

RESOURCE PLANNER

by

NAVYA RAYAPATI

B. Tech, Jawaharlal Nehru Technological University, India, 2014

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing & Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2016

Approved by:

Major Professor
Dr. Daniel Andresen

Copyright

NAVYA RAYAPATI

2016

Abstract

When a software development company wants to achieve its goals on time and efficiently use its staff on the projects, it is necessary for the company to have hands on information related to number of employees working on various projects along with their skill set and the number of employees still needed to complete the projects on time.

Resource Planner is a convenient tool to handle various projects in a software company efficiently. It is an online tool to manage projects currently running with the company as well as future projects. This tool tracks the employees working for the existing projects and details of new projects like no. of employees required, location, etc., and also estimates revenue which helps higher management to know the status of the various projects and work force. With this tool, HR can estimate the requirement of employees for the new projects and hence can recruit exact number of employees. The project manager can view the details of the employees in the company and the employees in his projects. He can also get the information of how many employees are there in a particular project and their skill set. When he wants employees for a particular project, he can add that information mentioning how many are working on the project and how many employees are totally needed for the project. This information can be had both for online and offline projects. This information is used in generating a special report called GAP summary.

This application is developed using Java Spring MVC Framework and MySQL database. The main goal of the project is to get familiar with Spring MVC Framework.

Table of Contents

List of Figures	vi
List of Tables	vii
Acknowledgements	viii
Chapter 1 - PROJECT DESCRIPTION	1
1.1 INTRODUCTION	1
1.2 INTENDED USERS	1
Chapter 2 - TECHNOLOGIES USED	2
2.1 SPRING FRAMEWORK ¹	2
2.1.1 CORE CONTAINER ¹ :	2
2.1.2 DATA ACCESS/INTEGRATION:	3
2.1.3 WEB:	3
2.1.4 IOC CONTAINERS ¹ :	4
2.2 SPRING MVC ³	5
2.3 MYSQL DATABASE ⁶	5
Chapter 3 - APPLICATION MODULES	6
3.1 EMPLOYEE MODULE	6
3.2 PROJECTS MODULE	6
3.3 REQUIREMENT AND ALLOCATION MODULE	6
3.4 ADMIN & REPORTING MODULE	7
Chapter 4 - APPLICATION FUNCTIONALITY	8
4.1 ADMIN	8
4.2 HR MANAGER	8
4.3 PROJECT MANAGER	8
Chapter 5 - SYSTEM DESIGN	9
4.1 USECASE DIAGRAM	9
4.2 CLASS DIAGRAM	10
Chapter 6 - DATABASE DESIGN	13
6.1 E-R DIAGRAM	14
Chapter 7 - CODING	17

7.1 APPLICATION CONTEXT.....	17
7.2 VALIDATIONS	18
Chapter 8 - TESTING.....	20
8.1 UNIT TESTCASE 1	22
8.2 UNIT TESTCASE 2	23
8.3 PERFORMANCE TESTING	23
Chapter 9 - SECURITY	26
9.1 SQL INJECTION	26
9.2 REMOTE CODE EXECUTION	26
9.3 USERNAME ENUMERATION	27
9.4 PARAMETER TAMPERING	27
Chapter 10 - IMPLEMENTATION.....	28
Chapter 11 - CONCLUSION.....	34
10.1 FUTURE WORK.....	34
Chapter 12 - BIBLIOGRAPHY	35

List of Figures

Figure 2.1: Spring Framework ¹	2
Figure 2.2: Spring Container ¹	4
Figure 5.1: Use case diagram.....	9
Figure 5.2: Class Diagram 1	10
Figure 5.3: Class Diagram 2	11
Figure 6.1: E-R Diagram.....	15
Figure 7.1: applicationContext.xml	18
Figure 7.2: Validating Username & Password	19
Figure 7.3: Login.jsp.....	19
Figure 8.1: Testing Hierarchy ⁹	20
Figure 8.2: Test case 1	22
Figure 8.3: Test case 2	23
Figure 8.4: Observation 1	24
Figure 8.5: Observation 2	25
Figure 8.6: Observation 3	25
Figure 10.1: Admin Login Page.....	28
Figure 10.2: List of Employees.....	29
Figure 10.3: Role-Location Report.....	29
Figure 10.4: Adding User	30
Figure 10.5: Removing User.....	30
Figure 10.6: Reset Password.....	31
Figure 10.7: GAP Summary	31
Figure 10.8: List of Projects.....	32
Figure 10.9: Adding new Project	32
Figure 10.10: Removing Project.....	33
Figure 10.11: Viewing Requirements	33

List of Tables

Table 6.1: Employee Table	13
Table 6.2: Projects Table	13
Table 6.3: Availability Table	14
Table 6.4: Requirements Table	14
Table 6.5: Allocations Table.....	14
Table 6.6: Login Table.....	14
Table 7.1: Lines of Code.....	17
Table 8.1: Test case Results.....	22
Table 8.2: Inputs to generate graph.....	25

Acknowledgements

I'm deeply grateful to my major professor Dr. Daniel Andresen for giving me an opportunity to do a project which enhanced my skills. I also want to thank him for trusting my abilities to complete my project

I take immense pleasure in expressing my gratitude to my committee members Dr. Mitchell Neilsen and Dr. Torben Amtoft for their encouragement and taking time to serve on my committee

I would like to thank my family and friends for their immense love and belief

Chapter 1 - PROJECT DESCRIPTION

1.1 INTRODUCTION

Resource Planner is an online tool to manage projects currently running with the company as well as future projects. It helps in avoiding more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the employees to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. As the data is centralized it is very easy to maintain the currently running projects with the company as well as future projects. This is an online tool so more than one user can login into system and use the tool simultaneously. This tool is very useful in estimating revenue, etc which helps higher management to know the status of the various projects and work force. For this project, we assume that the requirements are specified by a client.

The main objective in preparing this project is to learn Spring MVC Framework. Hence, the UI design is pretty much not descent.

1.2 INTENDED USERS

Administrator: The administrator of this software will be able to create new users and remove any user. He allots passwords and changes them.

HR manager: He should take a decision regarding whether to recruit any employees and whether there is dearth of any resources in a project.

Project Manager: The project manager can view the details of the employees in the company and the employees in his projects.

Chapter 2 - TECHNOLOGIES USED

2.1 SPRING FRAMEWORK¹

Spring could potentially be a one-stop shop for all the enterprise applications; however, Spring is modular, allowing us to pick and choose which modules are applicable to us, without having to bring in the rest. The Spring Framework provides about 20 modules which can be used based on an application requirement.

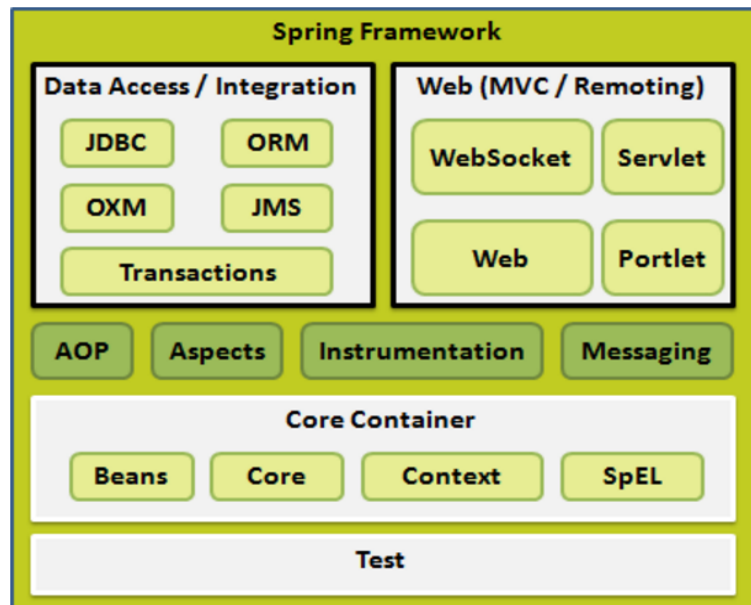


Figure 2.1: Spring Framework¹

2.1.1 CORE CONTAINER¹:

The Core Container consists of the Core, Beans, Context, and Expression Language modules. Core module provides the fundamental parts of the framework, including the IoC and Dependency Injection features. The Bean module provides BeanFactory which is a sophisticated implementation of the factory pattern. The Context module builds on the solid base provided by the Core and Beans modules and it is a medium to access any objects defined and configured. The ApplicationContext interface is the focal point of the Context module. The SpEL module

provides a powerful expression language for querying and manipulating an object graph at runtime.

2.1.2 DATA ACCESS/INTEGRATION:

The Data Access/Integration layer consists of the JDBC, ORM, OXM, JMS and Transaction modules. The JDBC module provides a JDBC-abstraction layer that removes the need to do tedious JDBC related coding. The ORM module provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis. The OXM module provides an abstraction layer that supports Object/XML mapping implementations for JAXB, Castor, XMLBeans, JiBX and XStream. The Java Messaging Service JMS module contains features for producing and consuming messages. The Transaction module supports programmatic and declarative transaction management for classes that implement special interfaces and for all your POJOs.

2.1.3 WEB:

The Web layer consists of the Web, Web-Socket, and Web-Portlet, Web-MVC modules. The Web module provides basic web-oriented integration features such as multipart file-upload functionality and the initialization of the IoC container using servlet listeners and a web-oriented application context. The Web-Socket module provides support for WebSocket-based, two-way communication between client and server in web applications. The Web-Portlet module provides the MVC implementation to be used in a portlet environment and mirrors the functionality of Web-Servlet module. . The Web-MVC module contains Spring's model-view-controller (MVC) implementation for web applications which I will be using to implement the project. A description of Spring's MVC is given below.

There are few other important modules like AOP, Aspects, Instrumentation, Web and Test. The AOP module provides aspect-oriented programming implementation allowing you to define method-interceptors and pointcuts to cleanly decouple code that implements functionality that should be separated. The Aspects module provides integration with AspectJ which is again a powerful and mature aspect oriented programming (AOP) framework. The Instrumentation module provides class instrumentation support and class loader implementations to be used in certain application servers. The Messaging module provides support for STOMP as the WebSocket sub-protocol to use in applications. It also supports an annotation programming model for routing and processing STOMP messages from WebSocket clients. The Test module supports the testing of Spring components with JUnit or TestNG frameworks.

2.1.4 IOC CONTAINERS¹:

The Spring container being the core of the Spring Framework creates the objects which are called as Spring beans, wire them together, configure them, and manage their complete lifecycle from creation till they are destructed. It uses dependency injection (DI) for injecting the data. The following diagram illustrates on how Spring works.

The Spring IoC container makes use of Java POJO classes (Plain Old Java Objects) and configuration metadata to produce a fully configured and executable system or application. The configuration metadata can be represented either by XML, Java annotations, or Java code.

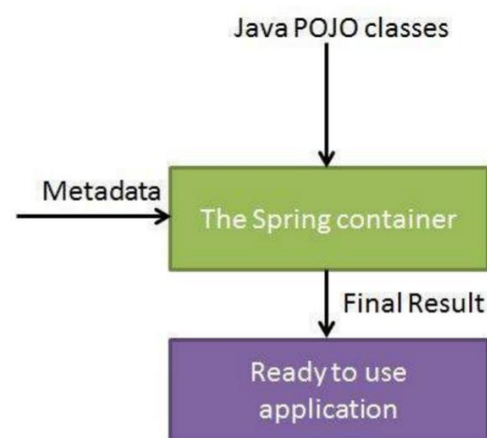


Figure 2.2: Spring Container¹

The container which I have used is `ApplicationContext`. This container adds more enterprise-specific functionality such as the ability to resolve textual messages from a properties file and the ability to publish application events to interested event listeners. This container is defined by the `org.springframework.context.ApplicationContext` interface.

2.2 SPRING MVC³

The Spring web MVC framework provides a model-view-controller architecture. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic). Spring makes and encourages us to work in loose-coupled manner.

The Model encapsulates the application data and in general they will consist of POJO(Plain Old Java Objects). The View is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret. The Controller is responsible for processing user requests and building appropriate model and passes it to the view for rendering. The Spring Web MVC framework is designed around a `DispatcherServlet` that handles all the HTTP requests and responses.

2.3 MYSQL DATABASE⁶

It is very important to have a back end database to which the application is connected. The database which has been used to connect with the application is `MySQL`. It is a SQL based relational database management system that is supported by many operating systems. It stores the information in the form of tables. Data retrieval is done from tables and data insertion is also done into tables.

I have also used a web server `Apache Tomcat 7.0` for deploying the project, `Eclipse IDE` as the workbench.

Chapter 3 - APPLICATION MODULES

It is very important to have different modules for a better understanding of the project and also to structure it properly. It would be easy for any developer to analyze the project carefully and do appropriate changes if something go wrong. For this project I assumed that the client provides requirements and the requirements are stated as the responsibilities of each user who uses the application. Authorizations should be given to each and every user of the application. Based on the authorizations, he should be able to access the modules specified below. The modules from my project are listed below

3.1 EMPLOYEE MODULE

This module deals with tracking the details of employees currently working with the company. In this module, only the HR Manager will be able to add/delete an employee record into the database. It allows all types of users to view the list of users existing currently in our company. It facilitates us to convert the employee report into excel format just by clicking *download to excel* button.

3.2 PROJECTS MODULE

This module maintains the details of projects currently associated with the company & the future projects which the company would be handling. In this module, a project manager will be able to add new projects and their details to the database and update the project as completed when it's done. It provides a user-friendly interface to add new projects. It also provides an option to convert projects report into excel format.

3.3 REQUIREMENT AND ALLOCATION MODULE

This module provides information about project-wise requirements which includes onsite and offshore that was entered by the project managers of different projects. It also allows any

type of user to view these project requirements. It helps the HR manager to view project-wise requirements and start recruiting the people. It also helps in allocating the people to a project after recruitment so that HR people can idea about the gap between requirement and allocation at any point of time very easily by generating HRD GAP Summary report. It provides all these reports to be converted and stored permanently in excel sheets.

3.4 ADMIN & REPORTING MODULE

This module provides interface to manage the tool like add/remove users, change privileges of users etc. It also helps to provide different reports required by the higher management for better analysis. It generates dynamic reports like Role-Location which displays role-wise employees report in different locations, Project-Location report which displays project-wise employees reports in different locations, Role-Skill reports which displays skill-wise role based employees list in different locations, Project-Skill reports which displays project-wise skills report of different employees and Skill-Location report which displays skill-wise employees report in different locations etc.

Chapter 4 - APPLICATION FUNCTIONALITY

Depending on the modules, the application functions in various ways. Different users who access the app perform different functions of the application. Functionalities are specific to users. Different users of the application are listed below.

4.1 ADMIN

The Admin users will be able to perform functions such as creating/removing/viewing the employees of the company, viewing Role-Location reports, Project-Location reports, Role-Skill reports, Project-Skill reports, Skill-Location reports. The admin will also be able to add different users of the application, change the password, resetting the password of other users in case of forgetting password and generating management reports.

4.2 HR MANAGER

The HR users will be able to perform functions such as changing the password, adding/viewing/ removing the employees in the company, viewing information of GAP summary and GAP RoleWise and Viewing Role-Location reports, Project-Location reports, Role-Skill reports, Project-Skill reports, Skill-Location reports.

4.3 PROJECT MANAGER

The Project Manager will be able to perform functions such as changing the password, viewing the details of employees in the company, adding/viewing/removing the details of projects, add/view the details of requirement in a project, add/view the details of allocation of employees to a particular project, viewing management reports.

Chapter 5 - SYSTEM DESIGN

4.1 USECASE DIAGRAM

A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. These are drawn as stick figures.

The goals of these actors are represented as use cases which describe a sequence of actions that provide something of measurable value to an actor and any dependencies between those use cases.

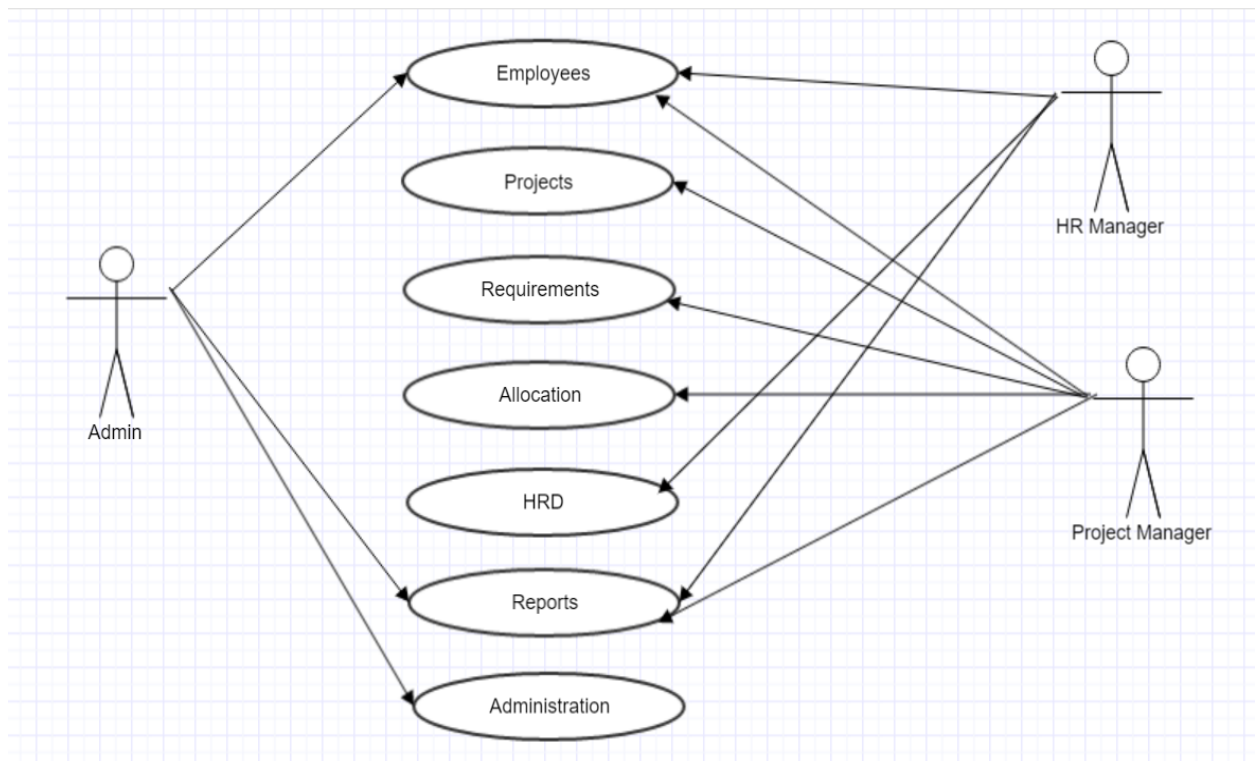


Figure 5.1: Use case diagram

Here, Admin can access all the modules. Project manager can access only Project modules. HR manager can access the HRD and report modules.

4.2 CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. A class diagram is defined in 3 parts. Class name is declared in the first part, then comes the attributes. Lastly the methods within the class.

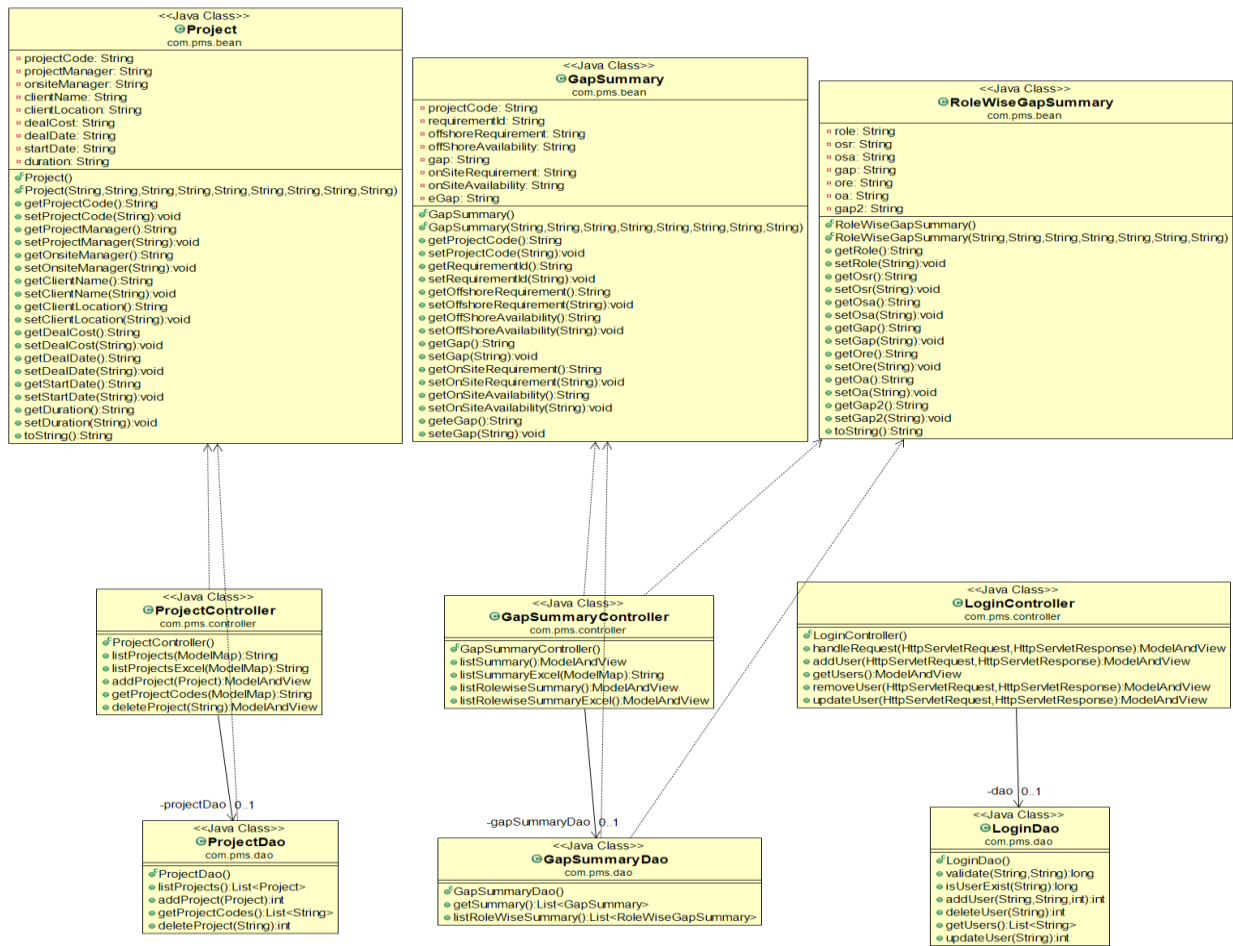


Figure 5.2: Class Diagram 1

The above classes are inter dependent to each other. ProjectController class interacts with ProjectDao and Project bean class. Similarly GapSummaryController class interacts with

GapSummary, RolewiseGapSummary bean classes as well as GapSummaryDao class. LoginController class interacts with LoginDao class. The bean classes are directly interacted to Dao classes respectively. Project class is responsible for creating the projects, deleting the projects. The query code is written in ProjectDao class. ProjectController creates the interaction between the model and the view. Similarly, GapSummaryController class is responsible for identifying the requirements based on projects and roles. The query code is written in GapSummaryDao class. GapSummaryController creates the interaction between the model and the view. Similarly for LoginController.

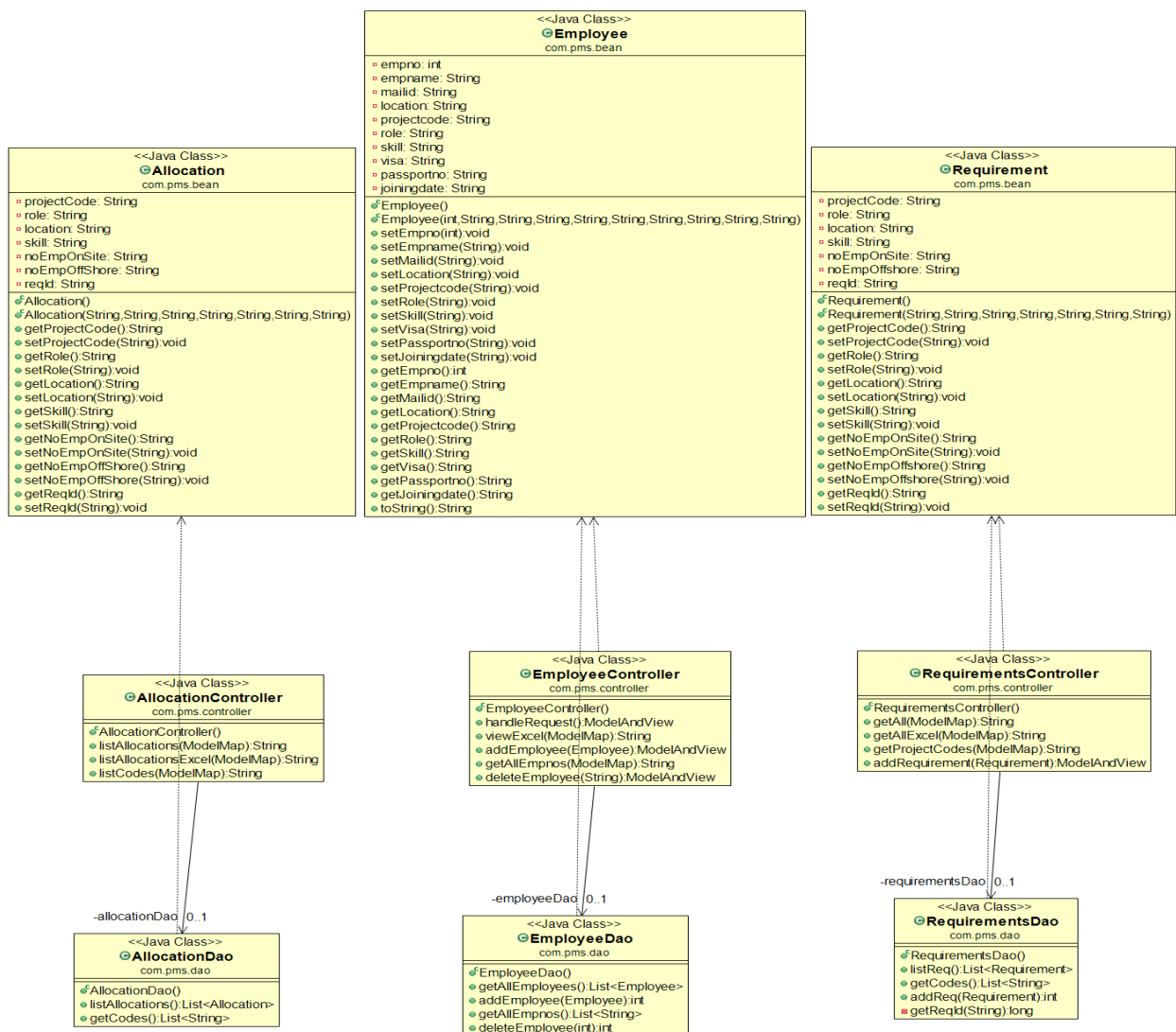


Figure 5.3: Class Diagram 2

Here, AllocationController class interacts with Allocation bean class as well as AllocationDao class. Allocation class is responsible for allocating the employees to a particular project. The query code is written in AllocationDao class. AllocationController creates the interaction between the model and the view associated with it.

EmployeeController class interacts with Employee bean class and EmployeeDao class. EmployeeController class is responsible for creating the employee details. The query code is written in EmployeeDao class. EmployeeController class creates the interaction between the model and the view.

Similarly, RequirementsController class interacts with Requirements bean class and RequirementsDao class. The bean classes are directly interacted to Dao classes respectively. Requirement class is responsible for checking the requirement for a particular project or checking for a requirement of a new employee. The query code is written in RequirementDao class. RequirementController class creates the interaction between the model and the view.

Chapter 6 - DATABASE DESIGN

The database which has been used is MySQL. I have used MySQL Query browser to run the database queries. The total number of database tables that were identified to build a system is 6. Each table has a primary key and one or more foreign keys.

- i. Employees table: Contains details of employees
- ii. Projects table: Contains details of projects
- iii. Login table: Contains login information & role information
- iv. Availability table: Contains employees availabilities
- v. Requirement table: Contains requirement information
- vi. Allocations table: Contains project allocation information

Table 6.1: Employee Table

EMPNO	Bigint(20)
EMPNAME	Varchar(100)
MAILID	Varchar(100)
LOCATION	Varchar(50)
PROJECTCODE	Varchar(20)
ROLE	Varchar(50)
SKILL	Varchar(100)
VISA	Varchar(50)
PASSPORTNO	Varchar(25)
JOININGDATE	Varchar(55)

Table 6.2: Projects Table

PROJECTCODE	Varchar(20)
PROJECTMANAGER	Varchar(50)
ONSITEMANAGER	Varchar(50)
CLIENTNAME	Varchar(50)
CLIENTLOCATION	Varchar(50)
DEALCOST	Decimal(25,2)
DEALDATE	Varchar(55)
STARTDATE	Varchar(55)
DURATION	Bigint(20)

Table 6.3: Availability Table

PROJECTCODE	Varchar(20)
ROLE	Varchar(50)
LOCATION	Varchar(50)
SKILL	Varchar(100)
NOEMPONSITE	Bigint(20)
NOEMPOFFSHORE	Bigint(20)
REQID	Bigint(20)

Table 6.4: Requirements Table

PROJECTCODE	Varchar(20)
ROLE	Varchar(50)
LOCATION	Varchar(50)
SKILL	Varchar(100)
NOEMPONSITE	Bigint(20)
NOEMPOFFSHORE	Bigint(20)
REQID	Bigint(20)

Table 6.5: Allocations Table

PROJECTCODE	Varchar(20)
REQID	Bigint(20)
EMPNO	Bigint(20)
LOCATION	Varchar(20)

Table 6.6: Login Table

USERID	Varchar(50)
PASSWORD	Varchar(15)
AUTH	Tinyint(4)

Various operations like searching for project, adding new user, adding an employee to a particular project, updating the fields of username, password, creating reports can be done by using queries in MySQL.

6.1 E-R DIAGRAM

The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modeling activity the attributes of each data object noted in the ERD can be described with a data object description.

The set of primary components that are identified by the ERD are Data object, Relationships, Attributes, Various types of indicators. The primary purpose of the ERD is to represent data objects and their relationships. Due to this, any viewer of the application can easily understand the flow of the project and the storage tables along with their names. I have stored my tables in my own database called ‘pms’ in MySQL. The E-R diagram of Resource Planner shows the relationship among all the tables present in the database.

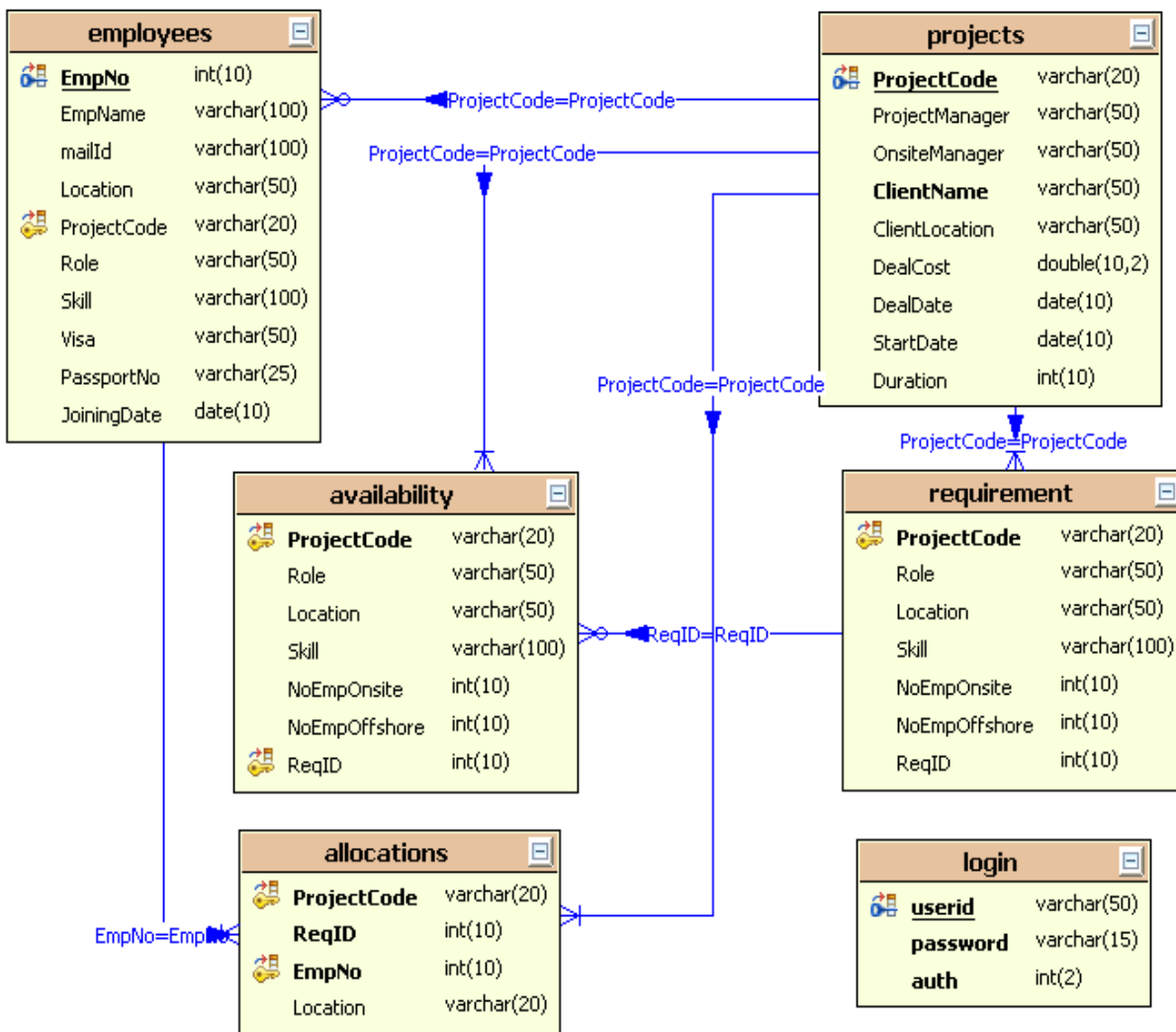


Figure 6.1: E-R Diagram

The 6 rectangular boxes in the table represent the 6 tables which are present in the database. The relationship among them is shown with a blue line mentioning the foreign keys which refer to the tables.

The primary keys for the database are as follows:

Employee table: EmpNo NOT NULL
Availability table: ProjectCode NOT NULL
Allocations table: ProjectCode, ReqID, EmpNo NOT NULL
Requirement table: ProjectCode NOT NULL
Projects table: ProjectCode, ClientName NOT NULL
Login table: userid, password, auth NOT NULL

The primary keys can never be duplicate or null.

The foreign keys of the database are as follows:

Employees table: FOREIGN KEY (ProjectCode) REFERENCES Project(ProjectCode)
Availability table: FOREIGN KEY (ProjectCode) REFERENCES Project(ProjectCode)
Allocations table: FOREIGN KEY (ProjectCode) REFERENCES Project(ProjectCode)
 FOREIGN KEY (EmpNo) REFERENCES Employees(EmpNo)
Requirement table: FOREIGN KEY (ReqID) REFERENCES Allocations(ReqID)

The foreign key constraints are represented in the E-R diagram in blue

The primary key constraints are represented by underlining the primary keys

Chapter 7 - CODING

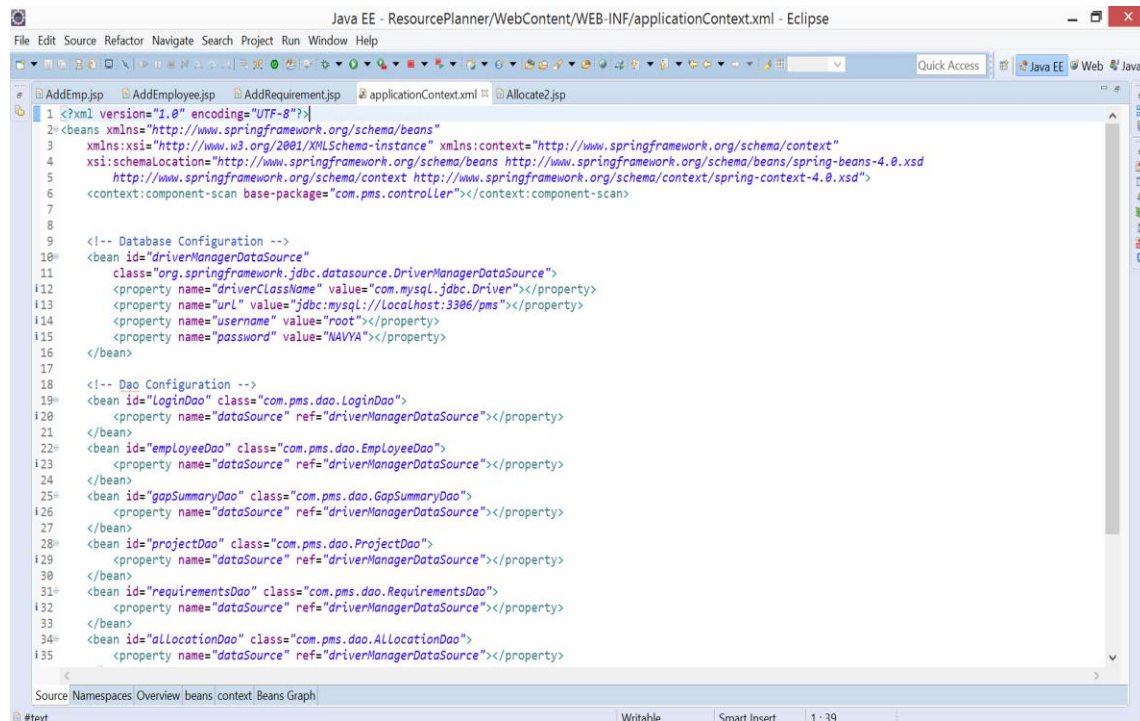
The total number of lines of code along with the language for the application is shown below. I have used various technologies like Servlets, JSP's, HTML, XML, JavaScript for developing the User Interface.

	Lines of Code
Java Server Pages (.jsp)	3679
HTML Pages (.html)	176
CSS pages (.css)	1008
JavaScript (.js)	716
Java code (.java)	1303

Table 7.1: Lines of Code

7.1 APPLICATION CONTEXT

This is the Spring configuration file which is in xml format. The ApplicationContext is the central interface within a Spring application for providing configuration information to the application. The ApplicationContext provides Bean factory methods for accessing application components, the ability to load file resources in a generic fashion, the ability to publish events to registered listeners, the ability to resolve messages to support internationalization and inheritance from a parent context. Below is the code of applicationalContext.xml file for the Resource Planner application. Database, DAO and View configurations have been defined in here.

The image shows a screenshot of the Eclipse IDE. The main editor window displays the content of 'applicationContext.xml'. The XML code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
5       http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd">
6   <context:component-scan base-package="com.pms.controller"></context:component-scan>
7
8
9   <!-- Database Configuration -->
10  <bean id="driverManagerDataSource"
11        class="org.springframework.jdbc.datasource.DriverManagerDataSource">
12    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
13    <property name="url" value="jdbc:mysql://localhost:3306/pms"></property>
14    <property name="username" value="root"></property>
15    <property name="password" value="NAVYA"></property>
16  </bean>
17
18  <!-- Dao Configuration -->
19  <bean id="LoginDao" class="com.pms.dao.LoginDao">
20    <property name="dataSource" ref="driverManagerDataSource"></property>
21  </bean>
22  <bean id="empLoyeeDao" class="com.pms.dao.EmployeeDao">
23    <property name="dataSource" ref="driverManagerDataSource"></property>
24  </bean>
25  <bean id="gapSummaryDao" class="com.pms.dao.GapSummaryDao">
26    <property name="dataSource" ref="driverManagerDataSource"></property>
27  </bean>
28  <bean id="projectDao" class="com.pms.dao.ProjectDao">
29    <property name="dataSource" ref="driverManagerDataSource"></property>
30  </bean>
31  <bean id="requirementsDao" class="com.pms.dao.RequirementsDao">
32    <property name="dataSource" ref="driverManagerDataSource"></property>
33  </bean>
34  <bean id="alLocationDao" class="com.pms.dao.ALLocationDao">
35    <property name="dataSource" ref="driverManagerDataSource"></property>
```

The IDE interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help), a toolbar, and a status bar at the bottom showing 'Writtable', 'Smart Insert', and '1:39'.

Figure 7.1: applicationContext.xml

7.2 VALIDATIONS

Validations are written in JSP's. Here, the client sends a request to a web server. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine. It then converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client. I have written many validation codes. Below is the sample of one of the validations.

```

Java EE - ResourcePlanner/WebContent/AddUser.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Web Java
AddEmployee.jsp AddUser.jsp
1 <%@ page language="java"%>
2 <%@ page session="true"%>
3 <%@ page import="java.sql.*,java.io.*,java.util.Random"%>
4 <HTML>
5 <head>
6 <LINK href="styles.css" type="text/css" rel="stylesheet">
7 </head>
8 <SCRIPT LANGUAGE="JavaScript">
9 <!--
10     history.go(+1);
11     function validate() {
12         x = document.NewUser.uid;
13         y = document.NewUser.pwd;
14         z = document.NewUser.auth;
15         var ed = x.value;
16         var pd = y.value;
17         var ad = z.value;
18         var pattern = /^[a-zA-Z0-9\_\.\ ]{4,8}$/;
19         var Apattern = /^[0-2]{1}$/;
20         if (!(pattern.test(ed))) {
21             alert("Invalid username");
22             return false;
23         } else if (!(pattern.test(pd))) {
24             alert("Invalid password");
25             return false;
26         } else if (!(Apattern.test(ad))) {
27             alert("Invalid Authentication");
28             return false;
29         } else
30             return true;
31     }
32 }
33 /-->
34 </SCRIPT>
35 <jsp:include page="MultiLevelMenu.htm" /><br>
36 <br>
37 <!--
HTML/SCRIPT/#text Writable Smart Insert 21:39

```

Figure 7.2: Validating Username & Password

```

Java EE - ResourcePlanner/WebContent/Login.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access Java EE Web Java
AddProj0.jsp Login.jsp
11 </SCRIPT>
12 <LINK href="styles.css" type="text/css" rel="stylesheet">
13
14 </head>
15 <BODY CLASS=SC onload="document.LOGIN.uid.focus();">
16
17 <center>
18 <BR> <BR> <FONT FACE="Century Gothic"> <!--Declaration of variables-->
19
20 <%!String errormsg;%> <%!String disluserid;%> <%!String dislpwd;%> <%
21 /*Retreiving user id and password*/
22
23 disluserid = request.getParameter("uid");
24 if (disluserid == null)
25     disluserid = "";
26 dislpwd = request.getParameter("pwd");
27 if (dislpwd == null)
28     dislpwd = "";
29 %> <!--Retrieve the error from request and display on screen--> <%
30 errormsg = request.getParameter("error");
31 //System.out.println("errmsg = "+errmsg);
32 if (errmsg == null) {
33     //System.out.println("Error messages was null so clearing it..");
34     errormsg = " ";
35 }
36 if (errmsg.equals("blankfields")) {
37 %> <FONT COLOR="Green" face=verdana> <B>Both the fields are
38 Mandatory.Please fill up both fields.</B><BR> <BR> <%
39 }
40 if (errmsg.equals("invalid")) {
41 %> <FONT COLOR="Green" face=verdana> <B>Invalid Username
42 and/or Password. Please re-enter.</B> <BR> <BR> <%
43 }
44 %> <BR> <BR> <BR> <BR> <BR>
45 <FORM NAME="LOGIN" ACTION="Login.do" METHOD="POST" align=right>
--
HTML Writable Smart Insert 1:1 Initializing ObjectAid

```

Figure 7.3: Login.jsp

Chapter 8 - TESTING

Testing is the process of detecting errors. Testing plays a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also. A good test case is one that has a high probability of finding error, if it exists. Testing can be done on various components like Code, Design, Requirements, Client needs. Levels of testing are as follows.

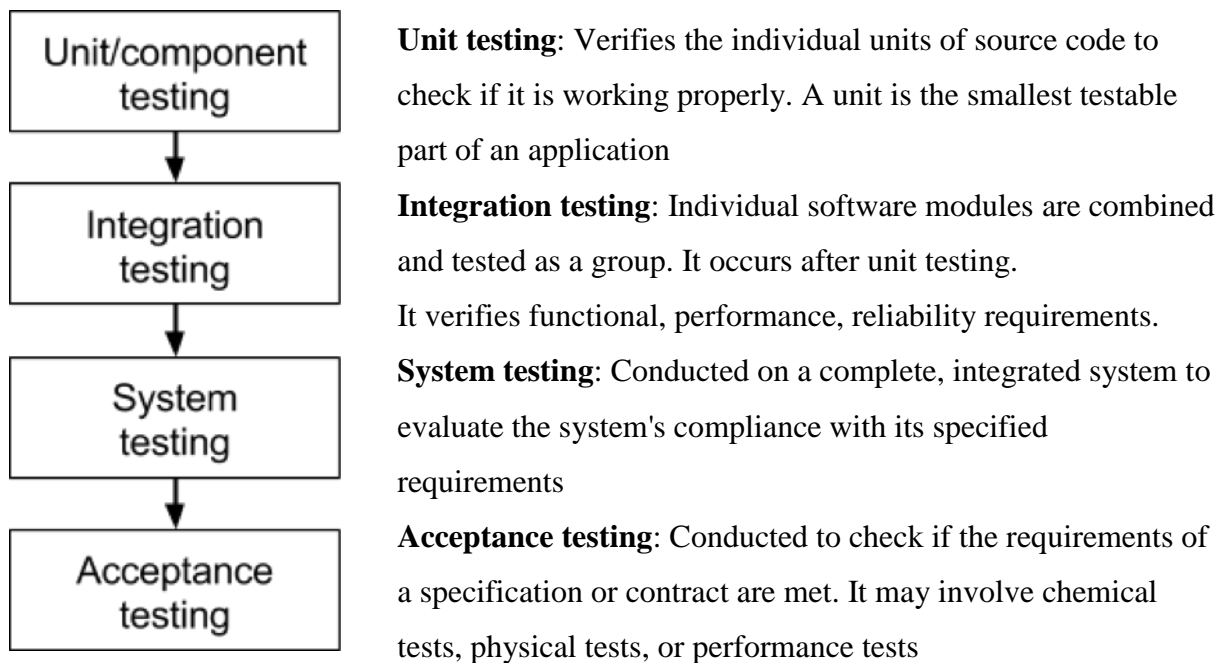


Figure 8.1: Testing Hierarchy⁹

As per the hierarchy, first unit testing is done on individual units. Once that is finished, Integrating testing is done by combining different modules tested by unit testing. Once integration testing is done, System testing is performed to evaluate the system's compliance with the requirements. Finally, acceptance testing is done which checks all the requirements specified by the client.

I have checked some unit test cases. Their expected results are as below.

S No	Unit Test Cases	Expected Result	Result
1	Enter Wrong username and password for Admin at the login form	Displays error message	Pass
2	Enter correct username and password for Admin at the login form	Displays Home Page	Pass
3	Create users of type HR * PM with the required parameters	New Users of type HR * PM will be created	Pass
4	Select remove user option and select a user from drop down list.	Selected user will be removed from the database.	Pass
5	Enter Wrong username and password for HRUser at the login form	Displays Home Page	Pass
6	Enter correct username and password for HRUser at the login form	Displays status as approved for the application	Pass
7	Select Add employee option and enter the required information.	New Employee info will be added in the database	Pass
8	Select Add Project option	Displays Authorization error message	Pass
9	Enter Wrong username and password for PM User at the login form	Displays error message	Pass
10	Enter correct username and password for PMUser at the login form	Displays HomePage	Pass

11	Select Add project option and enter the required fields	It stores the new project info in the database	Pass
12	Select Add Requirement option and enter the required fields	It stores the new requirement info in the database	Pass
13	Select Add Allocation option and enter the required fields	It stores the new allocation info in the database	Pass
14	Select Gap Summary Option	Displays Authorization error message	Pass
15	Login as HRUser and select Gap Summary Option	It displays the gap summary between requirement and allocation	Pass

Table 8.1: Test case Results

Some of the test cases results are taken and displayed below.

8.1 UNIT TESTCASE 1

When I give only username but not password, I get a message saying “Both fields are necessary. Please fill up both of them.”

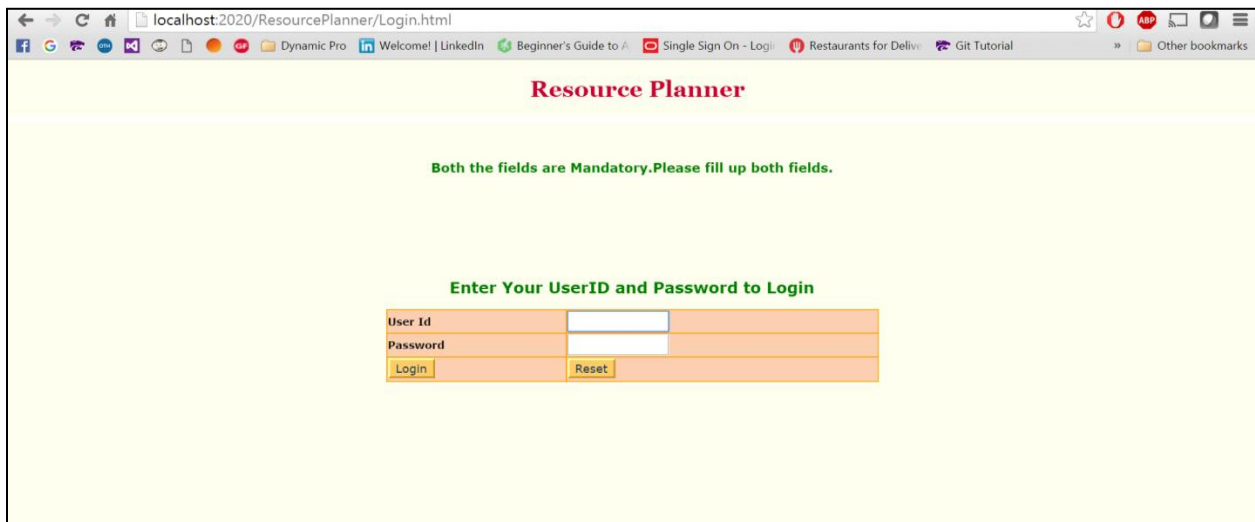


Figure 8.2: Test case 1

8.2 UNIT TESTCASE 2

After logging in as an Admin, if I try to view the projects, I get an error saying I'm not authorized to access the page.

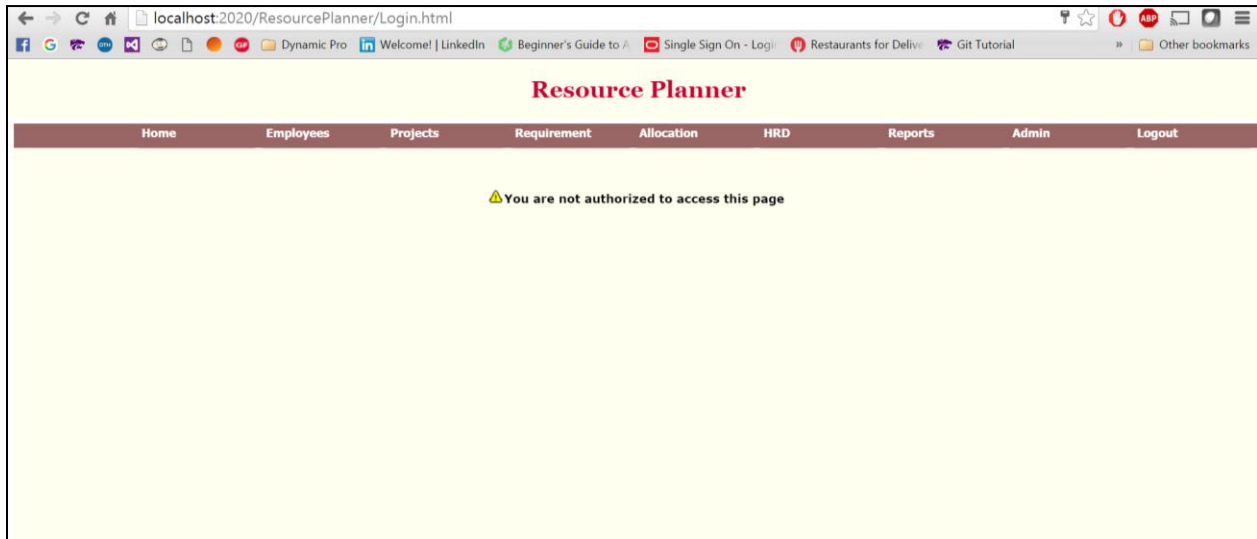


Figure 8.3: Test case 2

Here, I have selected unit test cases as I wanted to test each and every unit which is possible in unit testing.

8.3 PERFORMANCE TESTING

Performance testing is done to determine the speed or effectiveness of a program or an application. It involves many quantitative tests for calculating response time or number of millions of instructions per second at which the system functions. This testing is very useful as before giving the application to end user, we need to check if the app is not hanging if multiple users access it. To check that issue, we have to perform this testing, view the results and then deploy it on live. JMeter is one of the tools available to check the performance of application. I have used apache-jmeter-2.13 to check the performance. After performing the testing, the results got are as shown below in graphical format. All the below observations show the throughput,

Average, Median, Deviation and samples. I have taken throughput and average response time to calculate the performance. The users are dropped at a certain level. It's not being checked. That's why we get the average response time around the same time in all the 3 observations.

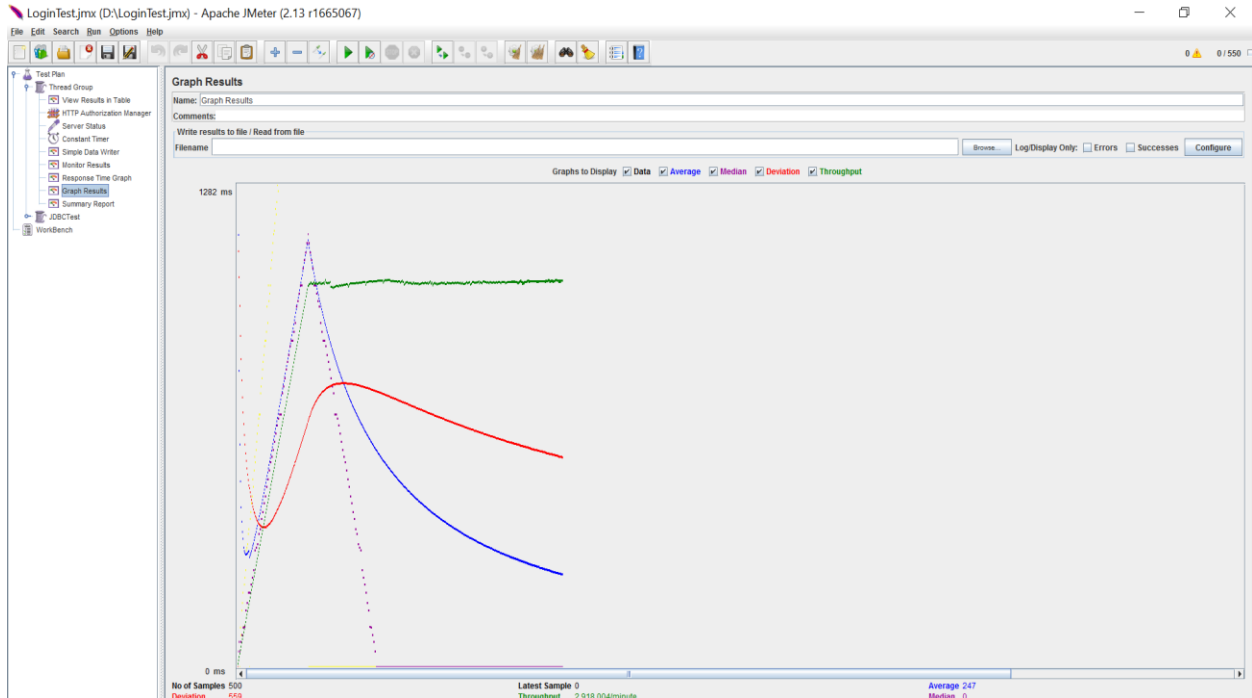


Figure 8.4: Observation 1

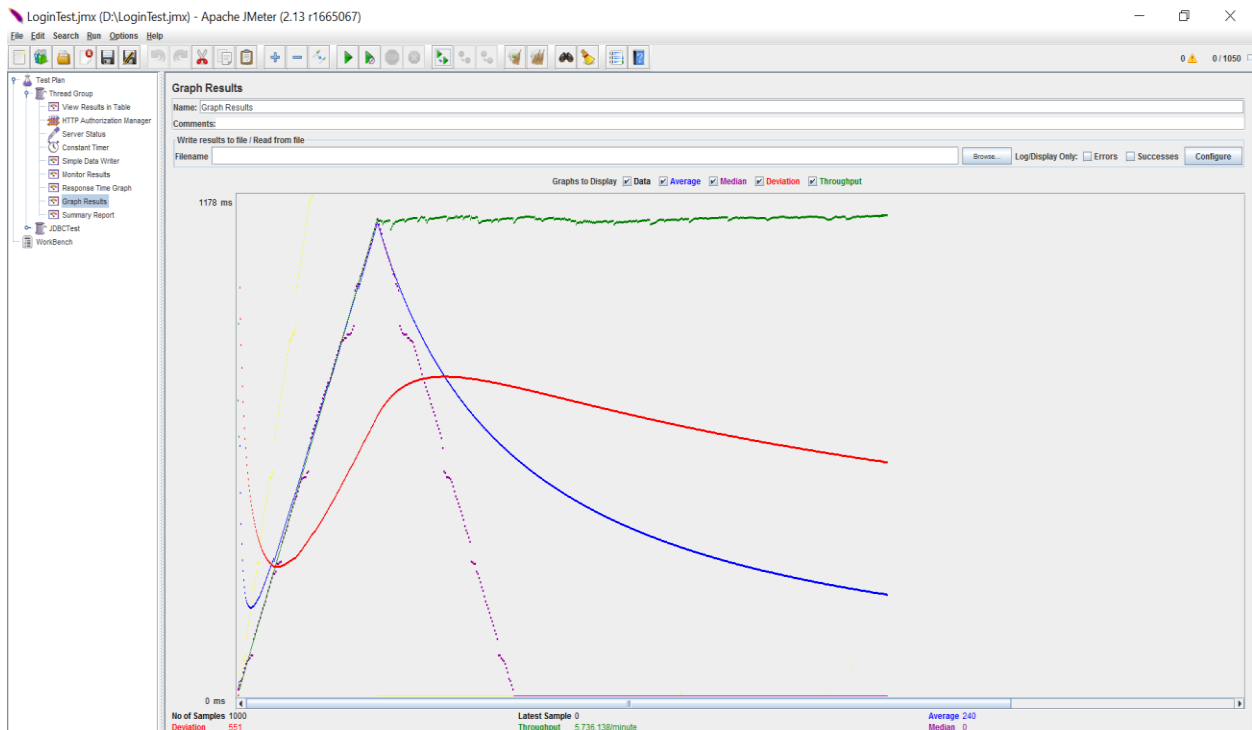


Figure 8.5: Observation 2

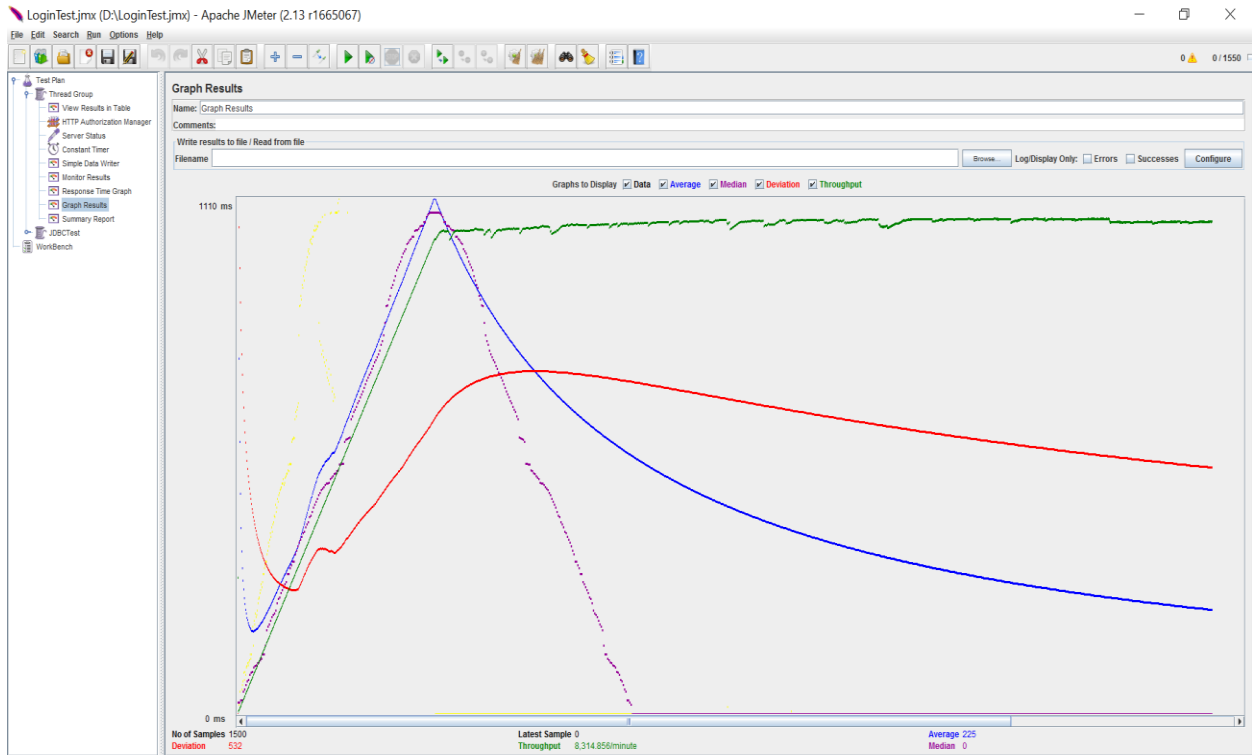


Figure 8.6: Observation 3

The table below shows some of the test cases I considered for generating the graphical view of performance analysis.

	Number of Users(Threads)	Ramp-Up Period	Loop Count	Avg Response Time	Throughput
Obsv 1	500	10	1000	2.47 sec	2918 res/min
Obsv 2	1000	10	1000	2.40 sec	5736 res/min
Obsv 3	1500	10	1000	2.25 sec	8314 res/min

Table 8.2: Inputs to generate graph

From the graphs we can infer that throughput is increasing with increasing no. of users and average response time is decreasing with increasing no. of users. As the Throughput is increasing, I can infer that my application’s performance is pretty descent.

Chapter 9 - SECURITY

Web applications run in the browser. Any security issue can lead to exploiting vulnerability in web application. There are some threats which should be addressed for the security purpose.

Some of them are:

9.1 SQL INJECTION

It allows an attacker to retrieve critical information from a web server's database. The impact of this attack can vary depending on the application's security measures. Using SQL Server can reduce this attack to occur. A malicious SQL injection query will result in the following attempt:

```
$qry = "SELECT * FROM users WHERE username = ' ' or '5=5'";
```

Since the OR condition is always true, this query returns records from the database. This error helps an attacker to get hold of any information which they are looking for.

Couple of countermeasures is given below:

- By avoiding connection with the database as a super user or as the database owner, we can address this attack.
- By always using customized database users with the minimum required privileges required to perform the assigned task.

9.2 REMOTE CODE EXECUTION

An attacker might run system level code on vulnerable server and get any desired information as the output. Improper coding leads to this attack. This can be reviewed when a source code is reviewed.

One of the countermeasures is to ensure that only filtered data is used to construct the string to be executed and make sure to escape the output.

9.3 USERNAME ENUMERATION

This attack occurs when the back end validation script tells the attacker if the given username is correct or not. An attacker can guess different usernames and determine the valid ones with the help of different error messages. Sometimes, a developer creates account with the username/password as admin/admin, test/test. This also leads to an attack.

Countermeasure for this would be displaying consistent error messages to prevent disclosure of valid usernames. The developer has to make sure that these accounts have been deleted after testing is done and before it is deployed it online.

9.4 PARAMETER TAMPERING

This is a hacking which creates a change in URL associated with the web page. A hacker would be able to gain access to any information entered by an end user and then redirect it to the hacker for unauthorized use.

Countermeasures for this kind of attack include the following:

- Filter the input fields i.e.; to remove all the characters that might lead to an input injection.
- Block the packets containing input attack code and other unwanted data using application firewall.

Chapter 10 - IMPLEMENTATION

I'm trying to ensure the authorizations of each and every user. There are 3 kinds of users: Admin, HR, PM. Admin's privileges are to view the employees and add the users. So, they are shown through the screenshots below. Similarly, HR Manager's authorization is to view the HRD data and the employee list which is also shown in the below screen shots. The Project Manager's authorization is to view/add the requirements, projects to the application. He has the maximum access of everything in the project. They are shown in the screen shots below.

Currently Admin is logged into the application. This is how the home screen of the admin login page looks like.

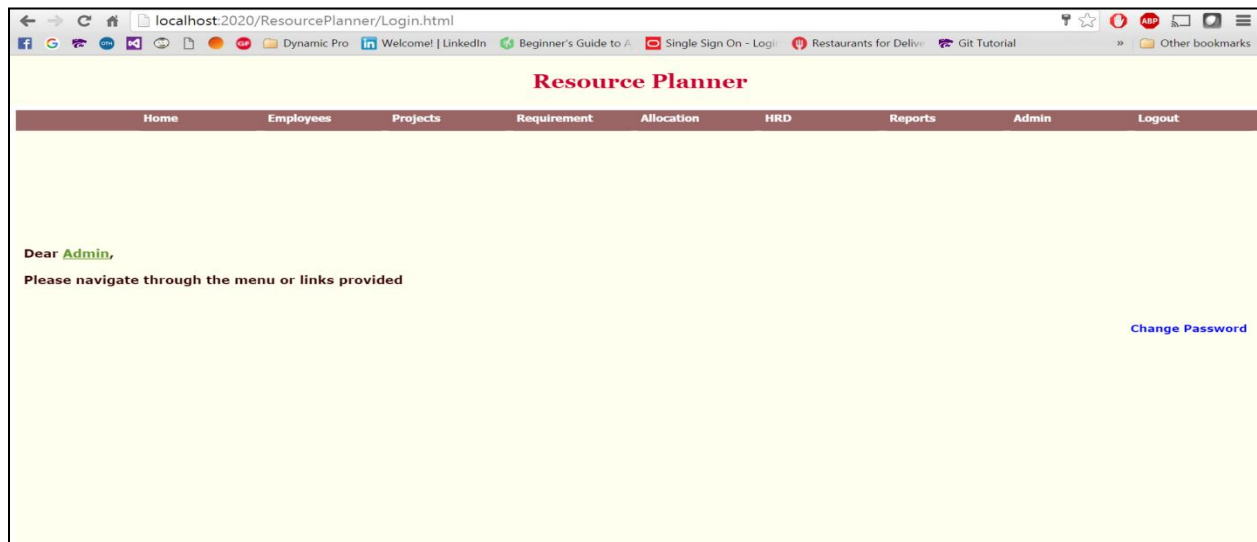


Figure 10.1: Admin Login Page

After logging into the application, the admin tries to view the list of employees in the company by clicking the employees tab and then View employees tab.

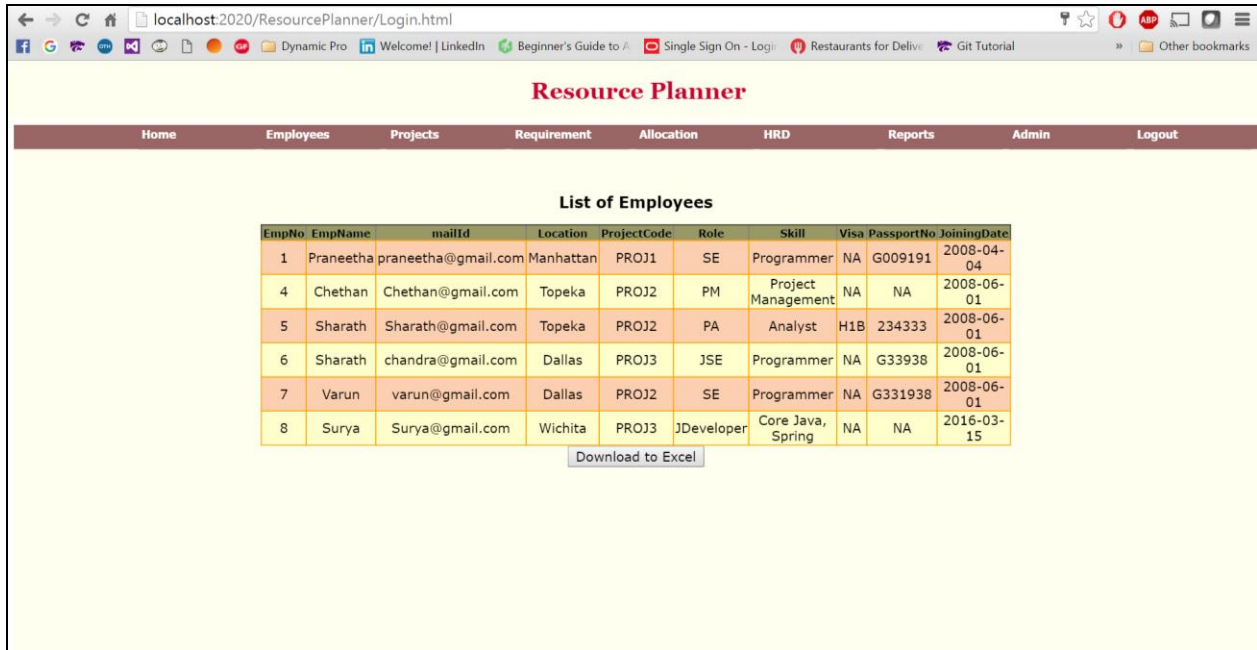


Figure 10.2: List of Employees

After the admin logs into the account, he is authorized to view the reports. So, he clicks on the reports tab to view various types of reports like Role-Wise, Project-Wise, Skill-Wise.



Figure 10.3: Role-Location Report

Admin has the authorization to add a user of the app. So, when he clicks on the Admin tab near the rightmost bar, he gets the options like ‘Adding User’, ‘Deleting a User’.

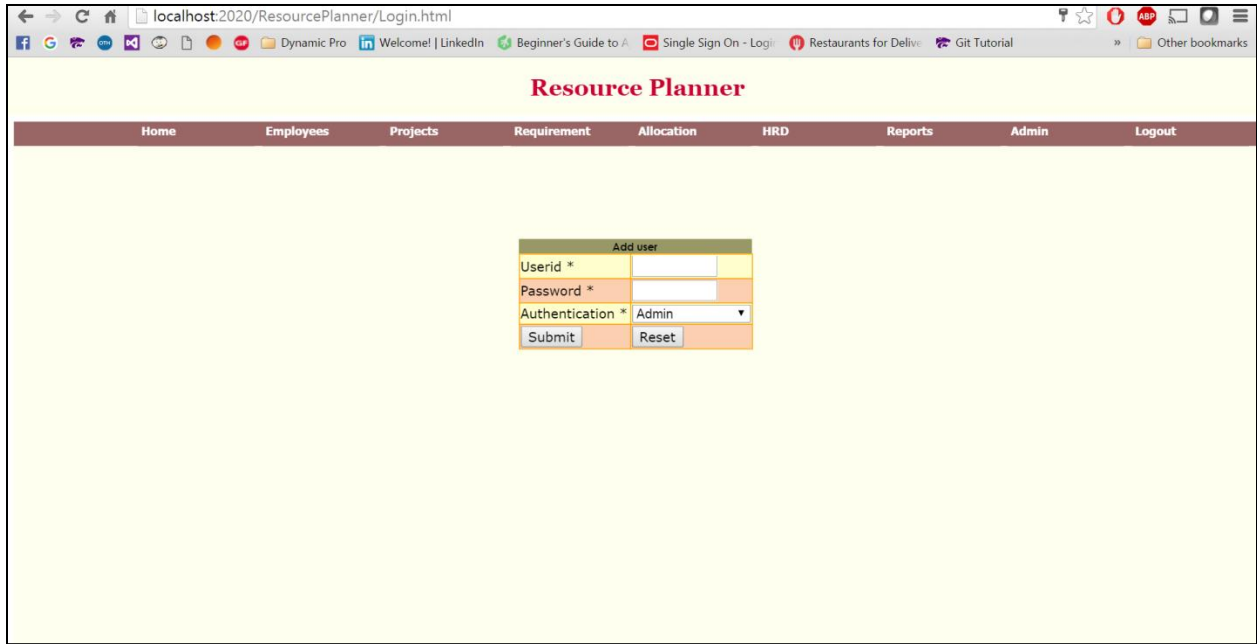


Figure 10.4: Adding User

Once the admin is logged in, he clicks on the Admin tool bar in the rightmost. He gets the options of adding/deleting the user. The admin decides to delete a user by clicking the tab.

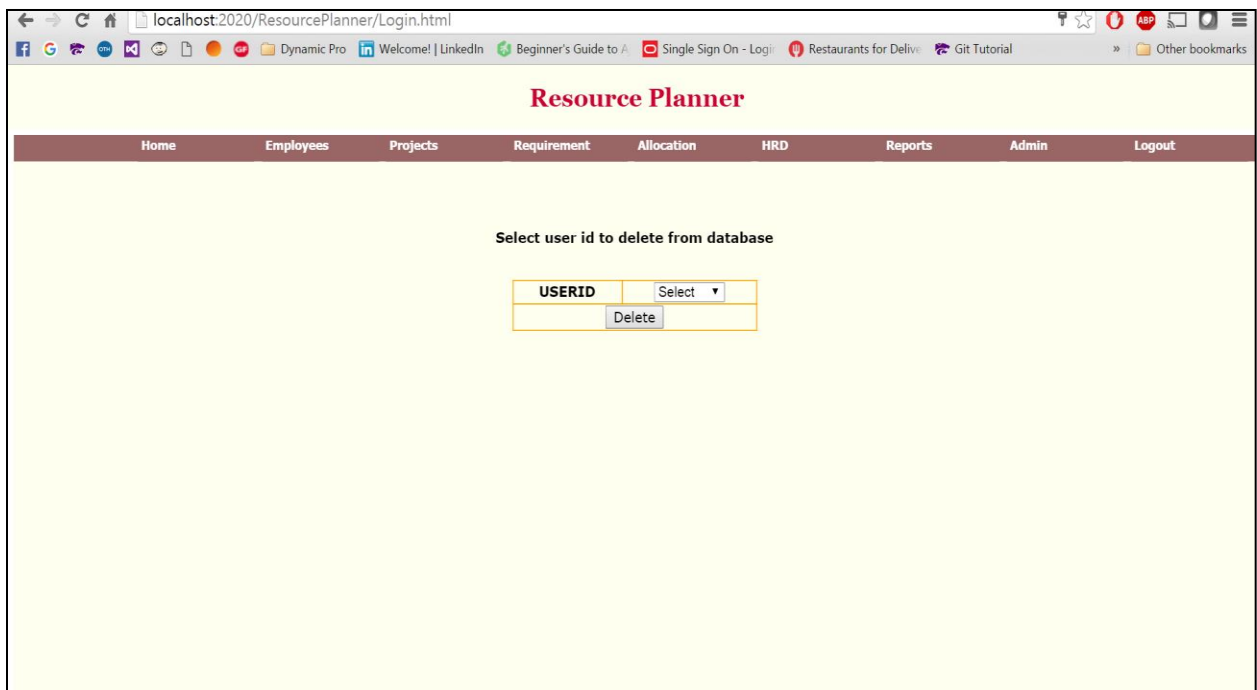


Figure 10.5: Removing User

Admin has the authorization to reset his/her password. After clicking the Reset password tab, he gets a page like this. When the user gives the userid, he will be allowed to change his password.



Figure 10.6: Reset Password

The admin will be able to view the GAP Summaries. I have given a screen shot of the GAP Summaries home page below.

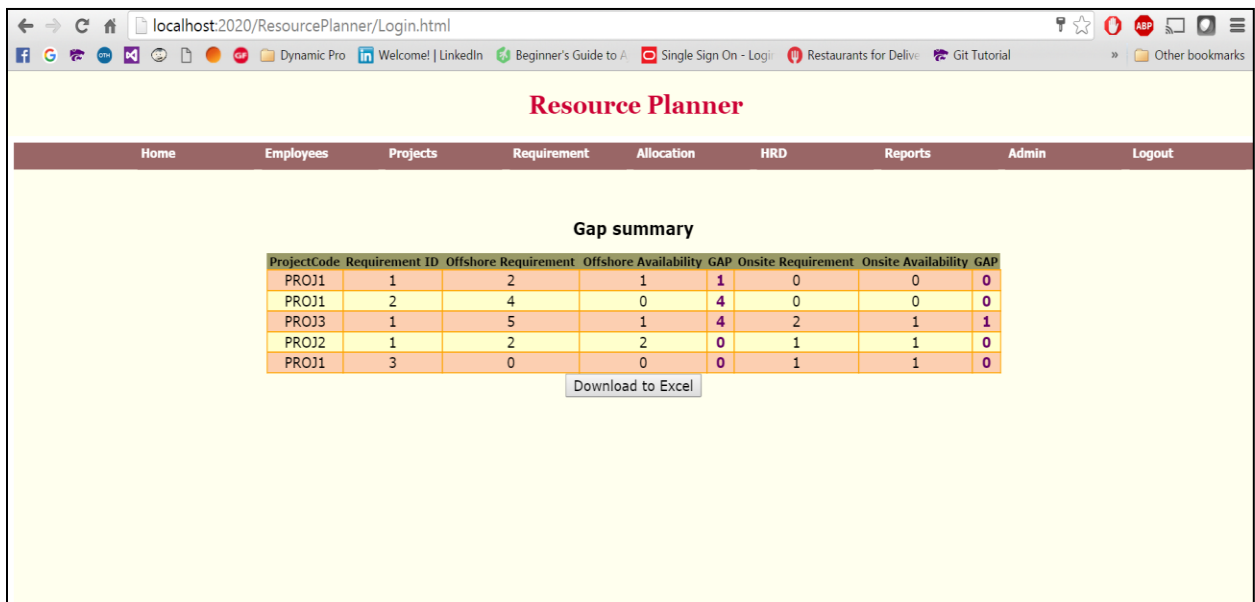


Figure 10.7: GAP Summary

Once the project manager logs into his account, he has certain privileges of viewing, adding the projects. Here, the project manager wants to view all the projects available by clicking the view Projects tab which is under Projects tab.

ProjectCode	ProjectManager	OnsiteManager	ClientName	ClientLocation	DealCost	DealDate	StartDate	Duration
PROJ1	Ankur	Rahul	ABC ltd	Manhattan	100000.00	2008-05-04	2008-06-01	20
PROJ2	Varun	Varun	Cerner Corp	Florida	1000.00	2008-08-08	2008-10-10	10
PROJ3	Sharath	Sharath	DST Systems	Seattle	234.00	2008-08-08	2008-08-08	5

Download to Excel

Figure 10.8: List of Projects

The project manager will be able to add a new project into the database by clicking on the tab ‘Add new Project’ which is present under the ‘Projects’ tab. He should be giving all the values of the attributes for which there is an asterisk.

Add new Project

Project Code *

Project Manager *

Onsite Manager *

clientName *

Client Location *

Deal Cost in US Dollors

Deal Date * mm/dd/yyyy

Start Date * mm/dd/yyyy

duration * in weeks

Figure 10.9: Adding new Project

The project manager is given privilege to remove any project from the database. He should specify the project code to remove the project from the database. He does this operation by clicking the Remove Project tab which is present under Projects tab.



Figure 10.10: Removing Project

The Project manager will be able to view the project requirement by clicking the Projects tab.

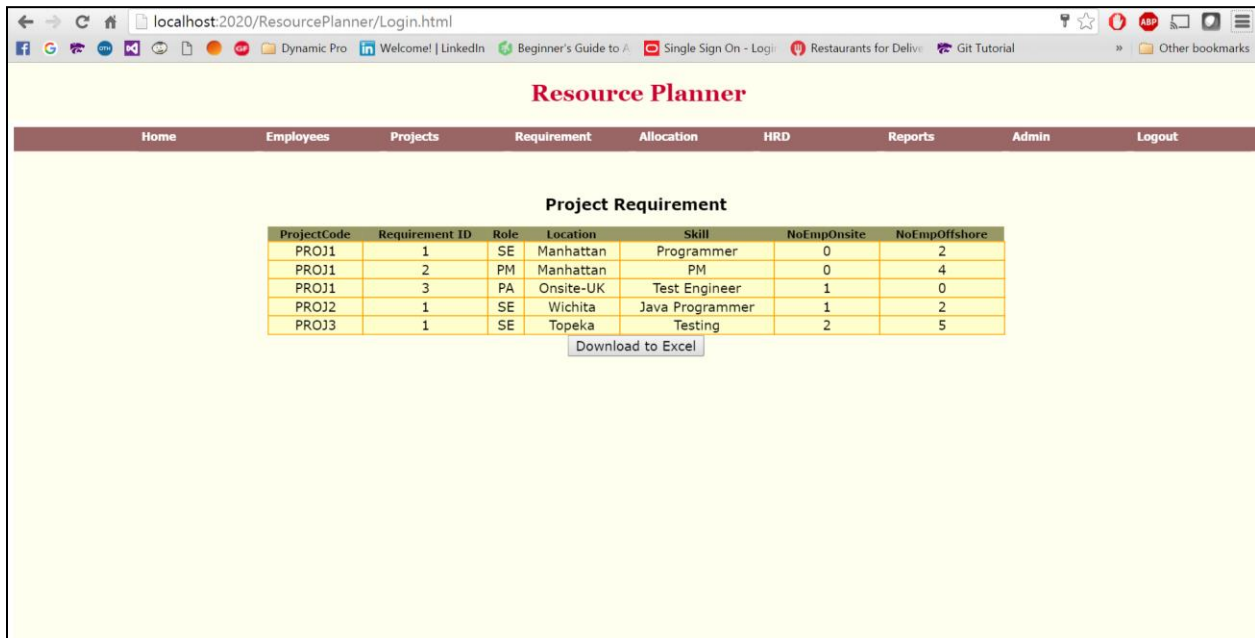


Figure 10.11: Viewing Requirements

Chapter 11 - CONCLUSION

The application has been designed successfully to meet user's requirements which have been assumed in the previous section. As per the requirements, the admin have been given privilege to view the employees of the company, to add/ remove the users of the application, resetting the password, view the management reports. The HR has been given privilege to view the employees list, add a new employee and to view the management reports. The Project Manager has been given privilege to add a new project, remove project, view/ add the requirement and allocate employees to the project. Implementing the project using Spring MVC framework is the challenge as I was pretty much new to the MVC architecture as well as Spring.

Even though I haven't made it accessible through the mobile app, I have deployed it on the Apache Tomcat Server to view it as a website. I haven't made it available as an online website as I did not deploy it in any domain.

10.1 FUTURE WORK

The application can be made more responsive even to use it in mobile. Based on the screen's size, the application should be adjusted properly in the mobile for the user to view it.

The User Interface can be enhanced further to make it look more user-friendly using latest web technologies like bootstrap, HTML5, etc.

Currently, I have provided the authentication values for each kind of user. Depending upon the authentication values, each user is given privileges to view the information. So, security is ensured for the users. But, if someone hacks the account, it would be easy for a hacker to access the account. So, this level of security can be done as a future work.

Chapter 12 - BIBLIOGRAPHY

[1] Spring MVC Framework, [Online], Retrieved April 8th and Available:

http://www.tutorialspoint.com/spring/spring_quick_guide.htm

http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm

http://www.tutorialspoint.com/spring/spring_ioc_containers.htm

http://www.tutorialspoint.com/spring/spring_architecture.htm

<http://howtodoinjava.com/spring/spring-core/different-spring-3-ioc-containers-with-example/>

[2] Spring, [Online], Retrieved April 8th, Available: <https://spring.io/docs>

[3] Spring MVC Project implementation, [Online], Retrieved April 8th and Available:

<http://crunchify.com/simplest-spring-mvc-hello-world-example-tutorial-spring-model-view-controller-tips/>

[4] JSP's, [Online], Retrieved April 8th and Available: <http://javabeginnerstutorial.com/jsp-tutorial/jsp-basics/>

[5] Java Servlets, [Online] Retrieved April 8th and Available:

<http://w3processing.com/index.php?subMenuId=97>

[6] MySQL database, [Online] Retrieved April 8th and Available:

<https://www.lynda.com/MySQL-training-tutorials/260->

[0.html?utm_source=google&utm_medium=cpc&utm_campaign=11-US-Search-Dev-](https://www.lynda.com/MySQL-training-tutorials/260-0.html?utm_source=google&utm_medium=cpc&utm_campaign=11-US-Search-Dev-)

[MySQL&cid=11-us:en:ps:lp:prosc:s50:260:all:google:mbm-](https://www.lynda.com/MySQL-training-tutorials/260-0.html?utm_source=google&utm_medium=cpc&utm_campaign=11-US-Search-Dev-MySQL&cid=11-us:en:ps:lp:prosc:s50:260:all:google:mbm-)

[learning_mysql&utm_content=64967313546&utm_term=%2Blearning%20%2Bmysql&device=](https://www.lynda.com/MySQL-training-tutorials/260-0.html?utm_source=google&utm_medium=cpc&utm_campaign=11-US-Search-Dev-MySQL&cid=11-us:en:ps:lp:prosc:s50:260:all:google:mbm-learning_mysql&utm_content=64967313546&utm_term=%2Blearning%20%2Bmysql&device=)

[c](https://www.lynda.com/MySQL-training-tutorials/260-0.html?utm_source=google&utm_medium=cpc&utm_campaign=11-US-Search-Dev-MySQL&cid=11-us:en:ps:lp:prosc:s50:260:all:google:mbm-learning_mysql&utm_content=64967313546&utm_term=%2Blearning%20%2Bmysql&device=c)

[7] JDBC Connections, [Online] Retrieved April 8th and Available:

<https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html>

[8] Web server, [Online] Retrieved April 8th and Available:

<http://www.ibm.com/developerworks/library/os-eclipse-tomcat>

[9]Testing, [Online] Retrieved April 8th and Available:

<http://www.thegeekstuff.com/2015/02/jmeter-load-testing/>

<https://www.digitalocean.com/community/tutorials/how-to-use-apache-jmeter-to-perform-load-testing-on-a-web-server>

<http://jmeter.apache.org/usermanual/build-monitor-test-plan.html>

<http://reqtest.com/uncategorized/differences-between-different-test-levels/>

[10] Security, [Online] Retrieved April 14th and Available:

<https://peterhgregory.wordpress.com/2008/03/02/countermeasures-for-web-application-parameter-tampering-attack/>

<http://www.symantec.com/connect/articles/five-common-web-application-vulnerabilities>