ISSN No. 0976-5697

Volume 5, No. 4, April 2014 (Special Issue)

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Efficient Monitoring and Maintenance of Project over the Global

Seshagirirao Boddepalli[1]
Dept. of Information Technology
Aditya Institute of Technology And Management
Tekkali, Andhrapradesh , INDIA
seshagiriraob9@gmail.com

GRS Murthy[2]
Associate professor, Dept. of Information Technology
Aditya Institute of Technology And Management
Tekkali, Andhrapradesh, INDIA
murthy.grs@gmail.com

Abstract: In existing project tracking system uses Java Server Pages as a presentation layer, Java Database Connectivity to establish a connection with the database and Struts as a development framework. In this development procedure it takes lot of time for implementation of an application and if once deletes the database tables need to create all tables manually. So, it is difficult to maintain the database of an application. In JSP's we need to embed the java script and Hypertext Markup Language code along with their implicit objects, action tags and elements. In this paper we are proposing development of project monitoring application as a pipeline of work with client side Smart Google Web Tool kit as its user interface widgets library, Google App Engine Server is responsible to dynamically generate java script code. Object Relational Mapping technology is used to perform the database transactions and server side spring frame work which develops the service layer of an application. We have various relationships for different jobs which are maintained by relational tables which consists relative data. We have to develop queries according to the requirement, those queries help us to take solutions from main queries and queries make it normal to develop reports and forms. In the proposing system database tables are dynamically generating even if we delete all tables in the schema. Using a Data-Source as a shared data definition also greatly reduces redundancy between our user interface code and server side code increasing agility and reducing maintenance effort. We can achieve the development of an application without using scripting technologies which reduces duration of an implementation of an application. This provides an environment to monitor details about the current running project in various locations and it allows entering the details regarding that project. This achieves high efficiency in performing jobs.

Keywords: widgets, services, presentation, object relational mapping, information storage and retrieval.

I. INTRODUCTION

Project monitoring is need for any kind of running project which ensures enhancement in performance of the project, efficient resource utilization, reduces rework and misuse of finance. Efficient projects monitoring enhances reputation of the development company over the global market. The organization will get large number of clients and huge profit. Traditional web development uses Java Server Pages (JSP's) as a presentation layer. JSP is used to avoid the writing of java code but here we need to embed the java script and HTML along with JSP scriptlets, JSP declarations, JSP Expressions, JSP Directives and JSP Actions. Generally JSP is used to generate both dynamic and static web pages. A standard Java Data-Base Connectivity API is a set of classes and interfaces that can be used to develop java application to deal with databases. As part of JDBC API java soft has provided a set of interfaces within a package which is java.sql.*, here * represents Driver, Connection, Statement, Prepared Statement, Callable Statement, etc. These interfaces are used to establish a connection with multiple databases and for performing database operations.

We propose to use the smart Google Web Tool kit (smart GWT) as a presentation layer, this GWT-based framework has lot of comprehensive widget libraries that allows us to develop an application User Interface (UI) and for server-side data management also. In this we need not write the java script and HTML code. Google App engine server is responsible for generating the script code. Hibernate is used as Object Relational Mapping (ORM) technology to deal database transactions. Smart GWT is a sophisticated technology which is used to build and maintain more practical, reliable, portable and efficient web applications. Smart GWT application is developing based on 3 tier architecture from User Interface components and business logic to server side transaction handling. It provides practical and reliable solutions to project management a significant design of User Interface components helps us to minimize server load and maximize responsiveness and its server components are designed to meet requirements of high productivity user interfaces. Smart GWT reduced development time due to no need of doing debugging of an application and cross browser testing. In the development of User Interface of an application, smart GWT provides lucid, object oriented approach that preserves us from browser whims and bugs. Smart GWT can absolutely integrate with our already existing applications, portals. Smart GWT earmark us to develop huge projects with the above set of skills and revolutions in development of a web application and easily integrate with other technologies. Smart GWT provides conciliatory, common interfaces to integrate with service layer or data access layer that we can access with the help of java code.

II. ARCHITECTURE AND FRAME WORK

Architecture of smart GWT spans client and server enabling Rich Internet Applications (RIAs) that communicate transparently with our data and service tiers.

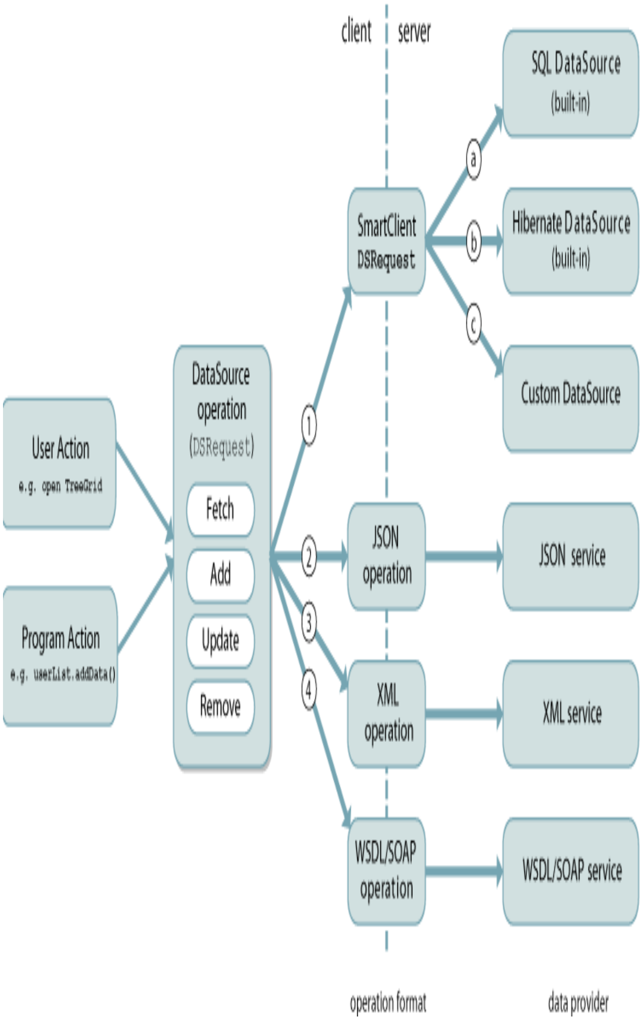


Figure. 1: Smart Google Web Tool kit Frame work

The client and server based components have a shared concept of Data-Sources, which describe the business objects in this application. By working from a single, shared definition of the data model, client and server side components can coordinate closely to deliver much more sophisticated functionality out of the box than either a standalone client based or server based solution can deliver. Using a Data-Source as a shared data definition also greatly reduces redundancy between our user interface code and server side code increasing agility and reducing maintenance effort.

Smart GWT is the highest performance platform available for web applications. Unlike many web sites, web applications are visited repeatedly by the same users on a frequent basis, and users will spend significant time actually using the application. To correctly assess the performance of a web application, we should be measured the performance when completing a typical series of tasks. For example, in many different types of applications a user will search for a specific record, view the details of that record, modify that record or related data, and repeat this pattern many times within a given session. To assess performance in this scenario, what should be measured are requests for dynamically generated responses - for example, results from a database query. Requests for static files, such as images and CSS style sheets, can be ignored since these resources are cacheable—these requests will not recur as the user runs through the task multiple times, and will not recur the next time the user visits the application. This improves the responsiveness and scalability. The possible approaches to data integration are summarized in the Fig1 diagram. Paths 2, 3 and 4 are client-side integration approaches, while path 1 includes all server-side integration approaches. Path 1 makes use of the Smart GWT Server Framework. The server framework is a set of Java libraries and servlets that can be integrated with any pre-existing Java application. It is highly recommended that we use the Smart GWT Server Framework for data integration. The server framework delivers an immense range of functionality that compliments any existing application and persistence engine. The data in the request and response objects has the same meaning. The key members of a DSR Request object are: Data, sortBy ,startRow and endRow ,oldValues. The key members of a DSR Response object are:Status, data, startRow and endRow, totalRows, errors.

III. SERVER CONFIGURATION

In order to develop and run the application we need to place our source files in a directory is called as webroot directory. Under web root it has WEB-INF usually it has lib, classes and web.xml files. In web.xml file provide the information about the web application, include the following code:

```
<?xml
version="1.0" encoding="UTF-8" standalone="no"?>

<web-app           xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

<servlet>
       <servlet-name>projectcontrols</servlet-name>
       <servlet-class>
org.springframework.web.servlet.DispatcherServlet
</servlet-class>
       <load-on-startup>1</load-on-startup>
 </servlet>

<servlet>
       <servlet-name>DataSourceLoader</servlet-name>
<servletclass>com.isomorphic.servlet.DataSourceLoader</s
ervlet-class>
</servlet>
 ------------------------------------------
--------------------------------------------
etc

<servlet>
       <servlet-name>screenLoader</servlet-name>
       <servlet-class>
com.isomorphic.servlet.ScreenLoaderServlet
</servlet-class>
 </servlet>

<listener>
<listener-class>
com.isomorphic.base.InitListener</listener-class>
</listener>
```

```
<servlet>
        <servlet-name>Init</servlet-name>
        <servlet-class>com.isomorphic.base.Init
</servlet-class>
        <load-on-startup>1</load-on-startup>
 </servlet>

<servlet-mapping>
         <servlet-name>projectcontrols</servlet-name>
         <url-pattern>*.pcs</url-pattern>
 </servlet-mapping>

<context-param>
<paramname>contextConfigLocation</param-name>
<param-value>
/WEB-INF/resource/springService-servlet.xml
</param-value>
 </context-param>

</web-app>
```

IV. SYSTEM DEVELOPMENT

According to proposing system here we need to add an Enterprise Project Structure (EPS) using menu or entry form. Smart GWT Tree grid is used in development of EPS which helps to create a node system to locate the location of current job of a project. Add users their profiles and privileges, the accessible privileges for various levels of users like IT administrators, Project Managers, High Level Users (Line Managers and Senior Project Engineers), Mid Level Users (Project Engineers and Superintendents), Site Supervisors, Site Administrators (Time keepers and Clerks), Site Engineers etc. Enter data using appropriate menu items and data entry forms. Create jobs base data for Manpower, Plants and Equipments, Materials, fuel and store items, costs, Schedules of Rates, Purchase Orders, Tangible quantities etc. Common reference data is maintained as a separate module to easy understanding and development. If necessary retrieve data from other databases to share the information. Create and assign baselines for the jobs under the new project as per approved original budgets and revised budgets. Attendance records for all personnel and plants entries from multiple users on daily basis (site supervisors and other users) using web based forms or standalone systems to track the project cost and inline development.

Work diary and Staff Time sheet entries from multiple users on daily basis (site supervisors and other users) using web based forms or stand alone systems. Capture date wise actual data or as build data for resources work progress and expenses. Validate the actual data or build data and approval process. The entered data should archive or backup data for future use upon completion of a project. This implements with smart GWT, Google App Engine server, Hibernate, spring framework, backend any data base. We can run this application on any web server like Tomcat or application server like Web Logic, Web Sphere and JBoss.

A. Component Layout:

Smart GWT has four kinds of layout classes. HLayout is to set the positions of components in horizontal and manage widths of a components, VLayout is to set the positions of components in vertical and manage heights of a components. But HStack and VStack to set the components in sequence of horizontal and vertical respectively.

B. Handling Events:

Smart GWT applications implement interactive behavior by responding to events generated by their environment or user actions. We can provide the logic for different events by implementing event handlers. The most commonly used smart GWT events include:

a. Click (for buttons and menu items)
b. RecordClick (for list grids and tree grids)
c. Changed (for form controls)
d. TabSelected (for tabsets)

a. For example:

```
import com.smartgwt.client.widgets.Button;
final IButton newButton = new IButton("Add");
newButton.setShowRollOver(true);
------------ etc
newButton.addClickHandler(new ClickHandler() {
@Override
public void onClick(ClickEvent event) {
RPCManager.setActionURL(GWT.getModuleBaseURL()+
"EpsHibernateOperations.rpc");
epsTreeGrid.startEditingNew(epsType);
}
});
```

Here upon click on add button an event is raised and the application generates a grid with fields and the user has to provide corresponding data.

C. Datasources:

Data-Source objects of smart GWT have set of data fields which are implemented at presentation layer provides independent presentation at user interface. Achieving the sharing of data models across components, various applications and both client and server can be done by using Data-Sources. Visual components are TreeGrids which helps us to display and manipulate persistent data and inheritance relationships. A built in mechanism in Data-Source provides us the standard data operations like add, update, sort, fetch, remove which are executed on both client and server for validating data, paging, data typing and unique keys and more. In smart GWT leverage automatic behaviors including validation of data which is provided by the user, data loading upon selection of some attribute, catching provides small memory to increase performance of an application, filtering to remove the unnecessary data, sorting of current data based on direction, paging .Based on Data-Source fields both grid and form components are automatically generated the respective component fields. The form has chosen specific controls for certain fields: valueMap is for selectItem, boolean is for CheckboxItem, date is for DateItem, length > 255 is for TextAreaItem etc. Fields are the building blocks of data bound components and data-sources.

D. Spring Execution:

End user selects the field from select box through smart GWT widget components and submits the request to web application. The controller servlet of web application traps and takes the request and read form data from form page. Activates spring container and gets spring bean class object from spring container. Servlet programme calls the method of SpringBean class. The business logic and persistence logic spring bean class collects the records from corresponding tables in database based on the given data(end user requirement). The business method takes the support of DataTransferObject (DTO) class bean and sends those records to controller servlet in the form of ArrayList object. Controller Servlet passes this result (ArrayList object) to the smart GWT. Finally result is displayed on presentation layer.

a. Code snippet for EPSHibernateController:

```
import java.io.Serializable;
import java.util.ArrayList;
etc -----
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.Controller;
import com.isomorphic.datasource.DataSource;
import com.isomorphic.rpc.RPCManager;
import com.isomorphic.rpc.RPCRequest;
etc…
public class EpsHibernateOperationsController implements
Controller {
@Autowired
          IEPSService iepsService ;
public ModelAndView handleRequest (HttpServletRequest
request, HttpServletResponse response) throws Exception
{------
       ------

      }
-------
}
```

E. Hibernate Framework:

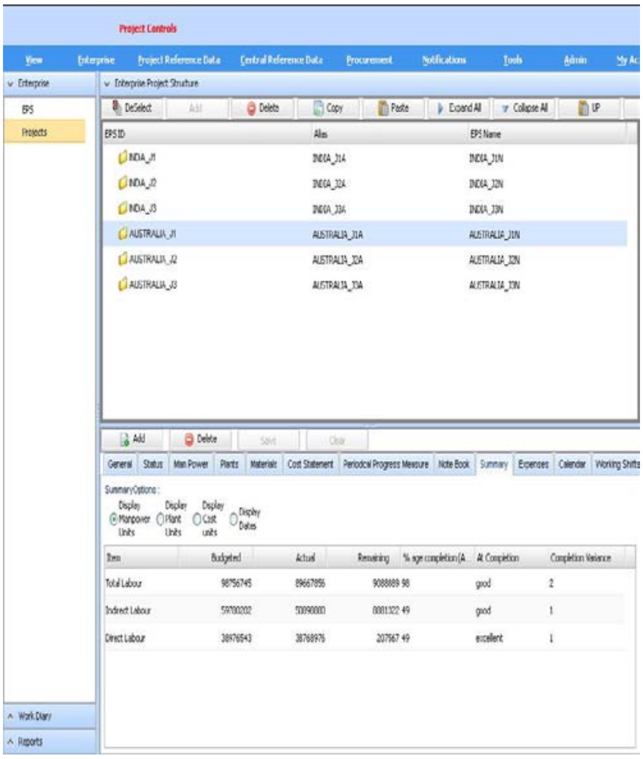
In order to perform the database operations using hibernate an application must get the reference of Session object. After performing the database operations the application must call session.close() method and call Configuration.configure() reads the information available in hibernate.cfg.xml file and information available in hbm or annotation file located in web root directory. Configuration.buildsesssionFactory() decides about various sql statesments that must be used to access the data and creates a SessionFactory object. These two operations are expensive. So we prefer to improve the performance of the web application we can use org.pcs.OraSF.CurrentSession() to get the reference of hsession object i.e. a session object. To close the session object we can use org.pcs.OraSF.CloseSession().When CurrentSession() is called for the time the following steps will be carried out.

a. Configuration() will be called.
b. buildSessionFactory() will be called.
c. SessionFactory objects reference will be stored in a static variable of OraSF class.
d. If the CurrentSession() is called at a later point of time 2nd time etc the code will use the existing SessionFactory object.

We are using Criteria Query, the interface org.hibernate.Criteria represents a query against a particular persistent class. The Session is a factory for Criteria instances.

Criteria crit=sess.createCriteria(Eps.class);

V. RESULTS

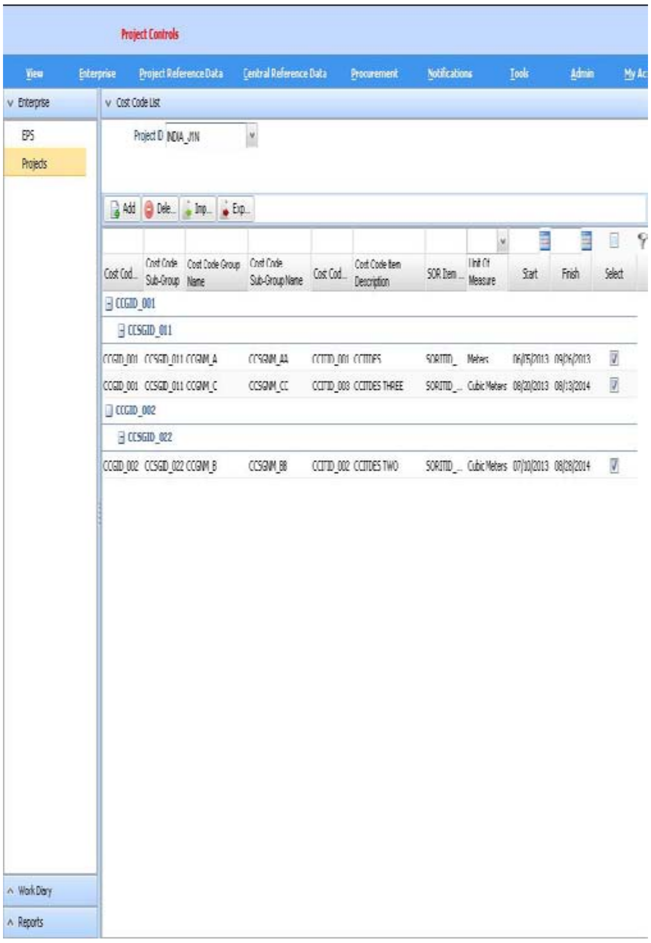


EPS ID	Alias	EPS Name
INDIA_J1	INDIA_J1A	INDIA_J1N
INDIA_J2	INDIA_J2A	INDIA_J2N
INDIA_J3	INDIA_J3A	INDIA_J3N
AUSTRALIA_J1	AUSTRALIA_J1A	AUSTRALIA_J1N
AUSTRALIA_J2	AUSTRALIA_J2A	AUSTRALIA_J2N
AUSTRALIA_J3	AUSTRALIA_J3A	AUSTRALIA_J3N

SummaryOptions : ● Display Manpower Units ○ Display Plant Units ○ Display Cost units ○ Display Dates

Item	Budgeted	Actual	Remaining	% age completion (A…	At Completion	Completion Variance
Total Labour	98756745	89667856	9088889 98		good	2
Indirect Labour	59700202	50090000	8001322 49		good	1
Direct Labour	38976543	38768976	207567 49		excellent	1

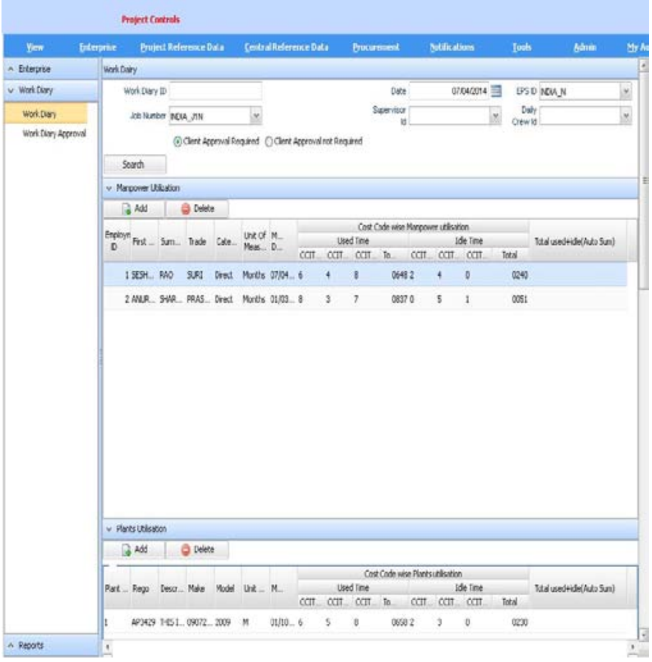
Figure. 2: Display Manpower units of a project which is currently running at AUSTRALIA location.



Project ID INDIA_J1N

Cost Cod...	Cost Code Sub-Group	Cost Code Group Name	Cost Code Sub-Group Name	Cost Cod...	Cost Code Item Description	SOR Item ...	Unit Of Measure	Start	Finish	Select
CCGID_001										
CCSGID_011										
CCGID_001	CCSGID_011	CCGNM_A	CCSGNM_AA	CCITID_001	CCITDES	SORITID_	Meters	06/05/2013	09/06/2013	☑
CCGID_001	CCSGID_011	CCGNM_C	CCSGNM_CC	CCITID_003	CCITDES THREE	SORITID_...	Cubic Meters	08/20/2013	08/13/2014	☑
CCGID_002										
CCSGID_022										
CCGID_002	CCSGID_022	CCGNM_B	CCSGNM_BB	CCITID_002	CCITDES TWO	SORITID_...	Cubic Meters	07/10/2013	08/28/2014	☑

Figure. 3: Display list of Cost Codes of a project in both group and sub group wise.



Work Diary

Work Diary ID:

Date: 07/04/2014

EPS ID: INDIA_N

Job Number: INDIA_JTN

Supervisor Id:

Daily Crew Id:

● Client Approval Required ○ Client Approval not Required

Manpower Utilisation

Employn ID	First ...	Sum...	Trade	Cate...	Unit Of Meas...	M... D...	Cost Code wise Manpower utilisation								Total used+idle(Auto Sum)
							Used Time				Idle Time				
							CCIT...	CCIT...	CCIT...	To...	CCIT...	CCIT...	CCIT...	Total	
1	SESH...	RAO	SURI	Direct	Months	07/04...	6	4	8	0648	2	4	0	0240	
2	ANUR...	SHAR...	PRAS...	Direct	Months	01/03...	8	3	7	0837	0	5	1	0051	

Plants Utilisation

Plant ...	Rego	Descr...	Make	Model	Unit ...	M...	Cost Code wise Plants utilisation								Total used+idle(Auto Sum)
							Used Time				Idle Time				
							CCIT...	CCIT...	CCIT...	To...	CCIT...	CCIT...	CCIT...	Total	
1	AP3429	THIS 1...	09072...	2009	M	01/10...	6	5	0	0650	2	3	0	0230	

Figure. 4: Generate and display Manpower Utilization, Plants Utilization based on date, EPS, and job numbers.

VI. CONCLUSION

Existing processes takes much time to develop an application means it is difficult to meet dead line. A developer who develops an application should have scripting capability also. If we want to change the backend we should have to create entire database schema and tables hence it takes lot of time and difficulty. This process can be easily applied only on predefined databases and medium size projects. Queries building become difficult on using traditional technologies. In the proposed system implemented the presentation layer using smart GWT ensures easy development in event handling, generating dynamic forms and pop ups. Data-Source is using as a shared data. It greatly reduces redundancy between our user interface code and server side code increasing agility and reducing maintenance effort. Modules of Spring Framework and smart GWT ensures easy enhancement of an application whenever a client wants to add new features to an application. Just we run the application to generate entire database tables automatically, this can achieve by using Hibernate annotations. This improves the process of development of an application in an easy and faster manner. There is huge scope to finish the project within timeline. This method improves the performance, look and feel of an application.

VII. REFERENCES

[1]. James Elliott, “Hibernate,” A Developer’s Note book.

[2]. Hemrajani, Anil, “Agile Java Development with Spring, Hibernate and Eclipse”.

[3]. Paul Tepper Fisher, Brain D.Murphy, “Spring Persistance with Hibernate.” Experts voice in Open Source.

[4]. Google Developers, “smart GWT”.

[5]. Thomas Risberg, Rod Johnson, Juergen Hoeller: “Professional Java Development with the Spring framework”.

[6]. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides “Design Patterns, Elements of Reusable Object Oriented Software. Foreword by Grady Booch.

[7]. Herbert Schildt “The Complete Reference JAVA”

[8]. Hans Bergsten, Java Server Pages O’Reilly

[9]. Bill Siggelkow, Jakarta Struts Cookbook

[10]. Yedidyah Langsan, Moshe Augenstein, Aalan M. Tenenbaum: “Data Structures Using Java”

[11]. Bernd Bruegge, Allan H. Dutoit: “Object Oriented Software Engineering Using UML, Patterns, and Java.”

[12]. Roger S Pressman: “Software Engineering.”, A Practitioner’s Approach.

[13]. Walker Royce: “Software Project Management”. A Unified framework.

[14]. Dr P.S.Deshpande, SQL & PL/SQL for Oracle 10g, Black Book.

[15]. Edward kit, “Software Testing in the Real World.” Improving the process.