

An Easy, Secure and Reliable Online Shopping & Payment System

Ripan Kumar Basak¹, Dr. Avula Damodaram²

¹JNTUH, School of Information Technology, Kukatpally, Hyderabad, Telangana, India-5000 85

²Professor of CSE, Chairman, BOS, CSE&IT and Director of University Academic Audit Cell, Kukatpally, Hyderabad, Telangana, India-5000 85

Abstract: *The On-line Shopping & Payment System is one of the fastest growing e-commerce businesses in India. The project is divided into three inter-operable components (Model, View, and Controller). We are using technology JSP/JSTL for front view, Servlets for controller, EJB for Model and JPA for back end connection. Using above technology the on-line system will be more secure, reliable, and reusable and can handle more number of traffics.*

Keywords: Shopping System, JSP/JSTL, SERVLET, EJB/JPA technology, MVC pattern.

1. Introduction

The term e-commerce is refers to the buying and selling of products or services over the Internet. The online shopping is part of the e-commerce. For making the shopping system secure and reliable we need to be carefully about two things **first** is carefully design the system and **second** is which technology should we use for making the system more secure, reliable and handle more load at a time. In this paper I am going to explain simple yet effective online shopping application that demonstrate various feature of java web and EE development. What is the existing system and why we need new system, that differentiate I have demonstrate very simple way.

2. System Study

2.1 Existing System

- 1)Inefficient in meeting the growing demands of people
- 2)The basic problems with the existing systems are the non-interactive environment they provide to the user
- 3)The use of traditional user interfaces which make continuous post backs to the web server; each post back makes a call to the server, gets the reply and then refreshes the entire web form to display the outcome. This scenario adds an added trade off causing a delay in displaying the outcome
- 4)A search engine that would display the outcome without allowing the users to further filter the results based on various parameters.
- 5)Use of traditional and non user friendly interfaces that are hard to use
- 6)Time consuming

2.2 Proposed System

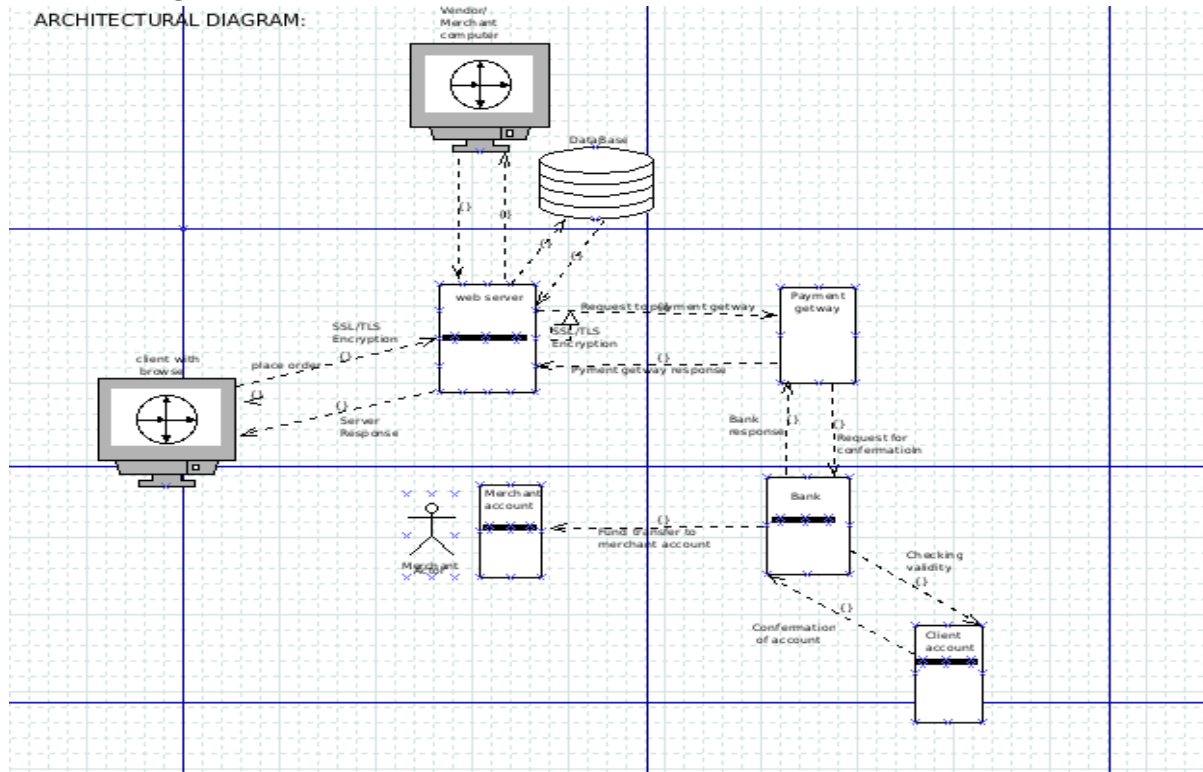
- 1)The purpose of this Online Shopping Application is to allow the user to play with the search tool and create different combinatorial search criterion to perform exhaustive search.

- 2)Making the application AJAX enabled gets rid of these unnecessary delays letting the user to perform exhaustive search. The users of this web application can simply feel the difference between the Ajax empowered user interface vs. usual user interface.
- 3)Provide Interactive interface through which a user can interact with different areas of application easily.
- 4)A search engine that provides an easy and convenient way to search for products specific to their needs. The search system would list a set of items based on the search term and the user can further filter the list based on various parameters.
- 5)Sitemap to help visitors find what they are looking for.
- 6)Provide Drag and Drop feature thereby allowing the user to add products to or remove products from the shopping cart by dragging the products in to or out of the shopping cart.
- 7)Comparing feature of the products for batter choice
- 8)Providing batter security.

3. Designing Application

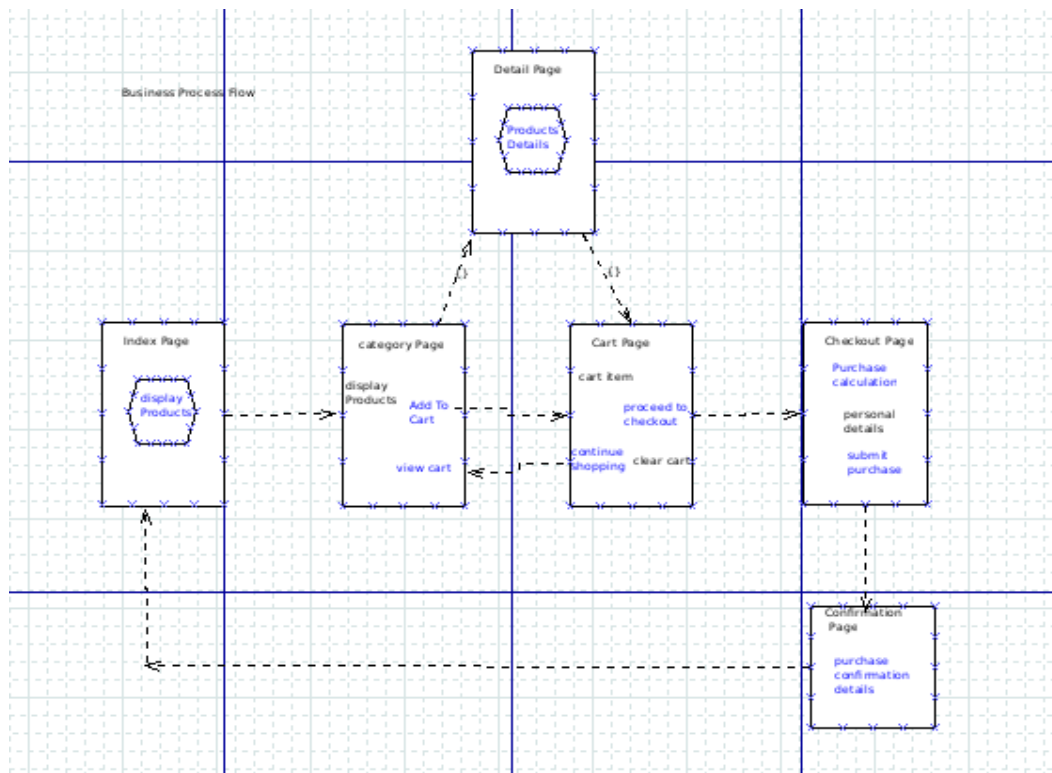
The application design is based on a real-world scenario. In this application we breakdown the requirements into a set of implementation tasks, and structure the application so that the responsibilities and interactions among functional components are clearly defined. We discuss the MVC (Model-View-Controller) design pattern. We use JSP, Servlet, EJB, JPA technology to the MVC architecture.

3.1 Architectural Diagram



3.2 Business Process Flow

The figure displays the visual and functional components of each page, and highlights the main actions available to the user in order to find the way through the site to absolute a purchase.



- **Model:** Represents the business data and any business logic that govern access to and modification of the data. The model notifies outlook when it change and lets the view query the model about its state. It lets the controller right to use application functionality encapsulated by the model.
- **View:** The view renders the contents of a model. It gets information from the model and specify how that data should be presented. The view also updates data presentation when the model changes. An analysis also in advance user input to a controller.
- **Controller:** The controller defines application behaviour. It dispatches user requests and selects views for presentation. The controller interprets user input and maps them into manners to be performed by the model. In a online application, user inputs are GET and POST requests. A controller selects the observation to show

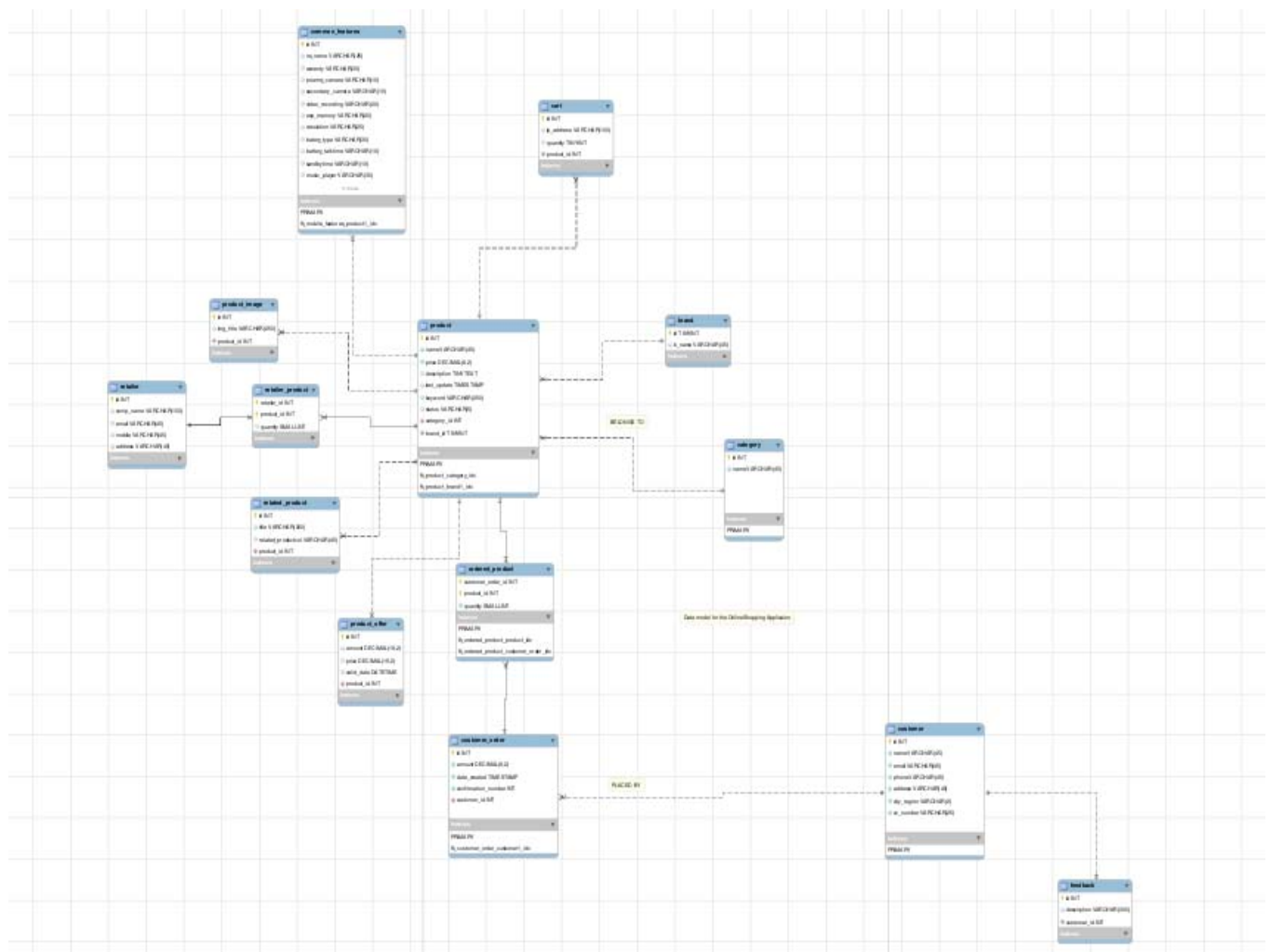
The diagram illustrates the MVC (Model-View-Controller) architecture. It is divided into three main sections by vertical lines:

- Client Layer:** Contains the **Client/ Browser** (represented by a circle).
- Application Layer:** Contains the **controller/ servlets** and **view/ jsp** (both represented by rounded rectangles), and a large vertical container for the **EJB** and **JPA** (both represented by ovals).
- Database Layer:** Contains the **Database** (represented by a cylinder).

The interactions are as follows:

- Client/ Browser** sends requests (dashed arrow with `{}`) to **controller/ servlets**.
- controller/ servlets** sends responses (dashed arrow with `{}`) back to **Client/ Browser**.
- controller/ servlets** sends requests (dashed arrow with `{}`) to **view/ jsp**.
- view/ jsp** sends responses (dashed arrow with `{}`) back to **controller/ servlets**.
- controller/ servlets** interacts with **EJB** (solid arrow).
- EJB** interacts with **JPA** (solid arrow).
- JPA** interacts with the **Database** (solid arrow).

The data model should enclose all the logical and physical design parameters required to generate a script using the Data Definition Language (DDL), which can then be used to create a database.



962

Enterprise JavaBeans technology as a "server-side component architecture that "enables rapid and simplified development of distributed application, also transactional, safe and portable applications. We can pertain EJBs (i.e., Enterprise beans) to the application, and the services provided by the technology remain transparent as a developer, thus eliminate the old and often error-prone task of adding a lot of boiler plate code which would otherwise be required.

- **Pooling:** For each EJB component, the EJB platform also creates a pool of item instances that are shared by customers. The EJB pooled instance is only allowed to be used by a single customer. As soon as an occurrence is finished servicing a customer, it is returned to the pool for reuse instead of being frivolously discarded for the garbage collector to reclaim.
- **Thread Safety:** EJB makes all types of components thread-safe and very much performance in ways that are completely unseen. This means that you can write your server components as if you were developing a single-threaded desktop application. It doesn't matter how complex the element itself is; It will make sure of thread-safety.
- **Transactions:** EJB supports declarative transaction management that helps you add transactional behaviour to components using simple configuration instead of code. In effect, we can designate any component method to be transactional. If the process completes normally, EJB commit the transaction and make the data change made by the method lasting. Otherwise the transaction is going to roll back
- **Security:** EJB chains combination with the Java Authentication and Authorization Service (JAAS) API, so it is easy to secure an application using simple configuration instead of cluttering up your application with security code

6. Securing the Applications

When securing web applications, there are two primary concerns that need to be addressed:

- Preventing unofficial user from acquirement access to confined content.
- Preventing confined content from read while it is being transmitted.

The first concern, access control, has typically involves two steps: (1) determining whether a user is who he or she claims to be (i.e., authentication), and then (2) either granting or denying the user access to the requested resource (i.e., authorization). A simple and common way to implement access control for web applications is with a login form that enables the server to compare user credentials with a pre-existing list of authenticated users.

The second alarm, protecting information while it is in transit, typically involves using Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL), in order to encrypt any data communicated between the client and server.

We generally need to secure the application in the following ways:

Step1: Set UP a authentication form for the administration console that enables staff member's access to the console's services, and blocks unauthorized users.

Step2: Organize secure information transport for both the customer checkout process, and for any data transmitted to and from the administration console.

7. Conclusion

With today's high tech business and E-commerce environment, it is crucial to have the capability to protect information assets by implementing security measures. Losses of huge amounts of money and System damage are examples of the negative effects resulting from weak security measures. Security threats cause serious incident to e-commerce firms such as revenue loss, reputation damage, legal consequence and loss of market share. Therefore, e-commerce companies should use proper techniques to secure their system and increase user awareness of those threats. To defeat the security obstacle for adoption in e-commerce in developing countries, decision-makers and IT Professionals should enhance the security of online payment and assure and educate the people about conducting online transactions. Dealing with the security issue will build and strengthen the trust in online transactions and lead to have a safe e-payment gateway for businesses. This will increase confidence in public and business to conduct online payment safely.

References

- [1] McAfee study: online security fears affect online shopping.
- [2] Ecommerce
- [3] Journal.
- [4] June29,2009.[Cited:0114,2011.]
- [5] [http://ecommercejournal.com/news/16510_mcafee_study_online_security_fears_affect_online_shopping]
- [6] M. J. Schniederjans, Qing Cao. e-commerce Operations Management.
- [7] s.l. : World Scientific Publishing Co., 2002.
- [8] BOILARD, ROBERT. HOW EXACTLY DOES ECOMMERCE
- [9] WORK? I4Market. [http://www.i4market.com/articles/d347.html]
- [10] Last retrived on March 15,2011
- [11] F.Belanger, Janine S. Hiller, Wanda J. Smith. Trustworthiness in
- [12] 2002, The Journal of Strategic Information Systems, pp. 254-
- [13] 270.Scientific Publishing Co., 2002.
- [14] L. Clemer. Information Security Concepts: Confidentiality, Integrity,
- [15] Availability, and Authenticity. May, 2010
- [16] S.M. Furnell, and Karweni, IT Security implications of electronic
- [17] commerce: a survey of consumers and businesses..., Internet

- [19] Research: Electronic Networking Applications and Policy, 1999, pp.
[20] EJB 3 in action addition 2012

Author Profile



Ripan Kumar Basak is M.Tech (Computer Science) Scholar at SIT-JNTUH, Kukatpally, Hyderabad, Telangana, India



Dr. Avula Damodaram is Professor of CSE and Director, University Academic Audit Cell, Kukatpally, Hyderabad, Telangana, India