

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

Product Defect Fixing - A Java Web Application Development Using Vaadin Framework

Shruti Ghatnatti¹, Namita Mirjankar²

Lecturer, Bachelor of Computer Applications, K.L.E'S BCA College, Vidyanagar, Hubballi, Karnataka, India¹

Lecturer, Bachelor of Computer Applications, K.L.E'S BCA College, Vidyanagar, Hubballi, Karnataka, India²

ABSTRACT: Web Applications in today's fastest growing technologies has evolved with the great impact on the fields of business, information industry. With a more secure and a user friendly interface of a web application development is a challenging aspect for developers to deliver the web application with high performance, security, usability. Vaadin Framework, which is a Java Web Application Development Framework, is used to create and design a web application that enables high performance of data rendering on the network. Vaadin Framework uses Google Web Toolkit for interface of the client models in web application. The Vaadin Framework is also used for Rich Internet Applications. This paper details on the architecture of the Vaadin Framework and implementation in Product Defect Fixing.

KEYWORDS: Vaadin Framework (VF), GWT, Web Application, Rich Internet Application (RIA).

I. INTRODUCTION

The Web Application renders set of information stored on the remote machine which is delivered through Internet using a browser interface. This focused more on client-server architecture where the work (task) was divided by several programs on different machines to serve the purpose of information rendering. The client-server model used the distributed application structure, where the task was rendered by a program stored on the machines with the service they provide. These service providers are termed as Server. The service response is neatly organized and presented by the service requester called Client. Server rendered a new page of information every time when the request is sent and this information is presented on the client. Server maintains a track of such pages which is termed as state information. Traditional client-server model used Synchronous mode of data transformation. Synchronous mode made client in a wait mode while server is computing the request and responds back. This approach of request and respond had a major drawback that made client interface a less user friendly and limiting the ease of access on the web application.

The applications with traditional client-server model are been replaced with new technologies that gives a Rich Internet Application (RIA) with a high quality maintenance of web based interfaces. The new technologies concentrate on the productivity of the web application with the logic that is used to render the information on both client side and server side effectively. Such applications use AJAX (Asynchronous JavaScript and XML) an Asynchronous mode of data transferring, where client request and send information using variety of formats like HTML, JSON, XML and also text files.

Vaadin Framework (VF) is a framework used to create such AJAX based web applications using JAVA as a platform. This framework uses Java language to develop web applications. VF is composed of 2 models: 1. Server Model. 2. Client Model. 1. Server Model development is robust which allows the logic solely being developed on server-side, with full utilization of AJAX-based Vaadin Client Engine to render the information. 2. Client Model allows the development of user interface components termed as widgets using Java. These components are compiled to

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

JavaScript and rendered on browser. Since the Server Model does the job of information gathering and Client Model organizes this information and represents it in a user friendly interface on the browser, it features all the characteristics of RIA. The two models share the same user interface widgets, styles, and back end data. VF has built-in GWT compiler for converting the Java Code into JavaScript. GWT has its own Library Classes which are used for conversion. Using the adverse technology like GWT, AJAX, and HTML, VF delivers a RIA. This paper attempts to describe the architecture of VF and application developed using VF for business solution: Product Defect Fixing.

II. RELATED WORK

The traditional client-server application targets to resolve one request at a time. This reduces the quality of performance of web application which renders the information. The web application which serves the purpose of high quality data rendering, along with maintaining the data respective to server and client side separately is a framework built in Java Language called as Vaadin Framework (VF). Using VF allows developers to have a flexible separation of client and server side code that increases the performance of web application.

The large scale software application/product that has heavy data to be computed comes in different release versions. These software applications/products are delivered end – to – end to the customer. The customer who purchases the software application/product under goes the manual work of maintaining the records of the software application/product on vendor side. The software application/product has to be updated on the customer side as and when the versions are released. This takes too much time for the customer as it is end – to – end delivery and holds up the work on the customer side. The software application/product records have to be highly secured since the software application/ product is completely owned by an organization and also has data that can be hacked easily from the customer side. The current process of maintaining such data of software application/product is as follows:

1. The version update termed as fixing the bug in the software product/application is done manually.
2. Manually the bug is detected using the sub version control tools like svn rep, git repo, perforce.
3. Checking the last version/fix updated and then manually creating the packages for the fixes release.
4. The customer then uses the packages and updates the software application/product version.

Thus, the current system is completely manual and tedious to work on. This paper attempts in defining an automation process to do the fixing of software application in less time and in more effective way.

III. ARCHITECTURE OF VAADIN FRAMEWORK

The Architecture of VF (Fig 1) is always updated with new versions. New classes are been added to main component class for effectiveness of the application. The logic remains same on the client-side as well as server-side. The general architecture has 2 main components: Server Model and Client Model.

A. Server Model:

Server Model is hooked upon the program that delivers the response to the request received. It takes help of 4 sub components: Vaadin Servlet, UI Logic, Built-In and Custom Widgets, and Themes. The Server-Side application is executed as a Java Servlet. The Vaadin Servlet uses the core Java Servlet API on the web server. The servlet receives the request from the client through HTTP and serves them with the required response. The servlet first identifies the events associated with the request and collects the results by the event handlers. The event handlers resolve the back-end request which involves the logic of database and business logic. The UI Logic renders the servlet response resolved by the handlers and organizes the interface by using the widgets and themes associated with the UI Logic. The *widget* is a component of the User Interface which gives the graphical effect of the action handled. VF Server makes best use of AJAX in development of the client widgets. Hence Server provides us the clear separation layer between client and back-end.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

B. Client Model:

Client-Model serves the purpose of the converting the Java Servlet response into JavaScript. This is achieved by using the GWT compiler. GWT provides the set of libraries which is used to deliver the compiled application. The Client Model of VF has two main components that deliver the response to the RIA. The Vaadin Client Side Engine delivers the main counterparts of the server response and organizes them in a bundle of widgets through browser. AJAX technique in Vaadin Client-Side Engine helps VF utilize the computation of client application and make web application more interactive and responsive. The Client Model can also be integrated with the latest interface technology like Twitter Bootstrap and so on. Thus the Vaadin Client Engine renders the User Interface of the server application on the browser.

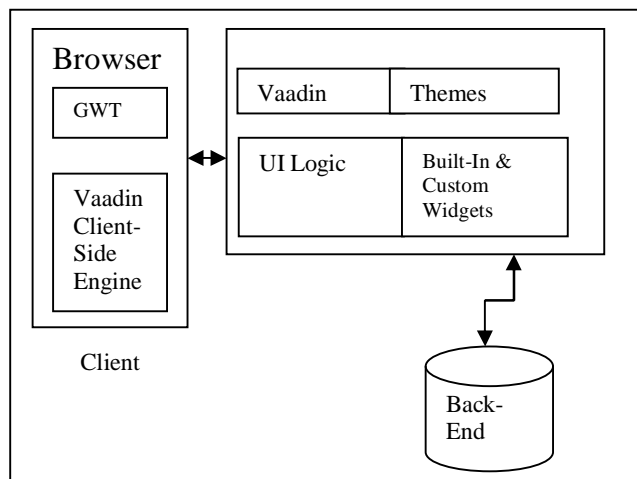


Fig 1

VF allows the flexibility and clear segregation between Server and Client programs. The web application developed using VF makes it a high performance, accurate and deliverable RIA. The VF provides vast collection of widgets and themes to design the application of your own choice.

IV. PROPOSED SYSTEM FOR VAADIN FRAMEWORK

Software Applications are high on demand in every business aspects. The products are updated with the latest versions and fixes. Once the maintenance of these products is started, we should make sure that it doesn't affect the existing product features. The error or fault in the product makes the product to behave in an unexpected way. Consider the X product which is delivered to the customer. The information of this product is stored in the JIRA atlassian. The X product updates feature y under the development area of the product. The y feature has to be updated on the customer side through JIRA atlassian. The customer cannot access y feature from X product with the old version of X. The product has a defect in accessing with it. Hence it is a bug in the product X. "Product Defect Fixing" is a web application identifies bug in a product and fixes the product with the new features without affecting the functionality of product.

The Fig 2 explains the architecture of the "Product Defect Fixing"; it is developed on the VF platform. The web application developed using VF includes interaction of the client and server, where major functional part of user interface like binding the information, control over the components is done on the client part of VF, storage of product information and event handling is done on the server.

The Product Defect Fixing web application uses the JIRA Rest API's to store the information of the product and keep track of the features and version of the product using a unique identifier called as iTrac. The client gets the

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

updates of the product by tracking this iTrac on the web application. The application maintains the session of each user who purchases the products.

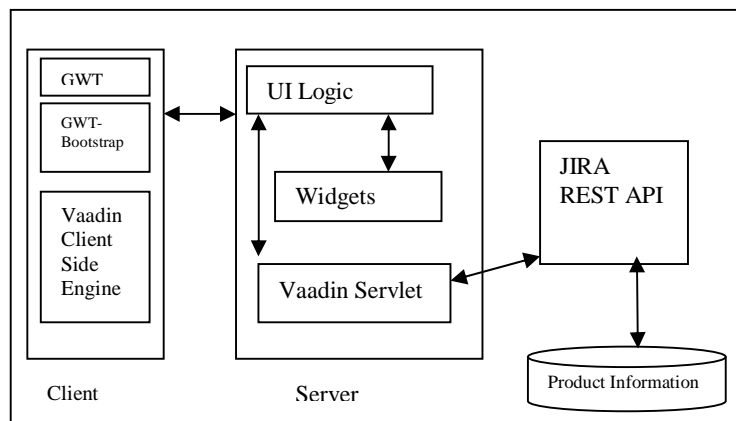


Fig 2

The JIRA Rest API's are used to update the information of the product. When user updates the product information, the action is completed with the help of Vaadin Servlet. The servlet communicates through the JIRA Rest API and updates the same on the back-end. The fixes of the product is sent in the form of a plain text file termed as *read me file* to the customer with the installation steps associated with the file. The installation is explained in the Fig 3.

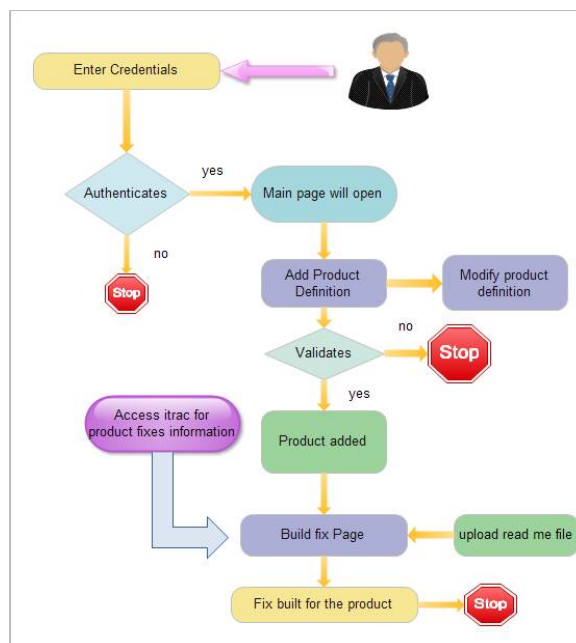


Fig 3

The workflow of the product is described in Fig 3. The user of the product logs in to the web application with the valid credentials. The session is maintained for each user. The user adds the product information if it's not updated.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

The user can proceed with the fix building for the product. The user who has the bug in product raises an iTrac for the fixes. Using this iTrac information of the product, the build for the fix has to be processed. Once the building is done on the server side, the file gets downloaded on the user side with the latest fix information on the product. This file is termed as *read me file*. The file contains the fix details and the information how to install this fix on the product. The user now can use this file to fix the bug on the product.

V. CONCLUSION

The old traditional web application served the purpose of the client-server communication but lagged in terms of performance, derivability costs and the quality of user interface. RIA replaces the old traditional web application by providing the handful flexibility in organizing the user interface and server programs separately. The RIA uses the technology like AJAX to make the application flexible enough to communicate between client and server. With the help of this flexibility of Java Programming in Vaadin Framework, a web application is developed for the business solutions to provide the defect free product for the customers.

REFERENCES

1. Vaadin Framework: <https://vaadin.com/docs/>
2. Oracle Docs on Client – Server Architecture: https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch20.htm
3. JIRA REST API: <https://docs.atlassian.com/jira/REST/latest/>