**Major Project Report on**

**WEB APPLICATION DEVELOPMENT USING SPRING-BOOT**

****

By

**Priti Kumari (Reg. No. 201600466)**

**Mayur Mahanta (Reg. No. 201600491)**

*In partial fulfilment of requirements for the award of degree in*

Bachelor of Technology in Computer Science and Engineering

(2020)

****

##### *Under the Project Guidance of*

**External Guide Internal Reviewer**

**Mr. Amit Ghosh, Mr. Suman K. Kar,**

**EA Delivery Lead of Sprint Account, Assistant Professor I,**

**Ericsson India Global Sikkim Manipal Institute**

**Service Pvt. Ltd.  Of Technology.**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

## (A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

**PROJECT COMPLETION CERTIFICATE**

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **06 January 2020 to 08 May 2020** and have successfully completed the project entitled **“Web Application Development Using Spring Boot”**in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No. | Name of Student(s) | Course |
| 201600466 | Priti Kumari | Computer Science & Engineering |
| 201600491 | Mayur Mahanta | Computer Science & Engineering |

Mr. Amit Ghosh,

EA Delivery Lead of Sprint Account,

Ericsson India Global Service Pvt. Ltd.

**PROJECT REVIEW CERTIFICATE**

This is to certify that the work recorded in this project report entitled **“Web Application Development Using Spring Boot”** has been jointly carried out by **Ms. Priti Kumari (Reg. 201600466) and  Mr. Mayur Mahanta (Reg. 201600491)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Major Project Viva Examination.

Mr. Suman K. Kar,

Assistant Professor I,

Department of Computer Science & Engineering,

Sikkim Manipal Institute of Technology,

Majitar, East Sikkim – 737136.

**CERTIFICATE OF ACCEPTANCE**

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Mr. Amit Ghosh, EA Delivery Lead of Sprint Account** of **Ericsson India Global Service Pvt. Ltd., Kolkata** from 06 January 2020 to 08 May 2020 on the project entitled **“Web Application Development Using Spring Boot”.**

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No. | Name of Student(s) | Project Venue |
| 201600466 | Priti Kumari | Ericsson India Global Service Pvt. Ltd. |
| 201600491 | Mayur Mahanta | Ericsson India Global Service Pvt. Ltd. |

Prof. (Dr.) Kalpana Sharma,

Professor & HOD,

Computer Science & Engineering Department,

Sikkim Manipal Institute of Technology,

Majitar, Sikkim – 737136.

**DECLARATION**

We the undersigned, hereby declare that the work recorded in this project report entitled **“Web Application Development Using Spring Boot”** in partial fulfilment for the requirements of award of B.Tech. in Computer Science & Engineering from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bona-fide project work carried out at KOLKATA under the supervision and guidance of **MR. AMIT GHOSH** ofEricsson India Global Service Pvt. Ltd., Kolkata.

The results of this investigation reported in this project have so far not been reported for any other Degree / Diploma or any other Technical forum.

The assistance and help received during the course of the investigation have been duly acknowledged.

**Ms. Priti Kumari (Reg. No. 201600466) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Mayur Mahanta (Reg. No. 201600491) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACKNOWLEDGEMENT**

We feel to acknowledge our indebtedness and deep sense of gratitude to our line manager, **Mr. Amit Ghosh,** **EA Delivery Lead of Sprint Account, Kolkata** who took time from his busy schedule and interacting with us and checking the progress of the project.

We would also like to thank our mentors, **Mr. Avik Dey,** Senior Solution Integrator, Kolkata and **Mr. Soumen Das,** Senior Solution Integrator, Kolkata who provided us the essential resources and materials along with proper understanding of the project and **Mr. Arpan Nandi,** Solution Integrator, Kolkata who continuously guided us in each and every step of the project and gave his insights on updates to the project.

We would like to thank our internal reviewer, **Mr. Suman K. Kar**,for guiding and helping us during the entire duration of the project and for the valuable time that he invested into reviewing our project and support during the tenure of the project.

We pay our deep sense of gratitude to **Prof.** **(Dr.)** **Kalpana Sharma**, **H.O.D**, **Computer Science & Engineering Department**, **Sikkim Manipal Institute of Technology** to encourage us to the highest peak and to provide us the opportunity to prepare for the project.

We are immensely obliged to our Major Project Coordinators, **Mr. Sourav Paul, Mrs. Chitrapriya Ningthoujam, Mr. Santanu Kr. Mishra and Mr. Biswaraj Sen** for elevating inspiration and kind supervision in completion of our project.

Last, but not least, we would like to thank our **Computer Science & Engineering Department**, **Sikkim Manipal Institute of Technology** and to all the faculty members for giving us continuous support and guidance that has helped us in completion of our project.

Priti Kumari (201600466) ………………….

Mayur Mahanta (201600491) ………………….

**DOCUMENTED CONTROL SHEET**

|  |  |  |
| --- | --- | --- |
| 1. | Report No. | CSE/Major Project/External/B.Tech./6/2020 |
| 2. | Title of the Report | Web Application Development Using Spring Boot |
| 3. | Type of Report | Technical |
| 4. | Author(s) | Ms. Priti Kumari (Reg. No. 201600466)  Mr. Mayur Mahanta (Reg. No. 201600491) |
| 5. | Organizing Unit | Ericsson India Global Service Pvt. Ltd., India |
| 6. | Language of the Document | English |
| 7. | Abstract | The system uses Spring Boot, HTML, JavaScript and CSS frameworks to help create a web-based application to manage and maintain employees’ data in a secured and efficient manner. |
| 8. | Security Classification | General |
| 9. | Distribution statement | General |

**LIST   OF   CONTENTS**

**SL. No. Title Page No.**

0 Abstract 11

1 Introduction 12

1.1 General Overview of the Problem 14

1.2 Survey 14

1.3 Problem Definition 16

1.4 Analysis of the Problem and SRS 17

1.5 Proposed Solution Strategy 18

1.6 Preliminary User’s Manual 18

1.7 Organization of the Report 19

2 Design Strategy for Proposed Solution 21

2.1 Architectural Diagram 21

2.2 Detailed Diagram: Object Oriented Design 22

3 Detailed Test Plan 27

4 Implementation Details 30

5 Results and Discussion 39

6 Summary and Conclusion 44

6.1 Summary of Achievements 44

6.2 Difficulties Encountered and How They Were Tackled 44

6.3 Limitation of the Project 45

6.4 Future Scope of Work 45

6.5 Any Special Observation 45

7 Final User’s Manual 46

8 Gantt Chart 47

9 References 48

**LIST   OF   FIGURES**

**Figure No. Figure Name Page No.**

1.1 Flowchart of RMS (Resource Management System) 13

2.1.1 Spring Boot Flow Architecture 21

2.1.2 Block Diagram of The Application 22

2.2.1 UML (Unified Modelling Language) Use Case Diagram 23

2.2.2 UML Class Diagram 24

2.2.3 UML Activity Diagram 25

2.2.4 Data Model for RMS 26

4.1 Project Structure of RMS 30

4.2.1 Application Properties File of RMS 32

4.2.2 Xampp Control Panel Version 3.2.4 32

5.1 Login Module for RMS 39

5.2 Resource Module for RMS 40

5.3 Admin Module for RMS 41

5.4 Manager Module for RMS 42

5.5 Validation Output of RMS 43

7.1 Diagram to Show the Flow of the Application 46

8.1 Gantt Chart 47

**LIST   OF   TABLES**

**Table No. Table Name Page No.**

1.1 Manual Survey 14

3.1 Test Cases of RMS 27

## ABSTRACT

RMS, which stands for Resource Management System, is a full stack web development project. This system is developed for the effective maintenance of the records of all the employees who work at Ericsson. The front-end pages are designed and developed using HTML, JavaScript and CSS. HTML is used to create web pages; JavaScript is used to control the behaviour of different elements and CSS is used for styling the webpage. These are used to create an interactive user-interface for a web-based application. In the back end, MySQL workbench is used for maintaining the database of the application and spring boot framework is used for connecting the user-interface with the database and to reduce the development, unit test and integration test time. The POJO class is used to improve the usage of the object and makes it easier to fetch and update details to the database from the frontend.

The system uses the frontend technologies and the spring boot framework with MySQL workbench for dataset storage to help Ericsson create a platform to manage resources and maintain their data in a secured and efficient manner.

# **1. INTRODUCTION**

Java 8 (Eclipse IDE) is a general-purpose programming language. It is class-based, object-oriented and platform independent, which is intended to let application developers write once, run anywhere (WORA). HTML stands for HyperText Markup Language and it provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript. CSS stands for Cascading Style Sheets; it is a simple design language and it helps to style the structure content of the website and make it look presentable. JavaScript is a logic-based programming language that can be used to modify website content and make it behave in different ways in response to a user’s actions. Common uses for JavaScript include confirmation boxes, calls-to-action, and adding new identities to existing information. These languages are front-end programming languages used for web development (front-end development of a website).

Spring Boot is a framework to ease the bootstrapping and development of Spring Application. It is a project built on top of spring framework, which provides a simpler and faster way to set up, configure, and run both simple and web-based applications. The resource management system is developed using the simple CRUD (Create, Retrieve, Update and Delete data) application with Spring Boot and Spring Data JPA (Java Persistence Api). It is a simple web application that will expose few REST endpoints for CRUD related operations. Spring Data JPA is a sub-project that comes under the Spring Data project; it is all about creating Entities and Repositories. Entity helps the java object to relational table and relational table back to java object. Repository helps to perform CRUD related operations with Entity, i.e., it can access data from the database by writing repository interfaces that extend Crud Repository or JPA Repository. The database connectivity is achieved by using JPA and Hibernate.

The resource management system is a maven project that is developed using the above-mentioned languages and frameworks. The important functionality of this system is to create an efficient and interactive user interface for the employees following a systematic order. All the employees have any of below three roles, i.e., Admins, Managers and Resources.

1. “Resources” are the employees who work at Ericsson. They have access to their own details, can update their details if required and can view their own feedback from their specified managers. The Admins and Managers also come under Resources.
2. “Managers” are the resources having another certain number of resources under them. They have access to maintain the work activities of their resources and keep track of their progress. He can view feedback given to all the resources.
3. “Admins” are the administrators that have the access to add new resources inclusive of both the employees and interns, update their details, maintain their roles, their departments and the projects associated with them. This system helps the admins to maintain these records in an orderly manner. He must provide roles to each resource i.e.; he must create a manager.

This application proves beneficiary to employees, as it is easily accessible. Moreover, it provides them an ease in manipulating the records following a proper procedure within their work hours without affecting their work. The figure1.1 shows the flow of the system according to the role given. View profile, update profile and search are common to all the roles.

A close up of a map

Description automatically generated

Fig 1.1:  FLOWCHART OF THE RESOURCE MANAGEMENT SYSTEM

**1.1 GENERAL OVERVIEW OF THE PROBLEM**

The main problem in the traditional method is the manual handling of the records and use of paper for storing the information of the employees. This method has more chances of human error while entering and modifying data. In addition, there is no direct access of employee’s personal details; the details are accessed after following certain lengthy procedures. Modification of the details is also time consuming. This method guarantees no full privacy and security of employees’ details. Therefore, this system is developed for overcoming all these problems faced by the employees. The system is automatic and is very timesaving.

# **1.2** **SURVEY**

This following are some of the manually researched insights related to the project. These findings are used to learn about the frameworks and used for the development of the application.

1.1 TABLE: MANUAL SURVEY

|  |  |  |  |
| --- | --- | --- | --- |
| **SL. NO.** | **Manual Research** | **Findings** | **Relevance to the Project** |
| 1. | “*Employee Management System*” - INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY (IRJET) VOLUME: 06 ISSUE: 05 | MAY 2019, E-ISSN: 2395-0056, P-ISSN: 2395-0072  Mr. Pratik Uday Shankar Singh, Mr. Hemant Singh Fartyal, Mr. Khan Abdul Ahad Zubair, Prof. Akshata Laddha. | Relevance of having an Employee Management System with clear objectives. | Less paper use and removal of redundancy and making the system interactive. |
| **SL. NO.** | **Manual Research** | **Findings** | **Relevance to the Project** |
| 2. | “Spring Boot Framework” | Provides several modules with a vast range of extending services and maintains the workflow structure. | Highly beneficial for making spring boot applications. |
| 3. | “Web Design” | Provides the basic understanding of the web pages and its layout structure. | Mostly useful for end-users in creating the frontend part of any system or application. |
| 4. | “MySQL Workbench” | Delivers visual tools for creating, executing and optimizing the SQL queries. | Helps in the database connectivity and supports functionalities like create, insert, alter, update, delete etc. |
| 5. | “Spring JPA Repository” | JPA handles most of the complexity of JDBC-based database access and object-relational mappings. | Spring Data JPA reduces the amount of boilerplate code required by JPA, making the implementation of the persistence layer easier and faster. |

# **1.3 PROBLEM DEFINITION**

Manual handling of information of the employees result in several challenges. The use of paperwork in handling some of these processes could lead to human error, papers may end up in the wrong hands and it is time consuming. The employees are not able to access and manage their personal information directly without having to go through their HR departments or their managers. Another challenge is that multinational companies will have all the employee information stored at the headquarters of the company making it difficult to access the employee information from remote places when needed at short notice.

This system will maintain employee information in a database by fully privacy and authority access. The project aimed at setting up an employee information system about the status of the employee, the educational background and the work experience in order to help monitor the performance and achievement of the employee through a password-protected system. This system should consist of an application program, on one hand, and a database (repository of data) on the other. The program should perform the basic operations upon the database as retrieving, inserting, updating and deleting data. The logical database model (tables, their content and the relationships between them) should respond to the given task and cover the basic requirements. The Interface of the program should be user-friendly, and the program should be as easy for use as it is possible. Controls and forms both logically and functionally be related within the program and fully respond to the structure of the database. Exception handling should be taken into an account during the system’s development due to eventual exceptions that may occur.

The sole purpose for designing this system is to help the employees use this technology in an accurate and timely process. The system also preserves the privacy and authority access for every employee.  It supports the maintenance of the record of resource data and information.

It makes easy controlling of the employee.  It also serves in eliminating or reducing the hardships of the existing system and lastly to avoid errors while entering the data.

**1.4 ANALYSIS OF THE PROBLEM AND THE SRS**

**1.4.1 PROBLEM ANALYSIS**

There are a few challenges faced by employee management system which are listed below -

A disciplined framework is required to manage the resource management system in the organization. The poor performance management is generally considered as a reactive action. Due to which most of the time it is deferred and along these lines transforms into an exchange that is hard to make significant. It requires the association to work off one all- inclusive strategy separated into applicable parts and regions of obligation. The manual process is a time-consuming task for handling various data and maintaining them simultaneously which reduces the productivity as well.

**1.4.2 SRS DOCUMENTATION**

Widely known as the Software Requirement Specification having a purpose, an overall description and some specific requirements. The purpose here is to build a resource management system. An SRS document collects all the specific requirements needed to build the system. The description of the rms system includes various features and functionalities. The document has an introduction, which introduces the basis about how the rms system will be built. Purpose is mentioned earlier. Scope describes the scope of the project i.e., the systems scope providing benefits, certain objectives and goals to the client. Some definitions and acronyms used for designing the code. The client’s requirement serves the main purpose since the system will be created based on the client’s requirement. Few assumptions and dependencies are needed which illustrates whether the intended time duration will be enough to build the system. An SRS document has some requirements, which includes the functional requirements, non-functional requirements, some system features and external interface requirements. The functional requirements are essentials needed to build the rms system. There are some external requirements, which include the hardware requirements, software requirements. There are some non-functional requirements, which include safety of the system; performance of the rms system, how secure is the system, how well the system works depicting the quality of it. These are few things required to create an SRS document.

**1.5 PROPOSED SOLUTION STRATEGY**

The proposed system is designed to eliminate all the drawbacks of the existing employee management software. The system shall be responsible for maintaining information about employees. The main features include a) Employee profiles, b) Segregated task management, c) Notifications, d) Search Fields and Employee Self-Service (ESS). The system should maintain the design consistency in every field. The design structure of the system should have the same header/logo, heading style, fonts, navigations etc. Every page layout of the system should highlight contrast between the text and background area fields. Monitor size should be taken into consideration. The provided features should be user-friendly and clear meaning users should not have a hard time trying to navigate the site. Navigation links should be consistent and clearly labelled. All navigation links should be uniform and only point to the intended page/site. These are some of the basic rules and regulations that needs to be implemented in order to build a proper functioning web-based system.

**1.6 PRELIMINARY USER’S MANUAL**

The preliminary users of the system are the developers of the web-application. The requirements of the application were clearly mentioned in the project document, so the system was built using a temporary dataset in the original database structure. The tasks were divided among these developers and the roles were assigned temporarily for testing the flow of the system. There are three roles mentioned, so among the group of developers, one was assigned as admin, two others were assigned as managers and the rest were given the role of the resource. As the roles had some specific functions to perform, the roles were divided to check whether all the functionalities of the system are performing in the required manner. The system was designed and tested keeping in mind the requirements gathered from the project document. The manual must not only describe everything the user does see, including all interfaces; it must also refrain from describing what the user does not see. That is the implementer’s business, and there his design freedom must be unconstrained. The architect must always be prepared to show an implementation for any feature he describes, but he must not attempt to dictate the implementation.

**1.7 ORGANIZATION OF THE REPORT**

**1.7.1 INTRODUCTION**

This section includes the overall view of the project i.e., the basic problem definition and general overview of the problem, which describes the project in layman terms. It also gives a brief about the techniques used for approaching the problem statement. Then is the survey, which includes the manual survey that is relevant to the project. This section also specifies the software used along with their versions and the proposed solution strategy i.e. the steps followed for the implementation of the project.

**1.7.2 DESIGN STRATEGY FOR THE SOLUTION**

This section consists of two main sub-sections; one is the Architectural Diagram that shows the architectural flow of process that handles the back end of the application and creation of the front end and its interaction with the back end. The other section includes of a Detailed Diagram that includes the Object-Oriented Designs, which includes the use case diagram, class diagram, activity diagram and the data model diagram.

**1.7.3 DETAILED TEST PLAN**

This section consists of the testing criteria used to test the application. It consists of a table that contains number of test cases and its result details. The table consist of five parts. First column contains the module names on which the test cases are prepared. Second column describes the testing purpose. Third column describes the testing condition and the last two columns are the details of the expected outcome and the actual result.

**1.7.4 IMPLEMENTATION DETAILS**

This section includes the explanation of the structure of the project and its algorithm for developing the different modules of both front end and the back end.

**1.7.5 RESULTS AND DISCUSSIONS**

This section has all the screenshots of the entire implementation of the application i.e., user interface and their description.

**1.7.6 SUMMARY AND CONCLUSION**

This section consists of five sections. The first section consists of summary of achievements that describes the implementation done successfully. Second the difficulties that were encountered during the implementation of the project. Then the limitations of the project describe certain constraints that was not met the project. Scope of future work describes what future changes can be incorporated in the project to enhance its functionality. Fifth, the conclusion that simply states the summary of the entire project report.

**1.7.7 GANTT CHART**

It illustrates the project schedule and the activities performed in time frame. The chart lists the activities to be performed on vertical axis and the time intervals on horizontal axis.

**1.7.8 REFERENCES**

This section states the research papers and the sites referred for this project in its study and implementation.

**2. DESIGN STRATEGY FOR PROPOSED SOLUTION**

**2.1 ARCHITECTURAL DIAGRAM:** The system is implemented using Spring Boot Application. The spring boot follows a layered architecture, which consists of four layers. The Presentation layer includes the front-end view which handles the Http request, translates the JSON parameter to object, and authenticates the request and transfers it to the business layer. The Business Layer performs authorization and validation, handles all the business logic. It consists of service classes, which use the services provided by data access layers. The Persistence Layer contains all the storage logic and translates objects from and to between the business and the database layers. Lastly, the Database Layer performs the CRUD (create, retrieve, update and delete) operations. These layers communicate with the layer above or below in a hierarchical structure. The figure4 shows the flow of the spring boot architecture. The flow contains all the four layers.

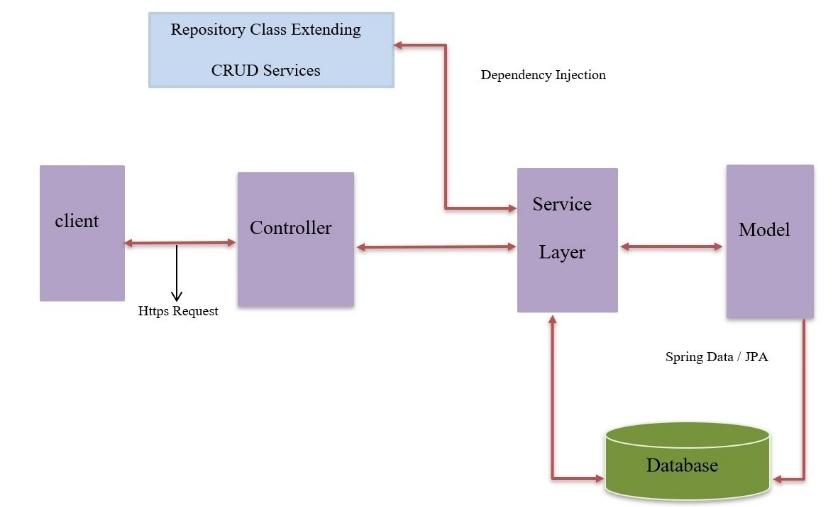


Fig 2.1.1: SPRING BOOT FLOW ARCHITECTURE

1. The clients are the users that make the http request (GET or POST) from the front end. This is the **Presentation layer** where users interact with the system through the system website.
2. The request goes to the controller, which maps the request. @Controller and @RestController, which then handles the response to the view, indicate the controller. The service class is called from the controller to perform any data related business logic. This is the **Business layer** where business logic is written in the function of service class and controller class.
3. The model class then contains the mapping of its repository with the actual database entity (table) using the getters and setters of each entity module. This is the **Persistence layer** where the repository connects service class to the model class to the database.
4. The service class is mapped with the JPA or CRUD Repository that contains simple select, create, insert, update and delete SQL queries. This is the **Database layer** where SQL queries are performed which is then mapped to the database using the application property file in spring boot.

A close up of a device

Description automatically generated

Fig 2.1.2: BLOCK DIAGRAM OF THE APPLICATION

**2.2 DETAILED DIAGRAM:** Object-Oriented Design is a software development approach to design and implement software systems as a collection of interacting stateful objects with specified structure and behaviour. The state is distributed among the objects, and each object handles its state data. There are several fundamental concepts defining OOD, some of them are mentioned below.

**2.1 Use-Case Diagram**: The following use-case diagram explains the interaction among the elements and helps to understand the system requirements. It is used to capture the dynamic aspect of the design. There are three actors named Resource, Admin and Manager. Nine use-cases represent specific functionality of the system.  1) Upload resource data: The actor “Admin” uploads the data to the database. 2) Login user: All the actors can login according to their specific role. 3) Search Resource: All actors can search other resources. 4) Add and update project: The Admin add and update the project of all the resources. 5) Modify resources: All the actors can modify their own details. 6) Approve or Decline modification of resource: Manager gets the modification request notifications from his resources and he approves or declines each request. 7) Allocate role to resources: Admin allocates roles to each resource. 8) Add and update features: Admin adds and updates the features. 9) Give feedback to resources: Managers give feedback to his resources and Resources view its own feedback given by his manager.

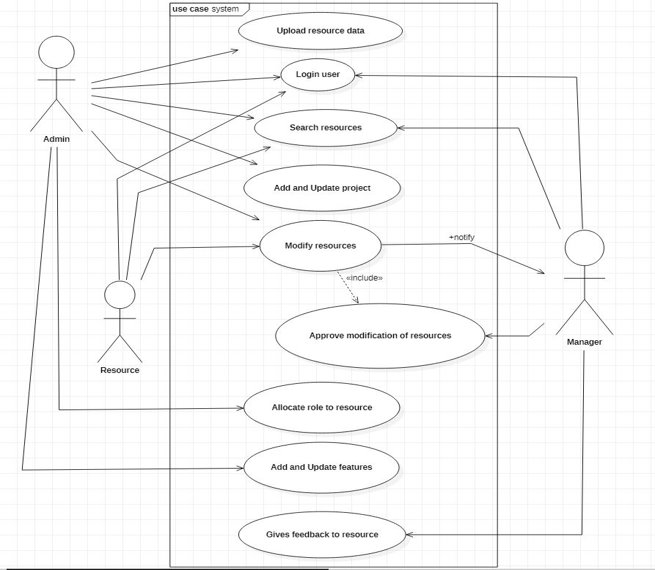


Fig 2.2.1:  UML USE CASE DIAGRAM

**2.2 Class-Diagram**: The following class diagram represents the static diagram explaining the static view of the program by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects. The rms class diagram consists of nine classes namely Resources, Department, Project, Role, Feedback, Feature, Project\_Resources, Role\_Feature, and Role\_Resource. These classes have their own attributes and operations. These classes have relations like generalization, association and reflexive association. These classes contain one-to-one mapping and many-to-one mapping with each other. The attributes of each class are the member variables whereas the operations are the services that the class provides.

**A close up of a map

Description automatically generated**

Fig 2.2.2:  UML CLASS DIAGRAM

**2.3 Activity Diagram:** The following activity diagram illustrates the flow control of the resource management system by referring to the execution of a use case. It focuses on the condition of flow and the sequence in which it occurs. A cause of an event can be shown using the activity diagram. The activity starts from the resource, since all employees come under resource. The resource starts the activity by entering login credentials, which is validated with the details of the resource from the database through the server. If the validation is incorrect, the system stops and redirects it to the login page with an error message. If the validation is correct, the resource is logged in according to the role assigned to his/her. If he/she is assigned as admin then he/she has some activity that he/she has to perform. He/she can add new resources, assign roles and projects to the resource, and can edit or delete resource details. If he/she is assigned as manager then he/she has the activity to view the resources under him/her, give feedback to them, view the feedback of the resource, and approve the data modification request from the resources under him/her. If he/she is just a resource then he/she has the activity to view his/her details, edit his/her own details and view the feedback given by his/her manager. The activity stops when the resources logout from the system.

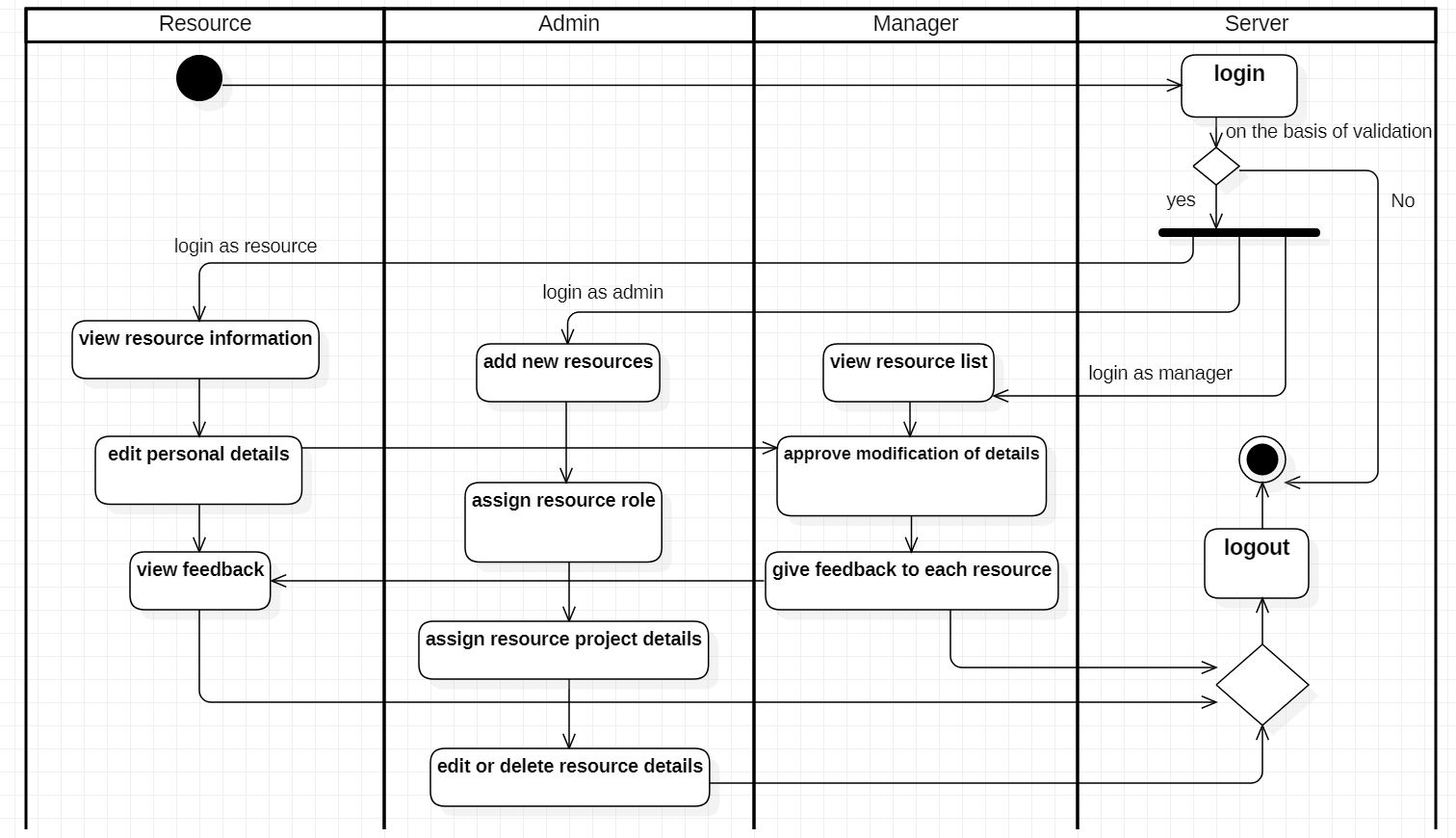


Fig 2.2.3: ACTIVITY DIAGRAM OF RMS

**2.4 Data Model:** Data Model defines the logical design and structure of a database describing how it is modelled. It introduces the abstraction of the entities, their relationship between themselves, meaning how each data is connected to each other, and how the data can be stored, accessed and updated in a database management system. There are certain sets of symbols used to describe a few relationships of the entities for the user to understand and those include the foreign key, the primary key etc. It also gives a brief about how the final view of a system looks after its complete implementation.

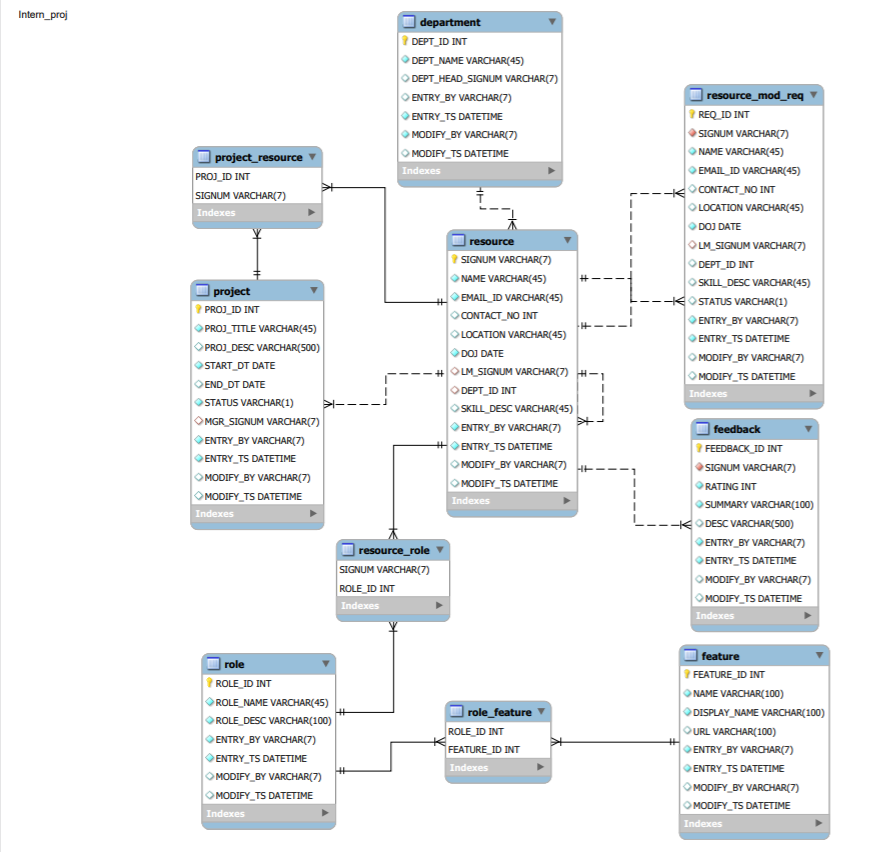


Fig 2.2.4:  DATA MODEL FOR THE RMS

**3. DETAILED TEST PLAN**

Detailed Test Plan can be considered as a document containing certain plans for all the testing activities to be done in order to achieve a quality product. A test plan document can be derived from a SRS document or any use case document. It is generally prepared by the testing integrators who decide what to test, what not to test, how to test, when to test and who will do what test. A test plan is generally kept up to date in order to achieve a successful testing of the project. The overall purpose of testing is to ensure that the rms system meets all its functional and business requirements. It also ensures validating the quality, usability, reliability and performance of the rms application. The Test Plan identifies the details of the tests, identifying the associated test case areas within the product. Testing for rms will follow the Blackbox approach. Certain test cases will be designed and arranged based on the requirements and functionalities. Suppose when a test fails because of the defect then that defect is reported, and a new version of the software is expected that has had the defect fixed. In such cases, execution of the test is done again to confirm whether the defect got fixed or not. This is known as confirmation testing and known as re-testing.

Few test cases are mentioned in the table below:

3.1 TABLE: TEST CASES OF RMS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL.NO.** | **Test Case** | **Test Purpose** | **Test Condition** | **Expected Outcome** | **Actual Result** |
| 1. | Login | Check the signum and password. | If the entered details are incorrect then error messages will be displayed. | Grant access to the applicable main system. | The resource successfully logs into the system on submitting correct login credentials. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL.NO.** | **Test Case** | **Test Purpose** | **Test Condition** | **Expected Outcome** | **Actual Result** |
| 2. | Add user | To ensure a new resource is added to the system successfully. | If a resource already exists in the system, an error message saying, “resource already exists” should display. | A new resource should be successfully added into the system. | If the signum of a new resource does not exist then it is successfully added, else an error message will pop up. |
| 3. | Edit Resource Details | To ensure that once different details are provided on the edit resource details form and submitted, these details are altered in the database to reflect the recent changes. | On the edit resource details form provide different information form what is currently being displayed. | When the form is altered, similarly, the details should be altered in the database as well and a confirmation message of the change should be displayed. | Once the data in the form is altered and the submit button clicked the details in the database are altered and a confirmation message of the change is displayed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL.NO.** | **Test Case** | **Test Purpose** | **Test Condition** | **Expected Outcome** | **Actual Result** |
| 4. | Search | To ensure whether the searched items are properly displayed on the screen. | The search button should work only for the list of fields to be searched else, an empty screen with no result will be displayed. | The search module should work for every available field used in the system. | On clicking the search button, the searched fields will be displayed on the screen and will be updated in the database as well. |
| 5. | Notification | Check whether the notification functions well or not. | In this module, approve and decline check boxes should work properly and show the updates accordingly else errors will be thrown. | On clicking the approve icon, the defined update should be done and updated in the database as well and the modified changes should be highlighted. It goes the same for the decline icon. | On clicking the decline icon, the changes will not be reflected but should ensure it functions well and similarly no changes in the database should be highlighted. |

**4. IMPLEMENTATION DETAILS**

The system is implemented using Spring Boot Application, which consists of four layers. The spring boot uses annotations to provide information about the program and it is not part of the program itself. Compile time annotations are checked by the compiler at compile time. The @SpringBootApplication, @EnableAutoConfiguration, @Configuration, @Controller, @RestController, @Repository, @Service, @Autowired, @Override @Query are some of the annotations used in the project.

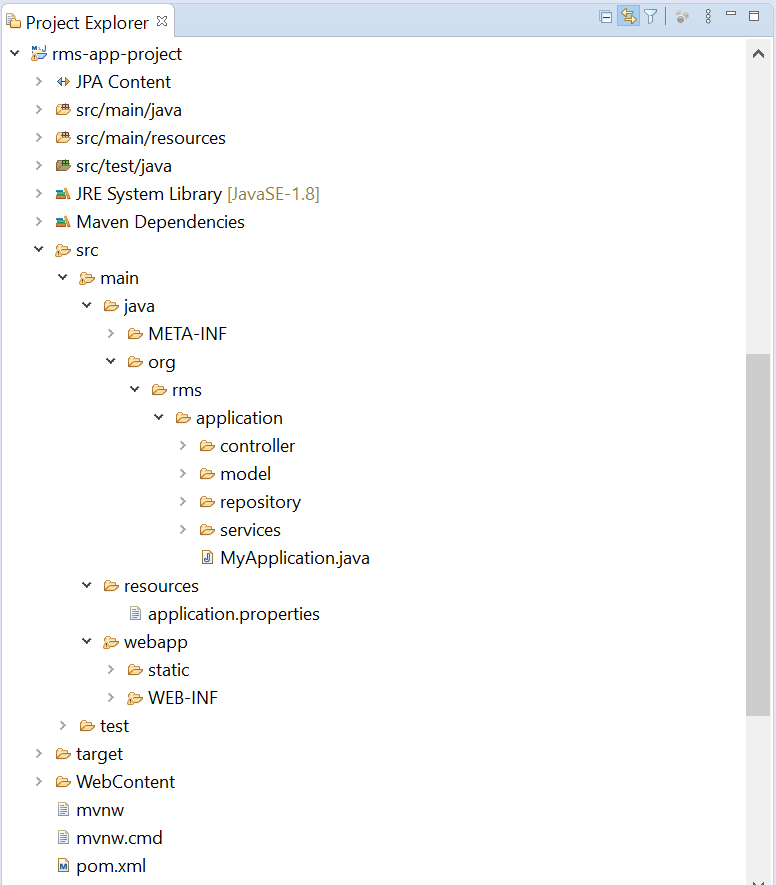
****

Fig 4.1: PROJECT STRUCTURE OF RMS

The above figure shows the project structure of the project. It is necessary to follow the project structure for better understanding and reusing the classes for performing the functionalities of the system.

Spring Boot annotation, @SpringBootApplication is used to mark a configuration class that declares one or more @Bean methods and triggers auto-configuration and component scanning i.e., it declares a class with @Configuration, @EnableAutoConfiguration ad @ComponentScan annotations. This class is very helpful in launching Spring MVC or Spring REST applications using Spring Boot. In the project, the Spring RESTful web service is created and deployed on Tomcat.

1. **Adding Spring Boot Maven Dependencies**: The first step to start with the Spring Boot project is to manage the pom.xml file. The project is a Maven project; it has a pom.xml file, which contains the following default configuration. Maven tool is a project management tool, which is used to manage the project. Maven dependency automatically downloads the dependent library and includes necessary jar files in the project. The Spring MVC dependencies are manually added in the pom.xml file and its versions are managed. The parent is added to the Maven dependency, which is used to declare that the project is a child to that parent project. The resource management system is the child of the spring framework so the parent having the groupID **org.springframework.boot** and artifactId **spring-boot-starter-parent** version **1.5.8 RELEASE** is added to the pom.xml file. Since it is a REST web service, which requires web dependency, so **spring-boot-starter-web** dependency is added. The Spring Boot provides **spring-boot-starter-data-jpa** to connect Spring Boot Application with relational database efficiently. The spring-boot-starter-data-jpa internally uses the spring-boot-jpa dependency. The Spring Boot provides **spring-boot-starter-tomcat** to start the tomcat for run-time environment for the controller class and **mysql-connector-java** to connect and execute SQL queries in the database. The other necessary jar files are added in the pom.xml file. The **spring-boot-maven-plugin** plugin is added, this plugin is used to run the java application as spring boot application.
2. **Creating application.properties file:** In Spring Boot, properties are kept in the application.properties file, which is located in the src/main/resources directory. By default, application.properties is used to run the Spring Boot application. So, the MySQL connection setting is updated in spring.datasource.\* like URL, username, password and other jpa hibernate setting details.

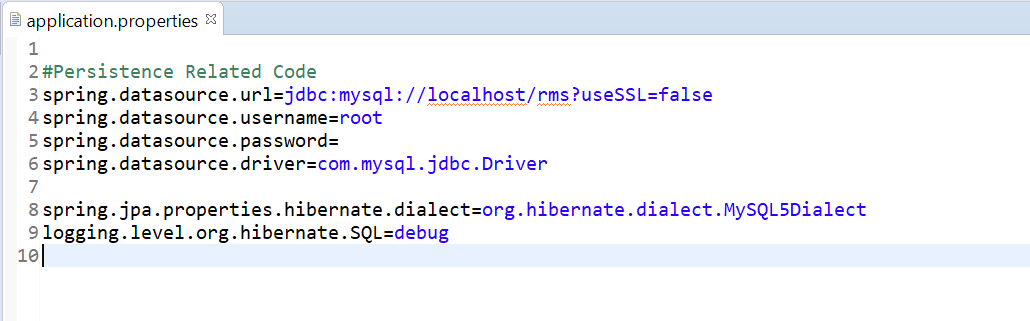


Fig 4.2.1: APPLICATION PROPERTIES FILE OF RMS

For this XAMPP is used for testing the MySQL, apache projects on the local computer. XAMPP is an open source cross-platform web server solution stack package software developed by Apache friends, consisting mainly of Apache HTTP Server, MySQL database, scripts written in PHP and Perl programming language. Apache Tomcat is used for pure java HTTP web server environment and MySQL database is used for managing the database of the RMS.

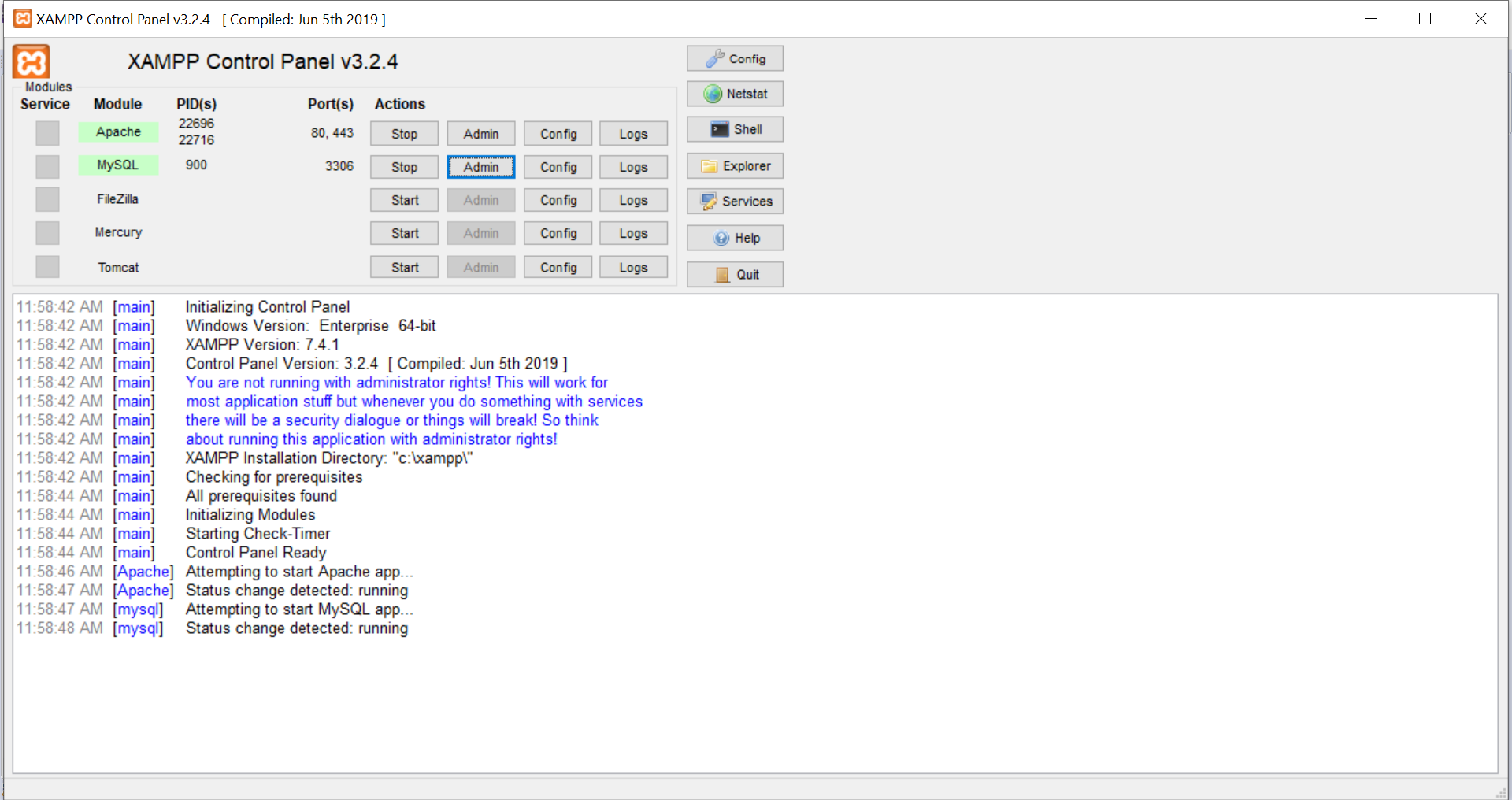


Fig 4.2.2: XAMPP CONTROL PANEL version 3.2.4

1. **Creating Spring Boot Application Class:** A java class with a main method is created that is marked with @SpringBootApplication annotation and it invokes SpringBootApplication.run () method. It is created under the application package, which is the main package of the entire project. It uses the default port localhost:8080, which means to start the application one should visit the route http://localhost:8080. The figure6.3 shows the main application class of the project, which enables the spring boot application method by calling SpringBootApplication.run method. It does the following things:
2. It detects spring-web that configures the default spring application beans. It helps in scanning and configuring @RestController and @Controller and similar annotations.
3. It detects embed tomcat jars so configure embedded tomcat for us.
4. It detects JSON jars so configure JSON support to APIs.
5. **Creating Controller Class:** @Controller annotation indicates that the annotated class is a controller. It is auto detected through class path scanning. It is typically used in combination with annotated handler methods based on the @RequestMapping annotation. It responds to the request from the client. The controller will be invoked with the request and it calls the model class for some information using Service class. In spring, a controller class, which can serve REST API requests, is called rest controller. It should be annotated with @RestController annotation. @RestController is a sibling convenience annotation for creating Restful controllers.  The controller sends the response to the request sent by the user. The controller contains @GetMapping, @RequestMapping, @PostMapping for mapping the action with the function that is to be performed.
6. **Creating the Model Class:** The model class is the POJO (Plain Old Java Object) class, which contains only private variables and corresponding public getters and setters methods. The POJO class is a JPA entity class i.e. an ordinary Java class that is marked (annotated) as having the ability to represent objects in the database. This class uses the annotation as @Entity, which means it is an entity class. @Table that means it is the POJO class of that table from the database. @Column is used to define the name of the columns in the database. @Id is used to declare the primary key of the table. This class contains its own default constructor and parameterized constructor, mainly used for calling the class and storing the values into the database using the class. It may override some methods like toString() from Object or some other interface like Serializable but does not have behaviour of its own. Each table in the database has its own model class.
7. **Creating Crud Repository Interface:** Crud Repository is an interface and extends Spring Data Repository. To use this Crud Repository, an interface of the model class is to be created which extends Crud Repository. The Crud Repository provides a generic CRUD method, which has two parameters, first is the model class of the repository and the second is the datatype of the model class’s primary key. The database queries of Crud Repository like findAll, findBySignum, etc. The spring @Repository annotation is a specialization of the @Component annotation, which indicates that an annotated class is a “Repository”, which can be used as a mechanism for encapsulating storage, retrieval, and search behaviours, which emulates a collection of objects. In the repository interface, @Query is used to write the more flexible query to fetch data. Every model class has its own repository to perform SQL queries to its own table.
8. **Creating Service Class**: Spring @Service annotation is used with classes that provide some business functionalities. Spring context will auto detect these classes when annotation-based configuration and class path scanning is used. In the service class, the business logics to store, retrieve, delete and update the required database fields are performed. This class the repository class to perform the query to the database.
9. **Creating html pages**: The HTML template of login, resource, admin and manager for resource management system are created. Certain design frameworks are added in the template like CSS, which styles the html pages adding the bootstrap technology for better view and JavaScript, for adding advanced functionalities while creating the system.

The above steps are followed while creating the project. The project runs from the web.xml configuration file, it is used to define how to deploy the web module to the Servlet containers like Tomcat, etc. The SpringBootApplication is first called from the web.xml file, where all the main method functions are performed.

**4.1 ALGORITHM BASED ON DIFFERENT MODULES:**

**1. LOGIN MODULE:**

**Step Algorithm:**

0 START

1 run application by calling SpringBootApplication.run () which is mapped with the landing page

2 read signum and password from web page

3 authenticate the credentials with the database do,

4 if credentials are valid

4.1 if signum id assigned as admin then

it enters admin module

4.2 if signum is assigned as manager then,

it enters manager module

4.3 if signum is assigned as resource then,

it enters resource module

5 if credentials are invalid

return error message

6 STOP

**2. RESOURCE MODULE:**

**Step Algorithm:**

0 START

1 to edit the personal details do,

1.1 if modification in any of the fields from the detailed form then,

1.1.1 modification request sent to resource's line manager

1.1.1.1 if request is approved by the manager then,

display the modification on the resource profile

1.1.1.2 if the request is declined by the manager then,

display the previous details on the resource's profile

2 to view the feedback do,

2.1 if feedback is given by the resource's manager then,

display the given feedback

2.2 if no feedback given by the resource's manager then,

display the feedback is empty

3 to search any person, department or projects do,

3.1 if searched value is present in the database then,

show all the details that includes the searched value

3.2 if searched value is not present in the database then,

display no result found

4 logout to exit from the application

5 STOP

**3. MANAGER MODULE:**

**Step Algorithm:**

0 START

1 to view resources under the manager do,

1.1 if the logged in signum matches with the manager’s signum in database then,

display the resource’s details in the page

1.2 if the logged in signum does not match with manager’s signum in database

then,

do not display the resource’s details in the page

2 to check the notification do,

2.1 if any modification request is notified then,

2.1.1 go to notification list and check the request

2.1.1.1 if the request is valid then,

approve the request

modify the requested details in the database

2.1.1.2 if the request is invalid then,

decline the request

3 to give feedback do,

3.1 if feedback is not given then,

add feedback to the resources

3.2 if feedback is given then,

update the feedback according to the performance

4 STOP

**4. ADMIN MODULE:**

**Step Algorithm:**

0 START

1 to add new resources do,

1.1 add the details of the resource according to the criteria of the fields

1.1.1 if the details do not follow the format of the field then,

return error message

1.1.2 if the criteria are fulfilled then,

save the details to the database

2 to assign the role to the resource do,

2.1 if resource’s signum is present then,

choose role option from the list

2.2 if the resource’s signum is not present then,

return error message

3 to view all the resources do,

3.1 fire query to the database to show all the resource details.

4 to view different roles do,

4.1 fire query to the database to show all the roles available.

5 STOP

**5. RESULTS AND DISCUSSION**

The following are the outcomes of the web-application. Each figure shows different functionality and behaviour in the rms application.

The figure 5.1 shows the login page, which is common to all the resources. This page displays the name of the system, RMS i.e., Resource Management System, and the purpose of the system, i.e., A Record-based application for Employees, in the slideshow beside the login form. It requires a signum of the resource to login to the system as the signum is unique for all the resources. The resource can login using only his/her own signum and password provided and instructed by the administrator. It returns an error when it gets some invalid entries. The resource will login to the page as per his/her role assigned by the administrator. The role of each resource is assigned and changed only by the admin. The higher authorities only choose the admin.

A close up of a computer

Description automatically generated

Fig 5.1: LOGIN PAGE

The figure 5.2 shows the details of the resource that has currently logged into the system, here managers and admins are also resources. There are two icons, one icon is to edit the profile details of the resource and the other is to view the resource’s feedback given by the resource’s manager. The edit form has some editable fields like name, signum, contact, email id, location and skills. These fields can be edited by the resource when necessary. When the resource edits his/her details, the modification request is sent to his/her line manager. The details of the resource are updated only after the approval from his/her line manager. If the line manager declines or fails to approve the request sent by the resource, the modified details are not updated in the database. Therefore, the resource must re-send the modification request again to the manager by modifying the field again. The edit form also has some non-editable fields like line manager name, line manager signum, department name, project name and date of joining. These fields are modified only by the admin with proper instructions by manager and higher authority. If any modification is made in these fields, the update is shown in the resource's profile along with his/her personal details. The resource can view his/her feedback by selecting the view feedback icon. This shows the feedback, i.e., ratings and remarks given by his/her line manager about his/her performance. Only the line manager and the resource can view this feedback. It is not visible to other resources. The search field searches any details from the database using query. The query used in the repository of the resource uses his signum, full name, and location, and email id, date of joining, skill description, department or project to search. This field is common in all the pages.

A screenshot of a cell phone

Description automatically generated

Fig 5.2:  RESOURCE PROFILE PAGE

The figure 5.3 below describes the Admin module summing up in a whole showing all the resources, which also includes the managers currently under the admin. The admin module plays an important role in the whole rms system. An admin can add a new resource; can update the resources details as well. Similarly, an admin can assign roles to a resource, since it is mentioned that a resource itself can be an admin or a manager. Only admin has some special features of assigning, modifying and removing. Admin can also view the resource details followed by the managers. As it is clearly visible in the figure below that, an admin module has three icons. These icons have a special meaning here followed by the task they support. First icon describes adding a new resource in the system, managing its department and projects as well. The second icon describes the role assignment where a resource will be assigned roles like an admin, a manager and resource. Similarly, the third icon describes the show roles, where the admin will be able to see what role he has assigned to a resource. An admin has the power to add a resource as well delete a resource that no longer belongs to the system. An admin is also responsible for managing a resources department followed by the projects that are or will be assigned. An admin can update the resource details or his details only if a notification is sent to the manager. An admin also needs to check which resource needs to be assigned to which manager. As mentioned earlier that an admin can also be a resource. The next is the manager’s page.

A screenshot of a computer screen

Description automatically generated

Fig 5.3: ADMIN PAGE

The figure 5.4 below describes a manager's module, which refers to a manager highlighting all the resources present under him. A manager is assigned with various roles related to its resource. A manager can view every detail of its resource, but detail modification features are not supported here. Only the admin and resource have the update details feature. Adding to that, there are few icons defining certain roles that a manager performs, likewise there is an icon beside every resource as shown in the figure. This icon represents the feedback, which is to be provided for every resource based on his/her performance. A feedback could be positive as well as negative. Depending on performance, it can be modified again. In addition, a secured feature is added such that a resource will be able to see its feedback but is forbidden to see the other resources' feedback. Only the manager can see feedback of each resource. Another icon represents the notification bell. The purpose of this icon is such, suppose when a resource or the admin wants to modify or update any details of them or admins resource, then a notification will be sent to the manager. The manager clicking the icon will receive a notification, which will redirect to the update of that resource containing two check boxes describing approval and decline, the manager will then check if the updates are correct or false. If the update is correct then the manager will approve and the modification will be automatically shown and will be updated in the database as well, else if the details are false, then the manager will decline the update and will be showing the resource details will the previous one. Therefore, these are a few functions of a manager. The next is the validation page.

A screenshot of a computer screen

Description automatically generated

Fig 5.4:  MANAGER PAGE

The figure 5.5 shows the JavaScript validation check methods used to validate the fields having specific requirements. If these requirements are not fulfilled, the validation error will be shown. These validations are checked by writing JavaScript on the html page for each field. There are different methods to validate each entry. One method is used to verify the entry of the email id. A valid email address consists of an email prefix and an email domain, where prefix appears to the left of the ‘@’ symbol and domain to the right. For example, ‘[example@mail.com](mailto:example@mail.com)’ where ‘example’ is the prefix and ‘mail.com’ is the domain. Another is used to verify the entry of contact number. The contact number should contain only numbers and should only consist of 10 digits. Else, it returns an error. Another verifies the entry of the signum. Signum length is fixed to ‘7’, should start with a specific alphabet and should contain only string values so if any special characters or numbers are entered, it returns an error. The figure shows the output of the authentication of the fields.

A screenshot of a computer screen

Description automatically generated

                                      Fig 5.5: VALIDATION OUTPUT

**6. SUMMARY AND CONCLUSION**

The main problem was manual handling of records that could lead to mishandling of papers containing the important details of the employees. This problem was solved as the system was designed to handle the details in a proper and well-organized way.

**6.1 SUMMARY OF ACHIEVEMENTS**

The system is fully automatic, which means it saves a lot of human resource work. As with the current system, all human resource work is needed to maintain and keep the record and personal details of every employee working in an organization. The administrator managed the database and the details were filled and modified in a timesaving manner. The information is accessed faster, and the privacy and security of the details are handled in a most efficient way.

**6.2 MAIN DIFFICULTIES ENCOUNTERED AND HOW THEY WERE TACKLED**

While developing this application, many new frameworks and technologies were used. The difficulties encountered during the development were: a) creating the object-oriented designs according to the specified functionalities mentioned. b) The working of HTML with CSS and JavaScript together for creating interactive pages. c) The understanding of the flow of the spring boot application as it contains many interior functions. d) Understanding spring boot functionalities like the CRUD (create, read, update, delete) and writing complex sql queries in the function.

The above-mentioned difficulties were tackled by understanding how the employee management system works that serves a role in creating the system’s workflow. In addition, the understanding concept of spring framework, which makes the development of web applications easy by providing the mvc module known as model, view and controller. A generic idea of bootstrapping the spring boot application was applied, which helped in designing the front-end part of the resource management system. The use of JavaScript for the validations in certain contexts like the email section, contact section etc. was also learned and applied.

**6.3 LIMITATION OF THE PROJECT**

The limitations of the system are: 1) the employees can just perform functions that are fixed to the roles given to them by the administrator. If the roles are not specified clearly then the employee may not perform accordingly. 2) Limitation of spring boot is that it might unnecessarily increase the deployment binary size with unused dependencies. 3) Offline access of information and modifications of data cannot be performed as the system works on internet connection.

**6.4 FUTURE SCOPE OF WORK**

This system is very flexible so that the maintenance and further amendments based on the changing environment and requirements can be made easily. The project will support a multi-user environment, which is more than one user can access simultaneously. It can be further developed to include more operations and analysis, as changes are required in the system to adapt to the external developments. Further enhancements can be made to the system at any later point in time. The versions of the framework may not affect the flow of the application as the system may automatically upgrade or update with the upgrade or update of the versions of the framework.

**6.5 ANY SPECIAL OBSERVATION**

There are few observations highlighted on achieving the rms application, which includes: a) to ensure how the performance objectives and standards were established. b) To observe resource performance. c) To provide them feedback based on their ability of performance. d) To strengthen better communication between the resource and their higher authorities. e) To develop the system with better features. f) Help resources with handling large amounts of data and information. g) Management of employee grievances or other external grievances. h) The need for a rms application was strongly highlighted.

**7. FINAL USER’S MANUAL**

The final users are the employees of the company. Since, the system was designed using a temporary dataset in the original database, the final original dataset of the employees was updated to the database by the chosen admin. The final dataset has all the fields required to fully function the application. The roles were chosen by the higher authority, mainly admins and managers were chosen, and the rest were given the role of resource. The admin assigns roles to all the employees according the list provided by the higher authority. The system was built keeping in mind the large dataset and multiple users and tasks performed simultaneously by them in day-to-day life. The versions of the framework will be modified accordingly, and the performance of the system will be checked by the developers of the application. The figure 7.1 shows the general flow of the application.

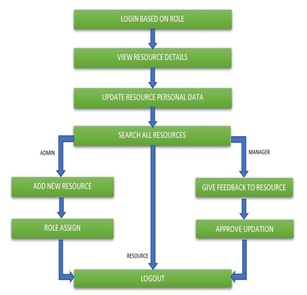
****

Fig 7.1: DIAGRAM TO SHOW THE FLOW OF THE APPLICATION

**8. GANTT CHART**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **6th Jan – 7th Feb 2020** | **8th Feb – 28th Feb 2020** | **29th Feb – 31st Mar 2020** | **1st April – 8th May 2020** | **8th May – 26th May**  **2020** |
| **Survey** |  |  |  |  |  |
|  |  |  |  |  |
| **Problem Identification** |  |  |  |  |  |
|  |  |  |  |  |
| **Design** |  |  |  |  |  |
|  |  |  |  |  |
| **Implementation** |  |  |  |  |  |
|  |  |  |  |  |
| **Documentation** |  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Proposed Activity** |  | **Ongoing Activity** |  | **Achieved Activity** |

Fig 8.1: Gantt Chart

**9. REFERENCES**

***[1]. Microsoft Windows, macOS, Linux, v3.0.2 / August 17, 2018 Tool to create UML diagrams.*** [***http://staruml.io/***](http://staruml.io/)**, *Last Accessed Date: Mon, 16 Mar 2020 05:50:10 GMT.***

***[2]. Apache POI includes a pre-defined Eclipse project file which can be used to quickly get set up in the Eclipse IDE. POI version 4.0.0 with jdk 1.8 or later.*** [***https://poi.apache.org/***](https://poi.apache.org/) ***Date: Mon, 16 Mar 2020 05:52:33 GMT, Last-Accessed: Sun, 08 Mar 2020 20:24:23 GMT.***

***[3]. Spring Boot is a framework to create better standalone web applications, with tutorials and references on how to create a spring boot web application using its functionalities like CRUD, JPA etc.*** [***https://spring.io/projects/spring-boot***](https://spring.io/projects/spring-boot)***Date: Friday 26 Feb 2020 20:24:23 GMT, Last-Accessed: Wed 18 Mar 2020 01:20:48 GMT.***

***[4]. W3Schools is a web developer’s site, with tutorials and references on web development languages such as HTML, CSS, and JavaScript.*** [***https://www.w3schools.com/html/default.asp***](https://www.w3schools.com/html/default.asp)**,** [***https://www.w3schools.com/css/default.asp***](https://www.w3schools.com/css/default.asp)**,** [***https://www.w3schools.com/js/default.asp***](https://www.w3schools.com/js/default.asp) ***Date: Mon, 16 Mar 2020 05:54:33 GMT, Last-Accessed: Mon, 16 Mar 2020 02:40:28 GMT.***

***[5]. Xampp version 3.2.4. MySQL 8.0 Generally Available (GA) Release Server for the local host.*** [***http://localhost/phpmyadmin/***](http://localhost/phpmyadmin/)**, *Last-Accessed: Thurs, 27 Feb 2020 12:35:48 GMT.***