PROJECT PART 1

CD STORE

DESIGN DOCUMENT

Prepared For:

Part1 CSI5380 Project submission

Submitted to:

Professor Dr. Hussein Al Osman

In partial fulfillment of the requirement for

CSI5380 - Systems and Architecture for Electronic Commerce

Submitted by:

Web Builders

SI.No	Team Member	Student ID	Contact ID
1	Aishvarya Arul Nambi	7758189	aarul094@uottawa.ca
2	Indhra Priya Shanmugam	7527045	ishan050@uottawa.ca
3	Priyanka Patel	7481432	ppatel038@uottawa.ca
4	Rachana Chandrashekar	7487187	rchan092@uottawa.ca
5	Shruthi Madhurika Naomi	7564806	snaomi011@uottawa.ca

Table of Contents

List of Figure

1. Abstract

The document represents the Team's work towards building an online CD Store. The aim of the project is to build a simple e-commerce based website where a customer can browse through a catalog of CD'S and purchase music online. This design document discusses the approach adopted by our team to design the web application and addresses in detail the steps that we followed to come up with a fully functional system.

2. Objectives

The main idea is to build an online music store where the visitors of the website can purchase CD's divided into categories. The visitors can scan through a product catalog of CD's. They can view and also add or remove products to their shopping cart. Once the items are added to the shopping cart, the customer can then proceed to checkout to purchase them. To purchase the CD's the customers would have to give their Credit Card and shipping information. Thus security plays a key role in our website and it is absolutely necessary to keep the customer's details secure and safe. This is achieved using the SSL configuration. The project was developed in Eclipse using MySQL database and Tomcat server.

3. Platform Specification

3.1. Server side

a) Hardware Requirements

- Windows 7 or Windows 8 operating system

b) Software Requirements

- Google Chrome or Mozilla Firefox Browser
- Download J2SE from http://java.sun.com/j2se/downloads.html

- Environment - Download Eclipse IDE from:
  https://www.eclipse.org/downloads/
- The database used in this project is MySQL version 5.6. It is an Open source database. The database can be mainly accessed using Sql statements in order to retrieve or alter data.
- Download MySql-installer-community-5.6
  http://dev.mysql.com/doc/refman/5.6/en/installing.html
- My Sql Connector J 5.2
- The web server used to host and serve websites is Apache Tomcat 7.55. Apache Tomcat is an open source software implementation of the Java Servlet and Java Server Pages technologies. It is considered to be a light-weight server with minimum configuration and installation effort.
- Languages used – HTML, CSS, JavaScript and JQuery.
- SSL Configuration - The main idea behind using an SSL certificate is to keep sensitive information encrypted so that only the intended user understands it. By using an SSL certificate, the information becomes unreadable to everyone except the server to which the information is sent. This way we can ensure the customers that the website is legitimate to do business with and is protected by Https.

Figure 1: SSL configuration

3.2. Client side

a) Hardware Requirements

- Windows 7 or Windows 8 operating system.

b) Software Requirements

- Google Chrome or Mozilla Firefox Browser.

4. Development Stages

The project undergoes several stages adopted from the software development cycle. The stages include Planning, Analysis, Design, Implementation, Testing and Maintenance. Each phase requires background study and a set of activities to complete and reach the next stage. The operations and the workflow in each phase can be explained as follows:

a) Planning & Analysis: This stage includes the background study and research work done prior to the commencement of the project by going related references, books, Internet etc. The main objective of this phase is to understand the fundamentals of e-commerce architecture, creating a rough sketch of the plan to implement the project and analyze various factors.

b) Design: The main objective of this phase is to determine the high level architecture of our project. The design hence gives us an overview of the entire project.

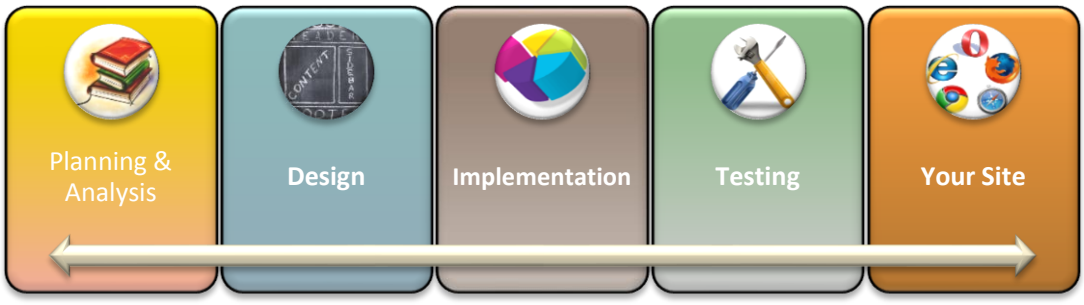

Figure 2: Development Stages

c) Implementation & Testing: During implementation, various modules responsible for different functionality are built and integrated. Testing phase includes reviewing the project and creating test cases for each phase to test the output of the project. The test plan is documented and the outputs are checked.

d) Maintenance: This is the final phase and shows the final website. The client can walk through the website and access the categories on the website and purchase the products they wish to. Maintenance focuses on modification of the website after project delivery to correct faults and improve performance.

5. Design Analysis

5.1. Model View Controller pattern

The CD Store architecture uses Model View Controller pattern, a widely used design pattern in graphical user interface (GUI) programming that separates application's concerns.

5.1.1. What is Model View Controller?

The MVC pattern separates the modeling of the domain, the presentation and the action. It is a fundamental design pattern for the separation of user-interface logic from business logic.

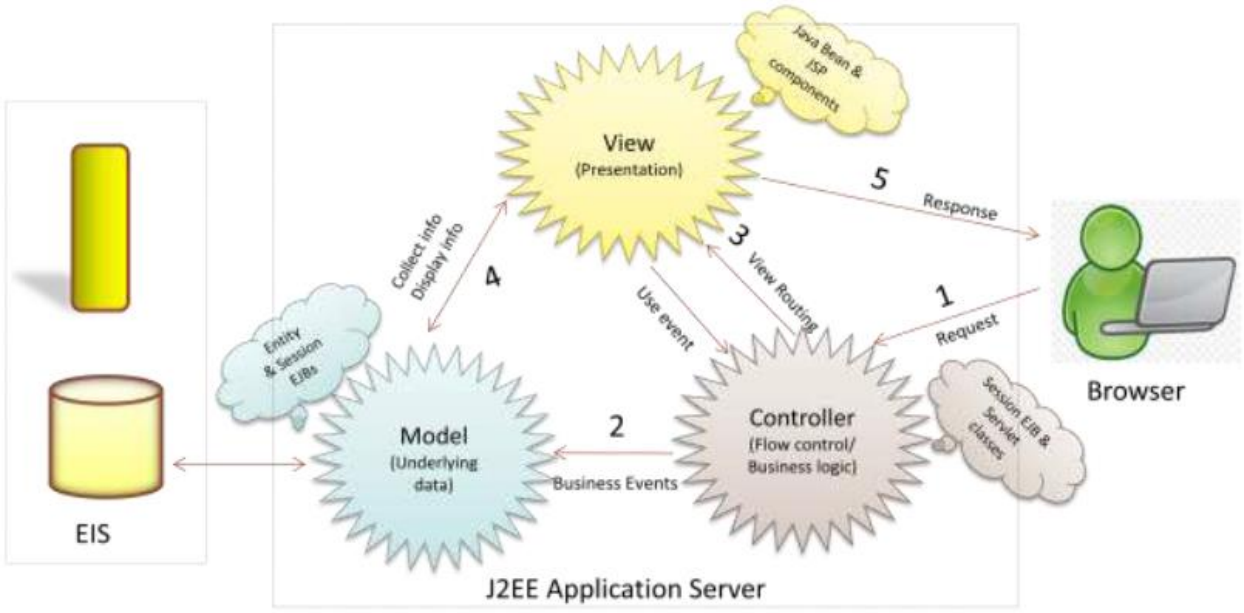

Figure 3: Model View Controller Pattern

MVC can be broken down into the following three basic elements:

- ✓ **Model:**
  - Represents the data and the rules.
  - Responds to requests for information about its state (from the view).
  - Responds to instructions to change state (from the controller).
- ✓ **View:**
  - Represents the visualization of the model-containing data.
  - Manages the display of information.
- ✓ **Controller:**
  - Keeps View and Model separate.
  - Controls the data flow into model object and updates the view when the data changes.
  - Coordinates interaction between View and the Model.

5.1.2. What does MVC provide?

The MVC design pattern is a well-established and compelling approach for building software. The MVC pattern can open up new levels of robustness, code reuse and is regarded as a good approach for building software for the following reasons:

- Provides separation of data from presentation layer.
- Provides ability to have multiple interfaces for an application.
- Provides a logical structure for interactive systems.
- Provides ability to have multiple view of single data.
- Involves less strain in changing business rules and data layer.
- MVC adheres to good engineering design principles and practices.
- Supports multiple clients, easier to implement, test and maintain.

5.1.3. System Architecture

The CD Store has four tiers:

- Client tier components run on the client machine.
- Web tier components run on the J2EE server.
- Business tier components run on the J2EE server.
- Enterprise Information System tier software runs on the EIS server.

This four-tier architecture is depicted in the diagram below

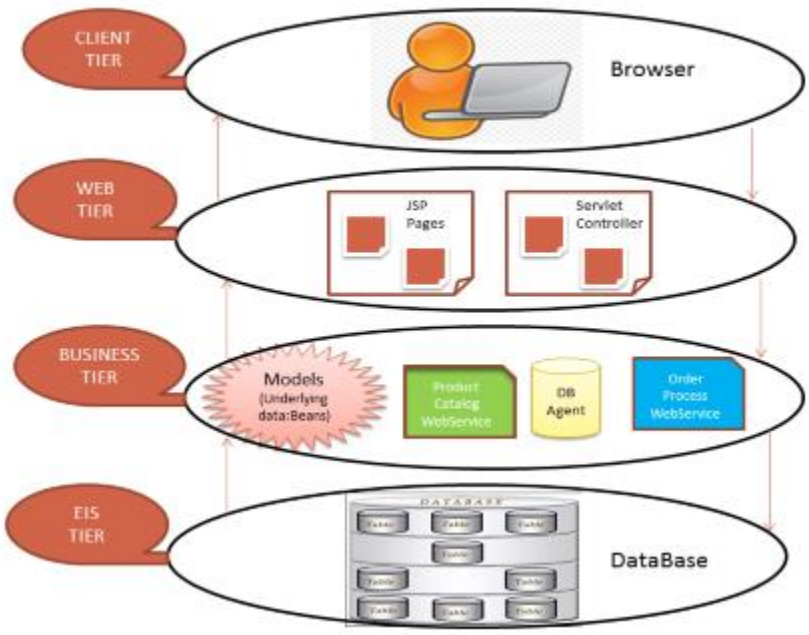

Figure 4: System Tier-Architecture

- **Client Tier:**

  The Client tier represents a Web browser or a Java application or other application like Applet. The CD Store is accessed by the client through the browser (Google Chrome, Internet Explorer, etc.) which fires request to the server. The client communicates with the server over https and gets a JSP page in return.

- **Web Tier:**

  The web tier consists of JSP pages and a Session Controller Servlet served by Apache Tomcat JSP/Servlet Container. The server receives http requests from the client and returns an appropriate JSP page back to the client as appropriate. The server communicates with the business tier (OrderProcessingWebService and ProductCatalogWebService) using REST (**Jersey**) Web Service protocol. A J2EE application's Web tier serves HTTP requests. The Web tier receives and interprets client requests, dispatches those requests to business logic, selects the next view for display, and generates and delivers the next view.

- **Business Tier:**

  The Business tier covers most of the application logic that is often implemented in Web Services using REST. This tier contains the business logic and the business data to provide business services. This tier communicates with the EIS tier using the database agent which encapsulates all access to the data store. Database agent is responsible for managing the database connection, queries, retrieves and updates.

- **EIS Tier:**

  The Database is located in this tier. EIS tier is responsible for storing critical data required by an enterprise to run its business operations. It is known to provide transaction support and enhanced security. EIS tier technologies include Java Database connectivity API. We use a MySQL database for storing data.

5.1.4. Deployment Diagram:

The figure below is the Deployment diagram of the CD Store. As seen in the diagram, the CD store has the Client, Web Server and the Database Server applications.

The Client browser communicates with the Web Server application through http/s. The servlet controller receives this http/s request from the browser and handles it. The Database Agent is responsible for the communication with the Database.

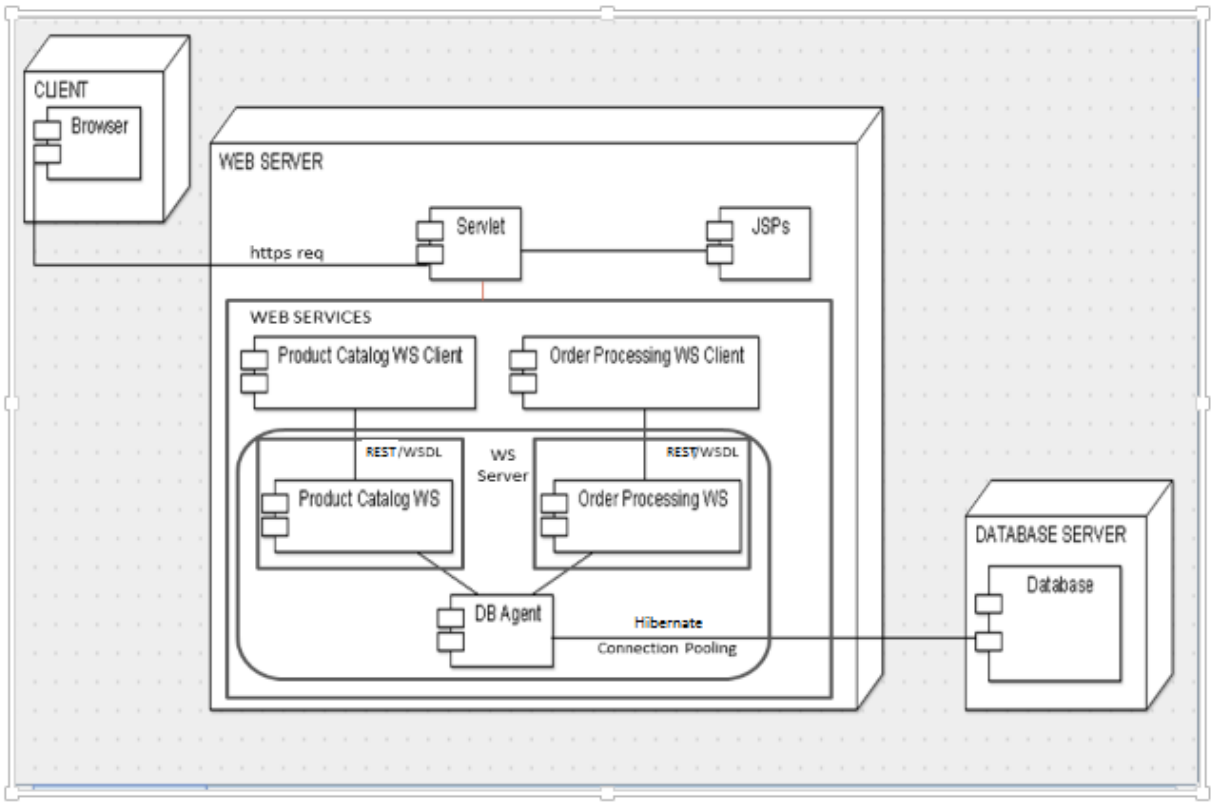

Figure 5: Deployment Diagram

5.1.5. Client Requirement Specification

- **GUI visibility:**

  The GUI should show Sign Up, Category list, Contact us and Shipping tabs successfully.

- **List of Categories:**

  All the categories are to be made available to the client for selection.

- **List of CDs:**

  All the CDs under the selected category have to be displayed to the client and made available for selection.

- **Product Description:**

  Product description should be made available to the client, if he wishes to know about the selected CD before he orders one. It is required to display the album artist, album name and description that would introduce the client to the selected music CD.

- **Add to Shopping Cart:**

  Once the client wishes to buy the music CD, it must be made possible for clients to add the selected items into their shopping cart, delete items from their shopping cart when they wish to discard their selection and view the shopping cart contents during the transaction.

- **Check for an existing account:**

  It is required to check if the operating user already owns an account with the Store. The client can check out his/her order only after logging into his/her CD store account.

- **Login existing user:**

  The users already having an account with the CD Store, can login to proceed with the checkout.

- **Create account for the new user:**

  The CD store demands its customer to have an account in order to make transaction with them. The new users are prompted to create an account before they proceed with the checkout.

- **Confirm Shipping Address:**

  The client is asked to verify his/her shipping address. Initially, the client is shown with his/her address that was provided during the registration as the shipping address. The client can choose to change the shipping address at this point of time or retain the same address provided during registration.

- **Change Shipping Address:**

  If the client prefers to ship his/her orders to the address different than the one provided during registration, he is prompted to fill in the form .

- **Payment Page:**

  The client can make payment for his/her order. The page collects the client's first name, last name, credit card number, card expiry date and lets him/her to make payment through credit card (of types: Visa/American Express/Master card).

- **Order Confirmation page:**

  Once the client has successfully placed his/her order, he is shown an order confirmation message on this page.

6. Logical Design

6.1. Database Design

The Entity-Relationship diagram is the graphical representation of entities and their relationships to each other. The CD store project has five entities, which can be listed as:

- **Store_CD**

  The Store_CD entity table stores all the information about the CDs that are for sale.

- **Store_Address**

  This entity table stores the user address details, which is provided during the registration.

- **Store_UserDetails**

  This entity object table stores all the user details provided during the registration.

- **Store_Order**

  The Store_Order table stores the order details. Order_OrderId is the primary key.

- **Store_OrderItems**

  This entity object table stores details of the order, placed by the user.

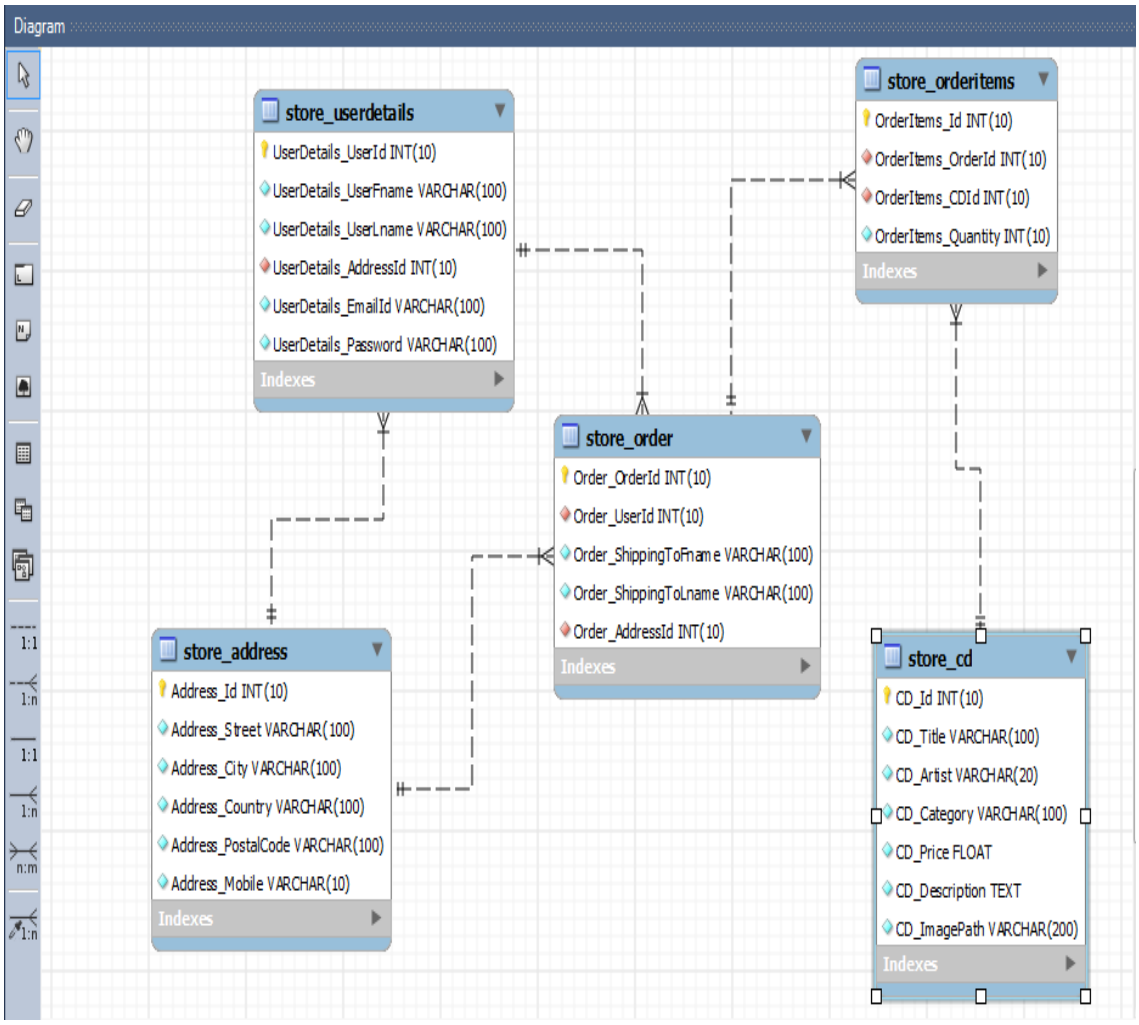

Figure 6: Database Design E-R Diagram

6.2. Package Diagram

The diagram below shows the system packages.

The system's packages include:

- Servlet Package.
- JSPs package.
- ProductCatalogWS package.
- OrderProcessingWS package.
- Models packaged in the Database Agent.

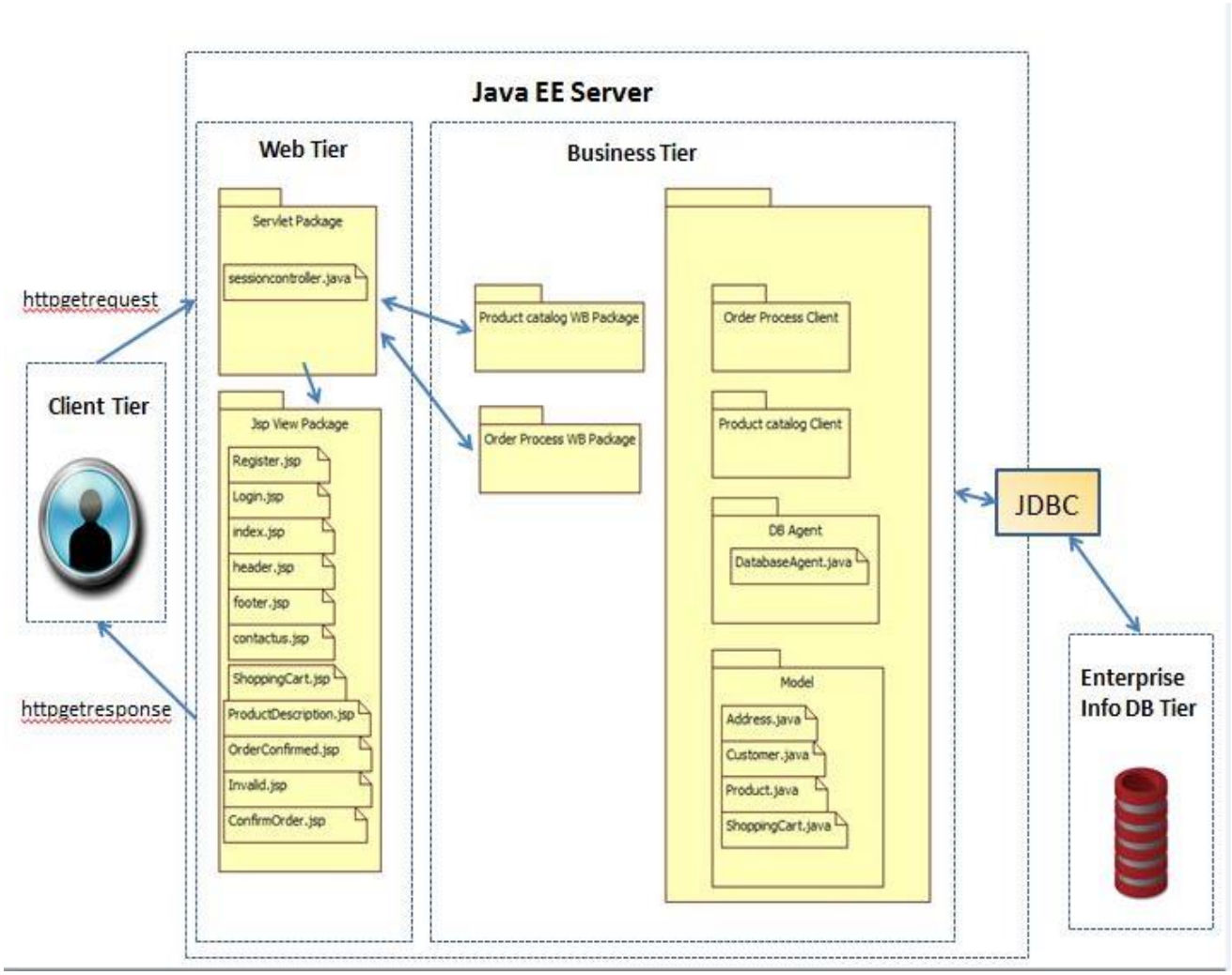

Figure 7: Package Diagram

7. Interaction Diagram

The figures in the following section show how and in what sequence our CD Store objects interact with one another. There are two sequence diagrams involving interaction objects: User/Client Browser, Views, Servlet Controller/Session Controller, 2 Web Services(ProductCatalogWS & OrderProcessingWS), database Agent containing the models and the database.

- Client traversing over the available products and selecting products of his/her choice.
- Client adding the products of his/her choice into shopping cart and placing an order.

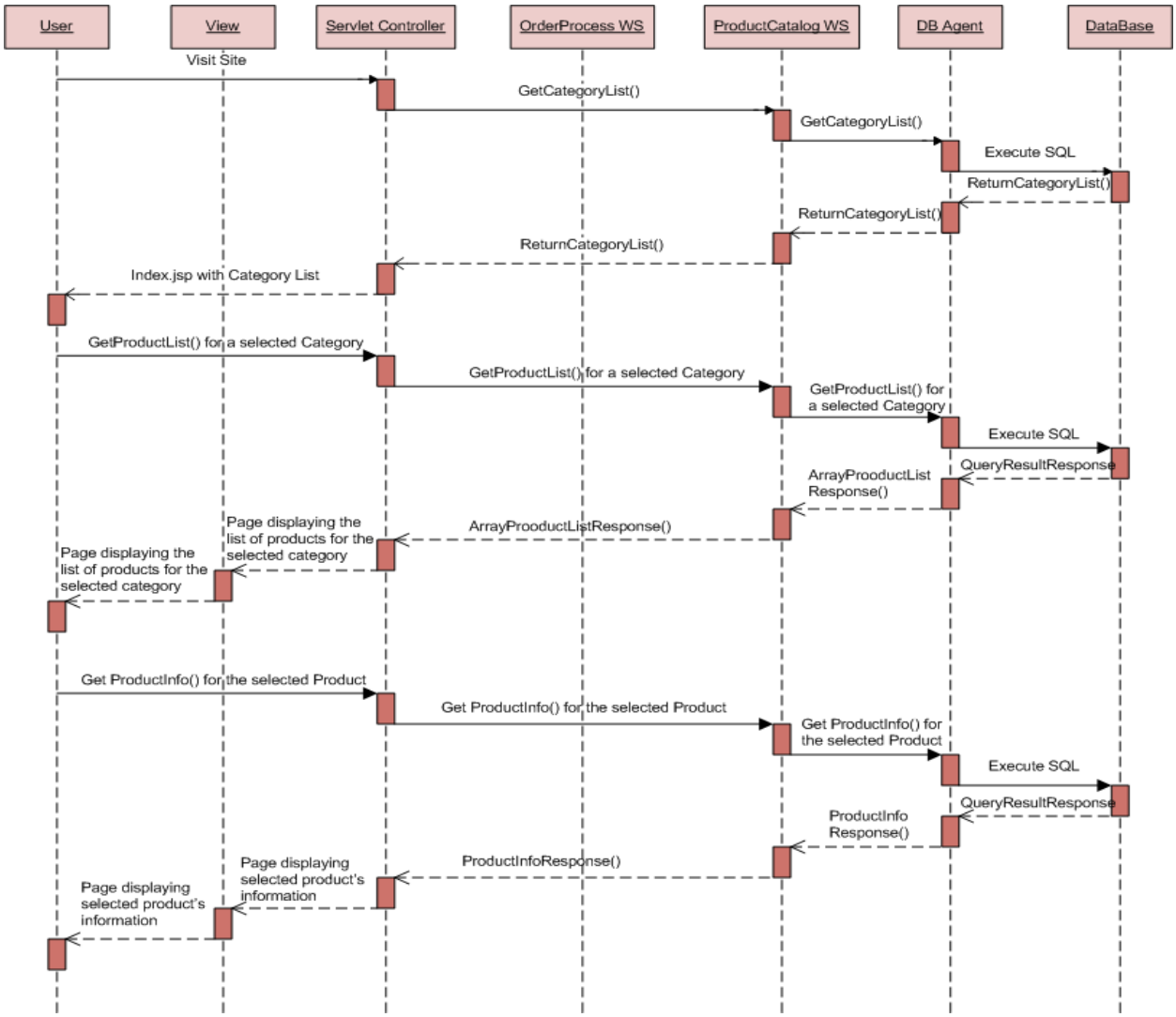

Figure 8: Interaction diagram for ProductCatalogWS transactions

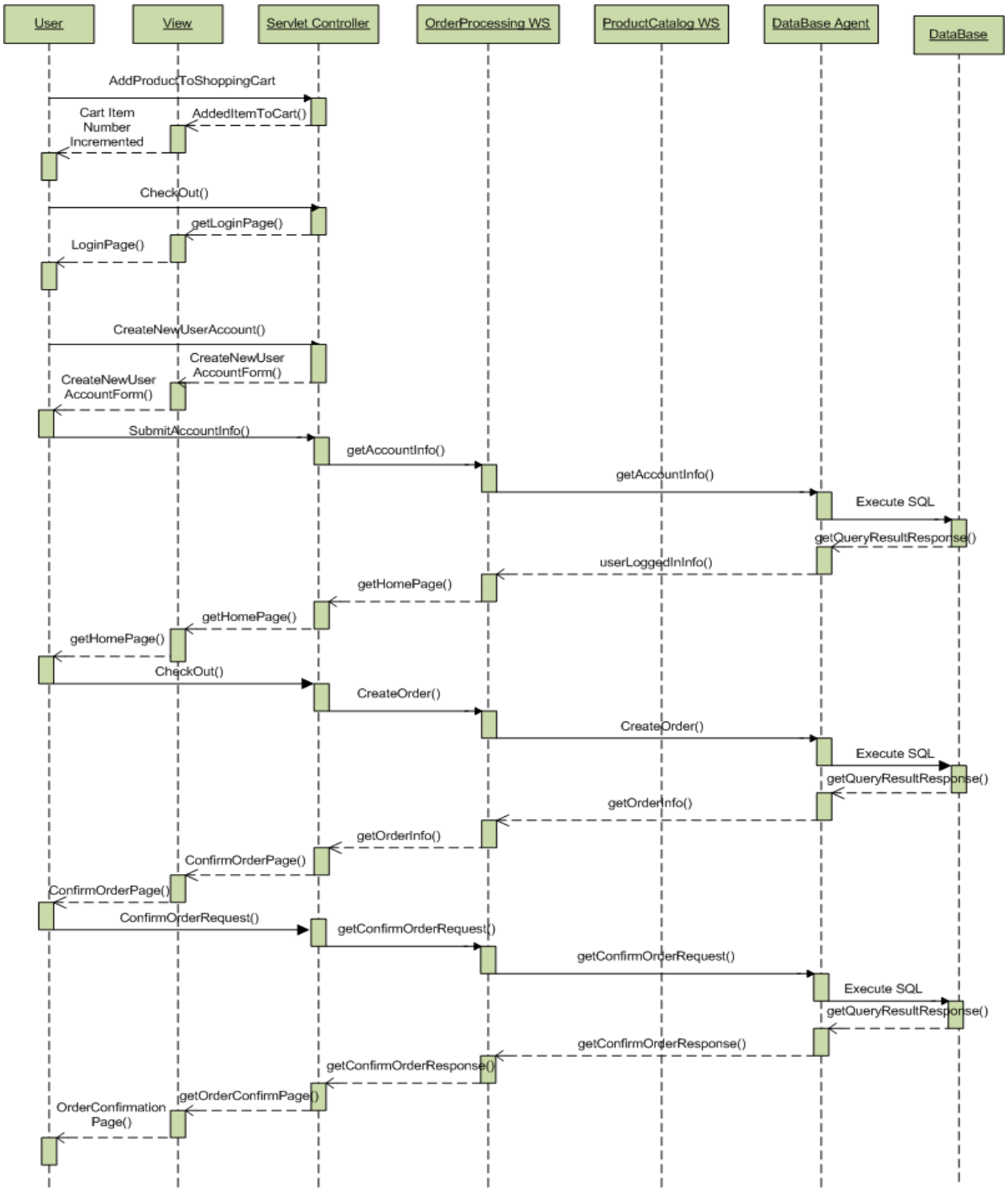

Figure 9: Interaction diagram for OrderProcessingWS transactions

8. Application Screenshots

Step 1: User navigates to CD store by providing the valid URL in the browser. The user is shown a home page displaying the category list on the left pane and the products of all the categories are displayed in the center of the home page.

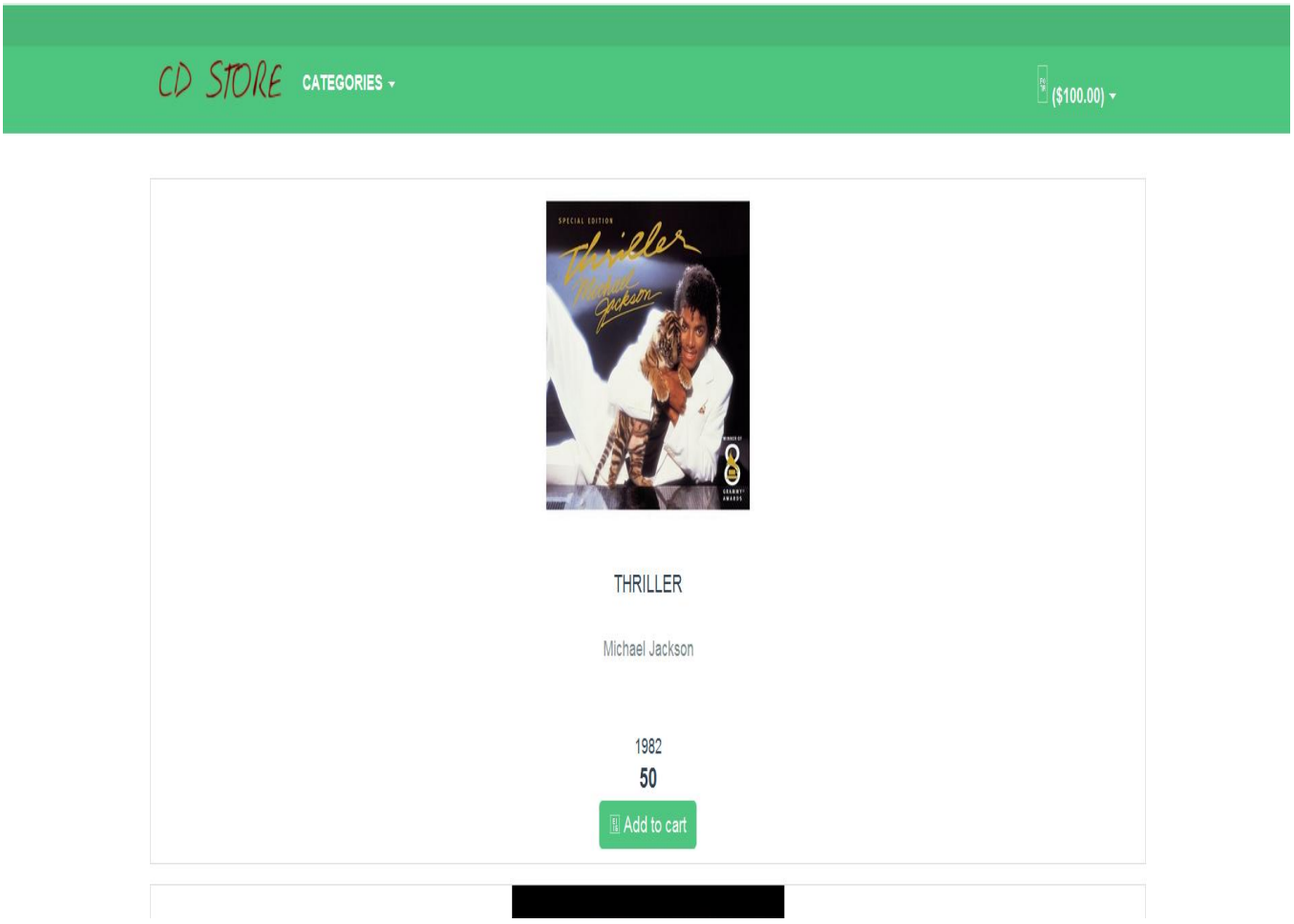

Figure 10: Displaying Home page with Category for CD Store Website

Step 2: The user can select the categories displayed on the left pane to display the products of that category. The below screenshot shows the entire category list to the user.

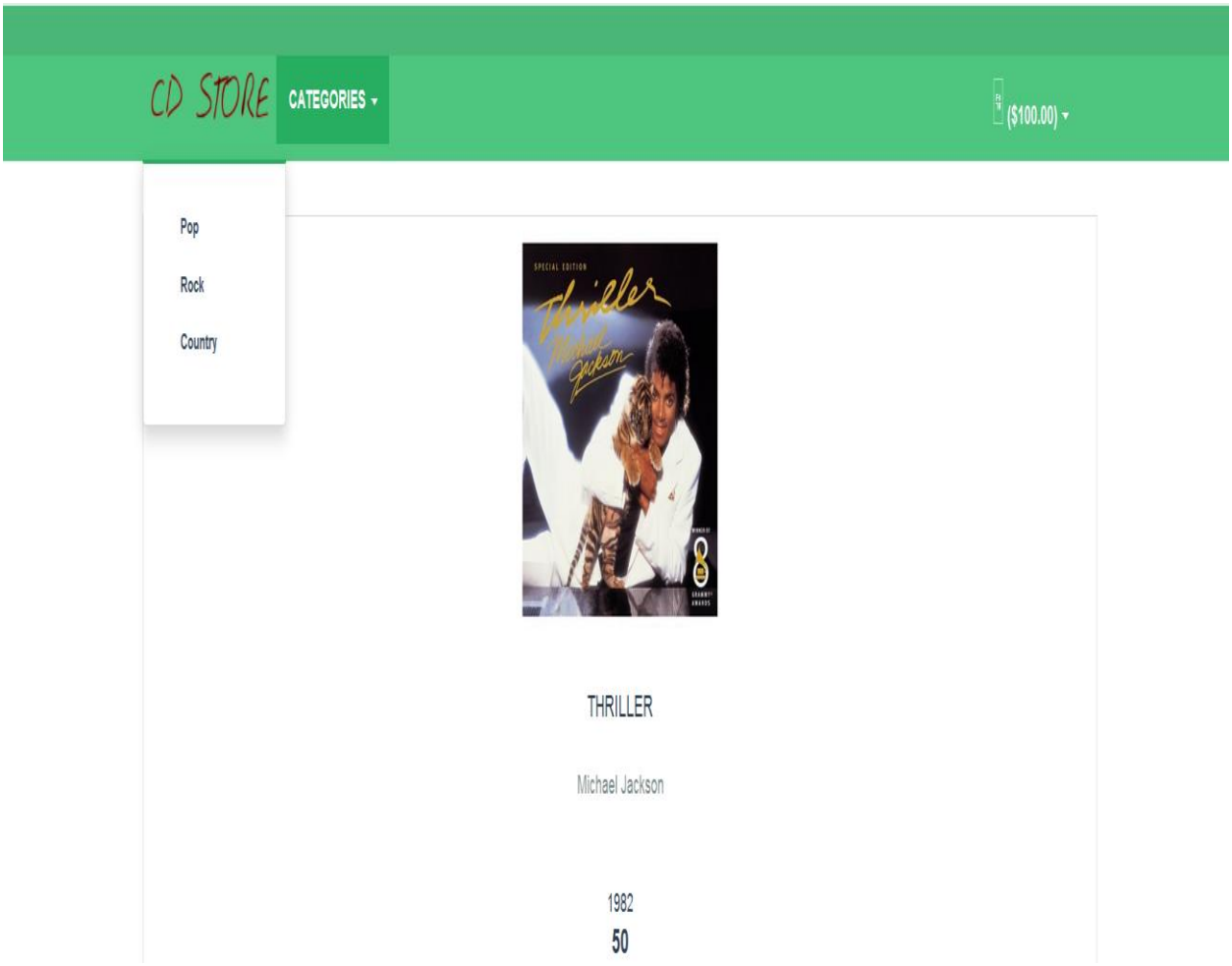

Figure 11: Displaying the Category List

Step 3: The user has selected category='ROCK' and hence all the products under 'ROCK' are displayed to the user. The user can also add the product to his/her shopping cart from this page.

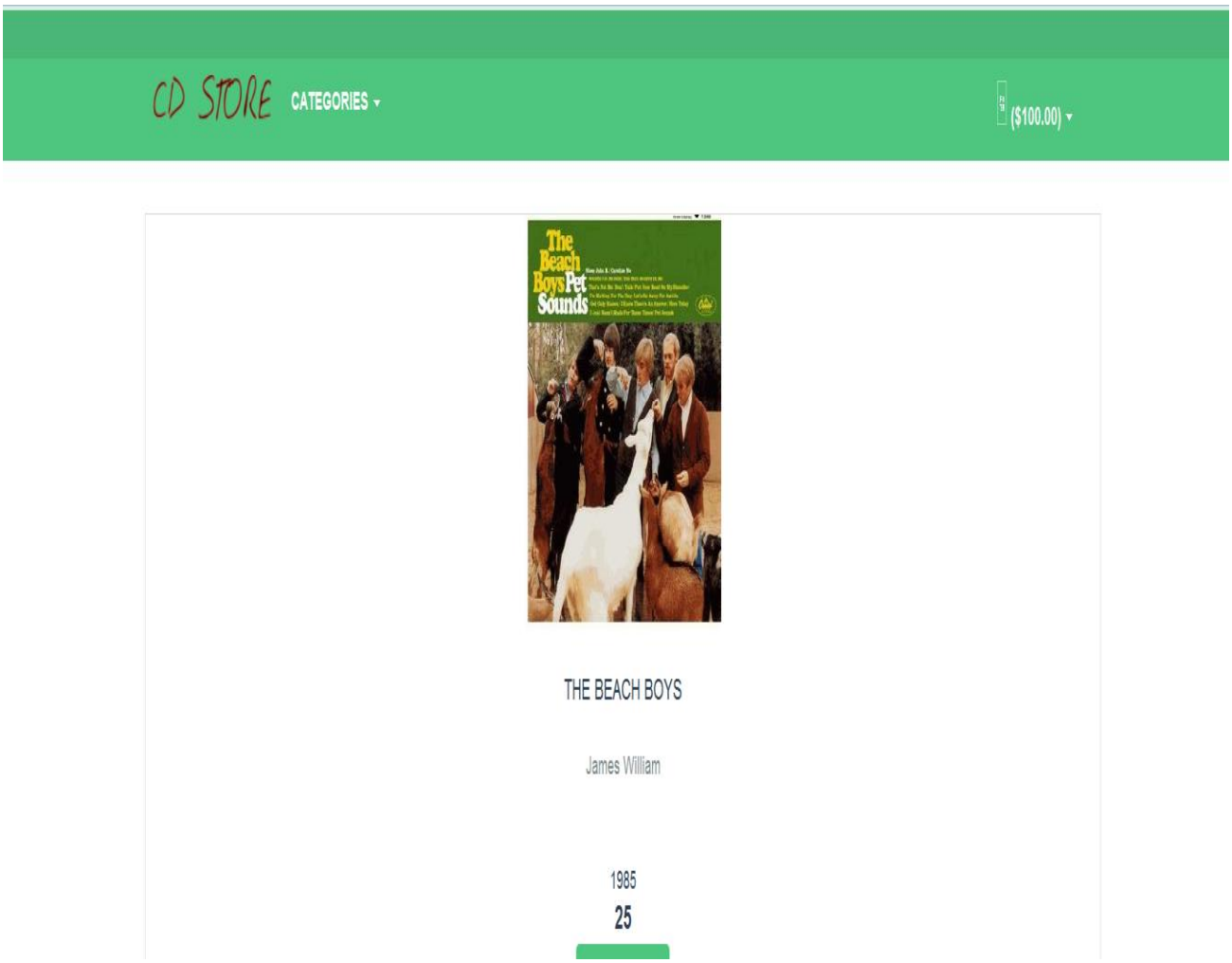

Figure 12: Select the Category to display the Product of that Category List

Step 4: The user adds the product to his/her shopping cart by clicking on "Add to Shopping Cart". Now Item got added in your cart, pop up message will be displayed to the user.

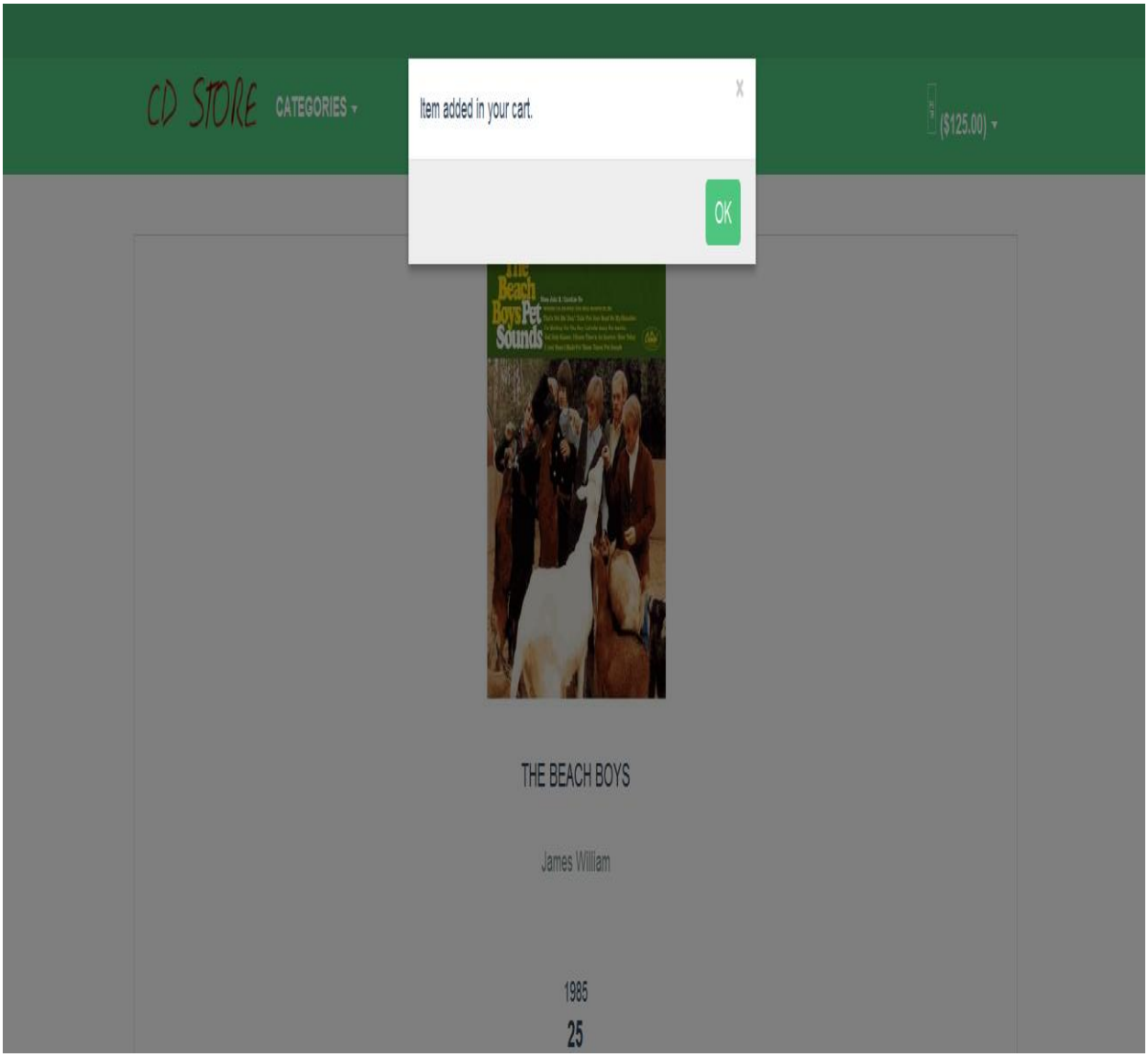

Figure 13: Add to Shopping Cart

Step 5: The user can view the shopping cart items by clicking on the shopping cart image, seen on the right side of the page.

Figure 14: View of Shopping Cart

Step 6: The user can delete the items from his/her shopping cart by clicking on "delete" button. On deleting the item, User gets a pop up window to confirm that item will be removed from the Shopping Cart.

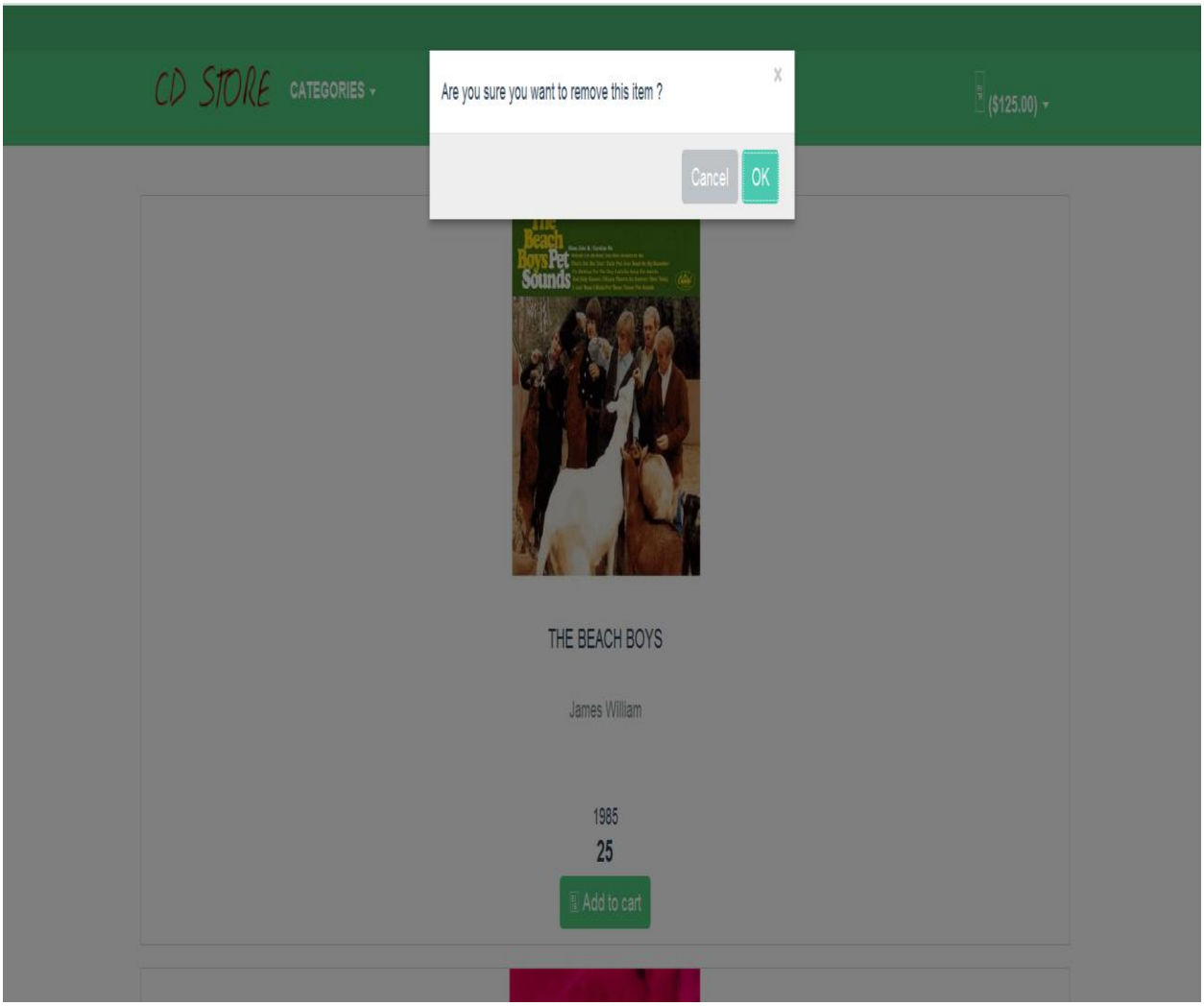

Figure 15: Conformation to Delete Product item in Shopping Cart

Step 7: Now click Ok in the previous screenshot to Confirm the item removed from the shopping cart. Users will be alerted with a message as shown below:

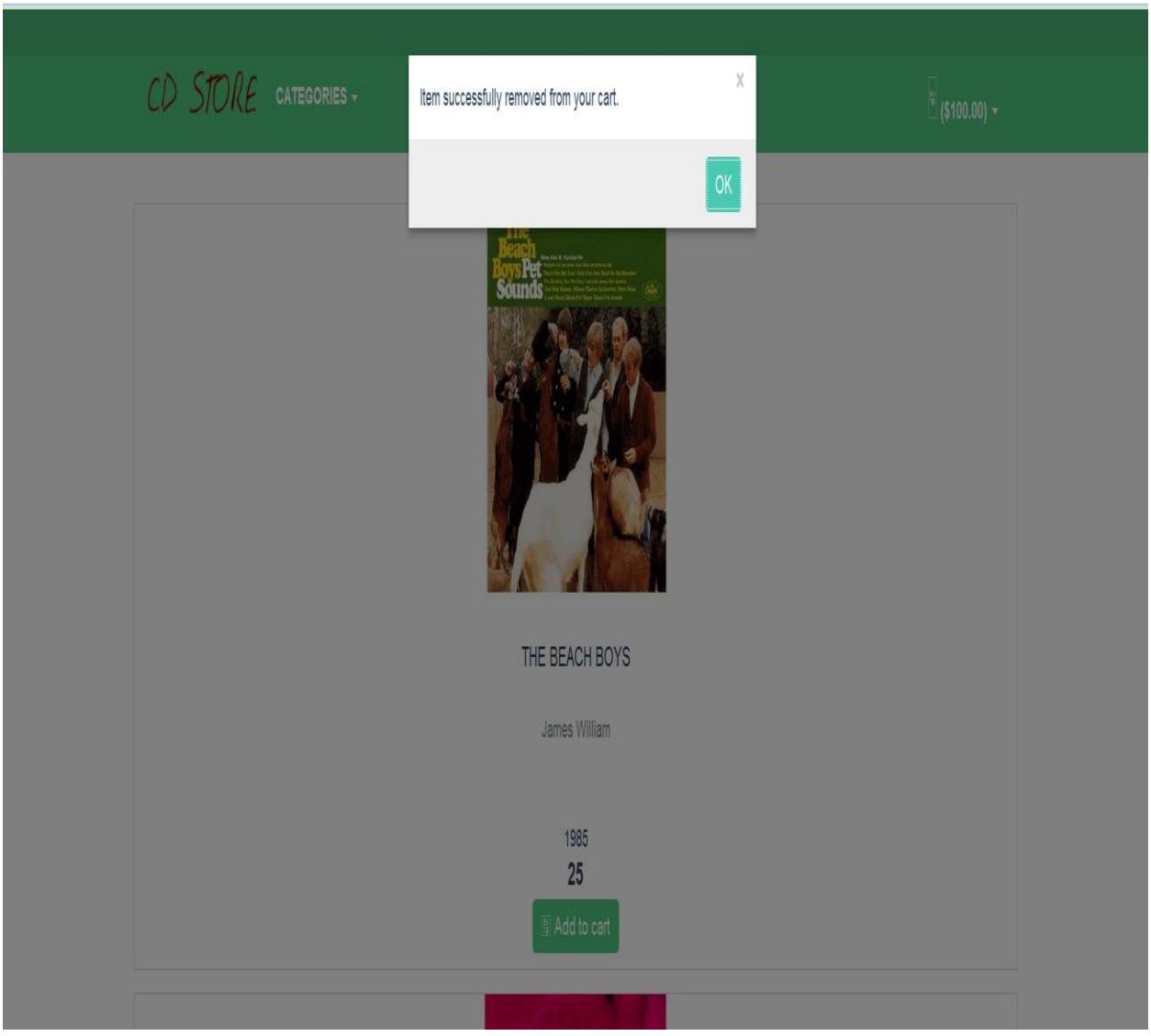

Figure 16: Delete the Product item in Shopping Cart

Step 8: When the user clicks on "Proceed to Checkout", he/she is navigated to the Order processing page. User can add the billing address that will be as same as the shipping address.

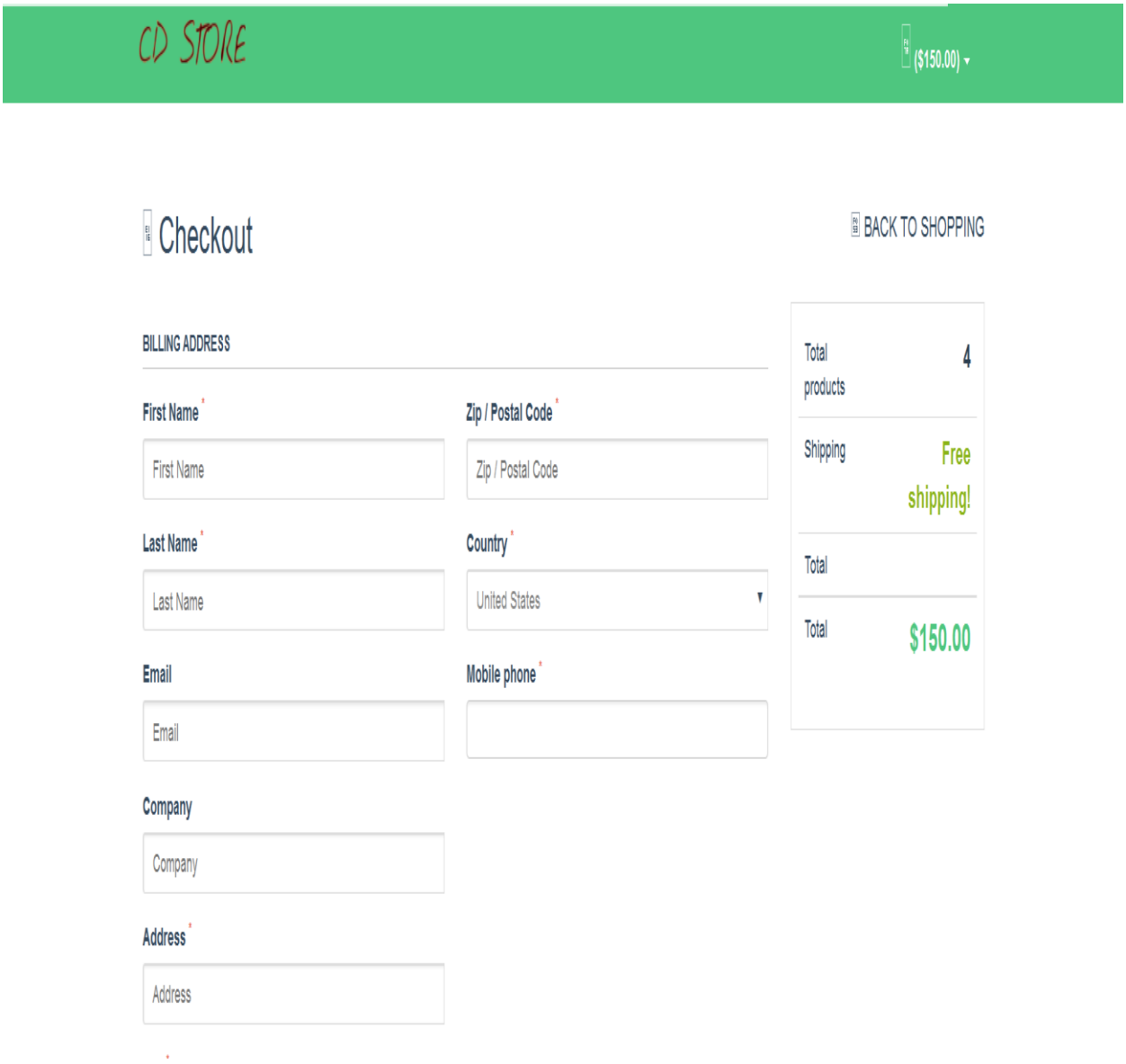

Figure 17: Order Processing Page

Step 9: If the user did not enter the required field in the Order Processing page, he/she will get a notification as shown below

a) Validation Check for ZIP code

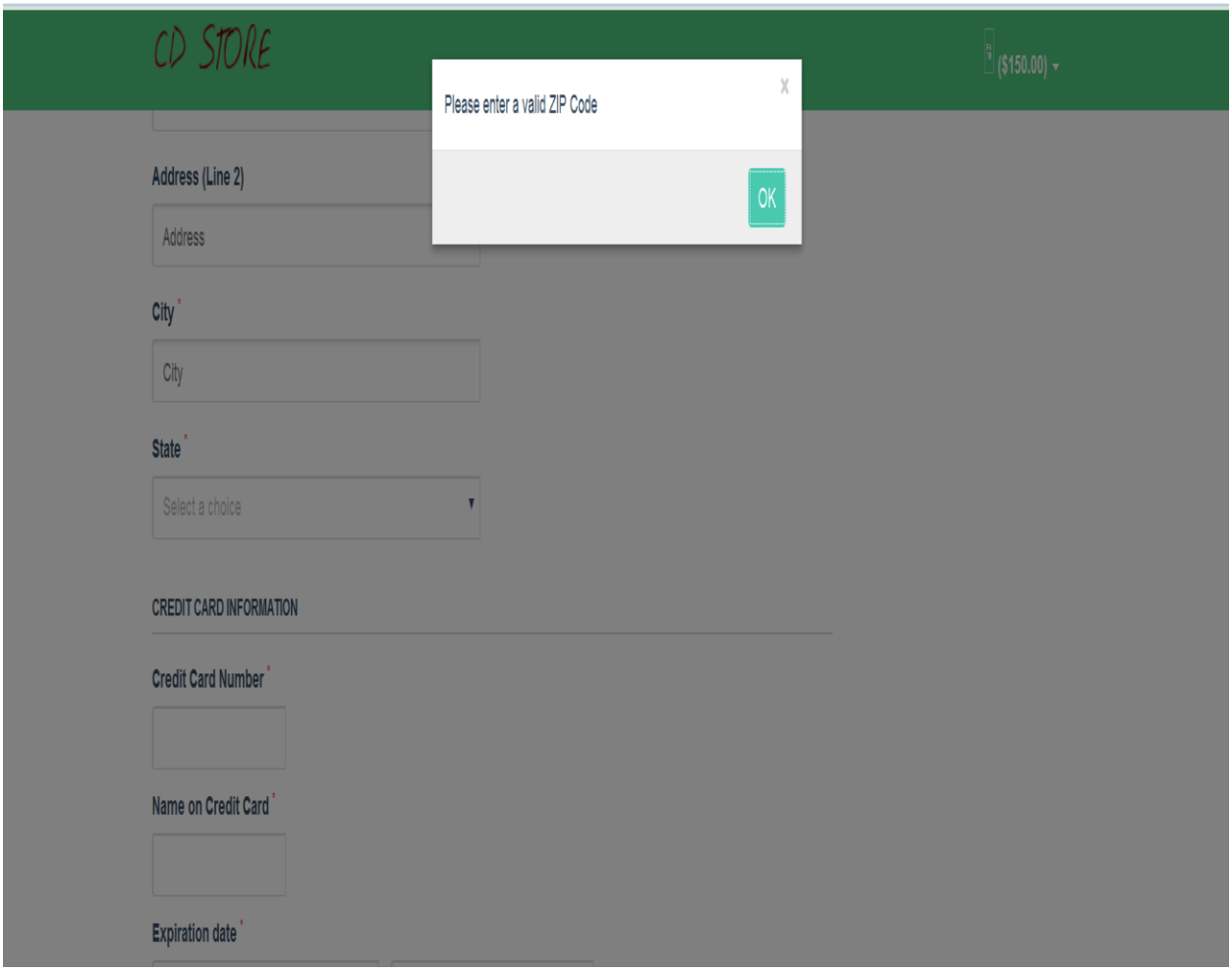

Figure 19: Validation Check for the Zip Code

b) Validation check for the first name

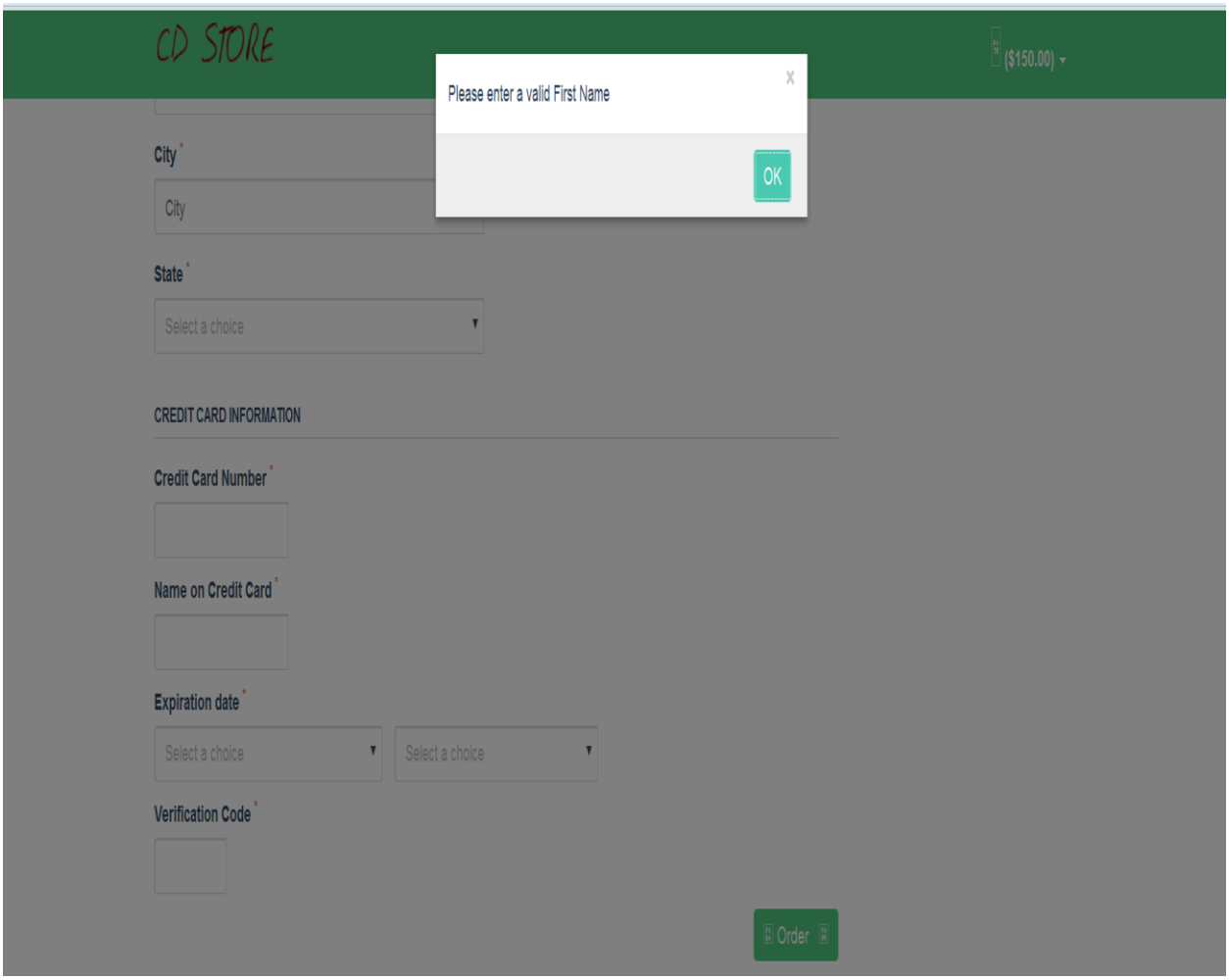

Figure 19: Validation Check for the First Name

Step 10: The user fills all the necessary user details and credit card details clicks on "Confirm Payment" button.

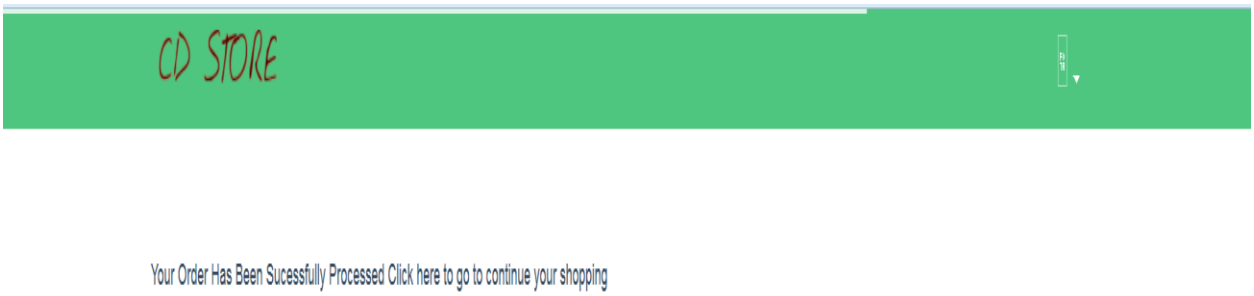

Figure 20: Order Success Page

Step 11: The project is also emphasizing on refusing every fifth request on the CD Store site.

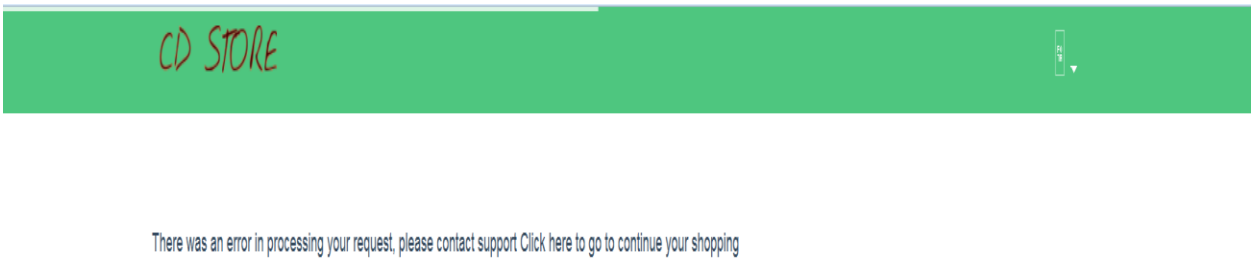

Figure 21: Order rejects for 5th request

- MVC Architecture

http://www.slideshare.net/TarinGamberini/mvc-and-struts-1

http://www.java-forums.org/blogs/j2ee/689-explain-mvc-architecture-relating-j2ee.html

http://www.oracle.com/technetwork/articles/javase/index-142890.html

- System Architecture

http://docs.oracle.com/javaee/1.4/tutorial/doc/Overview2.html

- Deployment Diagram is constructed using ArgoUML

http://argouml-stats.tigris.org/documentation/manual-0.32/ch23.html

- Interaction Diagram is constructed using Microsoft Office Visio 2013

http://www.youtube.com/watch?v=QsNuGB9mvJk

- Connection Pool setup

http://www.mkyong.com/hibernate/how-to-configure-the-c3p0-connection-pool-in-hibernate/

- SSL Configuration

http://www.youtube.com/watch?v=rVt6g5AZ6ko