

FAULTIFY - AN ISSUE TRACKING APPLICATION USING SPRING MVC

Project report submitted for partial fulfilment of the requirement for the degree
of Bachelor of Technology
in
Computer Science and Engineering & Information Technology

By
Raven Mehta (191303)
Under the supervision of

Prof. (Dr.) Shruti Jain

&

Dr. Amol Vasudeva

To



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh**

TABLE OF CONTENTS

CONTENT	PAGE NO.
CANDIDATE'S DECLARATION	i
CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
LIST OF ABBREVIATIONS	iv
LIST OF FIGURES	v-vi
ABSTRACT	vii
CHAPTER 1: INTRODUCTION	1
1.1 Problem Statement	3
1.2 Objectives	4
1.3 Methodology	5
1.4 Organisation	19
CHAPTER 2: LITERATURE SURVEY	24
CHAPTER 3: SYSTEM DEVELOPMENT	26
3.1 Analytical system development	26
3.2 Computational system development	27
CHAPTER 4: EXPERIMENTS AND RESULTS ANALYSIS	43
CHAPTER 6: CONCLUSION AND FUTURE WORK	49
5.1 Future Scope	50
5.2 Applications	51
REFERENCES	52
PLAGIARISM REPORT	53

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Faultify - An Issue Tracking Application Using Spring MVC**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Shruti Jain**, Professor, Department of ECE and **Dr. Amol Vasudeva**, Assistant Professor (Senior Grade), Department of CSE.

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream **Artificial Intelligence**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Raven Mehta

191303

This is to certify that the above statement made by the candidate is to the best of their knowledge.

Prof. (Dr.) Shruti Jain

Professor and Associate Dean (Innovation)

Department of ECE

Dated:

Dr. Amol Vasudeva

Assistant Professor (Senior Grade)

Department of CSE

Dated:



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

(Established by H.P. State Legislative vide Act No. 14 of 2002)
P.O. Wagnaghat, Teh. Kandaghat, Distt. Solan - 173234 (H.P.) INDIA

Website: www.juit.ac.in

Phone No. (91) 01792-257999

Fax: +91-01792-245362

CERTIFICATE

This is to certify that the work reported in the B.Tech project report entitled “**Faultify - An Issue Tracking Application Using Spring MVC**” which is being submitted by **Raven Mehta** in fulfilment for the award of Bachelor of Technology in Computer Science Engineering by the Jaypee University of Information Technology, is the record of candidate’s own work carried out by him under my supervision. This work is original and has not been submitted partially or fully anywhere else for any other degree or diploma.

Dr. Shruti Jain

Associate Dean (Innovation) and Professor
Department of Electronics & Communication Engineering
Jaypee University of Information Technology, Wagnaghat

Dr. Amol Vasudeva

Assistant Professor (Senior Grade)
Department of Electronics & Communication Engineering
Jaypee University of Information Technology, Wagnaghat

विद्या तत्व ज्योतिसमः

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for me to complete the project work successfully. I am grateful and wish my profound indebtedness to Supervisor Dr. Shruti Jain , Professor and Associate Dean, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Shruti Jain and Dr. Amol Vasudeva , Department of CSE, for their kind help to finish my project.

I wish to express my sincere gratitude to my manager Mr. Aashu Jaidka and my mentor Mr. Sarbjit Singh, for providing me an opportunity to do my internship and project work in Paxcom by Paymentus. I would like to thank all my team members and seniors in Paxcom who helped me in every aspect of this project. I would also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unique scenario, I'd want to appreciate the different staff members, both teaching and non-teaching, who have developed their helpful assistance and facilitated my project. Finally, I must express my gratitude for my parents' unwavering support and patience.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Raven Mehta

(191303)

List of Abbreviations

Abbreviation	Name
MVC	Model View Controller
ORM	Object Relational Mapping
DAO	Data Access Object
JPA	Java Persistence API
JSP	Jakarta Server Pages
JSTL	Jakarta Standard Tag Library
XML	Extensible Markup Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
HDD	Hard disk drive
SSD	Solid State Drive

List of Figures

Fig. No.	Figure	Page No.
1	Model View Controller	6
2	MVC Detailed view	7
3	Spring Dispatcher	8
4	JPA Working	18
5	Company Description	21
6	Database Tables	30
7	Issues Table Properties	31
8	User Table Properties	31
9	user_roles Table Properties	32
10	Dashboard	33
11	Issue Report Form	34
12	Login Page	35
13	Register Employee Page	35
14	showEmployee.jsp Page	36

15	viewProfile.jsp Page	37
16	Welcome.jsp Page	38
17	Assigning Issue To An Employee	43
18	Admin Dashboard	44
19	Login form	44
20	Employee Dashboard	45
21	Status Update Page	45
22	Updating status of an Issue	46
23	Updated Employee Dashboard	46
24	Employee Profile	47
25	Progress of an Employee	47
26	Show All Employees	48

ABSTRACT

While some major software development businesses utilize a bug-tracking tool to keep track of flaws, others rely on email and shared lists, which can be error-prone and lead to the neglect of less serious issues. To track issues in a product, solution, or application, a bug-tracking system—also called a defect tracking system—can be the best option. It allows individuals or groups of developers to effectively monitor outstanding bugs and increase productivity and accountability by providing a documented workflow and positive feedback for good performance. A software programme used to handle and keep track of issues or problems inside an organisation is called an issue-tracking application. The goal of this project is to create an application for recording issues that can be used to keep track of and handle problems across several departments. Users will be able to report issues, assign them to certain teams or people, and follow their progress until they are fixed using the programme. A user-friendly interface that can be accessed and utilised by various organisation stakeholders will be designed and developed as part of the project. Overall, it is anticipated that the programme would boost team communication, accelerate organisational efficiency, and expedite the problem-solving process. Users will be able to prioritise issues, add comments and attachments, and produce reports for tracking and analysing trends using the issue tracking application's other capabilities. A reliable and safe architecture that guarantees data privacy and security will be used to build the application. In order to make sure that the application satisfies each department's unique needs and expectations, the project will include working with stakeholders from several departments. In the end, the issue tracking application will be a useful tool for organisations to monitor internal concerns, quickly address them, and enhance performance. The issue tracking application will go through extensive testing and quality assurance procedures before being released to make sure it is efficient. The program will also be made to be versatile, scalable, and adjustable so that businesses may adapt it to their own requirements. A committed group of developers and project managers who will collaborate closely with stakeholders during the development process will be needed for the project. Overall, by expediting the problem-resolution process, the issue tracking application project has the potential to greatly increase organizational efficiency and productivity.

CHAPTER 1

INTRODUCTION

To track and handle internal issues across many departments, organisations can use an issue tracking application built using Spring MVC. A Java-based framework called Spring MVC offers programmers a variety of tools and capabilities for creating dependable and scalable online applications. The Model-View-Controller (MVC) architecture, on which it is based, divides the programme into three different layers: the model, the view, and the controller. This division gives the application more flexibility and maintainability.

User management is one of the main components of a Spring MVC-based issue tracking application. Administrators may use this feature to establish and manage user accounts, assign roles and permissions, and limit access to particular application capabilities. This is crucial to ensuring that the application is safe and that only users with the proper authorization may access critical data.

Issue reporting is a crucial component of a Spring MVC-based issue tracking application. Users may report issues using a simple and user-friendly interface. They can choose the issue's importance, provide further details, and attach relevant files and photographs. This feature is crucial for speeding up the process of problem-solving and ensuring that issues are handled as quickly as feasible.

When an issue is reported, it may be assigned to a specific team or individual for correction. The programme will notify the team or individual allocated to the problem and its importance right away. This feature is essential for ensuring that issues are handled properly, and that everyone is held responsible for each issue. Users may track the problem's progress from the time it was first reported until it is repaired. They are able to comment, read updates, and receive alerts about the issue's development. This task is crucial for enhancing team communication and ensuring that everyone is aware of how each issue is progressing.

The reporting and analytics features of a Spring MVC-based issue tracking application can provide useful information on team productivity and issue trends. These studies can help firms find areas for improving internal procedures. For instance, reports on issue patterns can help firms see reoccurring issues and take action to prevent them from happening again.

When creating an issue tracking application using Spring MVC, significant consideration must be given to the programme's architecture and design. One of the most significant design considerations is scalability. The programme should be designed to handle massive volumes of data and users in order to meet the organization's rising needs. Security is a key design element for a Spring MVC-based issue tracking application. When creating the application, security should be taken into account. Features like user authentication, authorization, and data encryption should be added. This is necessary to protect sensitive information and prevent the application from becoming vulnerable to security issues.

Flexibility is a key design element for a Spring MVC-based issue tracking application. The programme has to be flexible and adaptive so that organizations may modify it to suit their own requirements. To ensure that the application satisfies the unique requirements of each company, this is essential.

The user experience is a crucial component of an issue tracking application created using Spring MVC. The programme should be designed with the user in mind, with a focus on providing an easy-to-understand user interface. This is essential to making sure the programme is straightforward to use and that users can easily report and track concerns. The bottom line is that a Spring MVC-based issue tracking application may provide enterprises a variety of benefits, including streamlined issue resolution, improved communication, enhanced productivity, and data-driven decision-making. Scalability, security, flexibility, and user experience must all be carefully addressed in the architecture and design of such an application. Overall, a Spring MVC-based issue monitoring application may be a useful tool for businesses to monitor internal issues, address them quickly, and enhance performance.

1.1 PROBLEM STATEMENT

- The statement of the problem for a Spring MVC-based issue tracking application is that it is frequently difficult for organisations to effectively manage and track internal issues. Delays in problem resolution, disruptions in communication, and lower productivity may arise from this. Due to the complexity and error-proneness of the current issue recording techniques, such as spreadsheets and email, there is frequently a lack of accountability and transparency in the issue resolution procedures. Furthermore, these approaches lack scalability and the analytics and reporting elements necessary for streamlining internal operations and finding potential improvement areas.

- A centralised platform for reporting, tracking, and resolving internal issues can be offered through an issue tracking application built with Spring MVC to handle these issues. Such a tool might speed up the problem-solving procedure, improve teamwork, and offer insightful data on problem trends and team productivity. With a focus on offering a straightforward and understandable user interface, the programme must be user-friendly, versatile, scalable, secure, and user-friendly. Ultimately, a Spring MVC-based issue tracking application can assist organisations in managing internal issues, enhancing overall performance, and promoting data-driven decision-making.

Organisations also struggle with a lack of transparency in problem-solving procedures. It might be challenging to monitor a problem's development and determine who is in charge of addressing it using conventional issue tracking techniques. Users who may not be informed of the progress of their reported concerns may become frustrated as a result, which can cause delays in response and a lack of responsibility. Additionally, traditional problem-tracking techniques sometimes lack the elements required for reporting and analytics, which can make it challenging for organisations to see patterns, improve internal procedures, and make data-driven choices. Because it may be difficult to detect and handle internal issues, this is especially troublesome for organisations with a lot of them.

1.2 OBJECTIVE

A unified platform for handling and monitoring internal issues across organisations is the main goal of an issue tracking application built using Spring MVC. This contains tools for submitting issues, distributing tasks, and monitoring development during the issue resolution procedure. Organizations can improve communication, increase visibility, and guarantee that concerns are addressed quickly and effectively by offering a consolidated platform for handling and recording issues.

The provision of analytics and reporting functionalities that enable organisations to acquire an understanding of their internal problem-resolution procedures is another important goal of an issue tracking application built using Spring MVC. This consists of tools for creating reports on problem trends, team effectiveness, and problem resolution timeframes. Organisations may pinpoint areas for improvement, streamline internal procedures, and make data-driven decisions that enhance overall performance by offering these analytics and reporting options.

The user experience and usability of a Spring MVC-based issue tracking application should also come first. This consists of capabilities for tracking the status of reported issues, as well as a user-friendly interface that enables users to submit concerns quickly and simply. The program should furthermore offer real-time alerts to update users on the status of their reported concerns. Organizations may guarantee that users are engaged and motivated to report concerns by placing a high priority on user experience and usability, which can reduce the time it takes to resolve issues in general.

Last but not least, a Spring MVC-based issue-tracking solution should be flexible and adaptive to the unique demands and specifications of each organisation. This includes the ability to interact with other tools and systems that the organisation uses, as well as customization options like custom fields and processes. Organisations can guarantee that their issue-tracking system can grow and change along with their internal processes by offering a flexible and adaptable platform, eliminating the need for expensive and disruptive updates.

Security and compliance should come first in a Spring MVC-based issue-tracking application. This consists of tools for data encryption, access control, and auditing to guarantee the security of sensitive data and the adherence of the application to pertinent legal requirements. Organisations may prevent security risks or legal infractions from compromising their internal problem resolution procedures by giving security and compliance first priority.

Collaboration and teamwork across teams are other goals of an issue-tracking application built using Spring MVC. This contains tools for assigning jobs, distributing duties, and efficiently interacting within the application. Organisations may guarantee that their teams collaborate effectively to handle internal problems without the need for extra communication channels or technologies by offering these qualities.

1.3 METHODOLOGY

To provide a centralised platform for monitoring and tracking internal issues inside organisations, the Spring Issue Tracker MVC project was created. The Spring MVC framework, a potent and adaptable Java-based platform for developing web applications, is used to construct the project. The agile development technique, which places an emphasis on flexibility and cooperation between development teams and stakeholders, is the strategy employed in this project. This method entails testing the application and making numerous iterations to make sure it satisfies the needs of the stakeholders.

A creative approach that streamlines the problem-tracking procedure inside organisations is the Spring Problem Tracker MVC project. The purpose of this project is to provide a centralised platform that all workers of a business can use to manage and monitor internal concerns. By using this platform, organisations may track internal issues more methodically and efficiently. The project is built using the Spring MVC framework, a powerful and flexible Java-based platform for constructing online applications. The Spring MVC framework makes use of the Model-View-Controller (MVC) design, which separates the presentation layer, the

business logic layer, and the data access layer. This makes future updates and expansion of the programme straightforward.

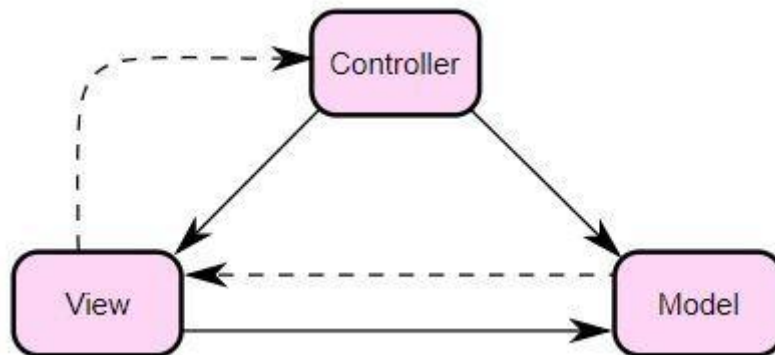


Fig: 1 Model View Controller

To ensure that the project meets the requirements of the stakeholders, a technique called agile development was used in this project. The flexibility and collaboration between development teams and stakeholders are highly valued in this method. The application is often tested and iterated to ensure that it fulfils the demands of the stakeholders. The Agile development style used in the Spring Issue Tracker MVC project includes a number of sprints, which are quick development cycles that typically last two to four weeks. The three phases of a sprint are planning, development, and testing. During the planning stage, the development team meets with the stakeholders to determine the needs of the sprint.

The Agile development technique contributes to a variety of benefits for the Spring Issue Tracker project. Due to the possibility of evolving stakeholder demands, it promotes flexibility and adaptation. In order to guarantee that the project meets the needs of all parties, it lays a significant focus on communication between the development team and stakeholders. Thirdly, it enables frequent testing of the application, resulting in high-quality results. In conclusion, the Spring Problem Tracker MVC project provides organisations looking to automate their internal problem-tracking processes with a robust and cutting-edge solution. By employing the Spring MVC framework and an Agile development approach, this project offers an adaptable, scalable, and reliable platform for managing internal concerns.

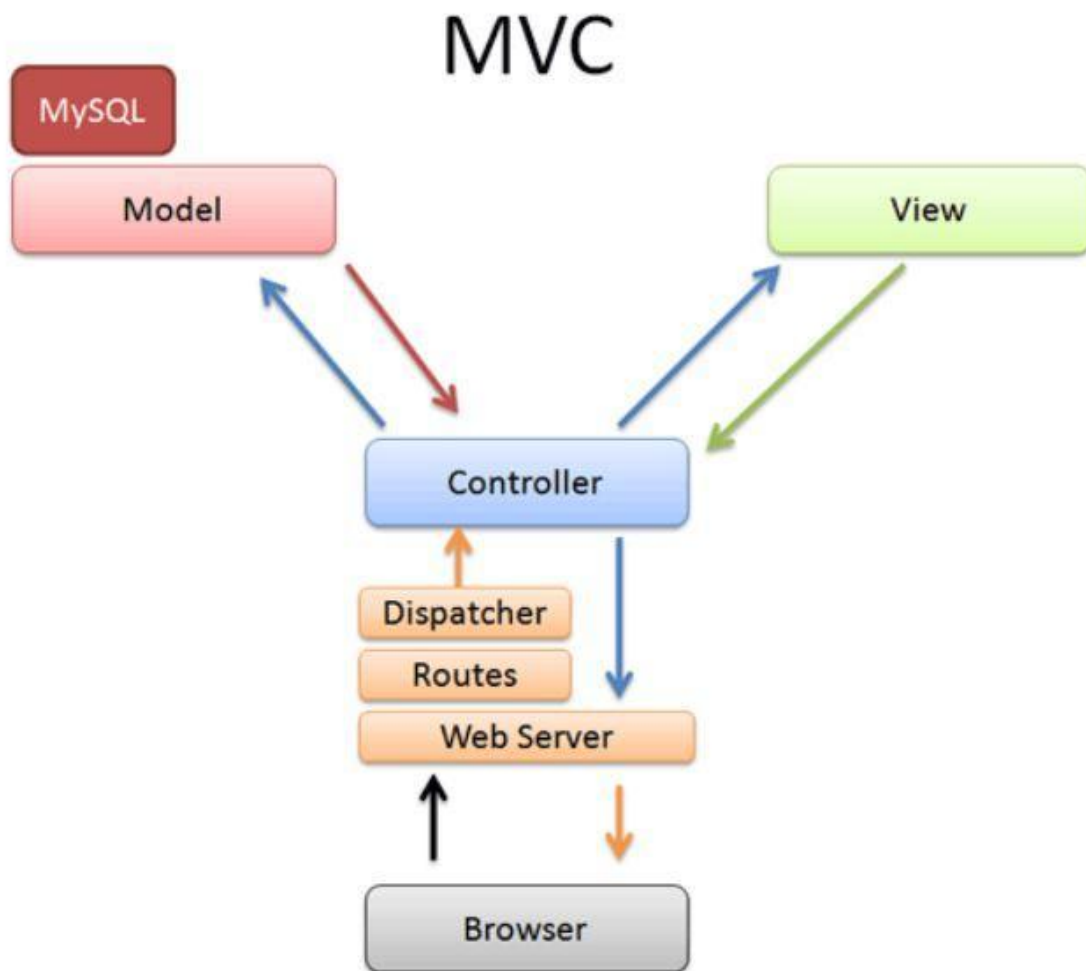


Fig: 2 MVC Detailed view

The Spring Issue Tracker MVC project makes use of a variety of different tools and technologies in addition to the Spring MVC framework and Agile development methodology to assure the project's success. JSP pages for the front end are one such solution that offers a reliable and scalable interface for users to engage with the application. JSP pages provide extensive customization options and are simple to modify to accommodate shifting organizational demands.

The project also makes use of JSTL, or JSP Standard Tag Library, a technology. A collection of industry-standard tags for XML, databases, and other web technologies are provided by JSTL. The development team may decrease the amount of code needed to construct the application and expedite the development process by

utilising JSTL. The business logic and data access layers are managed on the backend using the Spring MVC framework. An easy-to-use and adaptable architecture is offered by Spring MVC for creating web applications. Dependency injection is one of its many potent characteristics that makes the development process simpler and the programme easier to manage over time.

The project also supports Spring Security authentication to guarantee the application's correct security. Restricting access to certain resources according to user roles and permissions, this feature adds an extra layer of protection to the application. The programme and its data's security and integrity are supported by Spring Security, a potent technology. The project follows correct camel case naming standards, which contributes to the code's readability and uniformity. The Apache Tomcat server, a well-liked and dependable web server for Java-based applications, is used to host the project. The Spring Issue Tracker MVC project is a strong and dependable platform for tracking internal issues within organisations since it makes use of these techniques and technologies.

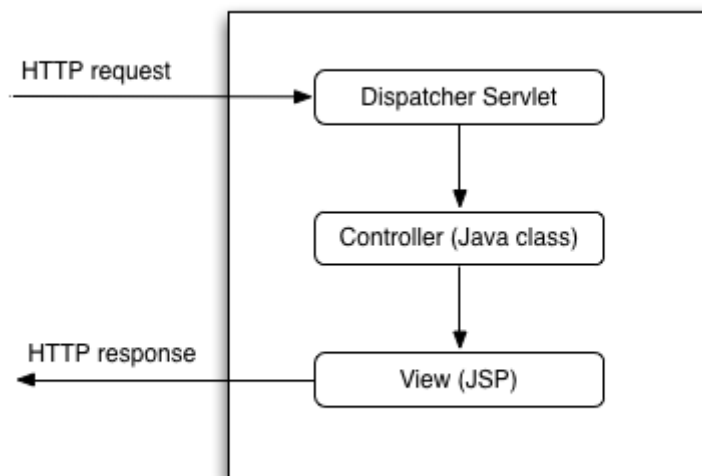


Fig: 3 Spring Dispatcher

The DispatcherServlet is the main servlet in Spring MVC, and it is responsible for handling all incoming requests and sending them to the proper controllers for processing. As the front controller of the Spring MVC application, the DispatcherServlet handles all incoming requests and routes them to the appropriate controller for processing. The Spring MVC framework's DispatcherServlet is in

charge of a variety of crucial duties. All incoming requests are intercepted, and it goes through a number of processing processes, such as resolving the handler, calling any interceptors set up for the request, and sending the request to the proper controller.

The flexibility and adaptability of the DispatcherServlet in Spring MVC are two of its main advantages. The DispatcherServlet may be tailored to suit the requirements of almost any Spring MVC application since it is very flexible. The DispatcherServlet is a great option for complicated web applications since it can be expanded to include specific processing logic.

In conclusion, a crucial part of the Spring MVC framework is the dispatcher servlet, which is in charge of sending incoming requests to the correct controller for processing. It is the perfect solution for complex web applications thanks to its flexibility and extensibility, and any high-performance Spring MVC application would be incomplete without it due to its ability to handle numerous requests at once.

Tools and Technologies Used:

The following tools and technologies were used in the development of the Spring Issue Tracker MVC project:

1. **Java:** The Spring Issue Tracker MVC project is mostly being developed using Java as the programming language. Java is a popular programming language that is well-known for being platform-independent, having object-oriented characteristics, and supporting a broad range of hardware and operating systems. The Java-based Spring MVC framework is a great option for creating web applications that need a robust and adaptable design. Platform independence is one of Java's main benefits. Any computer with the Java Virtual Machine (JVM) installed may run the bytecode that makes up a Java programming. This suggests that any platform that supports the JVM, such as Windows, Linux, macOS, and others, may be used to run Java applications. Due to its platform neutrality, Java is a popular choice for

creating web programmes that must function across several platforms. Another advantage is the vast array of libraries and frameworks available for Java. Java offers a vast ecosystem of tools and frameworks for developing web applications. Spring is one of the most well-liked Java frameworks and provides a powerful set of capabilities for building web applications. A few of the features that Spring provides to make it easier to develop and maintain complex web applications are aspect-oriented programming, dependency injection, and transaction management.

Java is recognised for both its speed and scalability. Bytecode is created from Java programming, which the JVM then runs. The JVM's capacity to optimise the bytecode during runtime may result in performance gains over interpreted languages. Java is used in the context of Spring MVC to build the application's backend. The backend of the programme is in charge of connecting to databases and other external services, performing business logic, and responding to front-end inquiries. Java is a suitable option for these tasks due to its speed, scalability, and durability.

Spring MVC provides a comprehensive set of functions for building Java web applications. Model, View, and Controller are the three components that make up an application according to the model, view, and controller (MVC) design pattern, which is utilised by Spring MVC. While the Model provides the application's business logic and the View its user interface, the Controller processes user requests and controls the program's flow.

Dependant management between application components is also made easier with the use of techniques like dependency injection provided by Spring MVC. Dependency injection allows components to be loosely coupled, which makes the programme simpler to maintain and test.

2. **Spring MVC:** The web application was created using the Spring MVC framework. A lightweight and adaptable framework called Spring MVC offers a robust set of capabilities for creating web applications. The support

for different view technologies, such as JSP, JSTL, and HTML, its strong support for data binding and validation, and its interoperability with other Spring frameworks like Spring Security and Spring Data are just a few of Spring MVC's standout attributes. The adaptability of Spring MVC is another crucial aspect. Because of the framework's great degree of configuration, developers may alter it to suit the requirements of a given application. Due to its adaptability, Spring MVC is extremely simple to connect with other programmes and frameworks like Hibernate and Struts. A strong collection of tools and functionalities are also offered by Spring MVC for the development of web applications. For instance, it has support for template engines that make it simple to create dynamic web pages, such as JSP and Thymeleaf. It also offers capabilities for handling errors and exceptions, validating user input, and processing forms. Spring MVC is a powerful and flexible web framework for building Java-based web applications. It is a part of the Spring framework and is popular among developers all around the world. Due to a number of significant benefits, including its lightweight architecture, flexibility, and extensive set of tools and capabilities, Spring MVC is a well-liked option for creating online applications. The lightweight nature of Spring MVC is one of its main benefits. In contrast to certain other web frameworks, which can be unnecessarily complicated and cumbersome, Spring MVC is intended to be compact and simple. Because of this, it is a popular option for developing web applications of various sorts, from little personal endeavours to substantial business applications. Along with supporting RESTful web services, Spring MVC offers strong support for processing HTTP requests and replies. As a result, creating online applications that can be used by a range of clients, including web browsers, mobile apps, and other web services, is made simple. Finally, Spring MVC is made to be very adaptable and modular. This implies that developers may quickly replace existing components with those from other tools or frameworks or add new features and functionality to the framework.

3. **JSP Pages:** Dynamic web pages are created using the JavaServer Pages (JSP) technology. It enables programmers to create HTML pages with integrated Java code. In order to execute the Java code and send the finished HTML to

the client's web browser, JSP pages are run-time-compiled into Java Servlets. JSP pages offer a versatile and effective method for creating dynamic information for the web. They make it simple for developers to combine Java code with HTML, CSS, and JavaScript to produce dynamic, interactive websites. The Java code may be used to do computations, produce dynamic content based on user input, and obtain data from databases. Custom tags, which developers may define and reuse across numerous pages, are likewise supported by JSP pages. To simplify code maintenance and reuse, sophisticated functionality, such as form validation, can be encapsulated using custom tags. Most web application servers accept JSP pages, which are often used in business online applications.

They offer a potent tool for creating scalable and stable web applications, and developers frequently choose them because of their connection with Java. JSP pages' simplicity of usage is one of its key benefits. JSP pages offer a potent set of tools for creating dynamic web pages as well. Java objects called JavaBeans, which may be used to store and modify data, are supported. JavaBeans may be used to encapsulate intricate data structures and make it simpler to transfer data across various application components. The JavaServer Pages Standard Tag Library (JSTL), which is supported by JSP pages, is another important feature. For typical operations like formatting, conditional statements, and looping, JSTL offers a collection of reusable tags. JSTL tags minimise the amount of Java code that has to be written and make it simpler to develop code that is clean and maintainable.

Expression Language (EL), which offers a streamlined syntax for accessing data contained in JavaBeans, is likewise supported by JSP pages. EL lowers the amount of boilerplate code that has to be produced and makes it simpler to develop code that is both clear and simple to understand.

4. **JSTL:** A set of unique tags called the JavaServer Pages Standard Tag Library (JSTL) may be used in JavaServer Pages (JSP) to make routine activities easier. These tags can be used for a variety of tasks, including iteration and

conditional processing. The library is made up of a variety of tag libraries that may be used to various situations. By offering a common set of tags that can be used across many web applications, JSTL is designed to make it simpler for developers to create JSP-based apps.

JSTL simplifies JSP development by lowering the amount of Java code that has to be written, which is one of its main advantages. Each of the several tag libraries that make up JSTL offers a set of tags for a particular use. Iteration, conditional processing, and variable manipulation are just a few of the fundamental operations that the core library offers a set of tags for. While the SQL library offers tags for dealing with relational databases, the XML library offers tags for working with XML data. To carry out simple tasks like string manipulation and date formatting, the functions library offers a selection of functions.

Any JSP container that supports version 1.2 or later of the JSP standard can be utilised with JSTL. This implies that a variety of web application servers, such as Apache Tomcat, IBM WebSphere, and Oracle WebLogic, may be utilised with JSTL. It may be simpler to create JSP-based apps because JSTL is also compatible with other web application frameworks like Spring MVC. The formatting library is another popular tag library in JSTL. A selection of tags for formatting data, including dates and numbers, are provided by this library. The library has tags like `fmt:formatDate` and `fmt:formatNumber` that may be used to format dates and numbers, respectively. The formatting library offers support for formatting data in accordance with various locales, which makes it particularly helpful when working with internationalisation and localization.

In conclusion, JSTL is a strong and adaptable library that may be utilised to make developing JSP easier. The library offers a selection of tags for carrying out typical operations including iteration and data formatting.

5. **Spring Security:** Web applications, including those created using the Spring MVC framework, may be secured with the help of the strong and adaptable Spring Security framework. Any online application must have authentication

and authorisation, and Spring Security offers a full range of capabilities for putting these security precautions in place.

Authorization is the process of figuring out what activities a user is permitted to take within the programme, whereas authentication is the process of confirming a user's identity. A variety of authentication methods, including as username/password authentication, token-based authentication, and Single Sign-On (SSO) authentication, are supported by Spring Security. The integration of external authentication providers like LDAP or OAuth is also supported by Spring Security.

The use of security filters to protect particular portions of an application is one of the fundamental components of Spring Security. These filters stop incoming requests before they can be processed and perform security checks. As an example, the `UsernamePasswordAuthenticationFilter` handles username/password authentication, the `OAuth2AuthenticationProcessingFilter` handles OAuth2 authentication, and the `BasicAuthenticationFilter` handles basic authentication. Spring Security offers a variety of security filters for various use cases.

Along with supporting a number of security-related activities including password encryption and hashing, user account lockout, and session management, Spring Security also offers support for the fundamental security filters. In order to securely store user passwords in a database, the framework, for instance, supports a variety of password encoder implementations, such as `BCryptPasswordEncoder` and `StandardPasswordEncoder`. Additionally, Spring Security offers a variety of integrated login and logout features that may be tailored to fit the appearance and feel of the application. This comprises assistance with form-based login and logout in addition to Single Sign-On (SSO) assistance using tools like OAuth2.

6. **Apache Tomcat Server:** The Apache Software Foundation created the free and open-source Apache Tomcat web server and servlet container. It offers a strong and scalable framework for hosting web applications and is used for

launching Java-based web applications. Because it is simple to use, dependable, and offers a lightweight and scalable architecture, Tomcat is frequently utilised in business contexts.

Tomcat can handle HTTP requests and answers since it is built on the Java Servlet API. It offers a web-based administrative interface that enables users to install web applications, configure the server, and manage users and roles. Additionally, SSL/TLS encryption, virtual hosting, and load balancing are supported by Tomcat. Scalability is one of Tomcat's primary advantages. It may be simply set up to run on several servers in a clustered environment and is intended to manage numerous concurrent connections. As a result, Tomcat can offer a fault-tolerant and highly available framework for hosting web applications. Various Java-based web frameworks, such as Spring, Struts, and Hibernate, are also supported by Tomcat. This enables web application developers to create them using their favourite framework and then deploy them on Tomcat.

Tomcat's security features are an additional key component. Basic authentication, form-based authentication, and certificate-based authentication are just a few of the several authentication and authorization options it supports. Role-based access control is another feature supported by Tomcat, allowing administrators to manage access to web resources according to user roles. To provide more features, Tomcat is readily linked with other web servers like the Apache HTTP server. This makes it possible for developers to create intricate web applications that need extra capabilities like load balancing, caching, and content delivery.

All things considered, Apache Tomcat is a strong and adaptable web server that offers a compact and scalable framework for hosting Java-based web applications. Both developers and businesses choose it because to its simplicity, dependability, and scalability.

- 7. Hibernate ORM:** Java objects may communicate with databases using the Object-Relational Mapping (ORM) framework called Hibernate. By mapping Java objects to database tables and vice versa, it makes database access simpler.

Enterprise-level applications frequently perform complicated data operations using the Hibernate ORM. Hibernate supports a large number of databases, including Oracle, MySQL, PostgreSQL, and Microsoft SQL Server, which is one of its main features. Developers may easily swap databases without having to modify their code thanks to this. Additionally, Hibernate offers sophisticated capabilities including automated object-relational mapping, lazy loading, and caching.

Both XML-based mapping and annotation-based mapping are supported by Hibernate. For the XML-based mapping, a mapping file that specifies the connection between Java objects and database tables must be created. On the other hand, the annotation-based mapping uses Java annotations to specify the mapping connection. Hibernate also has the benefit of offering a wide range of APIs for data querying. With the help of the robust query language known as Hibernate Query Language (HQL), programmers may create Java objects that resemble SQL queries. Working with Java objects rather than raw SQL queries is simple because of the runtime translation of HQL queries to SQL queries.

Additionally, Hibernate enables transaction management, a vital component of data integrity. Database activities benefit from transactions' atomicity, consistency, isolation, and durability (ACID) features. Hibernate enables programmatic or declarative transaction handling for developers. In conclusion, Hibernate is a potent ORM framework that makes managing complicated data operations simple while also streamlining database access. It is a popular option for enterprise-level applications because to its support for numerous databases, comprehensive query API, and transaction management.

A well-liked object-relational mapping (ORM) framework for Java-based applications is Hibernate ORM. By offering a high-level object-oriented API for converting database tables into Java objects and vice versa, it streamlines the development process. Hibernate ORM is a well-liked option for developers creating enterprise-level applications because of its simplicity, use, and adaptability. Hibernate ORM's ability to decrease the amount of boilerplate code needed for database operations allows developers to concentrate on the business

logic of their applications, which is one of its main advantages. Lazy loading, cascading, and querying are just a few of the tools that Hibernate ORM offers to manage intricate connections between entities.

- 8. JPA(Java Persistence API):** The Java Persistence API (JPA) is an Object-Relational Mapping (ORM) standard. In order to map Java objects to relational database tables, it offers a set of interfaces and annotations. Numerous Java Persistence providers, including Hibernate, EclipseLink, and OpenJPA, implement the JPA standard API. JPA offers developers a straightforward and user-friendly API to communicate with databases. Without creating SQL queries, it enables developers to execute CRUD (Create, Read, Update, Delete) actions on Java objects. Database tables are mapped to Java objects using JPA, which also offers transparent data persistence.

JPA's capability to handle many database suppliers without requiring code modifications is one of its benefits. It is also simpler to move between databases or even modify the database schema thanks to JPA's provision of a level of abstraction between the application and the database. JPA also supports connections between entities, which is a benefit. One-to-one, one-to-many, many-to-one, and many-to-many relationships are all possible. JPA also supports lazy loading of relationships, which can enhance the application's speed. Additionally, JPA supports caching, which can enhance application speed by minimising database requests. Either the entity level or the application level can set caching.

Additionally, JPA enables transaction management, a feature that guarantees that every database operation is completed as part of a single transaction. Between concurrent database activities, transactions offer consistency and separation. Finally, JPA offers a robust and user-friendly API for communicating with databases. Relationships, caching, and transaction management are all supported by it. It is simple to move between databases or alter the database schema thanks to JPA, a standard API that is supported by various Java Persistence providers. JPA is a fantastic option for developers who want to create dependable, scalable apps with an easy-to-use API.

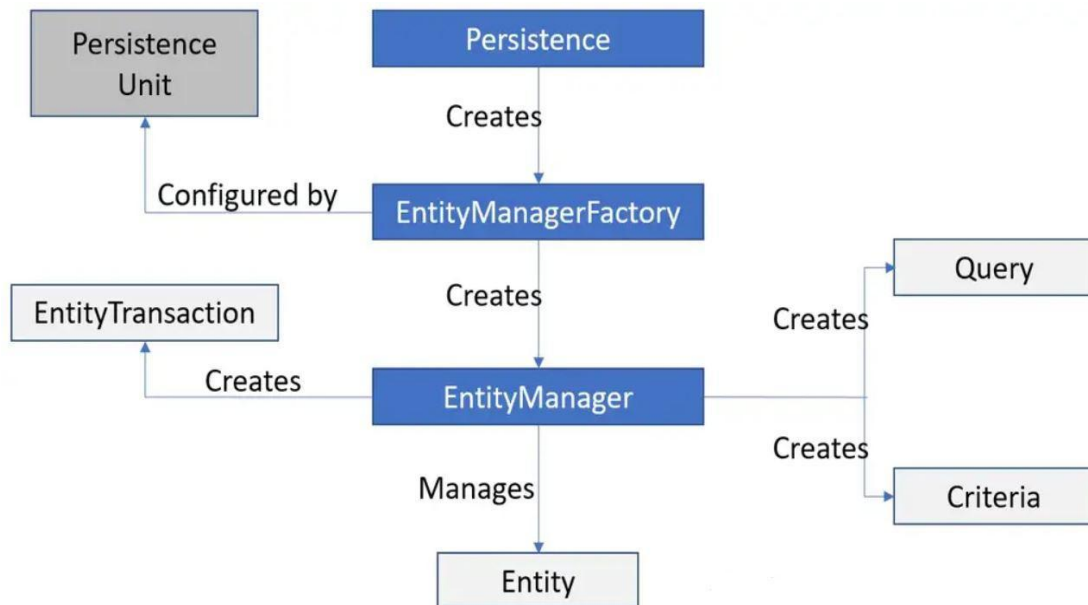


Fig: 4 JPA Working

9. MySql Database: A popular relational database management system (RDBMS) for storing and managing data is called MySQL, which is open-source. Since its first release in 1995, it has grown to be one of the most widely used databases worldwide. We'll examine MySQL's main attributes in this post and discuss why so many applications favour it. The scalability of MySQL is one of its key benefits. As the programme expands, it can simply be scaled up to take even more data. It can already manage a lot of data. Additionally, it is quite dependable and has no problems supporting several concurrent users. Numerous data kinds, such as integers, strings, dates, and times, are supported by MySQL. In order to enhance performance and optimise searches, it also supports a variety of index types. In order to guarantee data availability and recovery in the event of failures, it also offers a variety of backup and replication functions. The security features of MySQL are yet another crucial component. To guarantee that only authorised users may access the database, it offers a variety of authentication and encryption techniques. To monitor and manage database access, it also provides capabilities like auditing and access control. Additionally, MySQL is quite compatible with a wide range of frameworks and programming languages. For well-known programming languages like Java, Python, PHP, and others, it includes drivers.

Additionally, it integrates with well-known frameworks like Django, Spring, and Hibernate.

MySQL has a web-based graphical user interface (GUI) called MySQL Workbench for management and administration. Administrators can quickly administer and keep an eye on the database with this application. Additionally, it offers functions like database migration, performance tweaking, and backup and recovery. The fact that MySQL is open-source is one of its main features. This indicates that it is openly accessible and that it may be altered as necessary. As a result, a sizable developer community has contributed to the creation and upkeep of MySQL.

1.4 Organisation

PAXCOM

Paxcom is a software development firm that offers clients all over the world end-to-end e-commerce solutions. The business was established in 2004 and has its main office in New Delhi, India.

Paxco specialises in creating distinctive e-commerce solutions that are catered to the individual requirements of its customers. They provide digital marketing, cloud hosting, mobile application development, and website design and development. In a variety of industries, including retail, fashion, healthcare, and more, the organisation has worked with a diverse spectrum of customers, from tiny startups to big international corporations. Over 300 highly qualified individuals who are experts in a variety of technologies and frameworks, including Java, Spring, Hibernate, Angular, React, and more, make up the Paxcom team. They adhere to the Agile development process, which enables a flexible and iterative project development approach and guarantees that the finished product satisfies the client's expectations.

Paxcom offers consultation and support services in addition to e-commerce solutions to assist its clients in streamlining their online presence and e-commerce operations. Overall,

Paxcom is a respected business in the e-commerce sector with a proven track record of providing customers with high-quality solutions.

1. *Honesty* : Integrity and honesty are the guiding principles for Paxcom's work. In all business transactions, Paxcom is dedicated to providing genuine outcomes and upholding openness. Sincerity fosters trust among Paxcom's customers, staff members, and business partners, which is crucial for the organisation's long-term success.
2. *Customer Attraction*: Paxcom views its customers as partners in their success and cherishes their business. Paxcom wants to satisfy its customers' demands and win their confidence by offering superior services that go above and beyond their expectations. Paxcom's development and viability depend on developing solid client relationships.
3. *Cooperation*: According to Paxcom, success requires both teamwork and cooperation. Paxcom encourages its staff to collaborate as a big family since it understands the value of each employee's contribution to the overall success of the business. Mutual respect, trust, and support are important to Paxcom, which fosters an atmosphere where workers are inspired, empowered, and respected.
4. *Create the Paxcom logo*: Paxcom appreciates its brand and considers that each employee embodies the company's principles and objectives. Paxcom encourages staff members to uphold the organisation's basic beliefs and ideals, guaranteeing uniformity at all levels of the business. Paxcom can build a distinctive market position and draw in new clients by developing a powerful and cohesive brand identity.

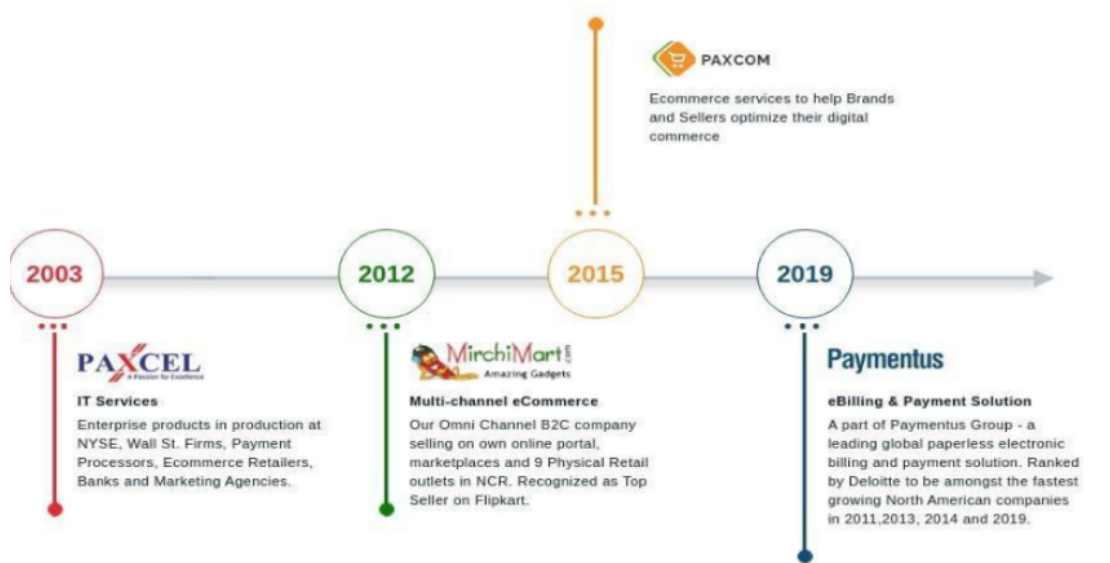


Fig: 5 Company Description

Paymentus

A technology startup called Paymentus offers commercial and governmental entities cloud-based bill payment solutions. Charlotte, North Carolina serves as the company's headquarters. It was established in 2004.

Online bill payment, smartphone payment, IVR (Interactive Voice Response) payment, walk-in payment, and text payment are just a few of the payment options provided by Paymentus. The company's platform accepts a broad range of payment options, including ACH (Automated Clearing House) payments, credit/debit cards, e-checks, PayPal, and PayPal payments. The flexibility of Paymentus to interact with existing billing and payment systems is one of its core benefits, enabling companies and organisations to quickly embrace the platform without having to make significant modifications to their present systems. The company also offers configurable branding options to assist businesses in preserving their own identity and messaging throughout the payment process. Paymentus prioritises security and compliance, employing technologies such as tokenization and encryption to protect sensitive payment information.

With the help of Paymentus, customers may use any financing source to pay billing firms in real-time. It provides a tier 1 PCI-compliant cloud-resident solution. Additionally, the business offers solutions for electronic bill presentment, client self-service, communication management, and revenue management. The utility, municipal, healthcare, insurance, and financial services sectors all benefit from it.

The company offers a user friendly platform that may assist organizations and will also enhance customer satisfaction and expediting payment operation.

This tactic ensures that businesses may benefit from a system that is efficient, streamlined, and cost-effective.

1. *Emphasis on configuration:* Instead of emphasising personalization, Paymentus focuses a large emphasis on setup. This indicates that their improved, law-based engine handles business requirements and payment regulations without requiring a great deal of customisation. This strategy guarantees that companies may take advantage of a system that is effective, streamlined, and cost-efficient.
2. *Accessible Platform:* Paymentus offers a cloud-based platform that can be quickly accessed via a variety of APIs, iFrames, and completely unique solutions. Businesses now have complete control over how their customers are treated. The platform provides a number of features, such as flexible billing choices, automatic payment reminders, and extensive reporting capabilities, that assist businesses in managing their payment operations better.
3. *Pci Level I Compliance:* Paymentus assists companies in lessening and removing a load of PCI as a PCI Level I compliant service. Businesses may rest easy knowing that their payment procedures are safe and uphold the strictest regulatory requirements as a result.

4. *Safe And Reliable System:* Paymentus provides a very safe and dependable system with a 100% uptime guarantee. To guarantee the dependability and security of transactions, the system is outfitted with a multi-layer detection and blocking system, multi-level authentication, encryption, and token processing. Businesses will be able to handle payments confidently knowing that their data is secure thanks to this.

5. *Seamless Integration:* Paymentus provides seamless payments, reporting, and reconciliation to important financial systems and activities by covering more than 350 essential programmes, such as the CIS, accounting, and ERP systems. This makes it simpler for organisations to manage their financial operations by enabling easy integration of payment processes with current systems.

6. *Innovative Approach:* Paymentus was created from scratch, using a single piece of code, and there are no versions. Paymentus is able to continuously develop and keep ahead of the always-changing demands thanks to this novel methodology. This indicates that companies may profit from a system that is constantly updated and furnished with the newest features and functionalities. Businesses can stay ahead of the curve and remain competitive in a continuously evolving business environment thanks to Paymentus' creative strategy.

CHAPTER 02:

LITERATURE SURVEY

Software development requires issue tracking to manage and keep track of software flaws, bugs, and feature requests. Users may create, edit, and close issues, follow their progress and assign them to developers for resolution using an issue tracker system. Many problem-tracking systems have been created in recent years utilising a variety of technologies. We will concentrate on issue tracker systems created using Spring MVC in this literature review.

A well-liked Java-based framework for creating web apps is Spring MVC. It gives developers a framework for building dependable, scalable applications that are simple to combine with other frameworks and technologies. Due to its adaptability and user-friendliness, Spring MVC has been used to build several issue tracking systems. JIRA is one of the most used issue tracking systems created with Spring MVC. JIRA is an exclusive problem-tracking tool created by Atlassian. It offers a complete set of capabilities for keeping track of issues, including scalable dashboards, reports, and workflows. Git, Subversion, and Jenkins are just a few of the development technologies that JIRA interfaces with. Additionally, it offers REST APIs for interfacing with outside programmes.

The open-source issue/bug tracking tool Bugzilla enables developers to monitor still-open issues with their product. It uses a MySQL database and is developed in Perl. Although Bugzilla is a tool for recording defects, it may also be used to manage tests, and can therefore be readily integrated with other test case management platforms like Quality Centre, Testlink, etc. Users are able to maintain contact with their customers or staff members and efficiently communicate about issues along the whole data management chain thanks to this open bug tracker. An indispensable issue tracking system called Bugzilla was developed using Spring MVC. Bugzilla is an open-source issue tracking programme developed by the Mozilla Foundation. It's frequently used by software development brigades to keep track of blights, point requests, and other issues. Bugzilla offers a web-grounded interface for adding, modifying, and terminating problems.

MantisBT is an open - source issue tracking system and is created using PHP and Spring MVC. For creating, amending, and ending problems, it provides a simple stoner interface. MantisBT tracking system also supports may feature related to tracking issues. It also uses REST API is likewise accessible for communicating with outside software.

Last but not least, Trac is an open-source issue tracking system created with Python and Spring MVC. For tracking issues, Trac offers a full range of tools, including adaptable workflows, reports, and notifications. Git and Subversion are only two of the version control systems that Trac connects with. A roadmap planner, wiki, and forums are also available.

The management of software flaws, bugs, and feature requests is made flexible and scalable by issue tracker systems created using Spring MVC. These systems include a wide range of features, such as flexible workflows, sophisticated search tools, and email alerts. Due to their simplicity of use and compatibility with other development tools, issue-tracking systems like JIRA, Bugzilla, Redmine, MantisBT, and Trac have grown in popularity. The user interface of a project employing a Spring MVC issue tracker is one of its most crucial components. An effective user interface may be the difference between a project's success and failure. The user interface should be simple to use, offer clear and succinct information, and be intuitive. In order for customers to customise it to meet their unique needs, it should also be customizable.

Database architecture is a crucial component of a Spring MVC project that uses an issue tracker. The database has to be built to handle the project's unique requirements. It needs to be able to manage a lot of data and be designed for quick retrieval. Additionally, the security of the Spring MVC project's issue tracker is crucial. To make sure that only people with permission may access the system, the project needs a secure login mechanism. The database should securely hold encrypted passwords. In order to ensure that users only have access to the system components they really require, the system should also incorporate role-based access control. The Spring MVC project's problem tracker ought to include a robust reporting mechanism as well. Users should be able to access current information on the status of their concerns using this system. Users ought to be able to create personalised reports depending on their unique requirements.

CHAPTER 03:

SYSTEM DEVELOPMENT

3.1 Analytical System Development

An analytics system is a piece of software that gathers, works with and examines data in order to offer insights into how well a system is performing. When tracking issues with Spring projects, an analytics system may offer insightful information about the project's status, point out bottlenecks, and recommend areas for development. In this post, we'll go through how to use Spring projects to create an analytics system for monitoring issues. Finding the important metrics that must be tracked is the first step in creating an analytics system for issue monitoring. The number of open problems, the number of closed issues, the time it took to resolve an issue, the average time it took to resolve an issue, the number of issues by category, and other metrics are examples of these metrics. The next stage is to create a data collecting system that can extract this information from the problem-tracking system after the important metrics have been determined.

Using Spring Data JPA to establish a connection to the issue tracking system's database and obtain the necessary data is one technique to achieve this in Spring applications. Building an analytics system is excellent for Spring Data JPA as it offers a simple and effective approach to interface with databases. Using a variety of analytical tools and techniques, the data may be processed and analysed after it has been obtained. Data visualisation, which includes creating visual representations of data to spot patterns and trends, is one often used analytical tool. To create an analytics system, a Spring project can use a variety of data visualisation technologies. These consist of well-known applications like Tableau, Power BI, and QlikView.

The capacity to create reports is a crucial component of creating an analytics system for tracking issues. Reports may be used to convey progress and pinpoint areas for development. They can include an overview of important metrics and trends. Reports may be generated for spring projects using a variety of reporting systems, including Pentaho, BIRT, and JasperReports.

The capability to execute predictive analytics is a crucial component in creating an analytics system for tracking issues, in addition to data visualisation and reporting. Statistical algorithms and machine learning techniques are used in predictive analytics to analyse data and forecast future results. Predictive analytics may be used in the context of problem monitoring to detect possible issues before they arise, prioritise issues based on their potential impact, and more effectively allocate resources.

Several machine learning frameworks, including Apache Mahout, Weka, and TensorFlow, may be used to create predictive analytics in a Spring project. These libraries offer a variety of algorithms that may be used to issue tracking data to perform predictive analytics. Predictive analytics are implemented along with key metrics identification, data collecting system development, data visualisation and reporting tools, and the usage of Spring projects to create an analytics system for issue monitoring. Organisations may increase performance by utilising the strength of Spring projects and related technologies to acquire insightful knowledge about their problem-tracking procedures.

3.2 Computational System Development

1. ***System Requirements:*** This will entail obtaining from users and stakeholders all pertinent needs for the system. This entails comprehending the system's goal, its scope, its functional and non-functional needs, and any limitations.

- a. *Software Requirements:*

- i. HTML, JSP, CSS
 - ii. JavaScript
 - iii. Java
 - iv. Spring
 - v. Maven
 - vi. Mysql

- b. *Hardware Requirements:*

- i. 2.2Ghz CPU or higher
 - ii. 8GB RAM or higher

iii. 10GB HDD/SSD

2. ***System Architecture:*** The system architecture will describe how the different components of the system will interact with each other. This includes the design of the user interface, the server-side components, the database design, and the communication protocols.

With the help of Spring MVC, JSP, Hibernate, JPA, and a MySQL database, an issue tracking web application has a multi-layered architecture with a variety of components that interact with one another to provide a reliable and effective system. The MySQL database, which is used to hold all the data linked to the issue tracking system, is at the centre of the design. The database is in charge of keeping track of all connected data, including user identification, issue specifics, comments, and attachments. It is built with effective data indexing and retrieval techniques to provide high availability, scalability, and data integrity.

The data access layer, which is the next layer in the design, is in charge of connecting to the database, obtaining the necessary data, and persisting the fresh data. Utilising Spring Data JPA, which offers a standardised and effective method of maintaining and accessing the data in the database, this layer is created. JPA is used to guarantee that the data access layer is independent of the underlying database technology, allowing for easy future migration to a different database technology. The business logic layer sits on top of the data access layer and is composed of all the business rules and procedures that control how the system functions. Spring MVC, which offers a reliable and adaptable framework for creating web applications, is used to develop this layer. The system is scalable, manageable, and tested thanks to the usage of Spring MVC, and there is a distinct separation of responsibilities between the various components.

JSP is used to create the presentation layer of the architecture because it offers a versatile and potent method of creating dynamic web pages. The JSP pages are in charge of providing the user with a clear and user-friendly display of the data and enabling interaction with the system through the submission of forms, requests, and

other activities.

The system also has a security layer, which is in charge of making sure the system is safe and that only approved users are permitted access. Spring Security, which offers a complete set of tools and functionality for protecting online applications, is used to construct this layer. The system is safeguarded against prevalent security risks including cross-site scripting, SQL injection, and session hijacking thanks to the implementation of Spring Security. The system design also has additional parts for caching, logging, and monitoring. While logging and monitoring are intended to offer extensive information about the functioning of the system and to warn the administrators in case of any mistakes or difficulties, caching is used to increase system efficiency by keeping frequently used data in memory.

Using Spring MVC, JSP, Hibernate, JPA, and a MySQL database, an issue tracking web application has a multi-layered architecture with a variety of components that interact with one another to provide a reliable and effective system. The system is scalable, manageable, and tested because to the use of well-known and standardised technologies like Spring, JSP, and JPA, and it is dependable and effective thanks to the use of a strong and secure database like MySQL.

3. **Database Design:** This will involve designing the database schema, creating tables, defining relationships between tables, and defining constraints to ensure data integrity.

Any online application development process must include database design, but issue tracking apps require it more than others. In this example, we need to construct three tables: problems, user, and user_roles. We are using Spring MVC, JSP, Hibernate, and JPA with a MySQL database. All concerns that are reported by users will be kept in the first table, the "issues" table. A title, a description, a date/time stamp of when it was made, and a status indication (such as open, in progress, closed, etc.) will all be included in each issue. This table will also have columns for the person who reported the issue and the user who is in charge of fixing it. The details of all users who have registered with the system will be kept in the second table, the "user" table. The user's unique identifier, first and last names, email address, username, and password will all be included in separate columns in

this table. To store additional data, such as a user's job or contact details, we may also introduce columns.

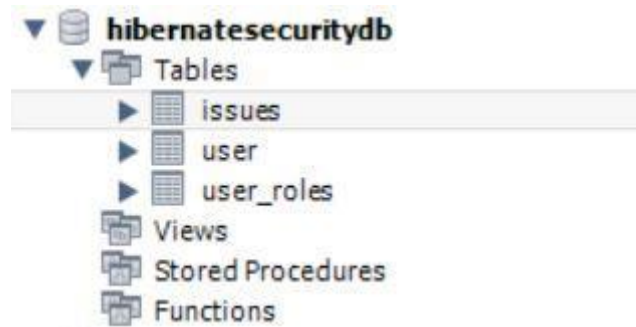


Fig 6: Database Tables

The "user_roles" database will also contain the places that have been assigned. The name of the part, the part's unique identifier, and the user's unique identifier will all have columns in this table. This table in the database is made by using a one-to-many relationship. In this case, the three tables stated above, and their liaison will make up the ERD.

The "user" table and the "issues" table will be related one too many times. The assigned user column will also create a many-to-one link between the "issues" table and the "user" table. This implies that while a user may be allocated to several issues at once, only one user may be assigned to any one issue at any given moment. The "user_roles" table and the "user" table will be related one too many times. This implies that even though a user may have several responsibilities, only one user may be assigned to each position at once.

With this database design, we can build an issue tracking web application that enables users to submit issues, see their own issues, and follow the status of each issue using Spring MVC, JSP, Hibernate, and JPA. By giving each user a role and limiting access to particular features based on those responsibilities, we can also create user authentication and authorization. In terms of implementation, we may use JPA to control the durability of these objects and Hibernate to map Java objects to database tables. The web requests and answers may be managed using Spring MVC, and the views can be rendered via JSP. We can make sure the application is scalable, dependable, and secure by using MySQL as the database.

Last but not least, the database architecture of an issue tracking web application utilising Spring MVC, JSP, Hibernate, and JPA with a MySQL database entails building three tables: problems, user, and user_roles. Relationships are built into the tables to guarantee data integrity and quick data retrieval. With the help of this database design, we can build an issue tracking system that is reliable, scalable, and capable of handling massive amounts of data while offering consumers a smooth user experience.

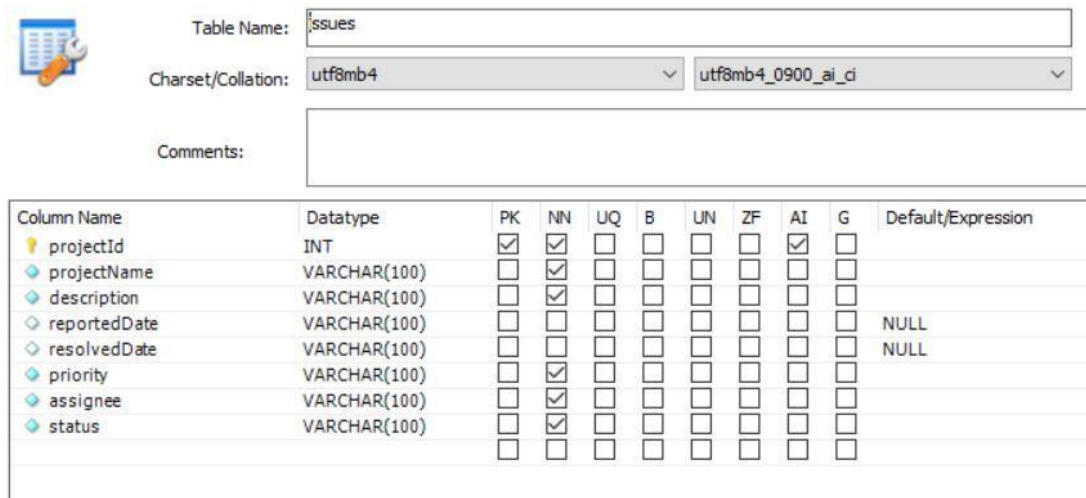


Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
projectId	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
projectName	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
description	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
reportedDate	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
resolvedDate	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
priority	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
assignee	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
status	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 7: Issues Table Properties

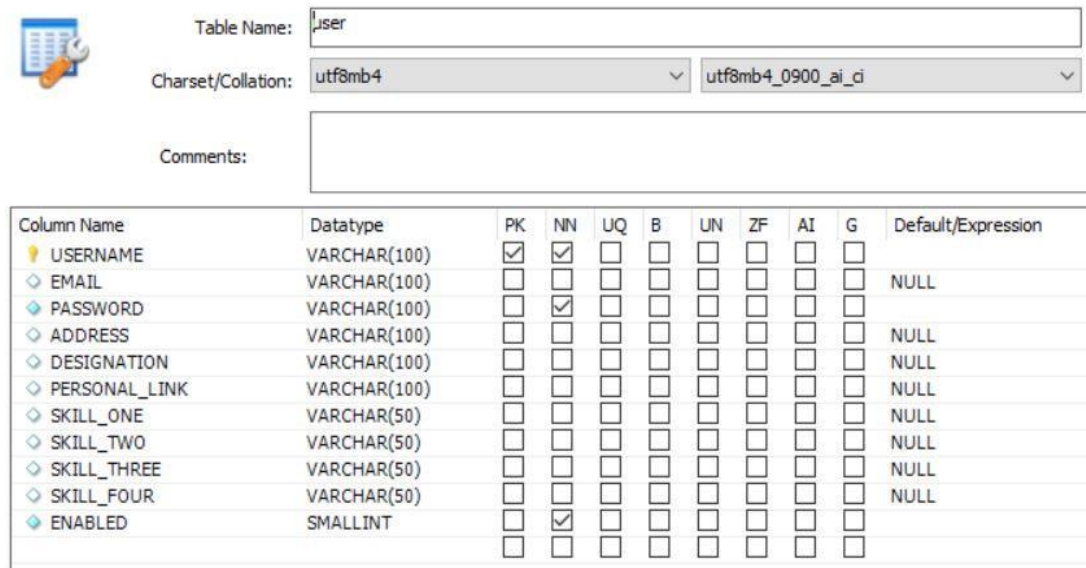


Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
USERNAME	VARCHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EMAIL	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
PASSWORD	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ADDRESS	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
DESIGNATION	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
PERSONAL_LINK	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SKILL_ONE	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SKILL_TWO	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SKILL_THREE	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SKILL_FOUR	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ENABLED	SMALLINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fig 8: User Table Properties

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ROLE_ID	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
USERNAME	VARCHAR(36)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
USER_ROLE	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Fig 9: user_roles Table Properties

4. **User Interface Design:** The user interface will include design of the web pages, user input forms, and navigation menus. The design will need to be responsive and user-friendly to ensure that users can easily use the system. The user interface design of an issue tracking web application using Spring MVC, JSP, and Hibernate is crucial for the user experience and overall success of the application. There are 18 pages comprising header footer, adminLogin, assignedEmployee, assignIssue, dashboard, editIssue, editProfile, issuerReport, loginForm, registerEmployee, resolvedIssue, showEmployee, unassignedEmployee, unassignedIssue, viewProfile, and welcomePage. Additionally, JavaScript, Bootstrap, and CSS are used to enhance the user interface and provide a more responsive and visually appealing design.

Every page has a uniform header and footer, making it simple for users to navigate and access content. The dashboard, profile, and logout links are often included in the header along with the company name and logo. Additional links to the terms of service, privacy policy, and contact information are provided in the footer. For administrative access to the programme, utilise the *adminLogin* page. An administrator may log in using their credentials on this page to access extra capabilities including the ability to assign and remove issues to staff members and read thorough reports.

A summary of all issues, including their status, priority, and allocated employee, is shown on the *dashboard* page. The user may sort and filter issues on this page using a variety of parameters.

The screenshot shows the FAULTIFY dashboard for an Admin user. The main content area is titled 'Issues Reported Admin view'. It features a search bar and a 'Show 10 entries' dropdown. Below is a table with 10 rows of issue data. At the bottom, it indicates 'Showing 1 to 10 of 16 entries' and has pagination controls for 'Previous', '1', '2', and 'Next'.

Appliance ID	Appliance Name	Date	Description	Priority	Assignee	Status	Edit
1	Refrigerator	02-05-2023	Not cooling properly	Medium	Raven	Closed	Click here
2	Washing Machine	17-04-2023	Not working	High	Himanshu	Open	Click here
3	Microwaves	17-04-2023	Not heating	High	Shubham	Closed	Click here
4	Dishwasher	18-04-2023	Leakage fault	Medium	Unassigned	Active	Click here
5	Air Conditioner	13-03-2023	Leakage	Medium	Shubham	Closed	Click here
6	Air Conditioner	17-04-2023	Not cooling properly	Medium	Shubham	Closed	Click here
7	Dryers	17-04-2023	Not heating properly	Medium	Raven	Closed	Click here
8	Dishwasher	19-04-2023	Not washing	Low	Unassigned	Open	Click here
9	Washing Machine	17-04-2023	not working properly	High	Unassigned	Open	Click here
10	Vacuums	18-04-2023	Not working	Medium	Shubham	Closed	Click here

Fig 10: Dashboard

All issues that are presently assigned to the logged-in employee are shown on the *assignedEmployee.jsp* page. The employee may examine the specifics of the assigned issues on this page, including their description, status, and priority. The employee can also update the status and fix the problem.

An employee can be given an issue using the *assignIssue.jsp* page. The administrator can choose a problem from a drop-down menu on this page and allocate it to a worker.

The page for the logged-in employee's assigned problems is located at *assignedEmployee*. The employee has access to the priority, status, and description of the assigned issues on this page. Additionally, the employee has the ability to update the situation and fix the problem.

To assign an issue to an employee, visit the *assignIssue* page. A problem can be chosen from a drop-down menu on this page by the administrator and then assigned

to a specific employee.

The priority, status, and description of an existing issue can all be changed using the *editIssue* page. Both administrators and workers have access to this page.

The user may alter their profile information, including their name, email address, and password, on the *editProfile* page.

All problems that have been fixed are listed on the *issuerReport* page. Information like the resolution date, the description, and the allocated employee are all included on this page.

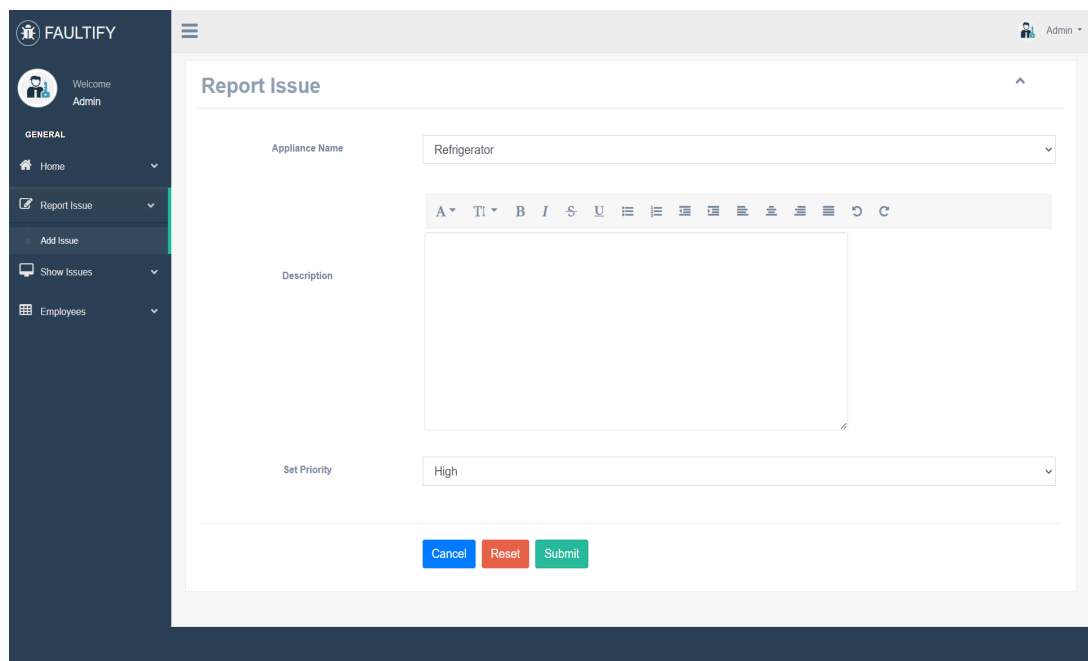
The image shows a web application interface for reporting an issue. On the left is a dark blue sidebar with the logo 'FAULTIFY' and a user profile for 'Admin'. The main content area is titled 'Report Issue' and contains three main sections: 'Appliance Name' with a dropdown menu showing 'Refrigerator', 'Description' with a rich text editor toolbar and a large text area, and 'Set Priority' with a dropdown menu showing 'High'. At the bottom of the form are three buttons: 'Cancel' (blue), 'Reset' (red), and 'Submit' (green). The top right corner of the page shows a user profile icon and the name 'Admin'.

Fig 11: Issue Report Form

User authentication takes place on the *loginForm* page. The user can access the application by logging in on this page using their login information.

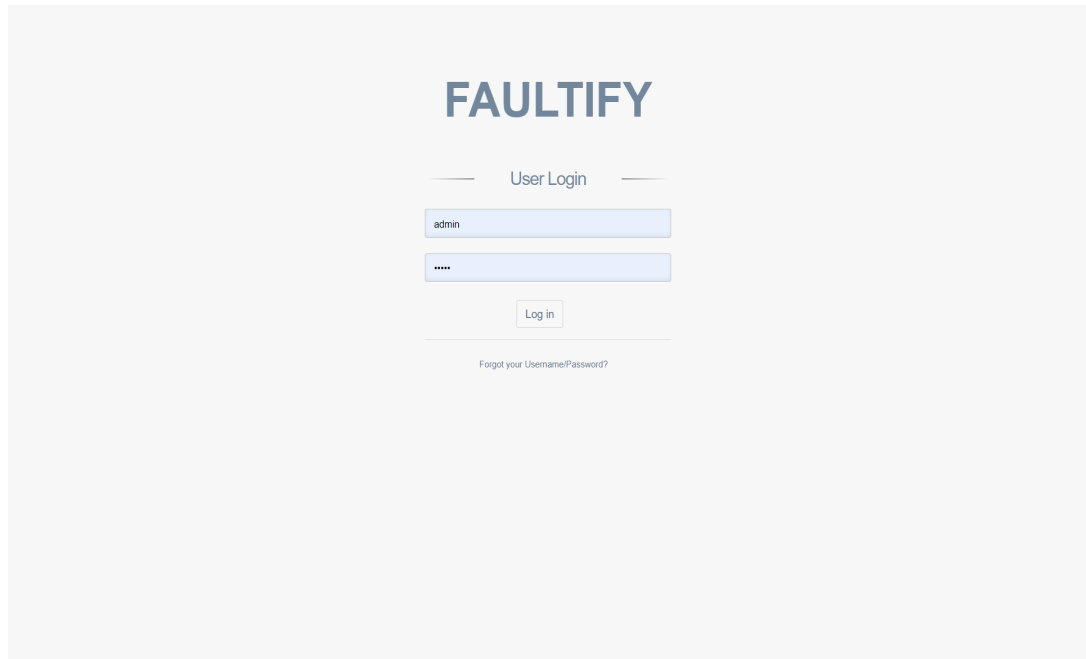


Fig 12: Login Page

A new employee is registered via the *registerEmployee* page. The name, email address, and password of the employee are all fields on this page.

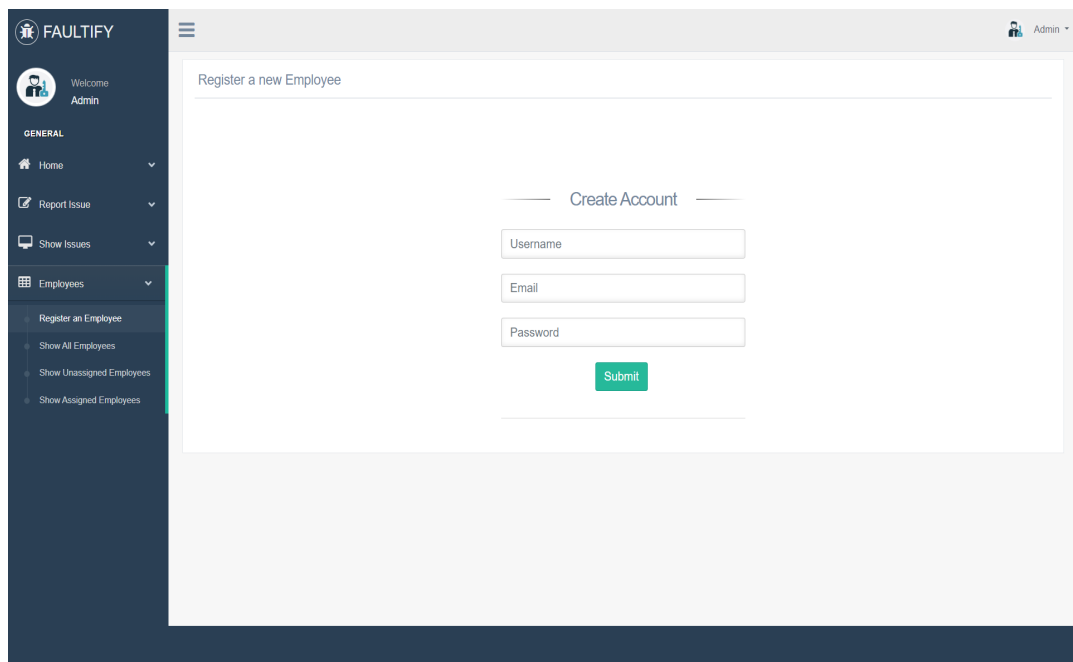


Fig 13: Register Employee Page

All issues that have been resolved are shown on the *resolvedIssue* page. The user may examine the specifics of the resolved issues on this page, including the resolution date, the issue's description, and the assigned employee.

All of the workers in the system are visible on the *showEmployee* page. The administrator may examine each employee's information, including name, email address, and assigned issues, on this page.

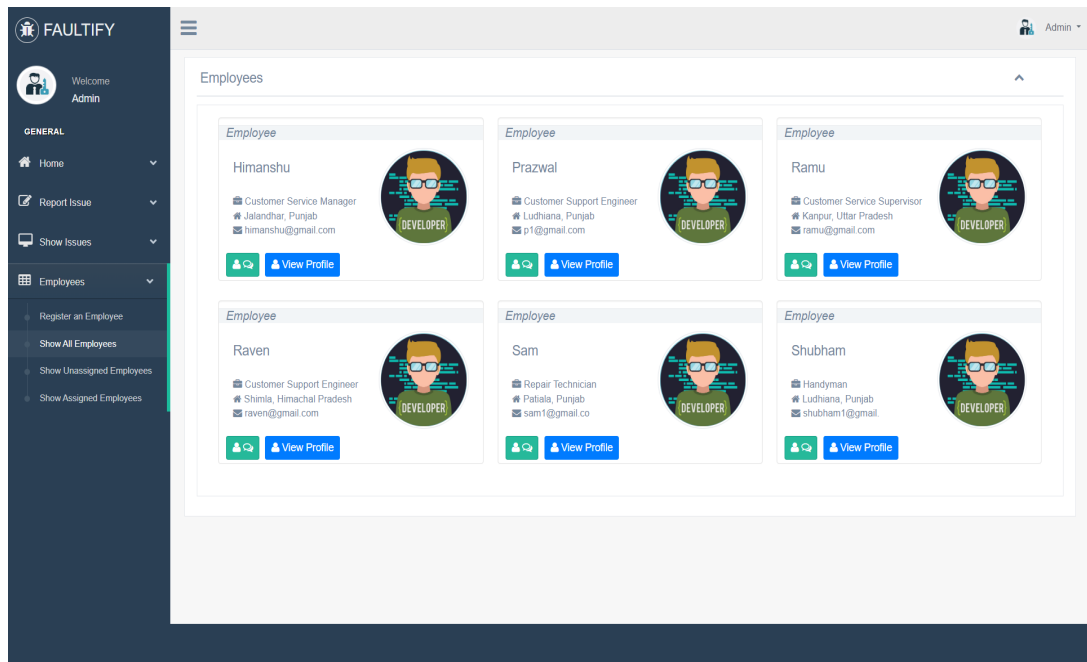


Fig 14: showEmployee.jsp Page

All workers who are not presently allocated to any problems are shown on the *unassignedEmployee* page. The administrator may examine each unassigned employee's data on this page and allocate them to a problem.

All issues that are not currently assigned to any employee are shown on the *unassignedIssue* page. The administrator may check the specifics of each open problem on this page and allocate it to an employee.

The user's profile details, including their name, email address, and assigned issues, are displayed on the *viewProfile* page.

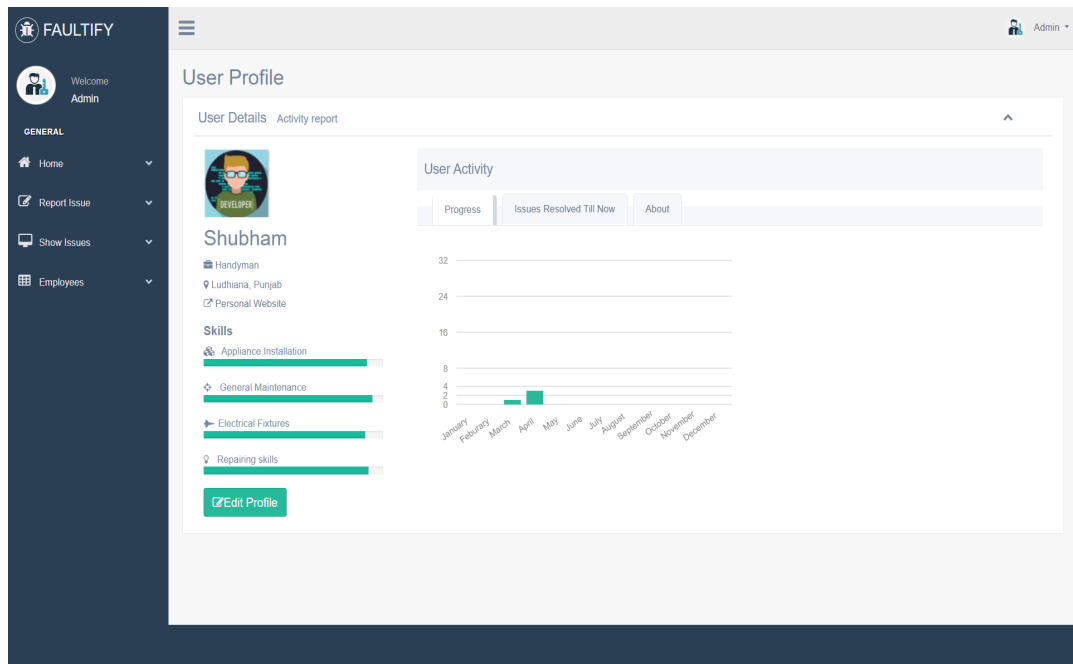


Fig 15: viewProfile.jsp Page

The programme's *welcomePage*, which serves as the landing page, gives a quick overview of the features and purpose of the application.



Fig 16: Welcome.jsp Page

To enhance the user interface's functionality and interaction, JavaScript is employed. JavaScript, for instance, may be used to validate user input, dynamically

update the website, and improve the user experience as a whole. The popular CSS framework Bootstrap is used to create responsive and aesthetically pleasing designs. Several pre-built UI elements, like buttons, forms, and alerts, are included with Bootstrap and may be simply modified to reflect the style of the application.

5. ***Business Logic Implementation:*** Business logic implementation is a critical aspect of any project. In the case of an issue tracker project, the business logic determines how the application should behave when a user interacts with it. In this article, we will discuss the business logic implementation of an issue tracker project using Spring MVC, Hibernate, and JPA, with a focus on the three main classes - user, user_roles, and issues.

User Class

User data like usernames, passwords, emails, and roles are managed by the user class. The user's level of access within the application is determined by their role. This programme has three different roles: admin, employee, and customer. All programme functions are accessible to the administrator, but workers only have access to assign and fix issues; consumers may only report issues.

Authentication and authorisation are both included in the business logic for the user class. By validating the user's login and password, authentication makes sure the user is who they say they are. The degree of access a user has to carry out particular operations within the programme is determined by authorization. Spring Security is one way the business logic for authentication and authorisation is implemented. With Spring MVC, Hibernate, and JPA, Spring Security offers a thorough security architecture that is simple to integrate. In-memory authentication, LDAP authentication, and database authentication are just a few of the several authentication providers that Spring Security may use to manage authentication and authorisation. Furthermore, Spring Security has built-in support for role-based access management, enabling developers to specify the amount of access needed for certain application operations.

User_Roles Class

The many roles that users can have inside the application are managed by the `user_roles` class. As was previously said, this programme has three different sorts of roles: admin, employee, and customer. The creation and management of roles, the assignment of roles to users, and the determination of the level of access for each role are all included in the business logic for the `user_roles` class.

Utilising Hibernate to control the persistence of roles and user-role connections is one way the business logic for the `user_roles` class is implemented. A well-liked object-relational mapping framework called Hibernate enables programmers to map Java objects to database tables. Developers may quickly build, change, and retrieve roles and user-role associations from the database by utilising Hibernate. Hibernate also supports slow loading and caching, which can enhance the speed of the application.

Issues Class

The many issues that users submit are managed by the `issues` class. Every problem has a name, a brief description, a status, and a priority. The priority might be low, medium, or high, and the status can be open, assigned, or resolved. The business logic for the `problems` class entails the creation and management of issues, the delegation of issues to personnel, and the tracking of each issue's priority and status.

JPA may be used to handle the persistence of issues as one implementation of the business logic for the `problems` class. The Java object-relational mapping specification JPA offers a high-level API for managing entities. Developers may quickly create, change, and retrieve problems from the database by utilising JPA.

In conclusion, the success of an issue tracker project utilising Spring MVC, Hibernate, and JPA depends on how the business logic is implemented. The business logic also creates and manages roles for the `user_roles` class, issues for the `issues` class, and authentication and authorisation for the `user` class. Developers

may design a reliable and secure issue tracker application that satisfies the expectations of its users by integrating these business logic components.

6. ***Authentication and Authorization***: The system will need to include a secure authentication and authorization mechanism to ensure that only authorized users can access the system and perform specific tasks. Authentication describes the procedure of confirming a user's identity when they attempt to access a protected resource in the context of online applications. On the other hand, authorization is the procedure used to ascertain if a user is authorised to access a certain resource.

Spring Security's `UserDetailsService` class is in charge of both authentication and authorisation. It supplies the authentication mechanism with user-specific data that is loaded from a database or another data source. In other words, it is used to extract user information from the database, including usernames, passwords, and authorities. Only one method, "loadUserByUsername," on the `UserDetailsService` interface accepts a username as an input and returns a `UserDetails` object. Information about a user, such as their username, password, and authority, is available in the `UserDetails` interface.

We must construct a class that implements the `UserDetailsService` interface and overrides the `loadUserByUsername` function in order to implement the `UserDetailsService` class. In this procedure, the user details are retrieved from the database and added to a new `UserDetails` object. We can utilise the `UserDetailsService` class in our Spring Security setup to authenticate and authorise users after we have created it. We normally do this by establishing a security filter chain, which is in charge of intercepting incoming requests and confirming that the user is authenticated and authorised to access the requested resource, and an authentication manager, which utilises the `UserDetailsService` to load user data.

Spring Security offers a number of other authentication methods in addition to the `UserDetailsService`, such as LDAP authentication, custom authentication providers, and OAuth 2.0. Every mechanism has its own advantages and disadvantages, and the best option relies on the application's particular needs.

Overall, the `UserDetailsService` class is an essential part of Spring Security since it offers a straightforward and adaptable method for importing user information from different data sources. We can make sure that our application is safe and that only authorised users have access to protected resources by implementing this class.

7. ***Data Access Layer Implementation:*** This involves implementing the data access layer using JPA and Hibernate. This layer is responsible for accessing and manipulating data stored in the database.

Any web application must include a Data Access Layer (DAL), which is in charge of accessing and modifying data kept in the database. The DAL is in charge of locating and updating information about users, issues, and user responsibilities in the case of an issue tracking web application. Using JPA and Hibernate, the DAL is implemented as part of the data access layer.

A Java framework called Java Persistence API (JPA) makes interacting with databases simpler. Developers may specify the database schema and use Java objects to communicate with the database using a set of interfaces and annotations that are provided. Popular ORM (Object-Relational Mapping) software called Hibernate offers a JPA implementation. Creating the entity classes that represent the database tables is the first step in putting the Data Access Layer into practice. There are three tables in the case of an issue tracking web application: users, issues, and user roles. There is a specified entity class that corresponds to each table's columns. JPA annotations that specify the table schema, connections to other tables, and other metadata are added to the entity classes.

The DAO (Data Access Object) classes that offer the API to communicate with the database must then be defined. The CRUD operations for each object are implemented by the DAO classes. The `UserDAO` class, for instance, has methods for creating, reading, updating, and deleting user entries from the database. To get a database session, the DAO classes use Hibernate's `SessionFactory`. The session

offers ways to carry out database activities including accessing, updating, or storing items, as well as removing them. The JPA EntityManager interface is used by the DAO classes to access the database. Executing queries, managing transactions, and persisting entity objects to the database are all duties of the entity manager.

Hibernate offers a caching technique that caches frequently requested data in memory in order to increase efficiency and decrease the frequency of database requests. The caching method can be set up to cache queries, entities, or a combination of the two. Several caching services, including EHCACHE, Infinispan, and Hazelcast, are supported by Hibernate. Hibernate also allows lazy loading, so related entities are only loaded the first time they are accessed. For instance, in the case of an issue tracking web application, the User entity and the Issue object have a one-to-many connection. Only when the User entity's getIssues() function is invoked are the issues loaded.

In conclusion, the Data Access Layer Implementation is a crucial component of a web application for tracking issues. Working with databases is made simpler by the JPA and Hibernate frameworks, and speed is enhanced by their lazy loading and caching features. The CRUD activities are carried out for each object by the DAO classes, which also offer the API to communicate with the database.

CHAPTER 04: EXPERIMENTS AND RESULTS ANALYSIS

Firstly, the admin has logged in to the web application who has the role of 'ROLE_ADMIN'.

Basically, all the issues reported are visible to the admin or any user who has an admin role.

So admin needs to assign the issues (unassigned issues) to the employee. Also, every issue has a unique project ID related to it which is auto-incremented implicitly by the database. Admin can also change information linked with the issue, like the project name, description, priority, assignee and status.

Here in this case, an issue with Project ID 8 and with the project name of Dishwasher is assigned to the employee Ramu.

The screenshot displays the 'Update Issue With Project ID: 8' form in the FAULTIFY application. The form includes the following fields and options:

- Project Name *: Dishwasher
- Date*: 19-04-2023
- Description*: Not washing
- Set Priority *: Low
- Assignee *: Raven
- Set Status *: A dropdown menu with options: Unassigned, Himanshu, Prazwal, Ramu (highlighted), Raven, Sam, Shubham.

Fig 17: Assigning Issue To An Employee

In the dashboard we can see that the Appliance/Project with Appliance ID 8 is now assigned to employee Ramu.

Issues Reported Admin view

Show 10 entries Search:

Appliance ID	Appliance Name	Date	Description	Priority	Assignee	Status	Edit
1	Refrigerator	02-05-2023	Not cooling properly	Medium	Raven	Closed	Click here
2	Washing Machine	17-04-2023	Not working	High	Himanshu	Open	Click here
3	Microwaves	17-04-2023	Not heating	High	Shubham	Closed	Click here
4	Dishwasher	18-04-2023	Leakage fault	Medium	Unassigned	Active	Click here
5	Air Conditioner	13-03-2023	Leakage	Medium	Shubham	Closed	Click here
6	Air Conditioner	17-04-2023	Not cooling properly	Medium	Shubham	Closed	Click here
7	Dryers	17-04-2023	Not heating properly	Medium	Raven	Closed	Click here
8	Dishwasher	19-04-2023	Not washing	Low	Ramu	Open	Click here
9	Washing Machine	17-04-2023	not working properly	High	Unassigned	Open	Click here
10	Vacuums	18-04-2023	Not working	Medium	Shubham	Closed	Click here

Showing 1 to 10 of 16 entries Previous 1 2 Next

Fig 18: Admin Dashboard

Now, employee Ramu will log in to the web application which is named Faultify.

FAULTIFY

User Login

ramu

Log in

[Forgot your Username/Password?](#)

Fig 19: Login form

When Ramu has logged in to the web application, he gets access to some limited features on his dashboard as the role of the employee is 'ROLE_EMPLOYEE'. So as Ramu logs in to his account, he can see the assigned issue in his dashboard and can now update the status of the issue.

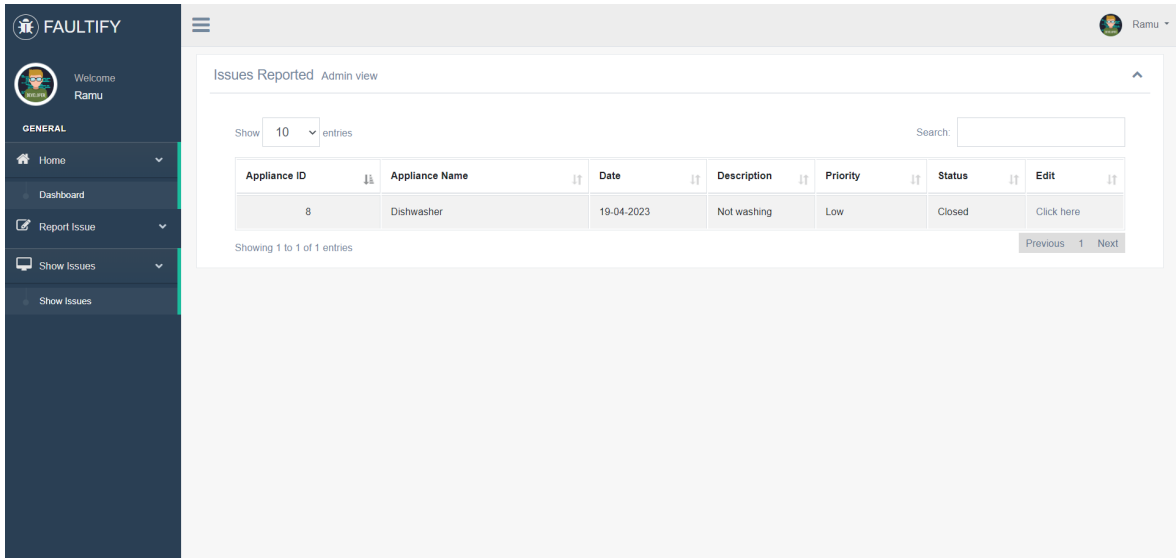


Fig 20: Employee Dashboard

Now when an employee clicks on the 'Click here' button in the edit column of the table, the employee can update the status of the issue assigned to him/her.

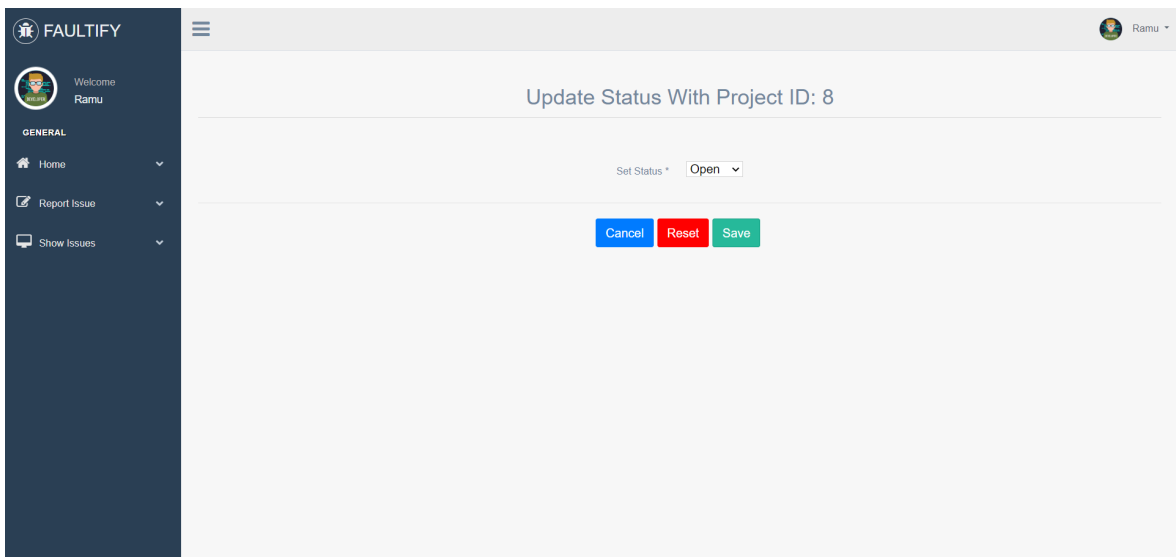


Fig 21: Status Update Page

Now the employee can set the status to Active, Testing and Closed. Here, Ramu sets the status to 'Closed' and saves the status.

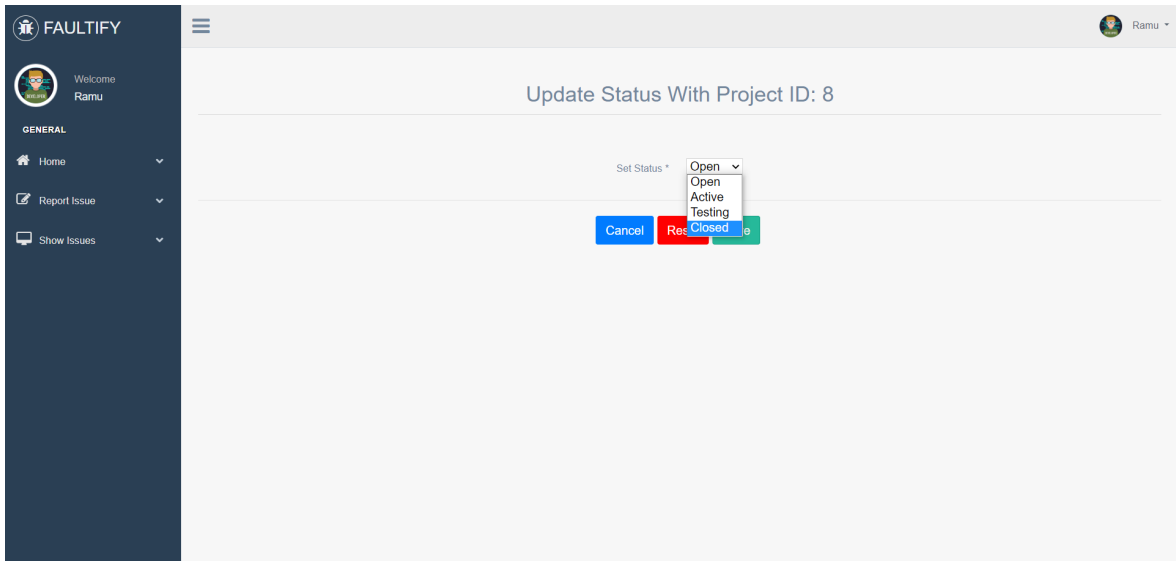


Fig 22: Updating status of an Issue

The issue is now marked as 'Closed' i.e. it signifies that the issue has been completely resolved and has been closed by the employee. Saving the status to 'Closed' by the employee also updates the status in the admin dashboard.

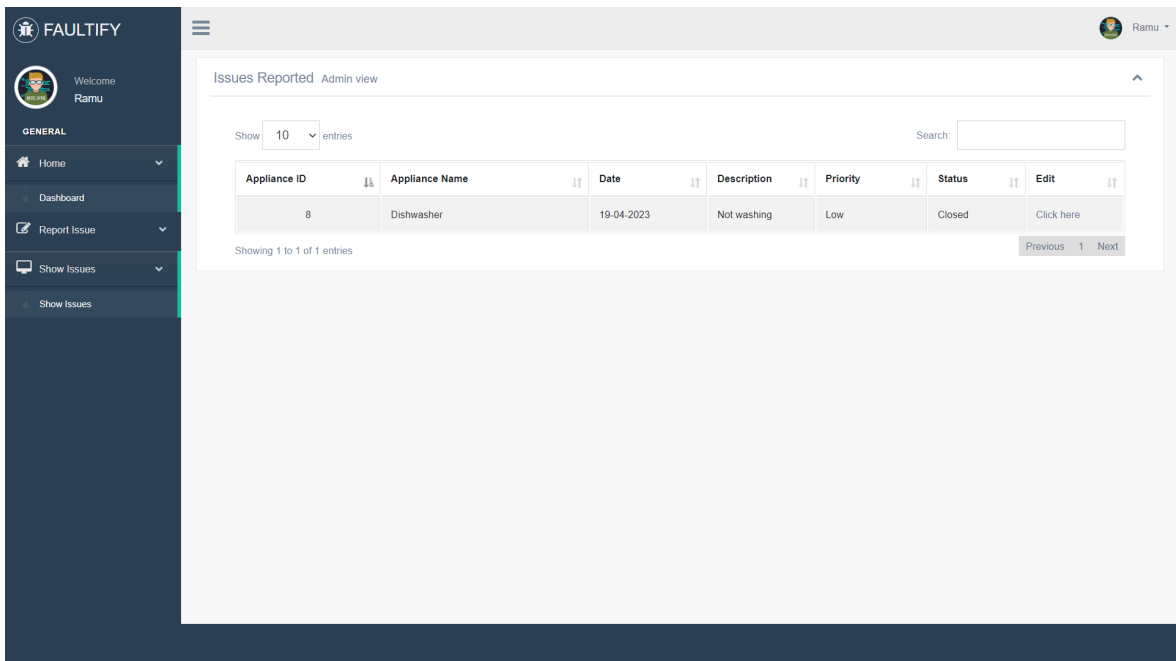


Fig 23: Updated Employee Dashboard

After marking the issue 'Closed', the employee can check his/her profile and see how many issues have been completely resolved by that employee.

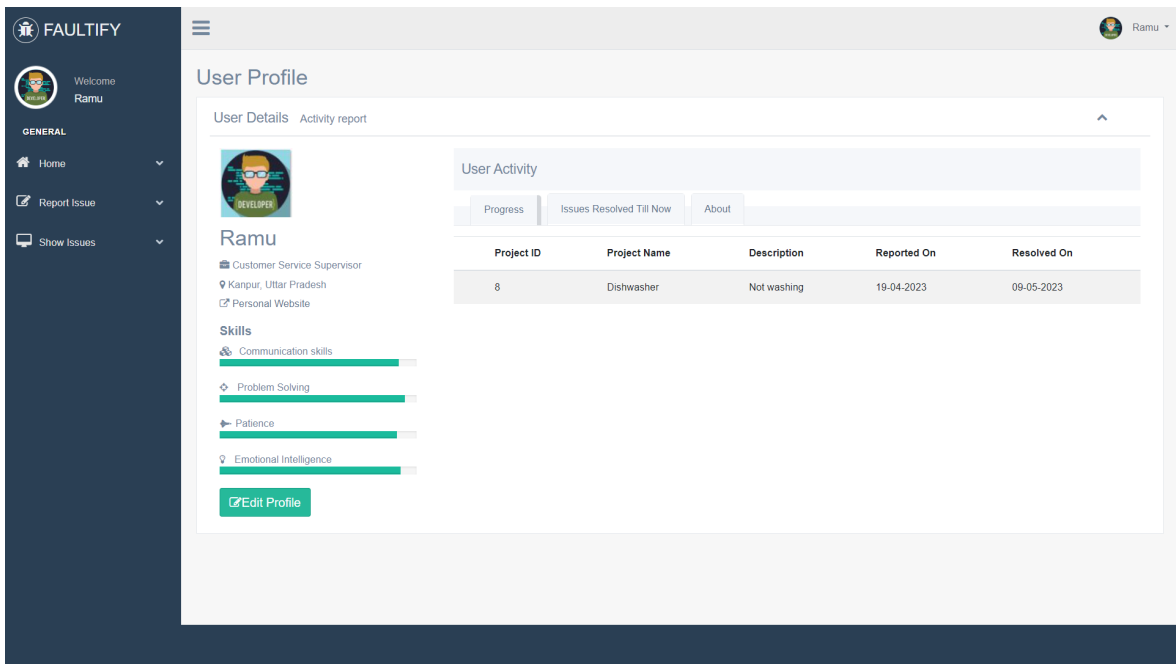


Fig 24: Employee Profile

And can also check the progress which shows a bar plot that shows the number of issues resolved by the employee in each month and the plot grow dynamically with the issues being solved by that employee whose profile is been seen.

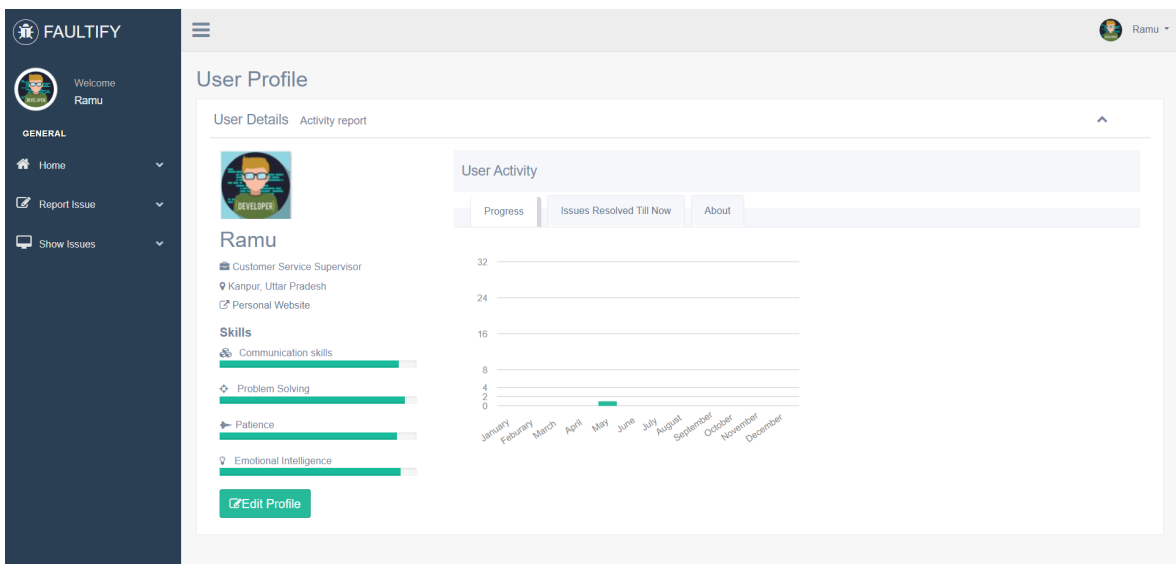


Fig 25: Progress of an Employee

Admin can also check every employee's profile, check unassigned employees and assigned employees, check resolved issues and unassigned issues.

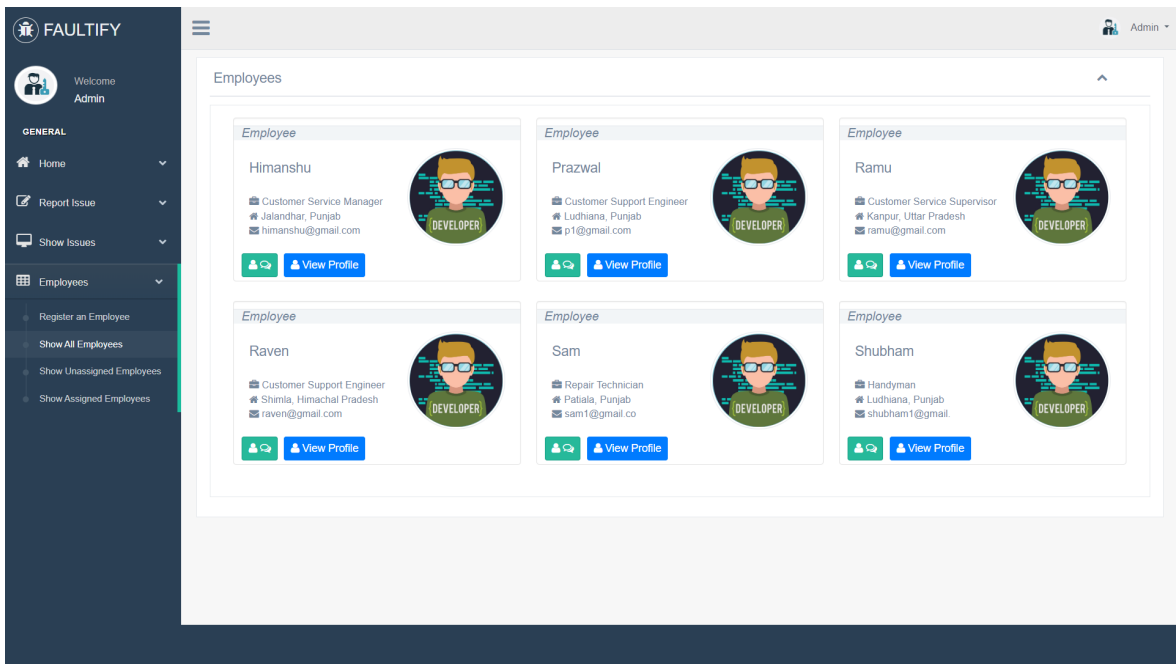


Fig 26: Show All Employees

The application allows users to create, track, and manage issues, as well as assign them to other users for resolution. It also includes a range of features that enable users to customize and filter their view of the issues, such as search and sort options.

CHAPTER 05:

CONCLUSION AND FUTURE WORK

Faultify is a web-based application basically an issue tracking application that is built using Spring MVC architecture. The issue tracking application will be a useful tool for organisations to monitor internal concerns, quickly address them, and enhance performance. The issue tracking application will go through extensive testing and quality assurance procedures before being released to make sure it is efficient.

The program will also be made to be versatile, scalable, and adjustable so that businesses may adapt it to their own requirements. A committed group of developers and project managers who will collaborate closely with stakeholders during the development process will be needed for the project. Overall, by expediting the problem-resolution process, the issue tracking application project has the potential to greatly increase organizational efficiency and productivity.

This is so that new features may be added or current ones can be changed as needed. Spring MVC enables modular and extensible programming.

The application can effectively save and retrieve data from a MySQL database thanks to the strong data access layer provided by Hibernate and JPA. Overall, a Spring MVC-based issue tracking application is a useful tool for any organisation wishing to optimise its issue management procedures and guarantee the speedy resolution of software bugs.

5.1 Future Scope

Integration with other tools and services: To offer a more complete solution for software development teams, the issue tracking application can be integrated with other tools and services like project management software, testing tools, continuous integration tools, and version control systems. The development process may be streamlined, and teamwork may be enhanced, with the aid of this integration. Custom reports and dashboards are only two examples of the additional reporting and analytics features that may be added to the

programme. This will make it easier to assess the performance of the development team, spot process bottlenecks, and gauge how well the issue tracking system is working.

Automated testing and quality assurance technologies may be coupled with the issue tracking application to guarantee that all issues are addressed effectively and accurately. Automated testing can also aid in problem detection and early intervention.

Accessibility on mobile devices: The programme may be made mobile-friendly so users can create, view, and amend issues while on the move. Team members will be better able to respond swiftly to problems as they emerge, which will increase productivity and responsiveness. Artificial intelligence and machine learning are two technologies that can be used to increase the precision and efficiency of problem tracking. Simple questions may be answered by chatbots powered by AI, which lightens the strain on the customer service personnel and raises customer satisfaction. Teams may proactively address possible issues before they arise by using machine learning algorithms to forecast the likelihood of issues based on historical data.

Improved security measures can be added to the programme to secure sensitive data and stop unauthorised access. Encryption, role-based access control, and multi-factor authentication are some examples of this.

Chat and messaging capabilities: Adding chat and messaging capabilities can enhance teamwork and communication by enabling participants to discuss and resolve problems more rapidly.

The issue tracking application built with Spring MVC, Hibernate, JPA, and MySQL offers a lot of room for growth in the future. The programme may be made a complete solution for software development teams by putting these upgrades and innovations into practice, which will boost efficiency, productivity, and cooperation.

5.2 Applications

There are many uses for an issue-tracking application built using Spring MVC, Hibernate, JPA, and MySQL across several businesses and sectors.

- The software development sector is one of the major industries where this kind of application is used. Software development teams can use this program to manage their projects and keep track of problems that occur throughout the development process. Team members may immediately identify and fix problems as a result, streamlining the workflow and lowering the time and effort needed to finish a project.
- In the customer service sector, this form of application is also used. This program may be used by customer care teams to handle and follow up on client complaints. This enables a better organized and efficient customer care procedure, resulting in quicker problem resolution and higher customer satisfaction.
- The healthcare sector may also make use of this kind of application. Using this tool, healthcare personnel may keep track of patient concerns and make sure they are promptly and effectively handled. This can enhance patient outcomes and lower the possibility of medical mistakes.
- The application can also be used to track equipment and also machine related problems and can also analyze the problem, maintenance and repair.
- Also, it can be used in the educational sector where admin can monitor any issues related to students which can help improve their problems.
- This application can also be used in the financial industry or sector where they can monitor any issues related to the industry.

REFERENCES

- [1] A. M. Majeed, A. Al-Qaysi, and H. M. Sabti, "A Web-Based Issue Tracking System," 2018 International Conference on Advanced Science and Engineering (ICOASE), Kuala Lumpur, Malaysia, 2018, pp. 1-6.
- [2] A. R. Ramdani and H. F. Kurniawan, "Design and Implementation of Web-Based Help Desk System Using PHP and MySQL" 2018 2nd International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2018, pp. 183-186.
- [3], "Design and Implementation of an Issue Tracking System for Healthcare," 2017 3rd International Conference on Information Management (ICIM), Chengdu, China, 2017, pp. 101-105.
- [4] A. Z. Moosa and N. F. Alhassen, "Development of Issue Tracking System Based on Web 2.0 Concept," 2018 3rd International Conference on Information Management (ICIM), Chengdu, China, 2018, pp. 188-191.
- [5] B. C. Patil, R. D. Mahajan, and S. K. Pawar, "Issue Tracking System Using Spring MVC and Hibernate," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016, pp. 1-5.
- [6] <https://www.baeldung.com/spring-mvc-tutorial>
- [7] <https://www.javatpoint.com/spring-mvc-tutorial>
- [8] <https://www.journaldev.com/3531/spring-mvc-hibernate-mysql-integration-crud-example-tutorial>
- [9] https://www.tutorialspoint.com/spring_boot/spring_boot_spring_data_jpa.htm
- [10] <https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>
- [11] <https://www.baeldung.com/hibernate-query-language>
- [12] https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm
- [13] <https://www.codejava.net/frameworks/hibernate/hibernate-one-to-many-relationship>
- [14] <https://www.baeldung.com/hibernate-one-to-many>
- [15] https://www.tutorialspoint.com/hibernate/hibernate_many_to_many_mapping.htm
- [16] <https://www.javatpoint.com/hibernate-many-to-many-mapping-using-annotation>
- [17] <https://www.baeldung.com/hibernate-cascade-types>
- [18] https://www.tutorialspoint.com/hibernate/hibernate_cascading.htm

PLAGIARISM REPORT

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: RAVEN MEHTA Department: CSE Enrolment No 191303

Contact No. 8091779004 E-mail. 191303@juitsolan.in

Name of the Supervisor: Prof. (Dr.) SHRUTI JAIN and Dr. Amol Vasudeva

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): FAULTIFY- AN ISSUE TRACKING APPLICATION USING SPRING MVC

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at5.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

nbv

ORIGINALITY REPORT

5%	5%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	3%
2	ir.juit.ac.in:8080 Internet Source	1%
3	Submitted to Arab Open University Student Paper	<1%
4	Submitted to Rochester Institute of Technology Student Paper	<1%
5	Submitted to Upper Iowa University Student Paper	<1%
6	Submitted to Washington University of Science and Technology Student Paper	<1%
7	norvegr.com Internet Source	<1%
8	upcommons.upc.edu Internet Source	<1%