Information Repository System For Web Base Applications

Aniket Bansal,Rijwan Khan

U.G. Student ,Department of Information Technology, ABES Institute of Technology, Ghaziabad, Uttar Pradesh, India
Associate Professor ,Department of Computer Science & Engineering, ABES Institute of Technology, Ghaziabad, Uttar Pradesh, India

ABSTRACT*:* Purpose of this paper is to modernize the existing manual work done which took a large amount of time, with 40% of minimum probable mistakes. This paper is for transferring the legacy into web based application using **J2EE architecture.** The objective of this modernization is to port some part of functionally along with mandatory data rules implementation. The main business objective for the modernization initiative for Information Repository System is to replace current monolithic legacy system, with a business-centric IT model where the rights are given to administrator to modify the details in an easy way. Information Repository System allows the users to provide Version Control Reports to Application Teams, Development Teams and other Senior Managements in a simple and hassle freeway. The main scope of this paper is to provide administrative capabilities to the user and allow him to manage the Code Drops, and help to proceed with deployments and its validation smoothly.

KEYWORDS*:* Information Repository System, Middle Tier, Change Control Manager, Version Control Report, Database.

I. INTRODUCTION

Information Repository System (IRS), provides a stage where we can generate reports for different platforms like Front End (FE), TIBCO or digital in order to compare labels between any environments and make sure correct and valid labels are deployed in the subsequent higher environment and post drop will prepare the summary report which shows the number of labels which are in-sync / out of sync.

This paper is **JAVA/J2EE** based using **MVC Framework,** i.e. – Model, View and Controller. For UI framework, JQuery and HML5 has been used. IRS DB is generated through **My SQL 5.5.50.**This paper is based on the concept of **DevOps.**DevOps focusses on collaboration, integration & automation of tools that can control the programmable and dynamic nature of the infrastructure & services oriented technologies.
DevOps movement has also focused on **Version control** and automating of code deployments, configuration management, monitoring and provisioning, application resilience, and self-heal. Technology Stack in the System

JQuery, HTML5
JAVA – J2EE
MY SQL

II. METHODOLOGY

MVC Framework IN JAVA-J2EE is used to develop Automated Tool for VC Report – Information Repository System.

MVC framework is used to divide the different layers, the data access layer, business logic code and the GUI that is defined and designed to let the people interact with the different applications. This application has three parts:

1. **MODEL –** This is a part of the framework which store the data of the IRS Application, which may be in the form of databases, text data, files and/or other web resources.

2. **View –** This is the graphical user interface of the IRS Application. It contains different kinds of buttons, text boxes and other controls which let the user interact with the application.

3. **Controller –** It is actual back-end code constitutes the controller of the IRS framework. A controller is which controls the data input from the users, or the data outgoing to the user from the model.

This leads to enter a condition of checking, because the stream of data (input from or outgoing to the user) is every time checked on the controller part. This is the reason it make the data more consistent by removal of false data entry or unauthorized data deletion from the application's data source.

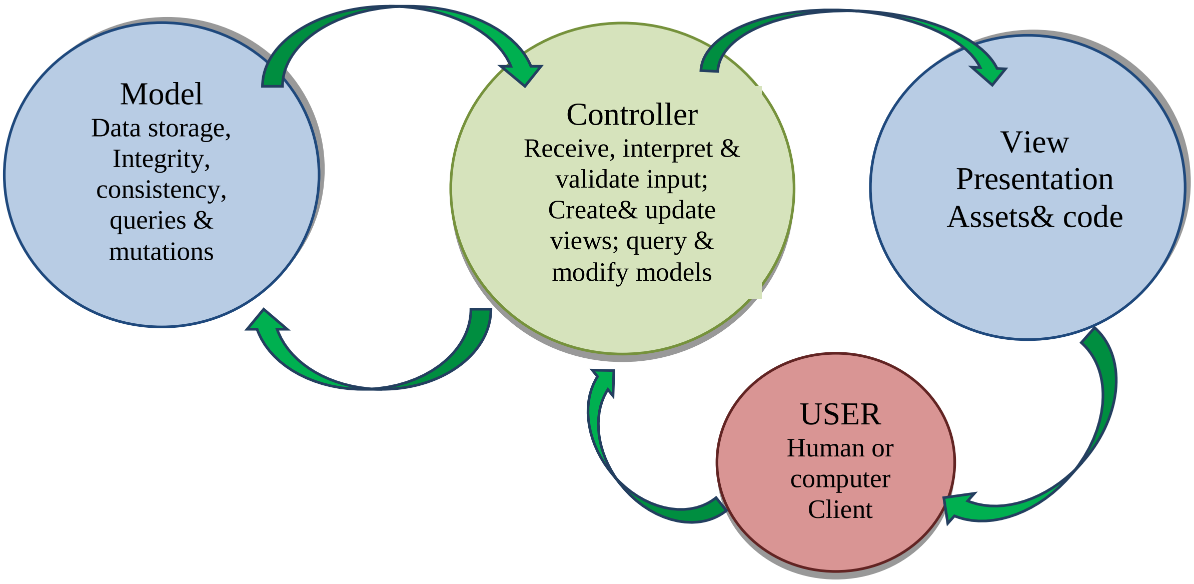

Fig 1.1 Block Diagram of MVC Framework

A. Implementation of MVC Framework

In Java-J2EE, we create packages. These packages authorize us to make collection of similar classes, or classes that are required in the same order or for the same cause. In our application we require to create an MVC pattern, for that we can create three different packages, Models, View and Controller, as discussed above.

After that we can add all of our Models, Views and Controllers in these packages and use them in different packages where ever needed.

This way, we can divide all of the sections into three different sections from one another, we can call them it in our sections; by importing the packages. In this way, the model would be left over to be used by the controller, view and should make a call to controller to get / update the data. Since we're working to develop the application first, and the note the changes by the application once it's been working.
Then there would actually be no requirement to follow, to create the view first then to create a controller to render the view and then finally to create a model for adding some data. We can start by creating any of the sections and then combine them into one main application and then make it run.

III. PROBLEM STATEMENT AND ITS SOLUTION

A. Problem Statement:

This paper was made with efforts which involved several steps to follow before final paper was completely submitted.

Steps were like:-

1. Running all the servers manually through putty. It consisted of about 20 servers. Took time for about 2 hours.
2. Running servers used to generate mails that contained all the details attached in the notepads. All the notepads used to be saved from all the mails in separate folders for FE, MT or Digital.
3. All labels/artefacts/versions were extracted manually for the 2 environments to be compared.

The above 2nd and 3rd step took around 2 hours for overall saving of notepad data and extracting of relevant details.

4. Once extracted, through formulas in excel, those were compared. Still manual intervention was required to check whether each and every label was matched properly. As formulas used to take incorrect data sometimes.

This step took around 5+hours to check and resolve the issues (if found)

5. After comparison was done, summary report was made keeping the count of all application. This step took 1 hour to prepare the table.

Precisely, Manual work on VC paper took 12-14hrs, of work, wherein a person had to extract all the artefacts, their labels with their versions for both environments to be compared, and then work on comparing them. The labels for any environment ranged up to 1500+ for all the 3 platforms- FE,MT and Digital, which took more resources in terms of time and efforts.

B. Solution:

IRS is a web based tool and will be in use for the change managers and their team at any point of time. This paper is designed to maximize the productivity by automating the version control report making, which was otherwise performed with hand (manually). By managing the work efficiency and the production, the system will meet the change manager's need while maintaining the work flow and easy to understand and use. It will help in reducing the work effort from 8-10 hours to 1 hour.

IRS will have below modules:

1. Front End Comparison
   - It will help us to select the 2 environment that need to be compared and submit the button to provide all FE labels compared of the selected environment. Will provide summary report as well for all applications.
2. TIBCO Comparison
   - It will help us to select the 2 environments that need to be compared and submit the button to provide all MT labels compared of the selected environments. Will provide summary report as well as for all applications.
3. Digital Comparison

- It will help us to select the 2 environment that need to be compared and submit the button to provide all Digital labels compared of the selected environments. Will provide summary report as well as for all applications.

4. VC Report Archive
   - It will archive all the previous reports in a separate tab from where the old reports can be retrieved as and when required.
5. Delete VC Report
   - It will help to select and delete any VC Report that is not of use.

IV. DESIGN

A. ARCHITECTURAL DESIGN

This architecture will represent the high level capabilities required for IRS implementation.

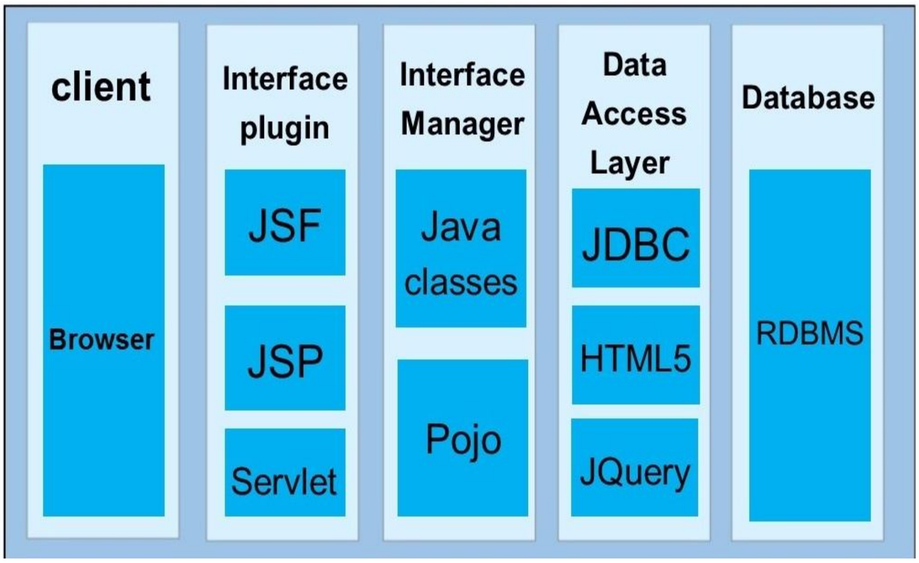

Fig 1.2 – Architectural Design of IRS Tool

ACTIVITY DIAGRAM

Activity diagram showing the flow of control in order to generate VC Report from this automated tool – IRS

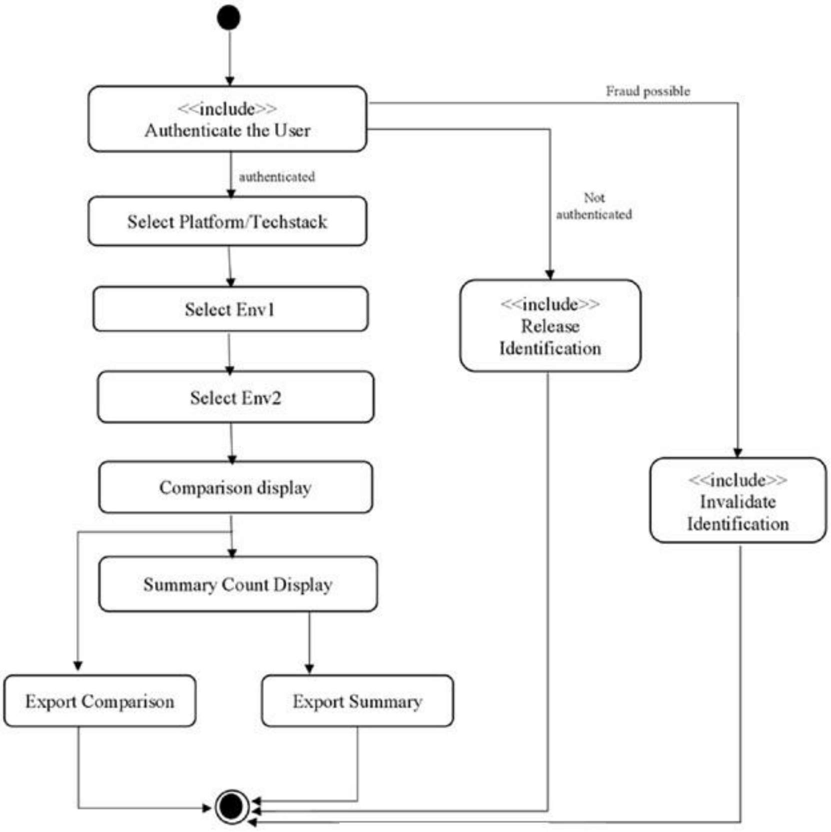

Fig 1.3 - Activity diagram of IRS Tools

USE CASE DIAGRAM

Following is the use Case Diagram of the IRS Application.

There are 2 actors involved in this system:

1. User
2. Admin

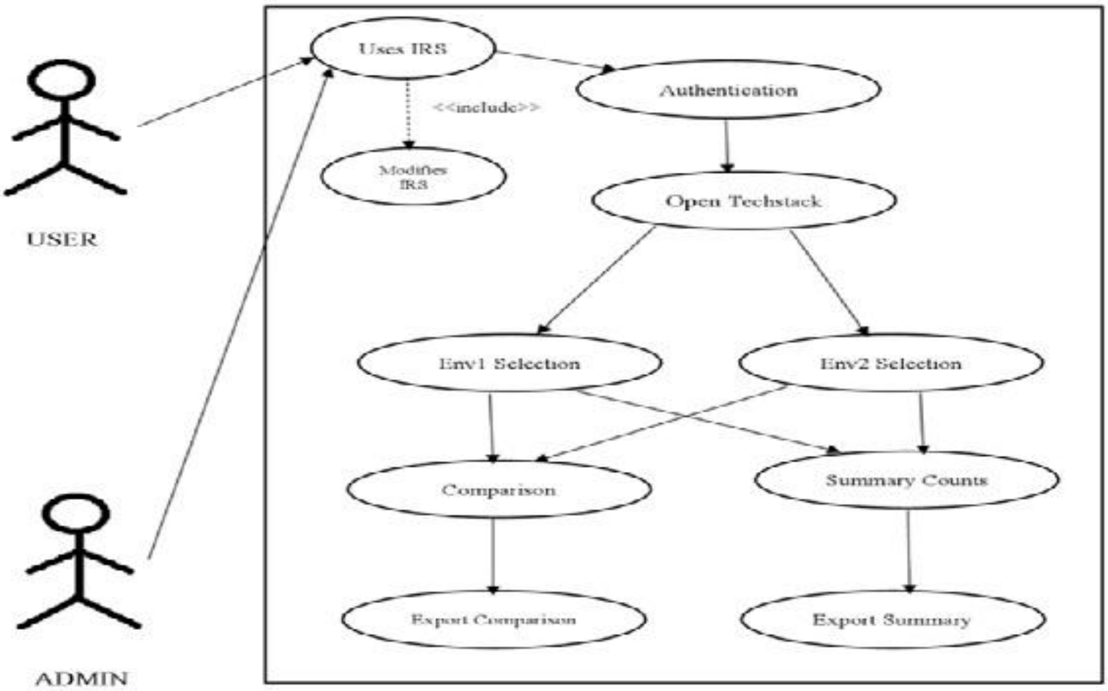

Fig 1.4 – Use Case Diagram of IRS Tool

V. EXPECTED BENEFITS

Real-time benefits to the customers.User Friendliness, as application has given various controls provided by the system Rich User Interface.High Manageability, as only the administrator can login to the application and do the required modifications.The system makes the overall management much easier and flexible, by managing the Code Drop deployments and other day-to-day deployments with the data provided by IRS.Reduction in manual efforts to 90%.Possibility of Human Errors have been totally dissolved now, as overall thing has been automated.

VI. CONCLUSION

Automated tool for Version Control: Information Repository System helps in successfully creating Version Control Reports for all the platform, i.e. – Front End, Middle Tier and Digital with its accurate summary table, as and when required. It helps us to download all these reports in excel format as well which can be further utilized for support purpose.Thus, it helps in managing the Code Drops in best possible way, by generating the reports and reflecting what all labels/versions are out of sync and deployment is required to make the 2 environments in sync.It also helps in reflecting what the latest label is deployed in the server for any environment and day to day deployment can also be validated through this. Manual work has been removed to a greater extent. Going into each and every server and getting out the details, all these workloads have been reduced lot. Additionally, it provides all the details with a proper UI on the screen.It is of great help for many different teams like – Application Teams, Deployment Teams, Change Management Teams, and various other higher management teams as well. They can use these data in order to fulfil their own purpose.

REFERENCES

[1] www.google.com
[2] http://guides.beanstalkapp.com/
[3] www.atlassian.com
[4] http://oss-watch.ac.uk/resources/versioncontrol
[5] http://ericsink.com/vcbe/vcbe a4 lo.pdf
[6] http://www.softwaretestinghelp.com/top-5-softwareconfriguration-management-tools/