

ZOPSMART WORKSPACE

PREPARED BY

Aditya Rathore (2018BTechCSE022)

FACULTY GUIDE

Mr. Gaurav Raj



Department of Computer Science & Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University, Jaipur

May 2022

ZOPSMART WORKSPACE

Practice School-2

Submitted in partial fulfilment of the requirements
for the degree of
Bachelor of Technology in Computer Science Engineering

By:

Aditya Rathore (2018BTechCSE022)

Faculty Guide:

Mr. Gaurav Raj



Department of Computer Science & Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University, Jaipur

May 2022

CERTIFICATE

This is to certify that the Practise School-2 project work entitled “**ZopSmart Workspace**” submitted by **Aditya Rathore (2018BtechCSE022)** towards the partial fulfilment of the requirements for the **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practise School-II examination.



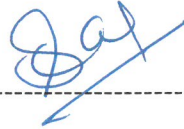
Mr. Gaurav Raj

Assistant Professor,

Department of Computer Science
Engineering

Institute of Engineering & Technology
(IET)

JK Lakshmipat University Jaipur



Dr. Sonal Jain

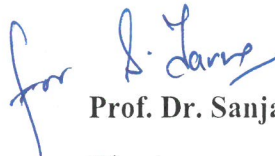
Co-Ordinator and Professor,

Department of Computer Science
Engineering

Institute of Engineering & Technology
(IET)

JK Lakshmipat University Jaipur

Date of Submission: 13th May 2022



Prof. Dr. Sanjay Goel

Director

Institute of Engineering & Technology

JK Lakshmipat University

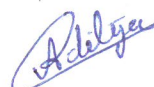
ACKNOWLEDGEMENTS

The internship opportunity I had with ZopSmart Technology Pvt. Ltd. was a great chance for learning and professional development. Therefore, I consider myself fortunate individual as I was provided with an opportunity to be part of ZopSmart. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

I perceive as this opportunity as a big milestone in my career development. I have taken efforts in completing this internship. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to express heartfelt gratitude to **Prof Dr. Sanjay Goel**, Director, Institute of Engineering & Technology for providing this opportunity to me. Also, I would like to thank my faculty guide **Mr. Gaurav Raj** for his constant support and guidance.

I am highly indebted to **ZopSmart Technology Pvt Ltd.** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express our gratitude towards our parents & the members of ZopSmart Technology Pvt Ltd. for their kind co-operation and encouragement which helped me in the completion of this project and internship. I would like to express my gratitude and thanks to **Mr. Vikash Kumar, Mr. Ujjawal Misra and Mr. Prashanth Kolakani** industry persons for giving me such attention and time.



Sincerely yours,
Aditya Rathore
(2018BtechCSE022)

ABSTRACT

In any organisation around the globe, it is extremely important to manage the workforce efficiently taking into consideration all the sector which could be benefitted because of well managed group of individuals. A well planned, properly executed and evaluated industrial working environment ensures professionalism, discipline and productivity within the organisation.

In today's fast-changing business environment, it's extremely important to be able to respond to client needs in the most effective and timely manner. To accomplish the mentioned goal, it is necessary to ensure progression in work with effective contribution of each individual in the team. Therefore, tracking contribution of each employee requires considerable attention. This allows to maintain active communication within the organization related to the project, contribution and goals.

The ZopSmart Workspace is an employee progress tracking web application with additional feature-set. The purpose of developing this application is to improvise the current practise of communicating the progress on the onshore and internal projects of the company with the help of Database management system concepts and computer software, fulfilling their requirement, so that the valuable data can be stored for easy accessing and manipulation. It is a web-based application developed using Java powered by Java based framework spring boot for handling APIs at backend and MySQL as database whereas the frontend is developed using React.

LIST OF FIGURES

Figure	Page No
Figure 1.1 ZopSmart Technology	1
Figure 1.2 Spring Boot	4
Figure 1.3 Docker Architecture	5
Figure 1.4 Spring Security Internal Flow	8
Figure 2.1 Use Case Diagram	11

LIST OF TABLES

Table	Page No
Table 4.1 Login Test	20
Table 4.2 Add Employee Test Version 1	21
Table 4.3 Add Employee Test Version 2	21
Table 4.4 Add Project Test Version 1	22
Table 4.5 Add Project Test Version 2	22
Table 4.6 Add Employee to Project Test Version 1	23
Table 4.7 Add Employee to Project Test Version 2	23
Table 4.8 Assign Role to Employee Test Version 1	24
Table 4.9 Assign Role to Employee Test Version 2	24
Table 4.10 Add Update Test Version 1	25
Table 4.11 Add Update Test Version 2	25

LIST OF ACRONYMS

RAD: Rapid Application Development

XML: Extensible Markup Language

HTTP: Hypertext Transfer Protocol

SCM: Source Code Management

CONTENTS

<i>CERTIFICATE</i>	<i>iii</i>
<i>ACKNOWLEDGEMENT</i>	<i>iv</i>
<i>ABSTRACT</i>	<i>v</i>
<i>LIST OF FIGURES</i>	<i>vi</i>
<i>LIST OF TABLES</i>	<i>vii</i>
<i>LIST OF ACRONYMS</i>	<i>viii</i>

CHAPTER 1: INTRODUCTION 1-8

1.1 About ZopSmart Technology	1
1.2 General Introduction	1
1.3 ZopSmart Workspace	2
1.4 Objectives	2
1.5 Modules of Proposed System	3
1.5.1 Admin	3
1.5.2 Employee	3
1.6 Technology Used	4
1.7 Special Technical Terms	4
1.7.1 React	4
1.7.2 Spring Boot	4
1.7.3 Docker	5
1.7.4 Github	6
1.7.5 Github Actions	6
1.7.6 Database Migrations	7
1.7.7 Unit Testing	7
1.7.8 Flyway	7
1.7.9 Spring Security	7

CHAPTER 2: REQUIREMENT ENGINEERING 9-17

2.1 Problem Statement	9
2.2 Requirement Elicitation	9
2.2.1 Requirement Identified	9
2.2.2 Use Cases	10
2.3 Use Case Approach	11
2.3.1 Use Case Description	11

CHAPTER 3: SOFTWARE DESIGN	18-19
3.1 Design Strategy	18
3.1.1 Function Oriented Design	18
3.2 Feasibility Study	18
3.2.1 Technical Feasibility	18
3.2.2 Time Schedule Feasibility	19
3.2.3 Operational Feasibility	19
3.2.4 Implementation Feasibility	19
 CHAPTER 4: CONSTRUCTION	 20-25
4.1 General Introduction	20
4.2 Test Scenario	20
4.2.1 Login	20
4.2.2 Add Employee	21
4.2.3 Add Project	22
4.2.4 Add Employee to Project	23
4.2.5 Assign Role to Employee	24
4.2.6 Add Update	25
CHAPTER 5: CONCLUSION	26
REFERENCES	27

CHAPTER 1: INTRODUCTION

1.1 ABOUT ZOPSMART TECHNOLOGY

ZopSmart Technology is a cutting-edge retail technology company that provides you all the tools to build your e-commerce business. Whether you are a conventional retailer intending to build your omni-channel business or an online-only retailer intending to build your e-commerce business, we have a suite of products that will quickly and easily empower you to achieve your goals. They envision to be the #1 retail technology platform of the world. They have their expertise in the retail domain is built on our team's immense experience in the field of FMCG, retail, supply-chain and logistics. All our experience has been baked into business processes that are embedded in our products. Their solution has been built over 7 years and is handling thousands of transactions every day. Our suite of products is one of the most advanced, stable and scalable solutions available in the world of retail technology. Their solutions include e-commerce, m-Commerce, Digital Marketing, Customer Management, e-wallet and Order Management etc.



Figure 1.1 ZopSmart Technology

Using e-Commerce they give customers a world-class shopping experience - Intuitive search to quickly find products. Personalized product listing for ease of ordering. Self-service rescheduling and returns functionality for frictionless order experience. M-Commerce provide your customers a superlative shopping experience on mobile devices - Responsive website that adapts to all screen sizes. Native Android and iOS mobile apps for world-class experience. Persistent cart across devices for convenience. E-Wallet uses integrated wallet to build customer loyalty - Integrated wallet to enable quick and hassle-free checkout. Refunds and cashback to wallet to encourage repeat purchase. Wallet-credit as gift-cards for acquiring of new customers at low cost. Customer Management manage your online as well as offline customers easily and efficiently - Single interface to view all your customers with their contact details. Available details of each customer's purchase history, orders and loyalty points. Option to contact the customer from the interface by phone-call, SMS and e-mail.

1.2 GENERAL INTRODUCTION

In any organisation from around the world workforce management is an essential bracket of practises followed to optimize the productivity of the employees. Workforce management engross adequately projecting labour requirements along with forming and managing crew agenda to attain a particular goal on day-to-day basis. This practise assists organisation to

secure insights into the corporate business metrics to ensure a suitable amount of work for each individual in the team for effective contribution. To ensure that each individual contributes to the expected potential, it is necessary to monitor the participation over the time.

Additionally in any organisation around the globe, it is extremely important to manage the workforce efficiently taking into consideration all the sector which could be benefitted because of well managed group of individuals. A well planned, properly executed and evaluated industrial working environment ensures professionalism, discipline and productivity within the organisation.

However, as many organisations adopted the practise of working remotely or in hybrid mode post pandemic, the notion of workforce monitoring became mainstream to ensure productivity and fruitful contribution by the employees. Also, this maintains a sync within the team working on a project. Although the communication barrier between the managers and the employees is evident and it turns out to be increasingly difficult to monitor each employee as expected. Therefore, there is scope of a portal to communicate work status, progress and contribution.

1.3 ZOPSMART WORKSPACE

The ZopSmart Workspace is an employee progress tracking web application with additional feature-set. The purpose of developing this application is to improvise the current practise of communicating the progress on the onshore and internal projects of the company with the help of Database management system concepts and computer software, fulfilling their requirement, so that the valuable data can be stored for easy accessing and manipulation.

ZopSmart workspace allows the employees of the organisation to be added into the projects and post their updates during their course of working on that project. Moreover, it allows the managers to monitor the progress of a project and contributions of each employee in the team working on a particular project. It maintains a timeline of updates of each employee month wise. This timeline turns out to be handy for a quick review of an employee.

Thus, the ZopSmart Workspace enables a smooth review process for managers and employees by integrating the employee updates and evaluation into a single view. The system allows employees to post regular updates to the teams they are working in for their immediate supervisor to be in sync with their progress. This enables communication between the employee and their team.

It is a web-based application developed using Java powered by Java based framework spring boot for handling APIs at backend and MySQL as database whereas the frontend is developed using React.

To achieve the effective performance different Database strategies are implemented. User gets to access this website by simple registration process through organisation google email id. This allows to authenticate the user and fetch the details available on google account.

1.4 OBJECTIVES

The system aims at being the bridge between the communication gap between the managers and the employees.

- Maintain record of employees working on different project.
- Maintain record of updates posted by employee on a particular project.
- Maintain monthly timeline of employee updates.
- Enable smooth review process by integrating employee updates and evaluation.

- Shaping goals to fulfill organizational objectives.
- Defining the right expectations from employees and managers.
- Maintaining reliable and credible details.

1.5 MODULES OF PROPOSED SYSTEM

Currently the ZopSmart Workspace comprises of two modules namely Admin and Employee, both varying in the terms of authority to manipulate the information available on the workspace.

1.5.1 Admin

Admin is responsible for configuration and dependable functioning of the ZopSmart Workspace portal. Following are the sub modules of Admin:

- Add Employee:** Admin has the authority to add information related to employee working in the organization on the workspace.
- Add Project:** Admin has the authority to add information related to any project undertaken by the organization on the workspace.
- Add Employee to Project:** Admin has the authority to add employee to a particular project which is assigned to the employee.
- Add Role:** Admin has the authority to add role to the workspace.
- Add Role of Employee:** Admin has the authority to assign a role to employee. By default, a new user will have the role of an employee.
- Remove Employee:** Admin has the authority to remove an employee from the workspace.

1.5.2 Employee

Employees are the real user of the workspace to communicate their contribution to project over the time.

- View Hierarchy:** Employee added to the workspace are authorized to view the hierarchy of the organization.
- Post and Edit Updates:** Employees on the workspace are authorized to post and edit updated related to their contribution in the projects assigned to them.
- View Timeline:** Employees on the workspace are authorized to view the monthly timeline of their updates.
- View Projects:** Employees on the workspace are authorized to view the projects undertaken by the organization.
- Apply for Leaves:** Employees on the workspace are authorized to apply for leaves on the workspace.

1.6 TECHNOLOGY USED

The structure of frontend of the project is implemented using REACT and designed using Cascading Style Sheets (CSS) and JavaScript. REACT provides the basic structure of the web application and provide dynamic functionalities as well as improve the responsiveness to the application. CSS gives the required visuals as per design sheets.

The Backend logic was done using Java powered by spring boot framework. For every use case REST APIs were created to hit request and fetch response on the server. Additionally, spring security was used for authentication and role-based authorization on the web application.

Operating System – Windows, Linux and MacOS

Frontend – REACT, JavaScript and CSS

Backend – Java

Database – MySQL

Framework – Spring Boot

1.7 SPECIAL TECHNICAL TERMS

Object oriented design pattern was followed in the making of this project so that whenever there was a need to fetch something from the database, a method could be called which would execute the required query and return the result, this reduced the code repetition and increased the code reusability.

1.7.1 REACT

React is an efficient, declarative and flexible JavaScript library used for building user interfaces. It emphasises on creating user interfaces which are declarative using component driven concept. It is utilized for managing the view layer and extensively used in web applications and mobile apps. React is often perplexed as a framework however, it is specifically a library. The clarification regarding previous statement is that React only handles rendering of the user interface and reserve several stuffs at the discretion of projects.

Following are the benefits of using React:

- Development process is fast.
- Components support
- SEO friendly
- One-way data binding

1.7.2 Spring Boot

Spring boot is a Java-based open-source framework majorly used to build microservices. It is built on top of the already existing framework entitles Spring. It imparts a simpler and quick way to configure, set up and run web based and desktop applications. Spring boot is a module of Spring which offers RAD feature to the Spring framework. It is utilized to implement stand-alone applications based on spring with minimal spring configuration.

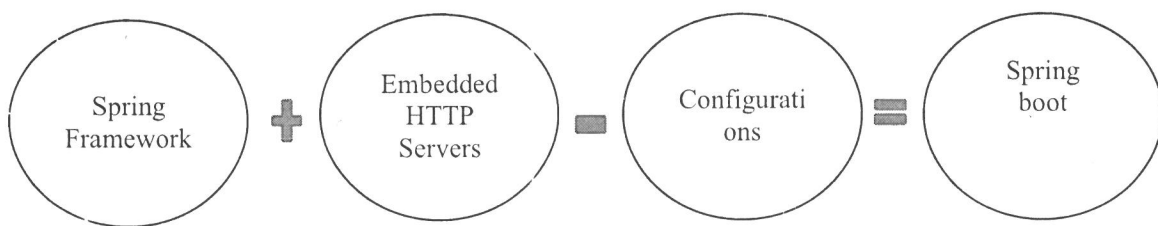


Figure 1.2 Spring Boot

In a nutshell, spring boot is a combination of Spring framework and Embedded servers. As described in the Figure 1.1 spring boot does not require XML configurations as deployment descriptors. It employs convention over configuration paradigm of software design which reduces the efforts of the software developer.

Following are the strong point describing the perks of using spring boot:

- It uses dependency injection approach.
- It comprises of capabilities of handling database transactions.

- It streamlines integration with other frameworks based on java such as hibernate/JPA ORM, struts, etc.
- It boils down the development time and capital of the application.

1.7.3 Docker

Docker is a centralized open platform for shipping, deploying and running software application. Docker allows to isolate the application from local infrastructure making it feasible to deliver software rapidly. In this way a developer can manage infrastructure in the similar fashion to that of managing the application. By utilization of the methodologies of docker for testing, shipping and deploying code, any developer could significantly bring down the delay between implementing code and running the implemented code in production.

Prior to docker the major issue faced by most of the developers was that a particular piece of code runs successfully locally on developer's system but not on the user's system. To resolve this issue docker was introduced which terminates all the configurations and ship the applications into containers which could be deployed anywhere.

1.7.3.1 Docker Architecture

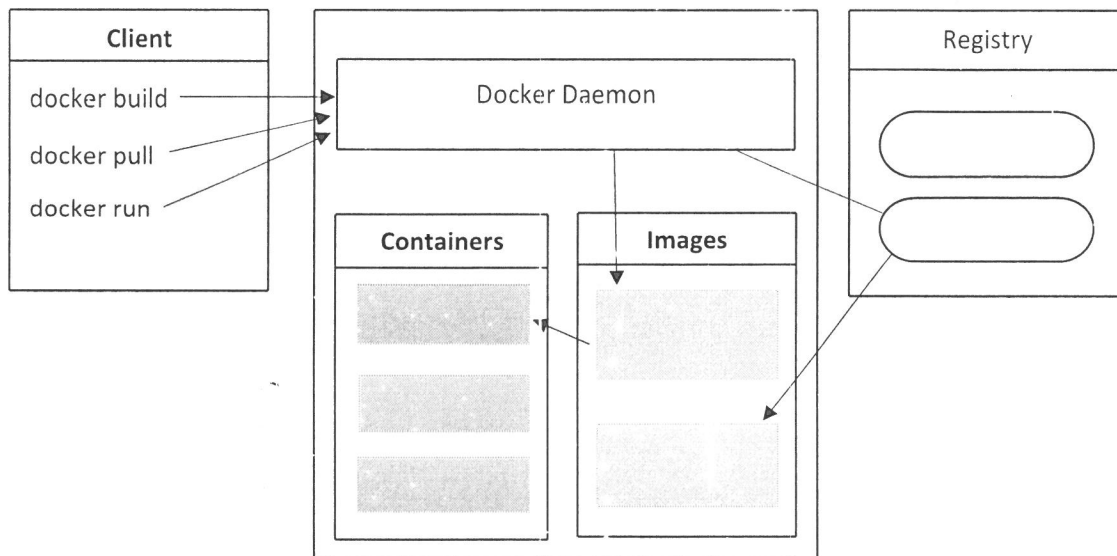


Figure 1.3 Docker Architecture

The architecture used by docker is client-server architecture where the client communicates with the docker daemon which carries out the hard labour of building, running and distributing the docker containers. Both the client and the daemon can run on the same system or can also a connection can be established between a remote daemon and client. The communication between docker client and daemon take place using REST API across a network interface or UNIX sockets.

The docker daemon tunes in for docker API demands and oversees docker objects such as networks, containers, volumes and images. A daemon is also capable of communicating with other daemons to oversee docker administrations.

The docker client is the fundamental way that numerous docker users' interface with docker. At the point when of using commands such as docker run, the client sends these commands to dockerd which executes them. The Docker client can speak with more than one daemon.

Docker registry comprises of the docker images. Docker Hub is one such community registry that anybody can utilize, and docker is designed to search for images on docker hub as a matter of course. When the commands such as docker pull or docker run, the expected images are pulled from designed registry whereas the command docker push is used to push the docker image to configured registry.

1.7.4 GitHub

GitHub is an immense platform for code hosting. It supports version controlling and collaboration and allows developers to work together on projects. It offers both distributed version control and SCM functionality of Git. It also facilitates collaboration features such as bug tracking, feature requests, task management for every project.

Essential components of the GitHub are:

- Repositories
- Branches
- Commits
- Pull Requests
- Git (the version control tool GitHub is built on)

GitHub service includes access controls as well as collaboration features like task management, repository hosting, and team management. The key benefits of GitHub are as follows.

- It is easy to contribute to open-source projects via GitHub.
- It helps to create an excellent document.
- You can attract the recruiter by showing off your work. If you have a profile on GitHub, you will have a higher chance of being recruited.
- It allows your work to get out there in front of the public.
- You can track changes in your code across versions.

1.7.5 GitHub Actions

With the help of GitHub actions developers can create a software development lifecycle workflow for their projects which allows to automate build, test and deployment pipeline.

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline. You can create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.

All GitHub Actions automations are handled via workflows, which are YAML files placed under the .github/workflows directory in a repository that define automated processes

Every workflow consists of several different core concepts. These include:

- **Events:** Events are defined triggers that kick off a workflow. They can be configured to look for one or more triggers and qualified as 4 needed by a developer. They can also be set to run on specific coding branches within a given repository on GitHub. [Learn more about what events can trigger a workflow.]
- **Jobs:** Jobs are a set of steps that execute on the same runner. Each run in its own VM and parallel to other jobs, unless otherwise specified.
- **Steps:** Steps are individual tasks that run commands in a job. These can be an action or a shell command. All steps in a job execute on the same runner.
- **Actions:** An action is a command that's executed on a runner—and the core element of GitHub Actions, which is named after it.
- **Runners:** A runner is a GitHub Actions server. It listens for available jobs, runs each in parallel, and reports back progress, logs and results. Each runner can be hosted by

GitHub or self-hosted on a localized server. GitHub Hosted runners are based on Ubuntu Linux, Windows, and macOS.

1.7.6 Database Migrations

It is like version control for database which keeps snapshot of every change in the database scripts. Along with the changing requirements, the schema keeps evolving. Migrations supports database schema to transition from current state to latest state. The purpose of migration to make changes in database repeatable, shareable, and testable without loss of data.

1.7.7 Unit Testing

In this project unit testing was implemented using JUnit and Mockito. Unit test case helps to check whether every service/component of code is working as expected or not and the other benefit is that the business logic is changed (if we find other optimized method), after that change the service should still work in the expected way. Before deploying the code on server unit test cases are executed to ensure that every service is working as expected.

Unit testing is done for every API endpoint using Mockito to verify that all the endpoints match standards before deployment. This practice helps to ensure that the best product is deployed and reaches users.

1.7.8 Flyway

Flyway is an open-source tool, licensed under Apache License 2.0, that helps you implement automated and version-based database migrations. It allows you to define the required update operations in an SQL script or as Java code. You can then run the migration from a command line client or automatically as part of your build process or integrated into your Java application. The good thing about this process is that Flyway detects the required update operations and executes them. So, you don't need to know which SQL update statements need to be performed to update your current database. You and your co-workers just define the update operations to migrate the database from one version to the next. And Flyway detects the current version and performs the necessary update.

1.7.9 Spring Security

Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications.

Spring Security is a framework that focuses on providing both authentication and authorization to Java applications. Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements.

1.7.9.1 Internal Flow

- a) **Authentication:** It is about knowing the identity of the user who is using the application. If the identity is known then only the user should be able to use the application otherwise, we should not permit the people to use the application.
- b) **Authorization:** The authorization is next to authentication. It is about giving authority to the user to use specific components of the application. For example, a typical web application contains various components. Some components are common components which can be accessed by every user of any role who is logged in whereas some components are role specific.

To understand the internal flow of spring security let us consider Figure 1.3,

- Suppose an end user throws a request to access an API then authentication filter takes all the user information from credentials and creates an authentication object which

binds all the user related information. This authentication object is a base where all the validation of the user credentials takes place in further steps.

- As soon as authentication filter gets the authentication object, it passes the object to authentication manager. Authentication manager is a component which will find out a proper authentication provider for the request.
- Further the object is sent to authentication provider where all business logic is available for validation of the details.
- Authentication provider internally uses two components namely user details service and password encoder.
- Once all the authentication checks are done, we get the user roles. Then the flow takes a U-turn to reach authentication filter.
- When the flow reaches authentication filter back, then it will take that object and pass it to security context which is used by spring security internally to place all the available information into a container. In case the same request comes again, it will access that request without asking for username and password.

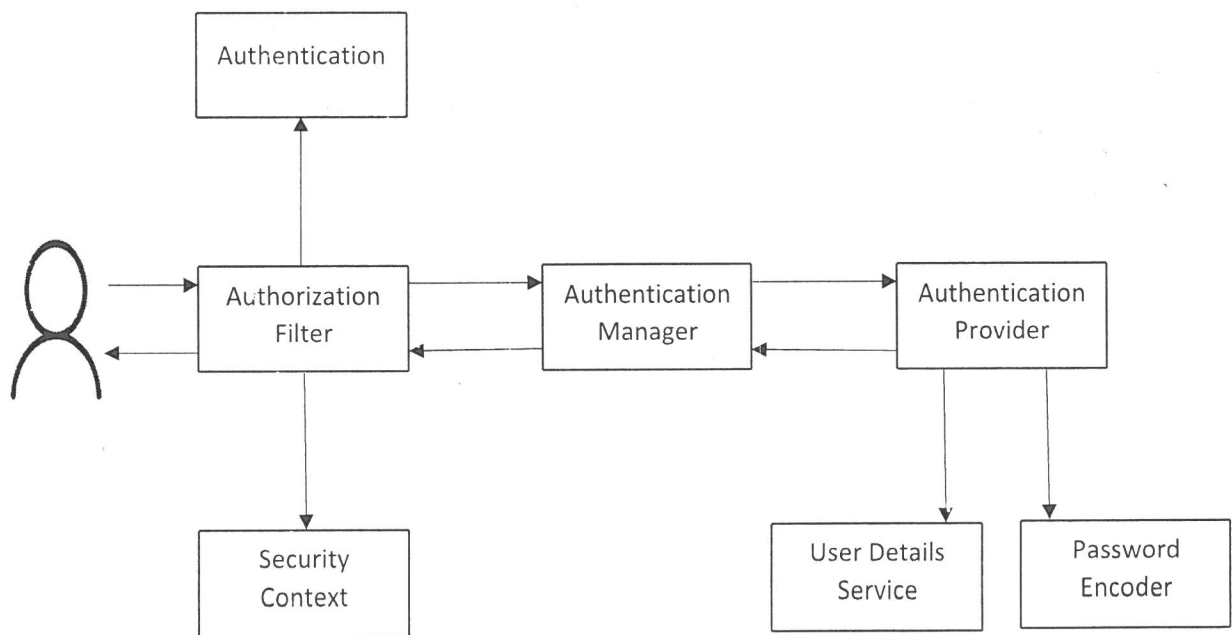


Figure 1.4 Spring Security Internal Flow

CHAPTER 2: REQUIREMENT ENGINEERING

2.1 PROBLEM STATEMENT

In today's advancing corporate world, workforce management is an important aspect of an organization's growth. It helps to ensure productivity and performance as per an individual's potential. It becomes necessary for an organisation to understand that the right number of employees are forecasted in the workflow at the right time and the right place.

However, keeping in mind, the drastic changes in the workforce over the past few years post pandemic, it becomes increasingly difficult to manage the workforce working from home or in hybrid mode. There is need of proper communication within the organisation and the teams working on the same project. Therefore, maintaining records of employee's contribution to the project is necessary for streamlining the goals and deadlines. There is requirement of a one stop portal for employees and managers to post their daily updates to track statuses of projects, teams and individuals within an organisation.

2.2 REQUIREMENT ELICITATION

2.2.1 Requirements Identified

a) Frontend

The frontend of the web-based application is build using React, JavaScript and CSS.

b) Backend

The backend of the workspace is implemented in Java powered by spring boot framework and the language used for database maintenance is MySQL.

c) Accessibility to updating

The workspace aims to ensures tracking of statuses of projects, teams and individuals by maintaining an engaging working community. There are a number of functionalities provided for best user experience. The user is allowed use relevant functionalities but do not have access to update any kind of information related to employees, projects, roles and their features

d) Timeline

ZopSmart workspace maintains a monthly timeline of updates posted by the employee regarding his/her contribution in the project assigned.

e) Hierarchy

The ZopSmart workspace allows every employee to take a look at the organisational hierarchy for better understanding of the communication paths between employees, departments and sections of the organisation.

f) Authentication

ZopSmart workspace allows new user to join the platform if the email id is of google and verified. Additionally, the domain name comprises of the organisation's domain name.

g) Role Based Authorization

Users of ZopSmart workspace are authorized to use a certain set of features based on their roles on the platform. For example, Admin has the authority to manipulate the details related to projects, employees and roles on the workspace whereas user is not authorized to modify any details related to any of these.

2.2.2 Use Cases

Functionalities accessible to Admin and User:

2.2.2.1 View Dashboard: Both the admin and the user can view the information available on the dashboard related to all the projects.

2.2.2.2 Login: Both user and admin will be able to login to the portal after creating account and will be able to use features accessible to them.

2.2.2.3 View Hierarchy: Hierarchy

The ZopSmart workspace allows every employee to take a look at the organisational hierarchy for better understanding of the communication paths between employees, departments and sections of the organisation.

2.2.2.4 Post Updates:

Every user on the workspace could post the updates related to his contribution to a particular project.

2.2.2.5 Timeline:

ZopSmart workspace maintains a monthly timeline of updates posted by the employee regarding his/her contribution in the project assigned.

2.2.2.6 Apply for Leaves:

All the employees working in the organisation should be allowed to apply for leaves on the workspace and the details of their leaves should be visible on project's calendar and summary.

Functionalities accessible to Admin:

2.2.2.6 Data Entry: This functionality allows admin to add, update or delete data from the workspace. This functionality ensures effective management of data.

2.2.2.7 Employees: Admin could add, update or delete information regarding employees enrolled in the workspace. Whereas other employees will be able to view the details but are not allowed to manipulate them.

2.2.2.8 Projects: Admin could add, update or delete information regarding project on the workspace.

2.2.2.9 Assign Roles: By default, the role is set to employee for every new user who join the workspace but the admin has the authority to assign any other role to the employees.

2.2.2.10 Add Employee to Project:

Admin has the authority to assign a project undertaken by the organisation to any particular employee.

2.3 USE CASE APPROACH

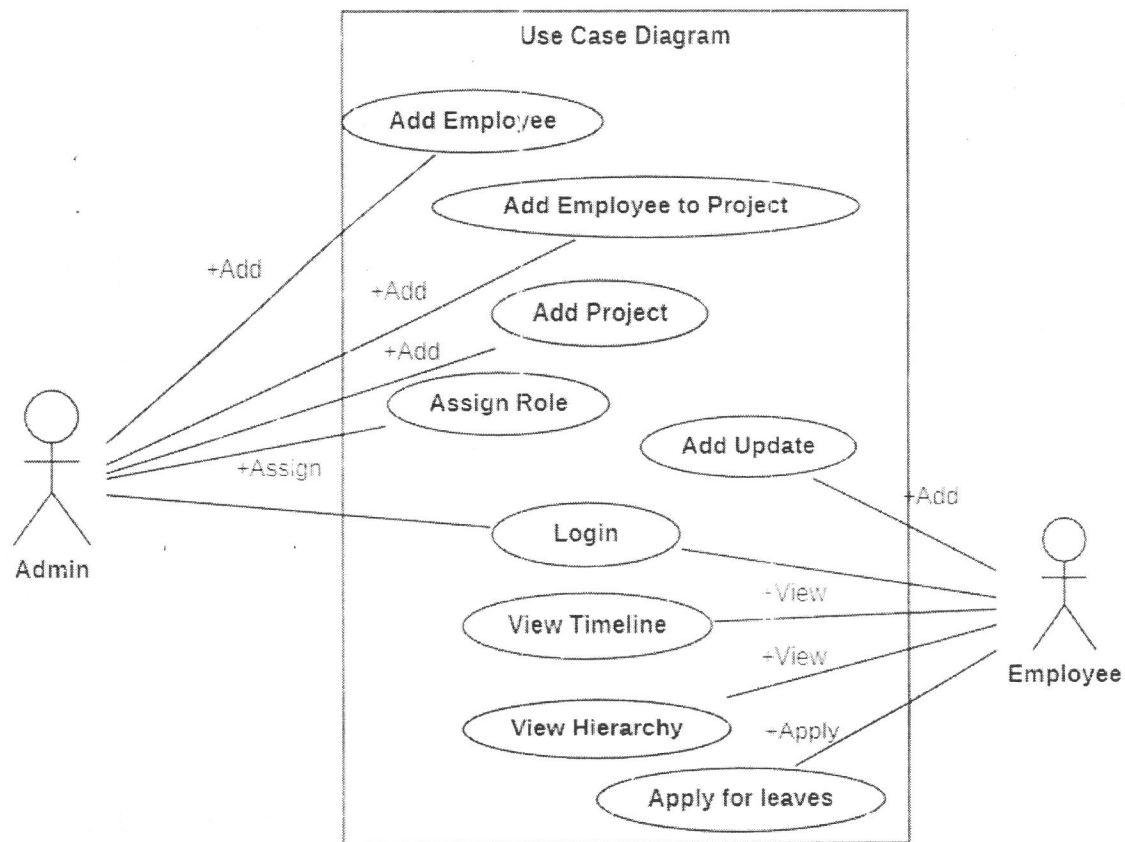


Figure 2.1 Use Case Diagram

2.3.1 Use Case Description

2.3.1.1 Login

2.3.1.1.1 Introduction

This use case describes the procedure for signing into the ZopSmart workspace based on privileges.

- Admin (Add employee, add project, add roles and assign project to employee)
- Employee (View hierarchy, View timeline, post updates and apply for leaves)

2.3.1.1.2 Actors

Employee and Admin

2.3.1.1.3 Pre-Condition

None

2.3.1.1.4 Post-Condition

If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

2.3.1.1.5 Flow of Events

2.3.1.1.5.1 Basic Flow

This flow begins when actor wants to log into the workspace.

- The system asks for email of the user
- The system verifies the email and its domain name to authenticate user.
- If the actor is registered user, he will get access to all the functionalities of the workspace.
- If the actor is not registered, the login flow checks the domain of the email and if it matches with the organization's domain then the user is added with default details.

2.3.1.1.5.2 Alternate Flow

If system receives invalid credentials, then an error message is displayed and the use case ends.

2.3.1.1.6 Special Requirements

None

2.3.1.1.7 Related Use Case

None

2.3.1.2 Add Employee

2.3.1.2.1 Introduction

This use case describes the procedure to post information of an employee on the workspace for the rest of community to get knowledge about it.

2.3.1.2.2 Actor

Admin

2.3.1.2.3 Pre-Condition

The actor must be admin because the employee doesn't have the authority to add employee.

2.3.1.2.4 Post-Condition

If the use case is successful, the employee information posted by the admin will be floated on the workspace for other to view.

2.3.1.2.5 Flow of Events

2.3.1.2.5.1 Basic Flow

- This use case begins when the actor wants to post information of an employee on the workspace.
- The actor fills the necessary details about the employee.
- After the submission of details, the employee could be added to projects.

2.3.1.2.5.2 Alternate Flow

- If the actor is not admin, then this functionality will not be accessible.

2.3.1.2.6 Special Requirement

Actor must be admin.

2.3.1.2.7 Related Use Case

None

2.3.1.3 Add Project

2.3.1.3.1 Introduction

This use case describes the procedure to post information of a project undertaken by the organisation on the workspace for the rest of community to get knowledge about it.

2.3.1.3.2 Actor

Admin

2.3.1.3.3 Pre-Condition

The actor must be admin because the employees doesn't have the authority to add project.

2.3.1.3.4 Post-Condition

If the use case is successful, the project information posted by the admin will be floated on the dashboard of the workspace.

2.3.1.3.5 Flow of Events

2.3.1.3.5.1 Basic Flow

- This use case begins when the actor wants to post information of any project undertaken by the organization on the workspace.
- The actor fills the necessary details about the project.
- After the submission of details, the project could be added to dashboard.
- Moreover, project could also be assigned to employees.

2.3.1.3.5.2 Alternate Flow

- If the actor is not admin, then this functionality will not be accessible.

2.3.1.3.6 Special Requirement

Actor must be admin.

2.3.1.3.7 Related Use Case

None

2.3.1.4 Add Role

2.3.1.4.1 Introduction

This use case describes the procedure to post information of a role on the workspace for managing the authorities.

2.3.1.4.2 Actor

Admin

2.3.1.4.3 Pre-Condition

The actor must be admin because the employees doesn't have the authority to add role.

2.3.1.4.4 Post-Condition

If the use case is successful, the role information posted by the admin will be added on the workspace.

2.3.1.4.5 Flow of Events

2.3.1.4.5.1 Basic Flow

- This use case begins when the actor wants to post information of any role introduced on the workspace to manage authorities for features.
- The actor fills the necessary details about the role.
- After the submission of details, the role could be added to the workspace.
- Moreover, role could also be assigned to employees.

2.3.1.4.5.2 Alternate Flow

- If the actor is not admin, then this functionality will not be accessible.

2.3.1.4.6 Special Requirement

Actor must be admin.

2.3.1.4.7 Related Use Case

None

2.3.1.5 Add Employee to Project

2.3.1.5.1 Introduction

This use case describes procedure to add an employee to any of the projects undertaken by the organisation.

2.3.1.5.2 Actor

Admin

2.3.1.5.3 Pre-Condition

The actor must be admin because the employees doesn't have the authority to assign projects to other employees.

2.3.1.5.4 Post-Condition

If the use case is successful, the information related to the new team member will be floated on workspace for other employees working on the project to know.

2.3.1.5.5 Flow of Events

2.3.1.5.5.1 Basic Flow

- This use case begins when the actor wants to add an employee to any of the projects undertaken by the organization.
- The actor fills the necessary details about the employee and project.
- The corresponding details will be saved in database.
- Details regarding new team member will be posted on the workspace.

2.3.1.5.5.2 Alternate Flow

None

2.3.1.5.6 Special Requirements

None

2.3.1.5.7 Related Use Case

None

2.3.1.6 Post Update

2.3.1.6.1 Introduction

This use case describes the procedure to post update regarding contribution to a project assigned to the actor on the workspace.

2.3.1.6.2 Actor

Employee

2.3.1.6.3 Pre-Condition

The actor must be employee because the admin is not assigned to any project.

2.3.1.6.4 Post-Condition

If the use case is successful, the updated posted by the employee will be available on the workspace.

2.3.1.6.5 Flow of Events

2.3.1.3.6.1 Basic Flow

- This use case begins when the actor wants to post an update of the contributions to any project.
- After the submission of details, the updated will be added to the timeline.

2.3.1.6.5.2 Alternate Flow

- If the actor is not part of any project, then the functionality to post update will not be displayed.

2.3.1.6.6 Special Requirement

Actor must be part of any one project.

2.3.1.6.7 Related Use Case

None

2.3.1.7 View Hierarchy

2.3.1.4.1 Introduction

This use case describes procedure to view hierarchy within the organisation to understand the communication paths.

2.3.1.7.2 Actor

Employee

2.3.1.7.3 Pre-Condition

None

2.3.1.7.4 Post-Condition

If the use case is successful, a hierarchy tree in the form of d3 organisational chart will be displayed with the feature to view upward as well as downward hierarchy.

2.3.1.7.5 Flow of Events

2.3.1.7.5.1 Basic Flow

- This use case begins when the actor wants to view the hierarchy of organization.
- Details from the database are fetched on the basis of ID of employee and of the person the employee is reporting to.
- D3 organizational chart is formed from the information fetched.

2.3.1.7.5.2 Alternate Flow

None

2.3.1.7.6 Special Requirements

None

2.3.1.7.7 Related Use Case

None

2.3.1.8 View Timeline

2.3.1.8.1 Introduction

This use case describes procedure to view timeline of updates posted by the actor to the projects assigned to him.

2.3.1.8.2 Actor

Employee

2.3.1.8.3 Pre-Condition

None

2.3.1.8.4 Post-Condition

If the use case is successful, a timeline of monthly updates is visible to take a look at all the updates related to different projects of which actor is part of.

2.3.1.8.5 Flow of Events

2.3.1.8.5.1 Basic Flow

- This use case begins when the actor wants to view the timeline of updates.
- Details from the database are fetched on the basis of ID of employee, start date, end date and page size.
- Monthly timeline is formed from the information fetched.

2.3.1.8.5.2 Alternate Flow

None

2.3.1.8.6 Special Requirements

None

2.3.1.8.7 Related Use Case

None

2.3.1.9 Apply for Leaves

2.3.1.9.1 Introduction

This use case describes procedure to apply for leaves on the workspace.

2.3.1.9.2 Actor

Employee

2.3.1.9.3 Pre-Condition

None

2.3.1.9.4 Post-Condition

If the use case is successful, information related to the employees leave will be visible in calendar and summary section of every project actor is part of.

2.3.1.9.5 Flow of Events

2.3.1.9.5.1 Basic Flow

- This use case begins when the actor wants apply for.
- Details regarding start date, end date and session is stored in database.
- Information of leave will be visible in calendar and summary section of every project actor is part of.

2.3.1.9.5.2 Alternate Flow

- If the leaves are not available then the use case terminates with an exception.

2.3.1.9.6 Special Requirements

The actor must have leaves available

2.3.1.9.7 Related Use Case

None

CHAPTER 3: SOFTWARE DESIGN

3.1 DESIGN STRATEGY

Software Design, is a process of implementing software solutions to one or more sets of problems. It transforms users and clients requirements into some suitable form, which helps programmer in coding and implementation.

Software design usually involves problem solving and planning a software solution. This includes both low level component and algorithm design, as well as a high-level architecture design.

Moreover, software design moves the concentration from problem domain to solution drama. It tries to specify how to fulfil the requirements mentioned in Software Requirement Specifications (SRS).

Software Design Strategy is a discipline that helps firms in deciding what to make, why to make it and how to innovate contextually, both immediately and over the long term. This process involves strategic design and interplay between design and business strategy.

The software design strategy is mainly about organizing design activities during the course of design.

3.1.1 Function oriented design

In function-oriented design, the system is comprised of many smaller sub-systems known as functions. These functions are capable of performing significant task in the system. The system is considered as top view of all functions.

In the function-oriented design, the system is designed according to the functionality it offers. The system is first widely observed and then each of its functions is observed to identify the sub-functions that it consists of that are responsible for performing that particular functionality.

In this type of design, there is a centralized function that consists of various functions. The software is designed in the same manner.

Considering the software for ZopSmart Workspace. Its main function is to maintain and store employee data. Now the sub functionalities which it can offer are the operations such as creating an engaging organisational network by providing interacting functionalities such as add employee, add project, view hierarchy, view timeline, etc. Now these functions further consist of the fundamental operations like login, register, assigning role to employee, etc.

In the same way in software, an element may be divided further into sub-elements for performing its inner functions and it may also act as a sub-element for some other module. However together they are responsible for providing the functionality that the entire software claims to offer.

3.2 Feasibility Study

3.2.1 Technical Feasibility

A technical feasibility study evaluates the details of how the developers intend to build a system or solution. Technical analysis evaluates technical merits of the system and at the same time collects additional information about performances, reliability, maintainability and productivity. In this section feasibility study of the project is analysed.

Basically, this study is to ensure that proposed system software is not a loss or burden to the user. It assesses the practicality of the proposed project. For this study, core knowledge of the system is required.

3.2.2 Time Schedule Feasibility

The Project has simple working and basic requirement can be satisfied with the allotted time period.

3.2.3 Operational Feasibility

The proposed system will increase the operational efficiency of all users. Therefore, the throughput of the system and reduction of time is as desired.

This system will ensure and satisfy the requirements identified by the requirement analysis phase of the system.

The API will be user-friendly and will work with any frontend system for the same definition i.e., it will be very useful and easy to operate.

3.2.4 Implementation Feasibility

User can directly access API with the Postman or any API testing software or also can create the frontend and API can be useful for the same.

Implementation will result in all the important information about all the available professionals and customers can hire any of them for their personal activities/needs.

CHAPTER 4: CONSTRUCTION

4.1 GENERAL INTRODUCTION

Software testing is a concept of determining the correctness of the software by taking into consideration all the attributes and assessing the implementation of components of the software to find bugs, errors or defects. Attributes of software are readability, scalability, portability, reusability and usability.

Software testing renders a standalone view and objective of the software and ensures accuracy of the software. It includes testing of all segments under the necessary administrations to affirm that if it is fulfilling the predetermined prerequisites. The process is additionally furnishing the customer with data about the nature of the product.

The manual testing technique is followed to test the ZopSmart Workspace. The process of testing software without using any automation tool is termed as manual testing. There are three types of manual testing namely white box testing, black box testing and grey box testing. The manual testing is done using functional unit testing approach which is a sub type of black box testing.

Functional testing emphasizes upon validation of the functionalities of the software by checking whether it is working as per the requirements specified. Each feature is tested by giving the input, determining the output and verifying the actual output with the expected output.

4.2 TEST SCENARIO

4.2.1 Login

- a) Test case 1: Valid email and password
- b) Test case 2: Invalid email and password
- c) Test case 3: Empty email and password field

Table 4.1 Login Test

Test Case Number	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC1	Check user login with valid credentials	Email: aditya.rathore@zopsmart.com Password: 12345	User should login into portal	As expected	Pass
TC2	Check user login with invalid credentials	Email: aditya.rathore@zopsmart.com Password: 123	User should not login into portal	As expected	Pass
TC3	Check user login with empty credentials	Email: Password:	User should not login into portal	As expected	Pass

4.2.2 Add Employee

- a) Test case 1: Input validation with invalid input
- b) Test case 2: Empty field
- c) Test case 3: Subordinate

Table 4.2 Add Employee Test Version 1

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on employee information with invalid input	Point out the error	As expected	Pass
TC2	Check employee information with empty fields	Point out the empty field	As expected	Pass
TC3	Check for invalid reporting to ID	Exception should be thrown	Not as expected	Fail

Table 4.3 Add Employee Test Version 2

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on employee information with invalid input	Point out the error	As expected	Pass
TC2	Check employee information with empty fields	Point out the empty field	As expected	Pass
TC3	Check for invalid reporting to ID	Exception should be thrown	As expected	Pass

4.2.3 Add Project

- a) Test case 1: Input validation with invalid input
- b) Test case 2: Empty field
- c) Test case 3: Subordinate

Table 4.4 Add Project Test Version 1

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on project information with invalid input	Point out the error	As expected	Pass
TC2	Check project information with empty fields	Point out the empty field	As expected	Pass
TC3	Check by entering already existing details	Exception should be thrown	Not as expected	Fail

Table 4.5 Add Project Test Version 2

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on project information with invalid input	Point out the error	As expected	Pass
TC2	Check project information with empty fields	Point out the empty field	As expected	Pass
TC3	Check by entering already existing details	Exception should be thrown	As expected	Pass

4.2.4 Add Employee to Project

- a) Test case 1: Input validation with invalid project ID
- b) Test case 2: Input validation with invalid employee ID
- c) Test case 3: Validation for employee already part of the project

Table 4.6 Add Employee to Project Test Version 1

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid project ID	Project not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for employee already part of the project	Employee already exist exception should be thrown	Not as expected	Fail

Table 4.7 Add Employee to Project Test Version 2

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid project ID	Project not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for employee already part of the project	Employee already exist exception should be thrown	As expected	Pass

4.2.5 Assign Role to Employee

- a) Test case 1: Input validation with invalid role ID
- b) Test case 2: Input validation with invalid employee ID
- c) Test case 3: Validation for role already assigned to employee

Table 4.8 Assign Role to Employee Test Version 1

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid role ID	Role not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for role already assigned to employee	Role already assigned exception should be thrown	Not as expected	Fail

Table 4.9 Assign Role to Employee Test Version 2

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid role ID	Project not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for role already assigned to employee	Role already assigned exception should be thrown	As expected	Pass

4.2.6 Add Update

- a) Test case 1: Input validation with invalid project ID
- b) Test case 2: Input validation with invalid employee ID
- c) Test case 3: Validation for empty remark

Table 4.10 Add Update Test Version 1

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid project ID	Project not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for empty remark	Empty remark exception should be thrown	Not as expected	Fail

Table 4.11 Add Update Test Version 2

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation with invalid project ID	Project not found exception should be thrown	As expected	Pass
TC2	Check input validation with invalid employee ID	Employee not found exception should be thrown	As expected	Pass
TC3	Validation for empty remark	Empty remark exception should be thrown	As expected	Pass

CHAPTER 5: CONCLUSION

In any organisation around the globe, the concept of workforce management differs in various aspects and over the time there has been a progressive awareness about the benefits of effective workforce management. The outcomes of this awareness are favourable in various aspects such as projecting labour requirement, managing crew agenda, employee evaluation, communication, etc. ZopSmart Workspace acts as one stop portal for achieving above mentioned organisational ambitions. This web application allows the employees of an organisation to understand the communication paths and concerned individual for every concerned individual directly associated with the nature of query.

Moreover, this web application allows the employees to communicate their contribution to a project, organisation and team over a period of time for records in the form of employee updates. These updates are stored in the form of a monthly timeline on the workspace. Thus, this web application acts as a user-friendly platform which is capable of providing various organisational insights such as hierarchy, update timeline and project related information.

REFERENCES

- 1) Baeldung., 2022. *Java, Spring and Web Development tutorials*. [online] Available at: <<https://www.baeldung.com/>> [Accessed 12 May 2022].
- 2) www.javatpoint.com. 2022. *Tutorials - Javatpoint*. [online] Available at: <<https://www.javatpoint.com/>> [Accessed 12 May 2022].
- 3) Docker. 2022. *Home - Docker*. [online] Available at: <<https://www.docker.com/>> [Accessed 12 May 2022].
- 4) GeeksforGeeks. 2022. *GeeksforGeeks | A computer science portal for geeks*. [online] Available at: <<https://www.geeksforgeeks.org/>> [Accessed 12 May 2022].
- 5) Stack Overflow. 2022. *Stack Overflow - Where Developers Learn, Share, & Build Careers*. [online] Available at: <<https://stackoverflow.com/>> [Accessed 12 May 2022].

