaq 6530b ...	Commercial traveler...
1170U367	CatEl	44	373.24	Hp Pro 3010 μT Pen ...	Invest today to grow t...
44953177	CatEl	32	142.45	Nvidia Quadro Fx 58...	Big on performance ...
5320A057	CatEl	876	497.12	Data Cartridge Lto4 ...	The actual values de...
Clothing-1	CatC	876	45.00	Stripe & Polka Dot C...	This sleeveless dre...
Clothing-2	CatC	876	30.00	HEART SOUL Cap S...	Heart Soul stretch ca...
Clothing-3	CatC	876	78.99	Diamond Knit Pointe...	Romeo & Juliet Cout...
Clothing-4	CatC	876	125.00	TWELVE K Rhinesto...	Twelve K bootcut jea...
Clothing-5	CatC	876	110.00	Embellished Long Sl...	Special Checks Em...
Clothing-6	CatC	876	118.99	One pience retail ba...	One pience retail ba...
En-1	CatEn	876	228.54	Professional Studio ...	*Professional-quality...
En-2	CatEn	876	425.00	Special Guitar with A...	*Agathis body with gl...
En-3	CatEn	876	75.00	Assassin's Creed 4	•*EXPLORE AN OP...
En-4	CatEn	876	45.00	The Bourne Trilogy (...	•*Actors: Matt Damo...
En-5	CatEn	876	750.00	5-Piece Black Junior ...	*Absoloutely Everythi...
En-6	CatEn	876	45.00	Mad Max Beyond Thu...	•*Actors: Mel Gibson...
F-1	CatF	876	1200.00	Modern leather doub...	* FEATURES: 1.Cov...
F-2	CatF	876	2599.00	Modern Sofa. Foam ...	* FEATURES: 1.Cov...
F-3	CatF	876	700.00	Solid wood Dresser	*Type: Bedroom Fur...
F-4	CatF	876	249.99	Premier antique wro...	*Type: Bedroom Fur...
F-5	CatF	876	178.99	Glass Dining table, d...	*Type: Bedroom Fur...
F-6	CatF	876	459.99	Antique Wardrobe	*Type: Bedroom Fur...
J-1	CatJ	876	1500.00	Luxurious CZ(zircon)...	*Jewelry Type: Drop ...
J-2	CatJ	876	1299.99	18k gold Wedding je...	*Jewelry Type: Ring ...
J-3	CatJ	876	10000.00	Wedding jewellery s...	*Jewelry Type: Neckl...
J-4	CatJ	876	99.99	Gold Crystal Hoop wi...	*Jewelry Type: Earrin...
J-5	CatJ	876	250.00	Wedding Ring with C...	*Jewelry Type: Ring ...
J-6	CatJ	876	8599.99	Diamond Necklace ...	*Jewelry Type: Neckl...
S-1	CatS	876	199.99	Suedette Afina Ruch...	*Afina pull-on boots ...

Fig. 28. Screenshot of the Database: Products.

Figure 29 displays the order information, including all the user information that was completed in the order form before the final order was placed. Right now, the database is empty, implying that no user has placed an order.

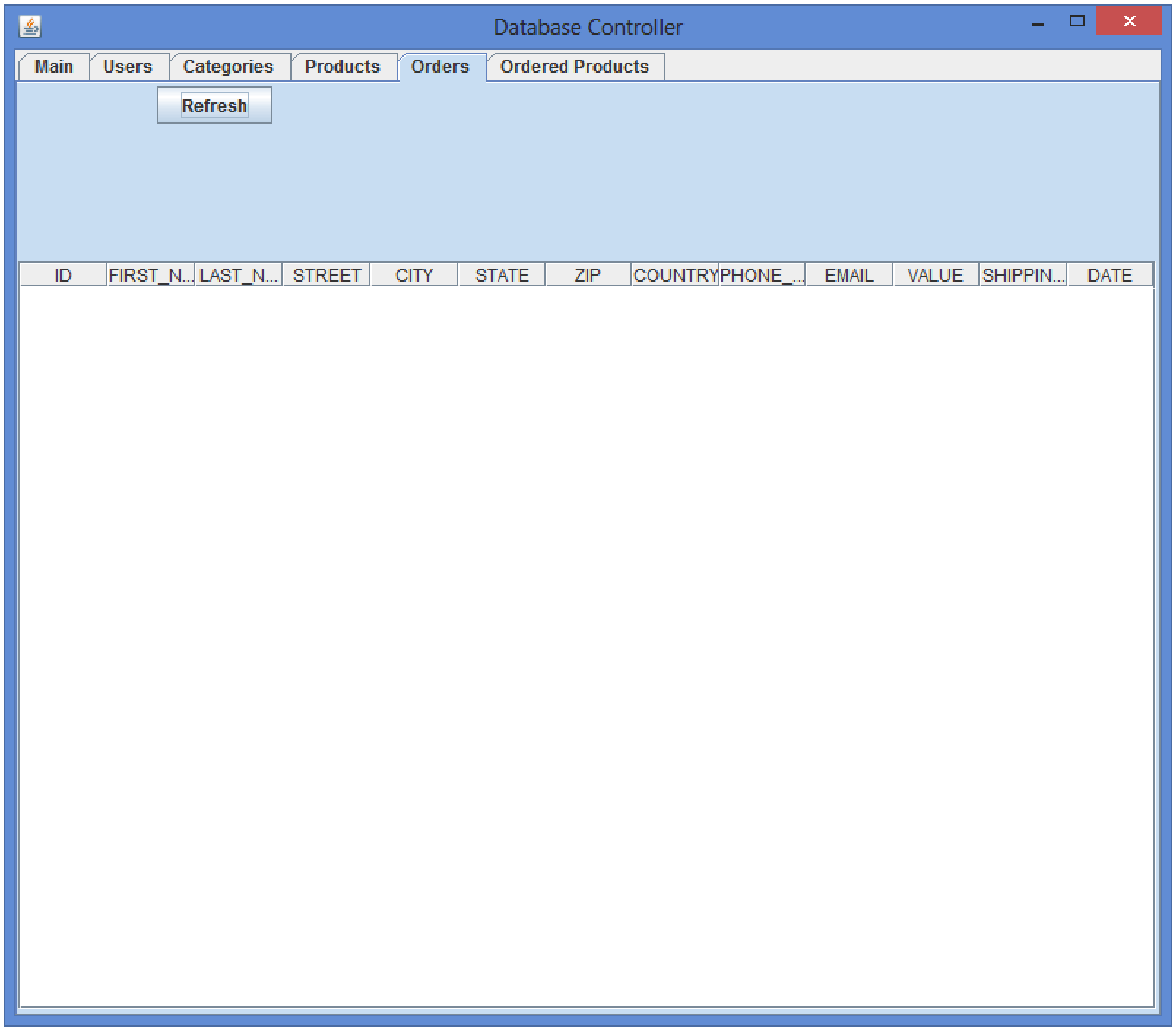
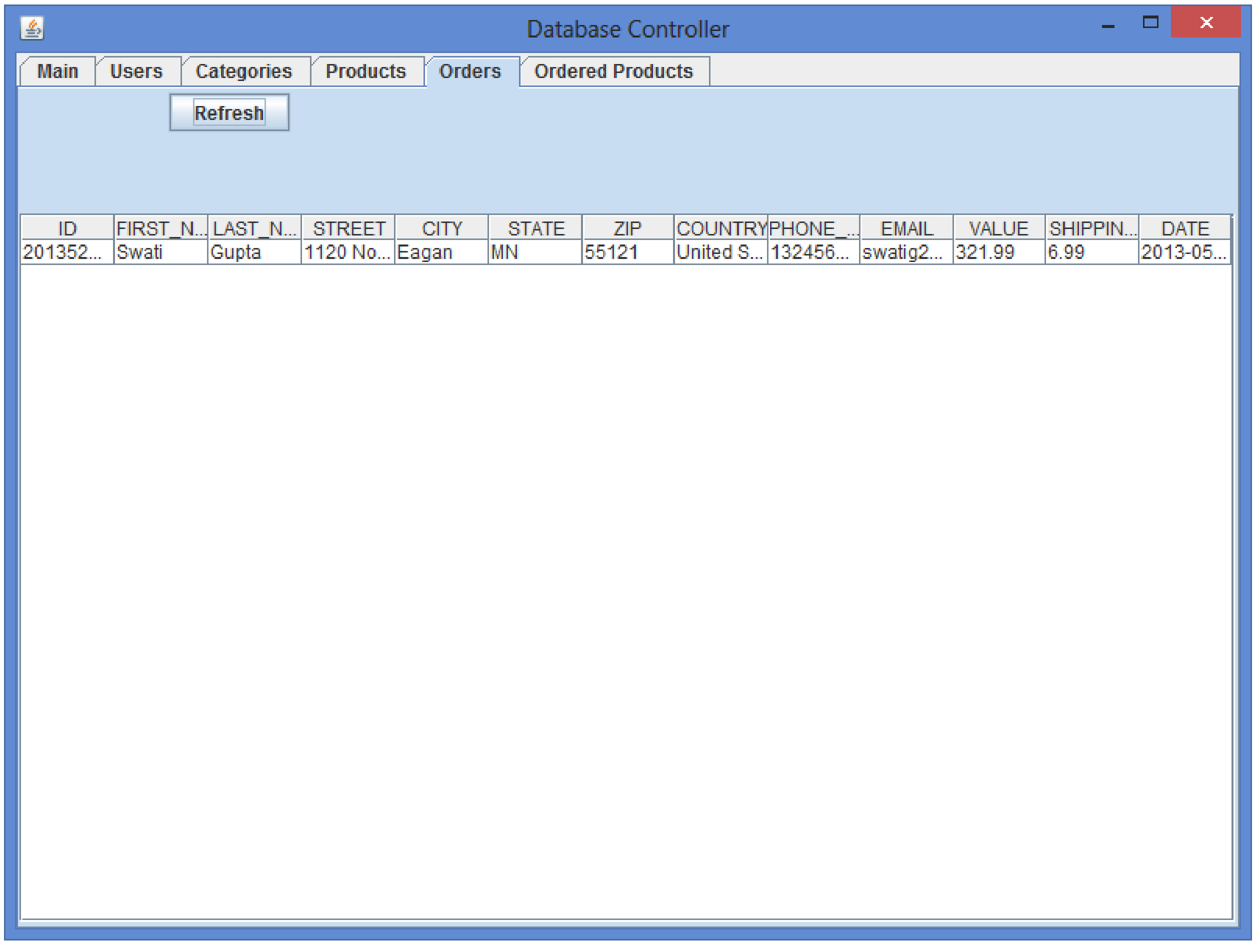


Fig. 29. Screenshot of the Database: Orders/Ordered Products Before Placing Order.

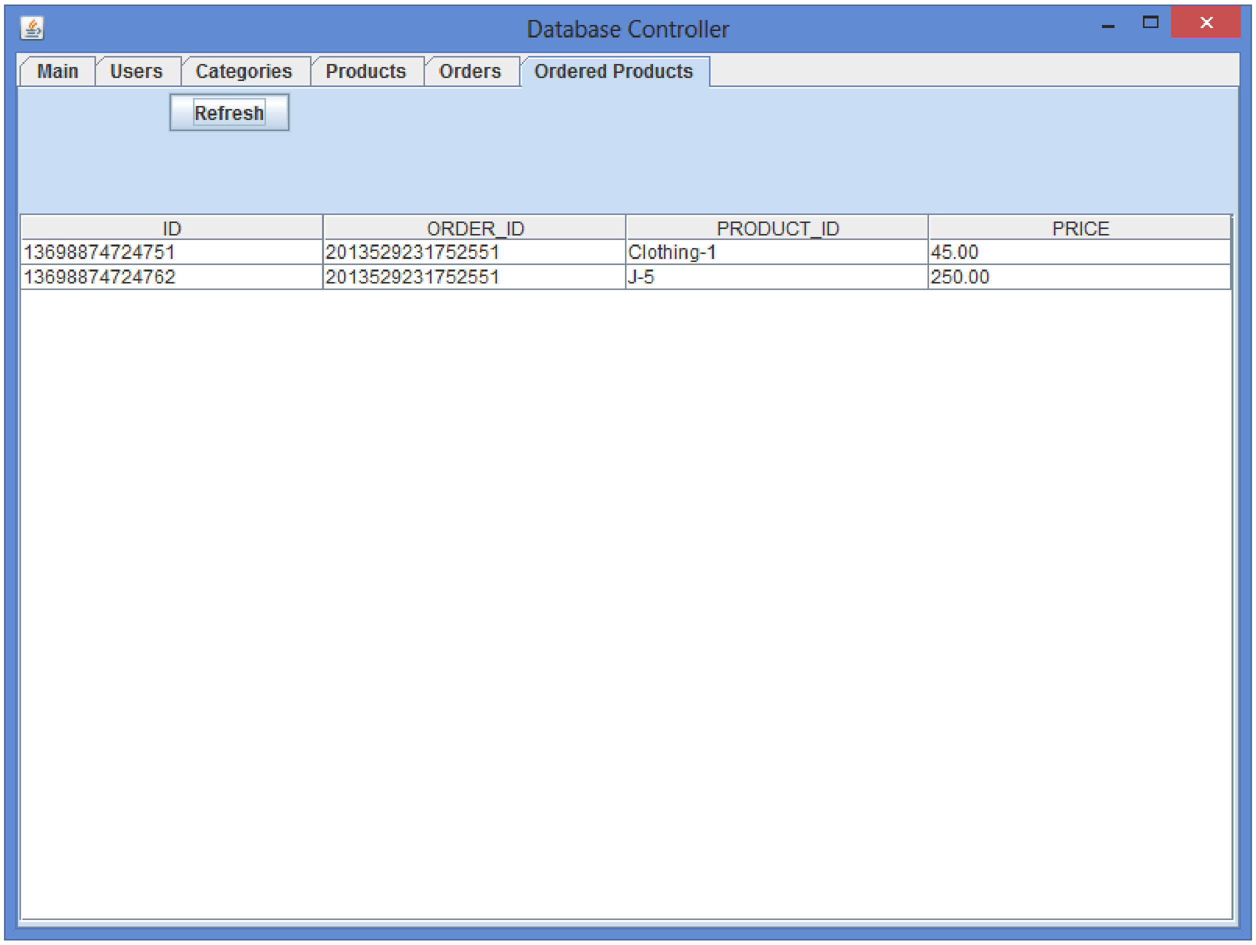
Figure 30 displays informaiton about the user once the order is successfully placed. All the columns are automatically updated with the appropriate informaton that was used on the order form in appropriate columns.



ID	FIRST_N...	LAST_N...	STREET	CITY	STATE	ZIP	COUNTRY	PHONE_...	EMAIL	VALUE	SHIPPIN...	DATE
201352...	Swati	Gupta	1120 No...	Eagan	MN	55121	United S...	132456...	swatig2...	321.99	6.99	2013-05...

Fig. 30. Screenshot of the Database: Orders After Successfully Placing an Order.

Figure 31 displays the ordered-products table in the database that contains information about all the ordered products. This table displays all items that have been successfully checked out. For example, in this screenshot, the user has ordered 1 item from the clothing category with a price of $45 and another item from the jewelry category with a price of $250. Similarly, if more successful orders are placed, then this table is updated, and the administrator can view the updated table by refreshing it.



ID	ORDER_ID	PRODUCT_ID	PRICE
13698874724751	2013529231752551	Clothing-1	45.00
13698874724762	2013529231752551	J-5	250.00

Fig. 31. Screenshot of the Database: Ordered Products After Successfully Placing an Order.

CHAPTER 4. TESTING

This chapter includes the methods that were used for testing, validating, and evaluating the system. The Conclusion and the Future Work for the software are also given.

4.1. Methodology

With this testing approach, all the specs were ready for a prototype, and a plan was already built to be shown; the tester started writing his or her code and saw if he or she could obtain the same results that the specs mentioned. This way, the specs were tested on each prototype, and continuous testing was applied. This also helped to minimize the testing that would have to be implemented at the end of the software lifecycle. In the process, all aspects of the software were tested. Steps to follow while implementing the methodology are as follows:

1. Start with a base function that you want to implement.
2. Create a document with the detailed requirement definition, an activity diagram with a description of the flow, database tables to be used, a component diagram, and a description of each component with the precondition and tables that would be affected by the component.
3. Give the document to the tester, and work with the tester while he or she writes the code to check if the steps in the document can be implemented and if the result of each use case can be achieved.
4. If the tester finds a step difficult to implement or thinks he or she is missing additional information to implement the functionality, then go to step 2; otherwise, go to step 3.
5. Ask the tester to log on all the errors and difficulties he or she encountered while working on the prototype implementation.
6. Once the prototype is done and the results between the developer's prototype and tester's prototype match, work on the other requirement, and expand the prototype to final software.

7. When the testing approach was implemented, the following pros and cons regarding the testing approach were realized.

   Pros of using the methodology

   - Helps give a better understanding about the requirements.
   - Better design at the end of the cycle.
   - Reduced testing to be performed at the end of the cycle
   - Documents produced would be of higher quality.

   Cons of using the methodology

   - The person working on the document should be experienced.
   - There are increased time and money involved with testing.
   - Different viewpoints for the same problem can lead to varying results.

4.2. Interface Testing

This section lists the functional requirements used for creating the test-case table, the test cases that were used to verify the interface table, and the results for the test-cases table.

Table 1 lists the functional requirements for the interface built for the online shopping-cart application, along with a short description of each requirement

Table 1. List of Functional Requirements.

Functional Requirement Number	Functional Requirement Short Description
FR01	The online shopping-cart application shall have two types of authentication: User authentication and Admin authentication.
FR02	The online shopping-cart application shall be accessible to all the users to browse all the categories and the items.
FR03	The users shall be able to view the items they added to the shopping cart.
FR06	The Admin shall be able to upload new/revised items as well as to add/modify the categories.
FR07	The Admin shall be able to view all the users registered in the system.
FR08	The Admin shall be able to view all the information abour users who placed an order from the shopping cart.
FR09	The users shall not be able to check out with an empty cart.
FR10	The users shall not be able to place an order without providing valid information for all rows in the order form.
FR11	The users shall not be able to place an order if any of the columns in the order form are left empty.

4.3. Test Cases

Table 2 shows the functional requirements used to write the test cases along with the test-case numbers for each test case and a short description of the test cases.

Table 2. List of Test Cases.

Functional Requirement No.	Test Case No.	Test-Case Short Description
FR01	TC01	To test the Login/Authentication interface for the Admin
	TC02	To test the Login/Authentication interface for the users
FR03	TC03	To test, users can view the items they add in the shopping cart.
FR06	TC04	To test, Admin can upload new/revised categories.
	TC05	To test, Admin can upload new/revised items.
FR07	TC06	To test, Admin can view all the users registered in the system
FR08	TC07	To test, Admin can view the information about all the users who successfully placed an order.
FR09	TC08	To test that users cannot check out with an empty shopping cart.
FR10	TC09	To test that users are not able to submit an order form if the information in any of the fields is not valid.
FR11	TC010	To test that users are not able to submit an order form if the information in any of the fields is left blank.

The following list includes the steps that should be taken by the user, the conditions that should be met for the successful execution of the test case, and the end result that should be met for the test cases to pass.

1. TC01: To test the Login/Authentication interface
    - Input: Username and Password
    - Output: Valid Destination Page
    - Valid Range: User Name →Alphanumeric, Password → Alphanumeric
    - End Messages/Result
        i. If (User == Valid User), an order form appears to complete the checkout process
        ii. If (User != Valid User), an error message is displayed on the Login interface.
2. TC02: To test, the users can view the items they add to the shopping cart.

- Description of Purpose: The system shows all the saved items in shopping cart for a particular user. The user can choose to check out the items or go back to continue shopping.
- Input: The user adds an item to the shopping cart from any of the available categories.
- Output: The shopping-cart page pops up, showing the item that is added by the user.
- End messages/Result
  1. If (Selection == Item and document == exists), the user is able to add that item to the cart, and the item shows up in the shopping cart, prompting user to delete the item, to continue shopping, or to check out the item.
  2. If (Selection = Item and Selection = View Cart), an empty shopping cart pops up with buttons to check out or to continue shopping.

3. TC03: To test, the Admin can upload new/revised categories and items.
   - Description: The Admin can add or upload more items to a category or can add a completely new category. The Admin can also modify the price, information and shipping taxes, etc. for the existing items and categories.
   - Input
     - i. User=Admin
     - ii. Selection=Items
     - iii. Selection=Categories
   - Output: New or modified items or categories in the shopping cart.
   - End messages/Result
     - i. If (User type = “Admin” &Selection = (Items || Category)&& Item/Category =existing), then display the modified items or categories in the shopping cart.

   ii. If (User type == “Admin” &Selection == (Items || Category) & Item/ Category=existing), then display newly added items or categories in the shopping cart.

4. TC04: To test, the Admin can view all the users registered in the system.
   - Description: The Admin can view all the users who are registered in the system in the database.
   - Input
     i. User Name →Alphanumeric, Password → Alphanumeric
     ii. User==Admin
     iii. Selection==View Database
   - Output: User List
   - End messages/Result
     i. If(login type == “Admin” & Database.clicked = ‘true’ and list.clicked=true and userlist.exists=true), then display users.
     ii. If (login type == “Admin” &Database.clicked = ‘true’ and list.clicked=true and userlist.exists=false), then display the empty database.
5. TC05: To test, the Admin can view the information about all users who successfully placed an order.
   - Description: For the Admin, a database, which contains all information about the users, is created after each user checks out the items and successfully places an order.
   - Input
     i. User Name →Alphanumeric, Password → Alphanumeric
     ii. User==Admin

iii. Selection==Database→Users Info

- Output: A database with the user's information or an empty database.
- End messages/Result

i. If (login type == "Admin" & Checkout.clicked = 'true' and Place Order.clicked=true and userlist.exists=true), then display the database containing the user's personal information.

ii. If (login type = "Admin" &Checkout.clicked = 'true' and PlaceOrder.clicked=true and userlist.exists = false), then display the empty database.

iii. If (login type = "Admin" & Checkout.clicked = 'true' and PlaceOrder.clicked=false and userlist.exists=true), then do not update the row in the database.

6. TC06: To test that users cannot checkout with an empty shopping cart.

- Description: If there are no items in the shopping cart, the checkout button is disabled, and the users cannot click the checkout button. A user cannot check out with an empty cart.
- Input

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Users

iii. Selection==View Cart→ No Items in the Shopping Cart

- Output: Disabled checkout button.
- End messages/Result

i. If (login type == "Users" & Items.AddToCart = 'false' &ViewCart.clicked='true'), then display the empty shopping cart with no items and a disabled checkout button.

ii. If (login type == “Users” & Items.AddToCart = ‘true’ &ViewCart.clicked=’true’ &Checkout.clicked==’true’), then display items in the shopping cart with the checkout button enabled so that users can check out.

7. TC07: To test that users are not able to submit an order form if the information in any of the fields is invalid.

- Description: The users cannot place a successful order if any information on the order form is invalid (i.e., A zip code is a 5-digit number, so any non-numeric value will be invalid.) or if any of the information is incomplete or left blank.
- Input

i. User Name →Alphanumeric, Password → Alphanumeric

ii. User==Users

iii. Selection==Checkout→Order Form → Place Order

- Output: User successfully or unsuccessfully places the order.
- End messages/Result

i. If (login type == “User” &CheckoutButton.clicked = ‘true’ andOrderFormInformation.Valid==’false’ or OrderFormInformation.Invalid==’false’&& PlaceOrder.clicked=true), then display an error message after the place order button is clicked.

ii. If (login type == “User” &CheckoutButton.clicked = ‘true’ andOrderFormInformation.Valid==’true’ and OrderFormInformation.Invalid==’true’PlaceOrder.clicked=true), then successfully place the order and display the success message.

4.4. Results

This section lists the results that were produced by running the test cases. Table 3 lists the test cases that were used while testing the interface along with the expected result and the actual results for each test case.

Table 3. List of Test-Case Results.

Test Case Number	Expected Result	Actual Result
TC01	Pass	Pass
TC02	Pass	Pass
TC03	Pass	Pass
TC04	Pass	Pass
TC05	Pass	Pass
TC06	Pass	Pass
TC07	Pass	Pass

CHAPTER 5. CONCLUSION/FUTURE WORK

This chapter includes the Conclusion reached after creating the current version of the software to meet the system objectives. The comparison is done between the system that was built and original requirements that were designed at the beginning of the project. It also describes the Future Work that is intended to be accomplished with later versions of the software.

5.1. Conclusion

The main objective of the application is to help computer science students understand the basics of Java, JavaScript, and HTML. By browsing through the application and looking at the code for each graphical interpretation, students should be able to easily understand the implementation. The following results have been achieved after the completing the system and relate back to the system's objective.

1. **Should allow computer science students to browse through the code and application:** This is achieved when users, i.e., computer science students, are able to run and install the application. When they run the application, they can browse through the implementation of different objects.
2. **Should allow users to browse through different product categories:** This is achieved when the user first runs the application and is directed to a home page that has categories available for all the different item types that can be purchased with this online shopping-cart application. The user can browse and click on any category to view the items listed for that particular category.
3. **Should allow users to save items to the cart and also to view detailed information about a particular item:** The users can add any number of items to the shopping cart from any of

the listed categories by simply clicking the cart icon at the right-hand corner of each item. Users can view a detailed description of the item and price by clicking on the more icon next to the cart icon.

4. **Should allow users to check out the items:** This is achieved when users click the checkout button in the shopping cart. The checkout button disappears when there are no items in the shopping cart. This implies that users can only click the checkout button when there are items in the shopping cart.
5. **System users should be able to place the order by filling out the order form:** This is achieved when a user clicks the checkout button and an order form appears on the same page, showing the detailed order total with shipping charges and sales tax, which requires the user to complete all the information to successfully place an order. The user is not able to place an order if any information on the form is invalid or empty.
6. **The user should see a success message after placing an order:** This is achieved when a user successfully places an order by completely filling in all the rows after which he or she can click the place-order button. The user receives a message that the order has been successfully placed.

5.2. Future Work

The following section discusses the work that will be implemented with future releases of the software.

1. **Detailed categories:** Future work could involve adding more categories which are more detailed and have additional items.
2. **Watch/Wish List:** Work can add a watch list or wish list so that users can add an item to a list to watch for item prices to go down or to see when there is a sale on any of those items.

3. **Enhanced User Interface:** Work on enhancing the user interface by adding more user-interactive features.
4. **Recommended Items:** Add a bar that would display the most-recommended items which would depend on the number of times an item has been purchased by any users.
5. **Payment Options:** Add different payment options, such as Visa, MasterCard, PayPal, etc., where a user can also save the card information for later checkouts.
6. **Shipping Options:** Add different types of shipping options: regular shipping, expedited shipping, international shipping, etc.
7. **Recent History:** Display the user’s recently browsed items in the recent-history tab.

REFERENCES

1. Howe, A. von Mayrhauser, and Mraz, R. T. Test case generation as an AI planning problem. Automated Software Engineering, 4:77-106, 1997.

2. Koehler, J., Nebel, B., Hoffman, J., and Dimopoulos, Y. Extending planning graphs to an ADL subset. Lecture Notes in Computer Science, 1348:273, 1997.

3. Treutner, M. F., and Ostermann, H. Evolution of Standard Web Shop Software Systems: A Review and Analysis of Literature and Market Surveys.

4. Jarvenpaa, S. L., and Todd, P. A. (1997). Consumer reactions to electronic shopping on the World Wide Web. International Journal of Electronic Commerce, 1:59–88.

5. Peterson, R. A., Balasubramanian, S., and Bronnenberg, B. J. (1997). Exploring the implications of the internet for consumer marketing. Journal of the Academy of Marketing Science, 25:329–346.

6. "More Consumers Using Tablets to Holiday Shop [STUDY]." Retrieved on *Mashable*. December 8, 2011. " http://mashable.com/2011/12/08/tablets-to-holiday-shop/"

7. Bigne, Enrique. (2005). The Impact of internet user shopping patterns and demographics on consumer mobile buying behavior. Journal of Electronic Commerce Research, 6(3).

8. King, Stephen F. en Juhn-ShiuanLiou (2004), A framework for internet channel evaluation. International Journal of Information & Management 24:473–488.

9. Falk, Louis K., Sockel, Hy, and Chen, Kuanchin. (2005). E-Commerce and consumer's expectations: what makes a website work. Journal of Website Promotion, 1:65–75.

10. Elliot, Steve, and Fowell, Sue. (2000). Expectations versus reality: a snapshot of consumer experiences with Internet retailing", International Journal of Information Management 20: 323–336.

11. Java vs.C#, Retrieved on 2013. "http://www.cs.yale.edu/homes/hudak/CS112F06/java.html"

12. Java Shopping Cart Powered by SoftSlate., Retrieved on 2013. "http://www.softslate.com/featureList.html"

13. Commerce4j, Java Based e-Commerce Application and Online Catalog Management System. , Retrieved on May 18, 2010. "https://code.google.com/p/commerce4j/"

14. CS-Cart.com (Simbirsk Technologies Ltd), © 2004-2013.http://www.cs-cart.com/

15. Ofbiz, The Apache Open for Business Project., Retrieved on 2013. "http://ofbiz.apache.org/index.html"

16. Comparison of shopping cart software., Retrieved on June 28, 2013. "http://en.wikipedia.org/wiki/Comparison_of_shopping_cart_software"