

Understanding Architecture and Framework of J2EE using Web Application

Devadrita Dey Sarkar ,Anavi jaiswal , Ankur Saxena

Amity University,UTTAR PRADESH
Sector-125, Noida, UP-201303, India

Abstract: This type of research represents an architectural procedural approach to design a Web application using Java framework. This framework is used to develop MVC to separate business logic and presentation logic. Java has three different platforms, or flavors, and each addresses certain programming requirements. The J2EE provides the environment to develop enterprise applications or services using multitier architecture. In this highly intensified technology the need for scalable, efficient, faster solutions for information management has increased. The JEE technology is rightly apt for meeting these requirements. Our research study show that applying multiple frameworks to design the Java application using MVC concepts makes applications easier compare to a single framework. In recent trends, more research papers have been proposed which employ a new and fast process to implement Web architecture and to avoid framework.

Keywords— Architecture,J2EE, JSP, framework, MVC, ,Servlet

I INTRODUCTION

Java is one of the most commonly used and mature programming languages for building enterprise applications. Over the years, Java development has evolved from small applets run on a Web browser to large enterprise distributed applications run on multiple servers. Now, Java has three different platforms, or flavors, and each addresses certain programming requirements.

J2EE (Java 2 Platform, Enterprise Edition) is Sun's preferred Java platform for multi-tier enterprise applications. It simplifies enterprise applications by basing them on standardized, modular components, by providing a complete set of services to those components, and by handling many details of application behaviour automatically, without complex programming [1].

J2ee uses a multi-tier distributed application model. There are three tiers in the typical J2EE application model: Web presentation tier, business logic tier and data tier. [2]. Java Platform, Enterprise Edition or Java EE is Oracle's enterprise Java computing platform, the platform provides an API and runtime environment for developing and running the enterprise software . It also includes network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications. Java EE extends the Java Platform, Standard Edition (Java SE) which provides an API for object-relational mapping, distributed and multi-tier architectures, and web services. The platform incorporates a design based

largely on modular components running on an application server, the software for Java EE is primarily developed in the Java programming language. The platform emphasizes convention over configuration and annotations for configuration in which optionally XML can be used to override annotations or to deviate from the platform defaults.

Web is the very complex issues these days. Since the desire of the companies and organizations are increasing so the complexity and the performance of the Web programming matters. Complexity with the different types of communication devices is increasing [3]. The business is demanding applications using the Web and many communication devices so that many of the companies use frameworks for making the development of their applications easier. The business today demands Web applications to advertise its company so it is very important to take care of the architecture used in development of the application. Framework can be considered as a set of functions helping the developers in creating the applications [3] . So with the increase load of the data on the internet we have to take care of the architecture issue [4][3].

MVC: Model-View-Controller ("MVC") is architectural design pattern for interactive applications. MVC organizes an interactive application into three separate modules [5] [5].it is a design pattern originally pioneered in the olden days of smalltalk. Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business logic without affecting the other. The MVC divides the application into three associated parts: Model, View and Controller, which makes the system development simple [6][6]. Controller: Handles navigation logic and interacts with the Service tier for business logic2. Model: The contract between the Controller and the View Contains the data needed to render the ViewPopulated by the Controller 3. View: Renders the response to the request Pulls data [7][7].

In this architecture (figure 1.2) browser send a request to jsp and jsp check setter and getter method inside JavaBean. Bean map the data from database and finally jsp send a response to browser [8][8].

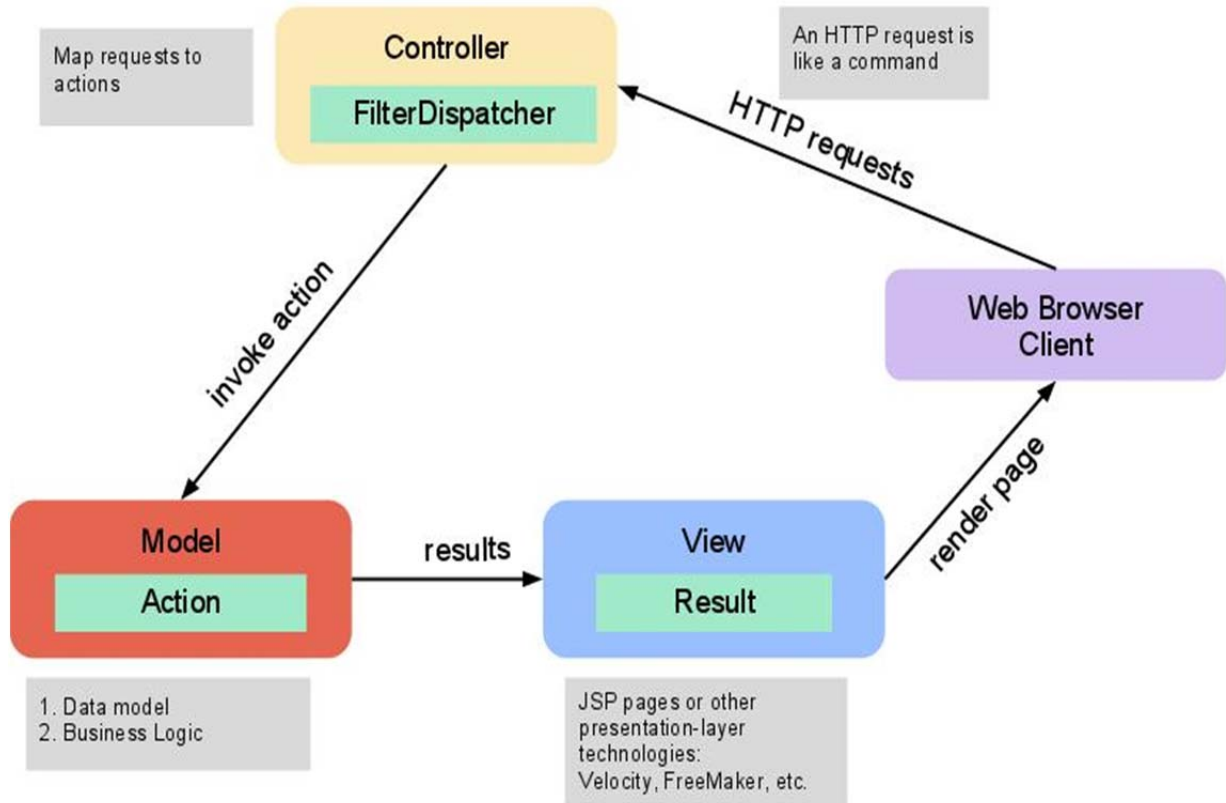


Figure 1.1 MVC Architecture

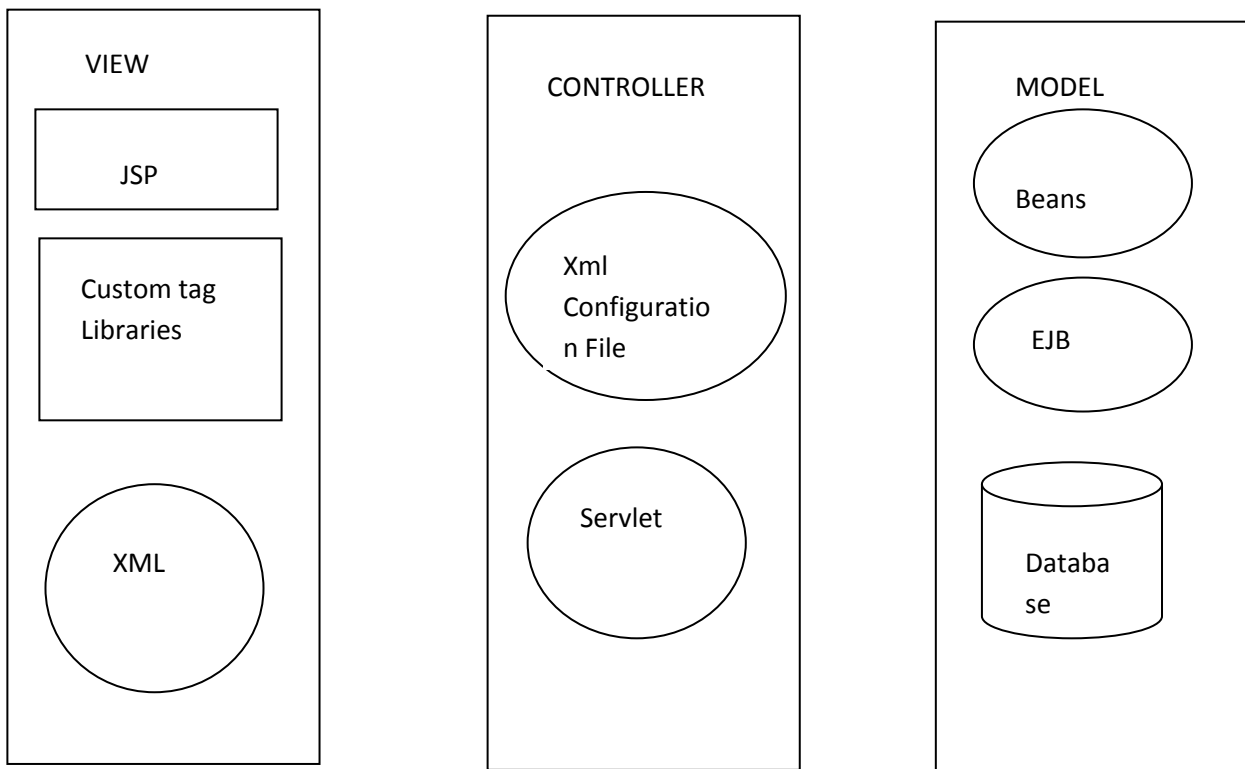


Figure 1.2 MVC2 Model

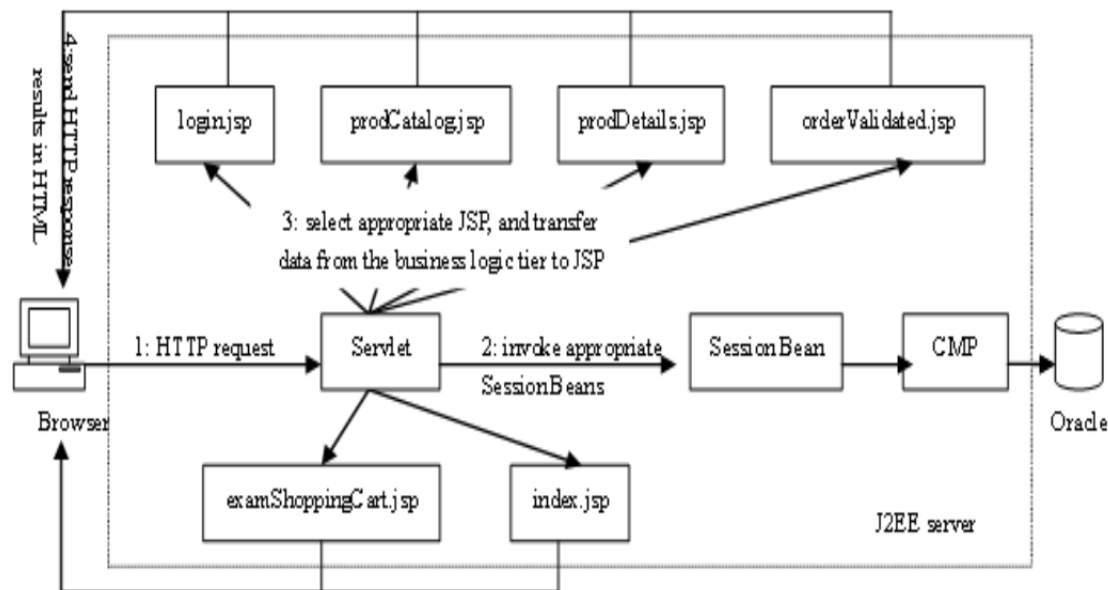


Figure 1.2 Architecture of Cart

II RELATED WORK

Web and Internet is ever growing area and the demands for the applications are growing. A single framework is not capable to handle the architecture of the application. To meet the current requirement of the applications it's necessary to design a architecture to implement the Frameworks.

Struts Framework have been designed and developed for the front end control of the Web applications. It provides the various features for the applications that interact to the users. It also follows the MVC 2 design features. Spring Framework is the designed to handle the various tasks. The spring work for the desktop and internet based applications also. It follows the principals of the MVC 2. [9]

The simultaneous use of the Struts and spring frameworks in the single application with the applying the MVC Design principals so that we can Improve the performance of the applications.

Struts Framework consists of three major blocks, Described in brief as follows

First is The View Block which controls the presentation part of the complete model. This contains following JSP files which you write for your specific application set of JSP custom tag libraries Resource files for Internationalization

Second Block is representing the Controller. This is for navigation the complete application. This contains XML configuration files; it contains the tags for the navigation of the paths.

Third Block is the Model. This part do the work of the Business Logic, Fetching and storing data to the database .This contains following Java Beans Enterprise Java Beans Database. Following figure shows the working of the components in the Struts framework [4].

In this section a computer components distribution system has been designed using the architecture proposed in the foregoing sections. The business features include user authentication, online product catalogs, shopping cart,

special functions, order generating, checkout features and Email confirmation. [10]According to system requirements the following Servlet and JSP documents are defined: login page, home page, product catalog page, the product details page, View Cart page, order confirmation page. The framework is shown in Figure 1.4. The MVC pattern is adopted in Figure 1.4, that is, business logic tier represents the Model, JSP documents are the View, and Servlet serves as the Controller. There're no direct calls among JSP documents. The Servlet receive all HTTP request, and call the appropriate business logic model, then run different JSP documents according to the processing results. End clients can receive HTML response which can be seen in a browser. In Figure.1.4 the steps of accessing the database are: firstly Servlet calls the Session Bean, then the Session Bean calls entity beans, finally the database is operated by the Entity Bean.

This approach not only separates the business logic from the data persistence logic and simplifies the development process, but also develops a stronger e-Commerce system with scalability and maintainability [11].

III IMPLEMENTATION

A web application framework is a software framework that is designed to support the development of dynamic websites, Web applications and Web services. The framework aims to alleviate the overhead associated with common activities used in Web development. A Web Application Framework (WAF) is a reusable, skeletal, semi-complete modular platform that can be specialized to produce custom web applications , which commonly serve the web browsers via the Http's protocol.

- WAF usually implements the Model-View-Controller (MVC) design pattern, typically in the **Model 2** architecture to develop request-response web-based applications on the Java EE and .Net models.

- Virtually all web applications have a common set of basic requirements, such as **user management** e.g., **secure user login, password recovery**, group management, and access authorization.
- A Web Application Framework usually includes all these functionalities, refined through hundreds of production deployments, freeing developers to focus on the needs of their specific application.
- WAFs store important data in a relational database and they interact with users via a web-based user interface.
- Any application written on top of a Web Application Framework can transparently and immediately take advantage of these basic services.

IV SAMPLE WEB APPLICATION

- 1) index.jsp a page that gets input from the user.
- 2) ControllerServlet.java a servlet that acts as a controller.
- 3) login-success.jsp and login-error.jsp files acts as view components.

web.xml file for mapping the servlet.

index.jsp

```
<form action="ControllerServlet" method="post">
Name:<input type="text" name="name"><br>
Password:<input type="password" name="password">
<br>
<input type="submit" value="login">
</form>
```

Controller

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class ControllerServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        String name=request.getParameter("name");
        String password=request.getParameter("password");

        LoginBean bean=new LoginBean();
        bean.setName(name);
        bean.setPassword(password);
        request.setAttribute("bean",bean);

        boolean status=bean.validate();

        if(status){
            RequestDispatcher rd=request.getRequestDispatcher("login-success.jsp");
            rd.forward(request, response);
        }
    }
}
```

```
    else{
        RequestDispatcher rd=request.getRequestDispatcher("login-error.jsp");
        rd.forward(request, response);
    }
}
```

LoginBean.java

```
package app1.devanasan;
public class LoginBean {
    private String name,password;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public boolean validate(){
        if(password.equals("admin")){
            return true;
        }
        else{
            return false;
        }
    }
}
```

File: login-success.jsp

```
<% @page import="ank.LoginBean"%>
<p>You are successfully logged in!</p>
<%
LoginBean bean=(LoginBean)request.getAttribute("bean");
out.print("Welcome, "+bean.getName());
%>
```

File: login-error.jsp

```
<p>Sorry! username or password error</p>
<% @ include file="index.jsp" %>
```

File: web.xml

```
<servlet>
    <servlet-name>s1</servlet-name>
    <servlet-class>app1.devanasan
    ControllerServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>s1</servlet-name>
    <url-pattern>/ControllerServlet</url-pattern>
</servlet-mapping>
</web-app>
```

IV FUTURE WORK

Open technologies are the best to attract the academic and research scholar to work. J2EE is the vast field now a day, its open technologies also. Architecture is never fixed its goes on changing with the change in the technology. There are many frameworks available to work with J2EE technologies, Single frame is never sufficient to provide the complete solution with all essential features of the application. There is a lot of scope to work further with many other frameworks to implement and enhance the MVC architecture.

CONCLUSION

This paper identifies the primary role of J2ee and its applications and also the role of MVC architectural framework. In the MVC framework, the controller mediates the flow of data between model and view objects in both directions. View objects use the controller to translate user actions into property updates on the model. In addition, changes in model state are communicated to view objects through an application's controller objects. Multiple framework architecture works better as compare to any single framework architecture with the effective of the multiple frameworks for the development of the large scale applications.

REFERENCES

- [1] A. Saxena, "Web Based Custom Validation Using Framework in Java," *International Journal of Computer Science Trends and Technology (IJCSIT)*, 2015.
- [2] Y. Wang, C. Guo and L. Song, "Architecture of E-Commerce Systems Based on J2EE and MVC Pattern," *IEEE Computer Society*, 2009.
- [3] M. G. P Gupta, "MVC Design Pattern for the multi framework distributed applications using XML, spring and Struts framework," *International Journal on Computer Science and Engineering*, pp. 1047-1051, 2010.
- [4] R. Garg, Y. Sood, B. Kottana and P. Totlani, "A Framework Based Approach for the Development of Web Based Applications," *World of Computer Science and Information Technology Journal (WCSIT)*, pp. 1-4, 2011.
- [5] A. Saxena, "Struts based Approach for the Development of Java Applications," *Journal of Software Engineering Tools & Technology Trends*.
- [6] T. Dey, "A Comparative Analysis on Modeling and Implementing with MVC Architecture," *International Conference on Web Services Computing (ICWSC)*, vol. 44, 2011.
- [7] G. P and G. M, "Spring Web MVC Framework for rapid open source J2EE application development: a case study," *International Journal of Engineering Science and Technology*, vol. 2, no. 6, pp. 1684-1689, 2010.
- [8] A. Saxena, "Java Server Pages," *MJP*, pp. 341-342, 2012.
- [9] D. Brown, M. C. Davis and S. Stanlick, "Struts in Action," *DreamTech*, 2010.
- [10] R. J. A Saxeian, "Securing Confidential Data using Java/J2EE", *International Journal of Science Technology and Management*, vol. 2, no. 3, 2011.
- [11] A. Saxena, "Handling of Synchronized Data using Java/J2ee," *International Journal of Management, IT & Engineering*, vol. 1, no. 7.