

Team Collaboration Management System

V.P.A.Iroshan

2020



Team Collaboration Management System

**A dissertation submitted for the Degree of
Master of Information Technology**

V.P.A.Iroshan

University of Colombo School of Computing

2020



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: **V.P.A Iroshan**

Registration Number: **2017/MIT/025**

Index Number: **17550252**

Signature:

Date:

This is to certify that this thesis is based on the work of Mr. **V.P. Amila Iroshan** under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: **Prof. G.K.A.Dias**

Signature:

Date:

Abstract

The team collaboration management system is online software solution package, which targeted the service oriented software company called Intervest Software Technologies (PVT) LTD.

The Intervest Software Technologies (PVT) LTD currently maintain their project details, defect/CR details and meeting scheduling information in manual way by using spread sheets. Because of the poorly maintained spread sheets, they faced more difficulties in day to day continuous defect/CR tracking works. As well as they have missed some important client meetings because of neglectful spread sheet handling. When it comes to team collaboration support tools, there are large variety of options to choose from. Some are open source and some are commercial. No doubt all these products offer some benefits.

This system intends to improve and automate their defect tracking, meeting scheduling and project management process by computerization. The system has followed MVC (Model View Controller) architecture and (OOP/OOD) object oriented programming/designing techniques. Unified Modelling Language (UML) was used for analysis and design. JAVA (JDK 1.8) language with Spring boot framework was used for implement the code base of this system. The Apache Tomcat 8.0 application server was used to deploy the web application archive file, meanwhile MySQL Server 5.5 was used as a database management system. Team collaboration system provides a centralized system and can be accessible by any authorized user and fulfills the purpose of the project. Team collaboration management system compatible with any web browsers including chrome 8+, Internet Explorer 6+ and Firefox 2+. After a critical analysis, the developed system can be considered as a acceptable project, which comply with all the expected requirements of our client as well as the standards of MIT degree programme.

Acknowledgement

I wish to express my gratitude to all who have assisted me to complete the final year project. First of all I would like to give my sincere gratitude to University of Colombo School of Computing (UCSC) academic and non academic staff who gave me a vast knowledge of information technology and facilities. Then I extend my gratitude to project supervisor Prof. G.K.A.Dias for invaluable guidance and impressive support given from beginning to end of my project works.

Then I offer my deepest gratitude to my client and project manager of Intervest Software Technologies, Mr. Harsha Rajapaksha for the guidance and assistance given during the whole life cycle of my project.

I wish to express my sincere gratitude to my dear colleges at MIT degree programme who shared their valuable ideas and knowledge and encouraged me throughout the programme.

Besides that specially I would like to thank staff of Intervest Software Technologies who were sacrificed their valuable time and precious support to make this effort success.

Eventually, I express my heartfelt gratitude to my family members and all other parties who directly and indirectly cooperate with me for successful this project

Table of Contents

Chapter 1	1
Introduction	1
1.1 Motivation.....	1
1.2 Objectives.....	2
1.3 Scope.....	2
Chapter 2	4
Background.....	4
2.1 Review Of Similar Systems	4
2.2 Feature Comparison of Similar Systems	5
1. Bugzilla.....	5
Advantages of Bugzilla.....	5
Disadvantages of Bugzilla	6
2. Jira	6
Advantages of Jira	6
Disadvantages of Jira.....	7
2.3 Implementation Tools and Technologies	7
Chapter 3	9
Analysis And Design.....	9
3.1 Introduction	9
3.2 Requirement Gathering Techniques	9
3.2.1 Discussions and Interviews	9
3.2.2 Provide Questionnaire.....	10
3.2.3 Study Similar Systems	10
3.2.4 Use Web Resources	10
3.3 Problem of existing process and propose solutions	10
3.4 Requirement Analysis	11
3.4.1 Actors of the System.....	11
3.5 Requirement for Proposed Solution.....	14
3.5.1 Functional Requirement	14
3.5.2 Non Functional Requirement	15

3.6 Proposed architecture for the system	16
3.7 Design Techniques.....	18
3.8 Propose System Design.....	19
3.9 Database design	21
3.10 Class diagram of the system	22
3.11 Data Flow Diagram(DFD)	23
Chapter 4.....	24
Implementation	24
4.1 Introduction	24
4.2 Development Tools And Techniques.....	24
4.2.1 Development Tools.....	24
□ MySQL Workbench 6.3.....	25
4.2.2 Development Technologies	25
4.3 Implementation Environment.....	27
4.3.1 Server Environment	27
4.3.2 Client Environment.....	27
4.4 Network Architecture	28
4.5 Project Structure	29
4.6 System User Interfaces	30
Chapter 5.....	34
User Evaluation And Testing.....	34
5.1 Introduction	34
5.2 Testing Procedure	34
5.3 Test Plan and Test Cases.....	35
5.3.1 Login Module	35
5.3.2 User Management Module	36
5.3.3 Project Management Module	37
5.3.4 Meeting Scheduling Module	38
5.4 User Acceptance Testing.....	40
Chapter 6.....	41
Conclusion.....	41

6.1 Introduction	41
6.2 Challenges Faced During the Project	41
6.3 Lesson Learned.....	42
6.4 Future Works.....	42
REFERENCES	43
APPENDIX A - SYSTEM DOCUMENTATION	44
Hardware.....	44
Software.....	44
SETUP	45
APPENDIX B - DESIGN DOCUMENTATION.....	46
Use case diagrams and descriptions	46
APPENDIX C - CODE LISTING.....	49

Introduction

This chapter describes the problem which is tries to address ,motivation to come up with a solution ,objectives and the scope of the proposed solution.This proposed solution can be used to any client who is need to collaborate their team members in software industry.

1.1 Motivation

The target audience of this system is software service oriented company called Intervest Software Technologies(PVT) LTD.The main worth of any company is their customer base and customer satisfaction.Most of customers expect seamless,immediate and high end product or service from company.It's difficult task to achieve customer's satisfaction purely withing tight dead lines.Therefore this proposed system is focused on track the client's expectation, objections and satisfaction related to services which are provided by software companies.Organization must make sure to meets customer's expectation as fast as possible.

Through this proposed system, service providers can monitor and analyze client's expectations via centralized system.The team collaboration management system is good solution for effectively handle the business as well as make very strong client relationship.Most of the software companies in Sri lanka facing various troubles when handle the issues reported by client side and accept the change requests. Some of the Sri lankan software companies uses a spread sheet or word documents for track the reported issues and note down the change requests.Besides that clients are directly reporting their complaints to business analysis or a project manager over the phone or via a e-mail. However, It is a very inefficient way when maintaining such kind of documents manually and it's takes more time and effort for update the documents.Such kind of inefficient manual document management process helps to make some client dissatisfaction.In other hand if client need some urgent meeting

with development team and then client inform it to project manager. Then project manager noted down it and when time passes project manager may forget about client call/appointment. The proposed team collaboration management system facilitates schedule appointment and set appointment reminders in-between client and development team.

Another significant problem of software development is unawareness of the release versions among the team members. Most of the local software companies are not using proper tools to track the information about release versions and release branches. The team collaboration management system may provide secure repository to keep above details and it is good solution to overcome such kind of problems.

1.2 Objectives

- To register users with system and provide authorization to each on-board user roles.
- To create projects and resource allocation for each projects.
- To report defects and make change request(CR), projects wise.
- To create sub tasks with related to user stories and able to time estimation for sub tasks.
- To schedule meeting/appointment with team members via meeting scheduling module.
- To visualize details of release versions.

1.3 Scope

The team collaboration management system is a all in one software solution for software companies. Which consists user management module, project management module, defect tracking module and meeting scheduling module. The proposed system is web based solution for make strong team collaboration to Software Companies. The system mainly consists of three types of users. Those are Admin User (System Admin/Project Manager), System User (Developers /QA) and Client. The system facilitate register users with the system and provide authorization for each on-board user roles. The admin user able to create new projects with sub tasks.

Then admin user able to allocate resources (Developers and QA's) to particular project without any clashes. In addition to that administrator able to handle master data. Besides that the system provides defects and change requests (CR) reporting facility to clients. The reported defects and change requests called as tasks/tickets and the administrator role has the privilege to assign these tasks to team members who are work in related projects. Then team members able to estimate time for particular tasks and add some special comments if needed. These comments can review client as well as all team members. In addition to that client can review project status and track team member's performance. The system provide another special functionality which is client able to make some appointment/call with team members. The appointment scheduling task is a work flow and the system will remind each users who have special meeting or appointment with client.

In addition to that development team put daily product deployments to UAT servers for user acceptance testing. Through proposed system user can add details related to UAT deployment. Those are UAT deployment's date/time, version, deployed server details and information of fixed issues related with deployment. It is very easy to track information of UAT deployment to client. Besides that team members can aware about clients meetings/calls via meeting reminders.

The important part of the system is client can get reports under several categories. The client able to get progress reports via proposed software application. The progress reports depicts how many tasks completed or failure in estimated deadlines. In addition to that client able to get reports about resource allocation for each projects. This means client can track number of employees and their working time on particular projects. It really helps to identify what are the projects are in under over budget. By consider such kind of features the team collaboration management system will really helps to make business much profitable and efficient client handling purposes.

Background

This chapter describes the background of Team collaboration management system and their usage. There are many Team collaboration management systems available and each of them have their own advantages and disadvantages.

2.1 Review of Similar Systems

There are several commercial bug tracking systems available with less reliability and non user friendly functions. These non user friendly functions make some vague and uncomfortable to users. Most software companies used commercial bug tracking systems to report defects, assign defects and to obtain as quality indicators. With the revolutionize of IT industry, now a days many software solution providers desire store their digital data on cloud storage. As well as most of software products comply with online web based solutions and micro service architecture for gain reliable and get real time experience from any where in the world at any time. The main objectives of bug tracking systems are raise a defects with severity, able to identify who is raise a bug and to whom reporter assign to it and able to attached user stories related to defects.

The defect tracking system plays vital role in software companies specially when developed product is in testing phase. The bug tracking system working as a dedicated communication channel in between quality assurance and software development personnel. When consider the currently available open source and licence version bug tracking systems, it is really challenge when select the best bug tracking tool which fulfill the user's requirements. There are many expectations that need to be considered as follow.

Application Installation

When consider the application installation perspective any software able to install on any kind of operating system platform with quick steps. Besides that software might be uses very less hardware resource allocation such as RAM, memory and etc. But most of freeware bug tracking tools are not easy to configure and customize as user expects.

User friendliness of the application

The key feature of any kind of software is user friendliness. Even software products consists with latest technologies and very popular brand names some time that products may be fail under user friendliness. Most of the licence version software products are too complex to handle and work on it.

Centralized system

Another top prioritize feature of software product is that user can be access the software application at any time from any where. Software product should not be depend on operating system and device hardware.

2.2 Feature Comparison of Similar Systems

1. Bugzilla

Bugzilla was released under open source and licence version categories by mozilla.org. This defect tracking system provides robust, simple and reliable project management and defect tracking services.

Advantages of Bugzilla

1). Open source Product.

The open source version of bugzilla serves most valuable and precious features with comparing other open source product versions.

2). Well manageable charts and reports can be exported.

Bugzilla provides the detailed and advanced reports and graphs for defects analyzing purposes. As well as export reports with different formats.

3). Avoid adding some duplicate defects.

When user going to entering the bug in bugzilla, the system will automatically find out the existing similar defects and facilitate user to attached old tickets related to new once.

4).Move defects between cross versions.

It is very easy to exchange bugs in between two installed versions without additional works.

5).Excellent search capabilities with high security(Authorization and Authentication).

6).Provides attachments, Add comments and patch viewer facilities.

7).Support multiple authentication methods and multiple DB servers.

The bugzilla application fully compatible with standard DB servers such as MySQL,PostgreSQL and Oracle.In addition to that many authentication methods like LDAP and Oauth2.

Disadvantages of Bugzilla

1).Lack of user friendliness

User interfaces are too much sophisticated and some of functions are make confusion with users.

2).Less customization options.

Some of most necessary functions are not able to customize.

2. Jira

Jira provides multipurpose software solutions for CI/CD (continues integration and continues deployment) perspectives as well as project management and defect management purposes.Besides that it is capable of advanced report generation as well as managing of work flows. Jira is tool which highly recommended and compatible with agile software development process.In addition to that jira is a not concentrated to software development industry and it can apply as help desk and release/sprint planning tool.

This amazing and most popular tool was developed by Atlassian family and Jira has acquired 60% market share on bug tracking software category.

Advantages of Jira

1).It can applicable for any type of industry.

This is a multipurpose software and it's features easily adapted with any kind of organizational expectations.Some times this works as log manager,help desk,bug tracking tool and project management tool.By the way atlassian team provides more supportable plugins and therefore jira tool much compatible with other standard CI/CD tools.

2).Project task management and time tracking

This software tool facilitate the maintain worklog for each tickets.So project management personnel can use time tracking worklog of each ticket and cost estimate for each tickets or sprint.

3).Advanced report generation and roadmaps

Jira provides various types of reports under various formats.Like that sprint burn up/down charts,version report,sprint report,velocity charts and etc.Through the roadmaps we can sketch out the detail picture of communication plans with stake holders.

4) .Provides rich API's and plugins for developer tool integration.

5) .Advanced searching and sorting facilities using jira query language filters.

Disadvantages of Jira

1).Sophisticated user interfaces

Without best user experience and familiarity,It makes really challenge to fresh users.Due to the each interface has lots of small icons in every where.

2).Not an open source software

Jira is a commercial /licence version software tool. Pricing is based on the usage and feature wise as well as monthly subscription fee have to pay.

2.3 Implementation Tools and Technologies

Team collaboration management system will be used by IT professional who are familiar with software development life cycle.

The system is comply with object oriented programming and design concepts.Latest tools and technologies are used to implement the front end and back end features of the system.When concentrate of the JavaEE (JAVA Enterprise Edition) programming, the spring community provides precious services and frameworks for simplicity the java programming.Therefore team collaboration management system is implemented by using spring boot framework.Then MySql choosen as the relational database management system and combination of HTML5 and JQUERY used as front end technologies.The thymleaf sever side template engine used for rendering data from back ens services to front end.In addition to that team collaboration management system used bootstrap framework 4.0 as a CSS framework and it supported to make

responsive pages. Basically jquery validation plugin used for form validation and MVC used to develop architectural pattern with spring boot frameworks.

Spring MVC will add many object oriented rich feature to system development. In addition to that by using ajax call system allows retrieve data from backend without loading whole page. The Team collaboration management system consisted with user friendly view with single page application concept. Which means the system change only the page contents without whole page load when page redirecting.

Analysis And Design

3.1 Introduction

This chapter describes the analysis of the problem domain and system design. The requirement gathering and feasibility study is the initial and most significant phase of the software development life cycle. This is the phase where designer or architecture decide the scope of the project. In addition to that in this level the designer identified stakeholders, methods used for requirement gathering, and identified functional requirements and non functional requirements. Software design is an iterative process and this is the milestone where we make the blueprint of our system. In brief this is the phase where we translate gathered information into working solution. Through proper software architecture we can elaborate functional and non functional requirement of system and the output of the design phase is the system specification.

3.2 Requirement Gathering Techniques

Requirements were gathered using multiple approaches for better understanding. The client meetings/discussions and interviews, questionnaire, observations and existing document reviews are the most successful and effective approaches to achieve this. The system designers are able to efficiently grab the real requirements through the client site visits, direct or indirect interviews with identified stakeholders. In addition to that the relevant documents of existing systems are the best way observe the drawbacks of existing systems. The requirement gathering techniques of team collaboration system are as follow.

3.2.1 Discussions and Interviews

Interviews were arranged with individuals that are familiar with the context. Further clarification were made through discussions with QA personals. After the discussion

with different level of QA people,discovered that their requirements are different based on the designation.

3.2.2 Provide Questionnaire

Provide questionnaire to different level of IT professionals,to identify what are the main functionalities they are expect through proposed system and what are special obstacles they are facing when working on existing systems.

3.2.3 Study Similar Systems

Studying similar systems such as bugzilla and Jira helped to identify requirements which are not considered by the existing system.The identified optional requirements as follow,

1. Provide open source system with Aouth2 high security.
2. Integrate appointment/meeting scheduling module with bug tracking system.
3. Provide simple dashboard with information of release versions.
4. User friendly, simple user interface design.

3.2.4 Use Web Resources

Reading about similar defect/CR tracking management system helped to accelerate the requirement capturing process.

3.3 Problem of existing process and propose solutions

There are several problems were encountered in requirement gathering phase and proposed some solutions/alternatives for identified problems. Table 3.1 display Problems encountered and solutions.

Problem	Proposed Solution
All bugs(defects) and change requests are handled by manually using excel sheets.	System provide online facility to handle the all bugs/CR.
No any place to tracking the project details.	System automated to tracking project details.
Currently project manger is only responsible person to handle all bug and CR details.	By using user management module system facilitate the give management privileges to other users.
Scheduling/arranging the meeting in between clients and team members by project manger via spreadsheets.	System automates the meeting scheduling via meeting schedule module.

Table 3.1 Problem encountered and solutions

3.4 Requirement Analysis

During the requirement analysis process actors of the system was identified. Then the use cases were identified and documented. There after the necessary diagrams were drawn and functional and non functional requirements are listed.

3.4.1 Actors of the system

The actor in software modeling describes as a “user or any other system which interact with the system”. Following actors were identified during the analysis.

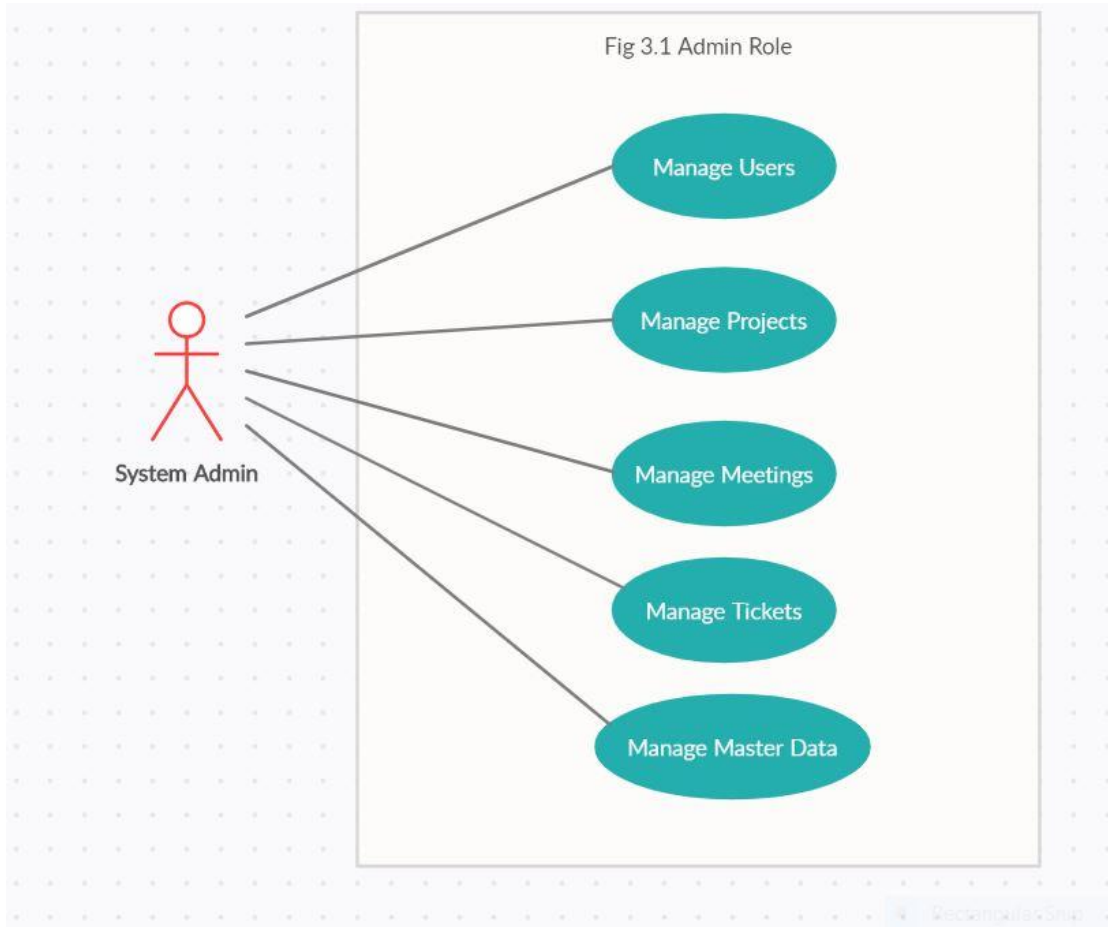


Fig . 3.1. Admin Role

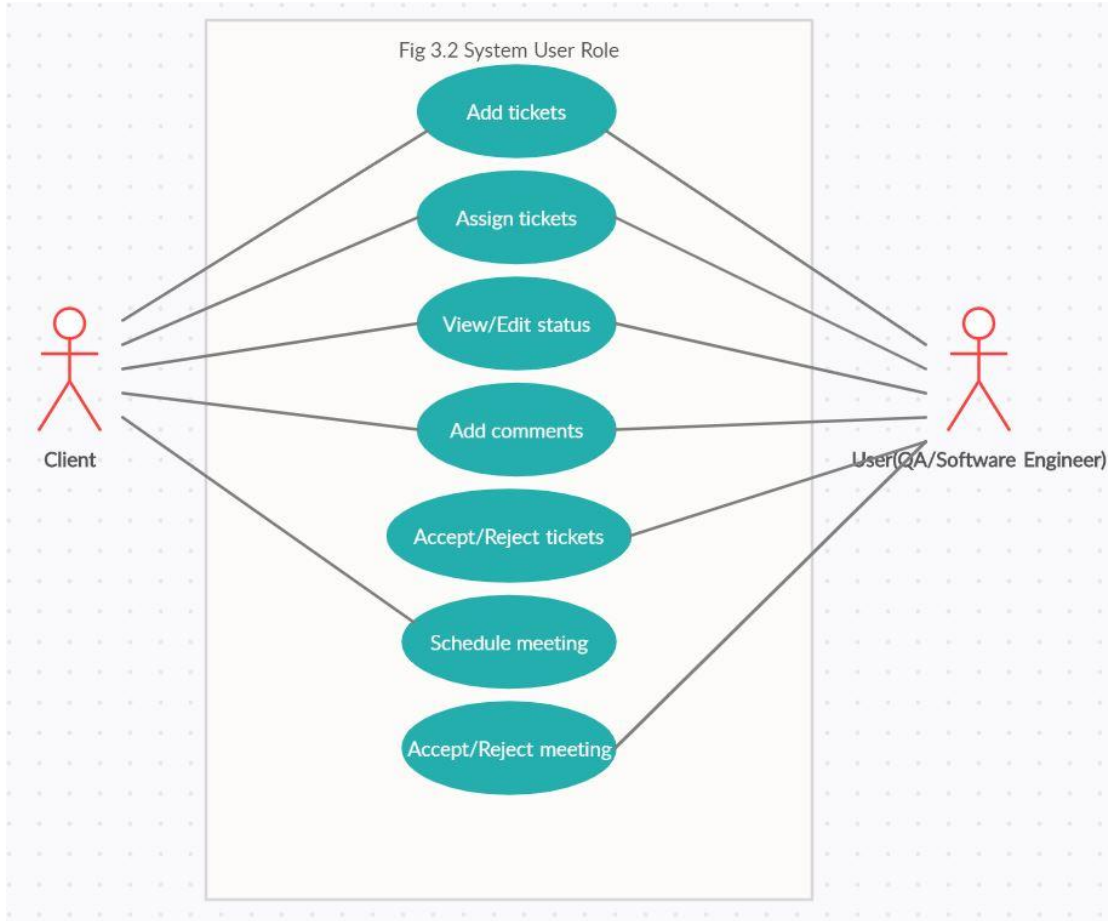


Fig.3.2. System User Role

3.5 Requirement for Proposed Solution.

3.5.1 Functional Requirement

Functional requirements describe the required behaviour of the system to be built or what the system supposed to do. The major requirements for the team collaboration management system were found via the questionnaire and interviews. Then studied and analyzed each requirements deeply and finally grabbed the feasible requirements with related to team collaboration management system.

- Generate defects and make change requests(CR).

The authorize users should be able to login to the system and report new defect or change request under particular project. User able to set the defect severity through the defect creation interface. Beside that user should able to assign created defect or change request to particular user who is responsible to fix it. Then user able to attached user stories or comment related defects.

- Create sub tasks for each change request and visualize sub task status.

If created change request is epic or some high weight user story, user able to divide it into sub tasks under main change request.

- Schedule meeting/appointment with team members via work flow.

The authorized users able to add new events to the system. The event types may be client meeting, interview, knowledge transfer session or etc. After create the event, user able to schedule sessions under particular event type. As well as user able to cancel the scheduled appointments and user can add existing users as a participants.

- Project management

User should be able to add new project to system. As well as user can be view project details and modify the project information. Then authorized users able to eliminate existing projects from the system.

- Visualize release version details.

Any user able to view the release version details via system. System shows the release version number, release date. Apart from that user able to view ticket numbers, which mean what are the defect and CR id's are concatenation with this release.

- User management

The team collaboration management system should allow system admin to add new users to system with different privileges. The system user has authority to change privileges of existing users and access denied of the on board users.

The purpose of Team collaboration management system is to improve the quality of software products and enhance the client relationship.

3.5.2 Non Functional Requirement

Non functional requirements are directly focus on the quality attributes of the system. If software product achieve hundred percent of functional requirements without fulfill the quality attributes, definitely that software product become unusable and unable to pass the user acceptance. The critical non functional requirements are portability, usability, responsiveness, security and etc.. Most of the functional requirements delivered through the system are highly depend on above quality attributes and quality services. Few of non functional requirements of the team collaboration management system are discussed under following categories. These requirements must be achieved at a system level rather than at a unit/component level.

Performance - System provides best response time and throughput.

Responsiveness - System used bootstrap as CSS framework. Therefore each and every interface in system are highly compatible with any screen size even mobile devices.

Availability - System should be available to access over the internet and should operate properly at IE, Chrome and Fire fox browsers and no any system downtime.

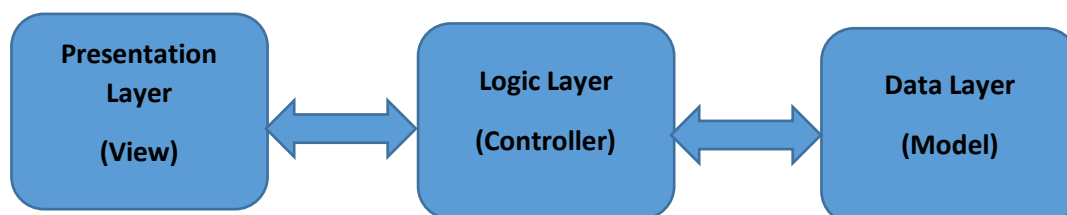
Security - Unauthorized users should not be able to access system or the production server. Used Aouth2 for handle authorization of the system.

Maintainability - System design should facilitate the implementation and deployment of requirement changes without complete re installation of the system.

3.6 Proposed architecture for the system

In order to achieve functional and non functional requirements, the team collaboration management system was developed using client server architecture. System was installed on central server and made accessible through a web browser. This enables users to access the system using various kinds of devices. As well as the application complied with responsive bootstrap template due to that user able to view the system interfaces on different devices. Software architecture concerns about separating system of interest into abstract sub systems. This enables the development of the system to be organized and traceable. In case of the proposed solution three tire architecture is selected and it is suitable for internet driven online web applications.

System was used MVC architecture as architectural style. The **Model, View, Controller** is a very popular and recommended software design pattern for developing web applications. The dispatcher servlet handles the all user requests and after receive particular request it chooses appropriate controller and then controller select the particular model(Entity). Then model class interact with database and send data to controller call again. After that controller class call the particular view and bind the date to it and render view page. This three interconnected layers, which helps to make clear and well organized code base for rapid web application development and dynamic interactivity with the database . As well as builder design pattern used for reduce the construct of complex objects. It allows to encapsulate the complex object creation logic. Fig 3.3 shows the interaction followed in the MVC framework.



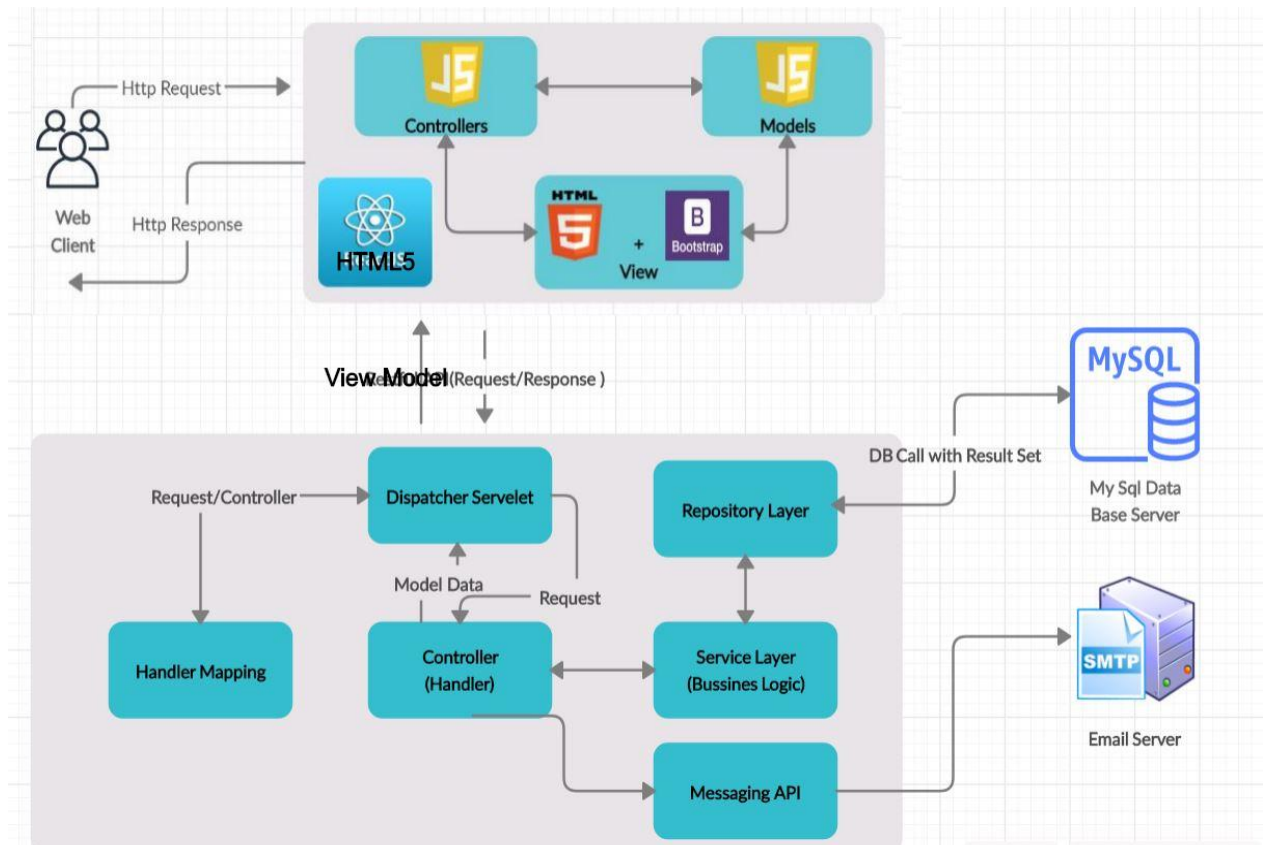


Fig. 3.3. Detail view of MVC usage

The primary actor is a web client who initiates the workflow of the system. After the web client triggers some event on the web interface, the HTTP request is sent to the dispatcher servlet. The dispatcher servlet is one of the most precious components of the MVC architecture, which is considered as a front controller. In the Spring framework, it cooperates with handler mapping and handler adapter to find the appropriate controller class. The `@Controller` and `@RequestMapping` annotations support this. The controller class consists of the business logic and is the responsible component for making interaction between the view and the model. After processing the incoming user request, the controller finds the best model class for the particular request. The Model class is a place where database-related logic and details of database entities are stored. In brief, the Model class is an object-oriented version of the relational table. After proceeding with all database-level transactions, the response is again redirected to the controller. Then the controller again redirects that response to the dispatcher servlet, and the dispatcher servlet forwards the request to the view resolver. The view resolver's responsibility is to find the particular view class. After finding the suitable view class, the view resolver forwards the request to the dispatcher servlet.

again. Then dispatcher servlet binds the model class data with view and renders the web page to the web client.

3.7 Design Techniques

Software design techniques are the strategies used to make a blueprint of the system architecture. There are several variants of software design techniques such as structured design, function-oriented design, object-oriented design, and rapid application development. The team collaboration management system is highly focused on the object-oriented design techniques, and its key factors are objects, classes, interfaces, inheritance, encapsulation, and polymorphism. Because of these key factors, the system will become easier to maintain due to the code reusability and recycling facility. The Unified Modelling Language (UML) is the foundation of the Object-Oriented Designing (OOD). UML provides some standards to model the system architecture via object-oriented design. The team collaboration management system used some of the structural and behavioral UML diagrams when designing the system.

Use-Case diagrams with some narratives – shows what the system needs to do.

- **Class diagrams** – shows the needed objects and relationships between them.

Data Flow diagrams (DFD) - shows the work flow of the system.

Java is the programming language which was used to implement the system code base, it does have a wide-range of object-oriented features. By using the Spring Boot framework with MVC architecture, it helps to make the development context reusable and much simpler.

3.8 Propose System Design

Bellow use cases are represent the way of involving actors with identified use cases.

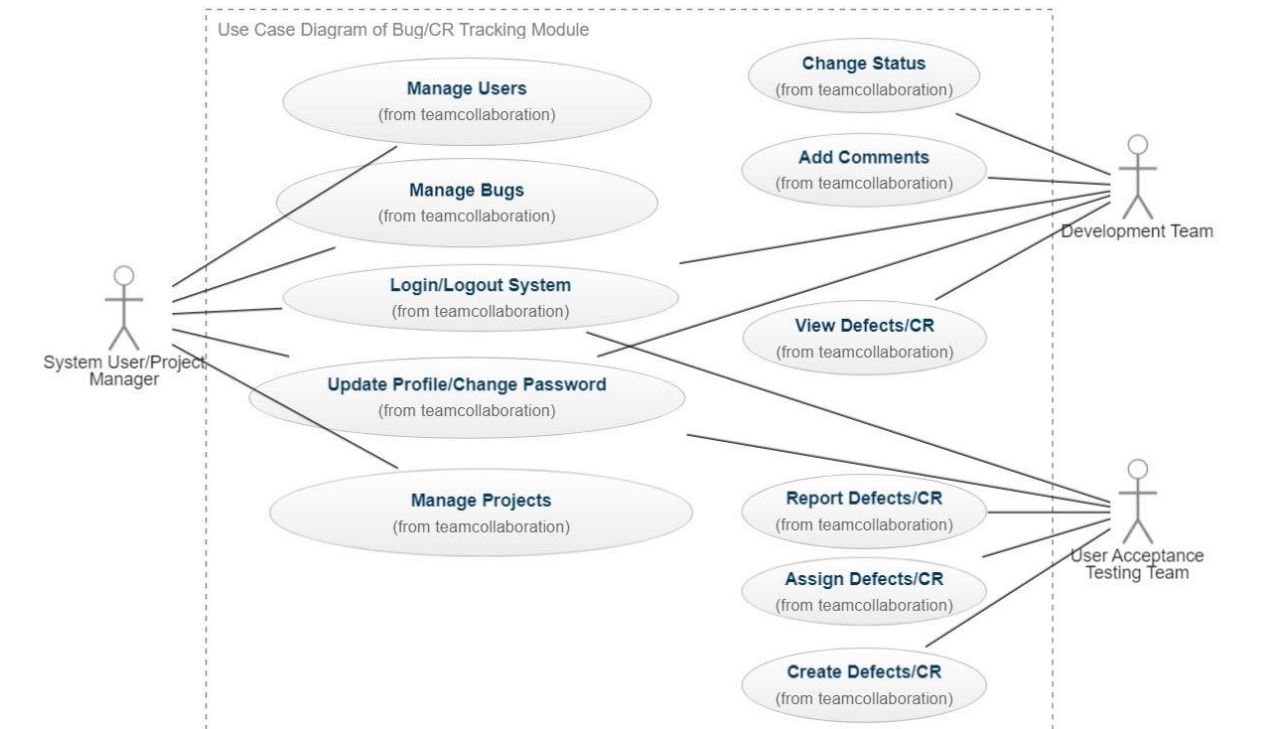


Fig. 3.4. Use case diagram of defect/CR tracking module

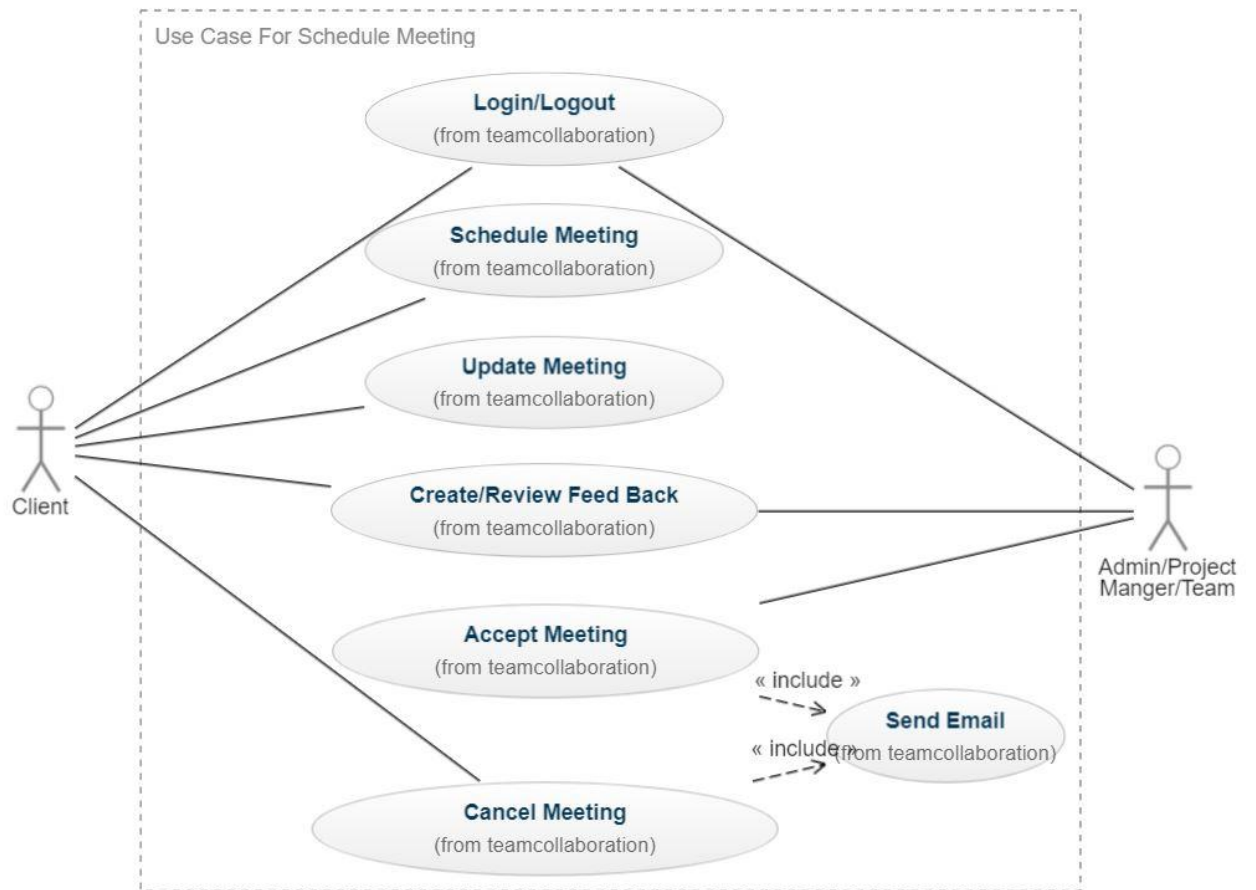


Fig. 3.5. Use case diagram of meeting scheduling module

3.9 Database design

Well design database structure is key factor of the system maintainability. The simple and normalized database will helps to easily extend the system features in future. The database design of the team collaboration management were normalized up to 3rd normal form.

Following figure shows the entity relational (ER) database design for the system.

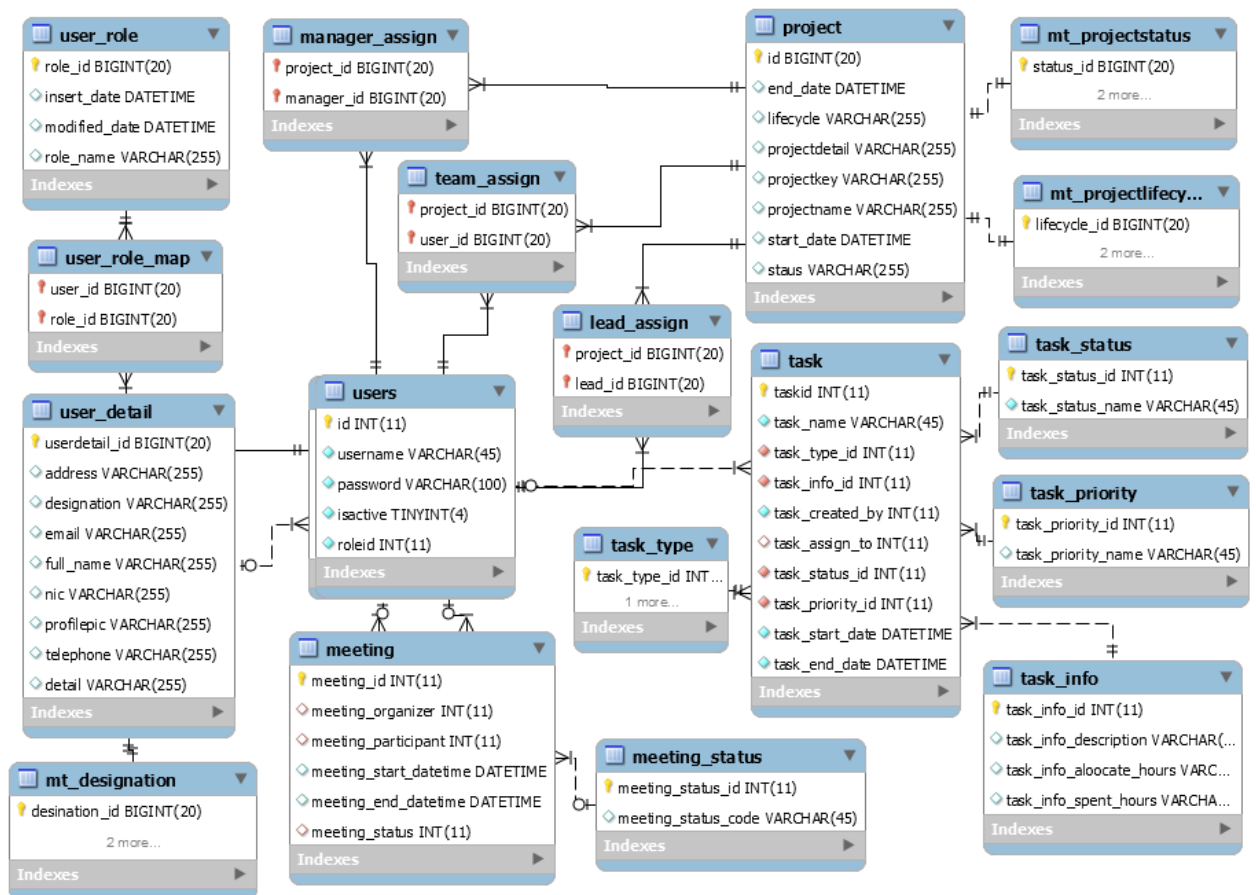


Fig. 3.6. Entity relational (ER) diagram of entire system

3.10 Class diagram of the system

Class diagrams depicts the static structure of the application. This class diagram describes how classes, interfaces are associate with each other in team collaboration management system.

The following class diagram depicts the overall classes of the system.

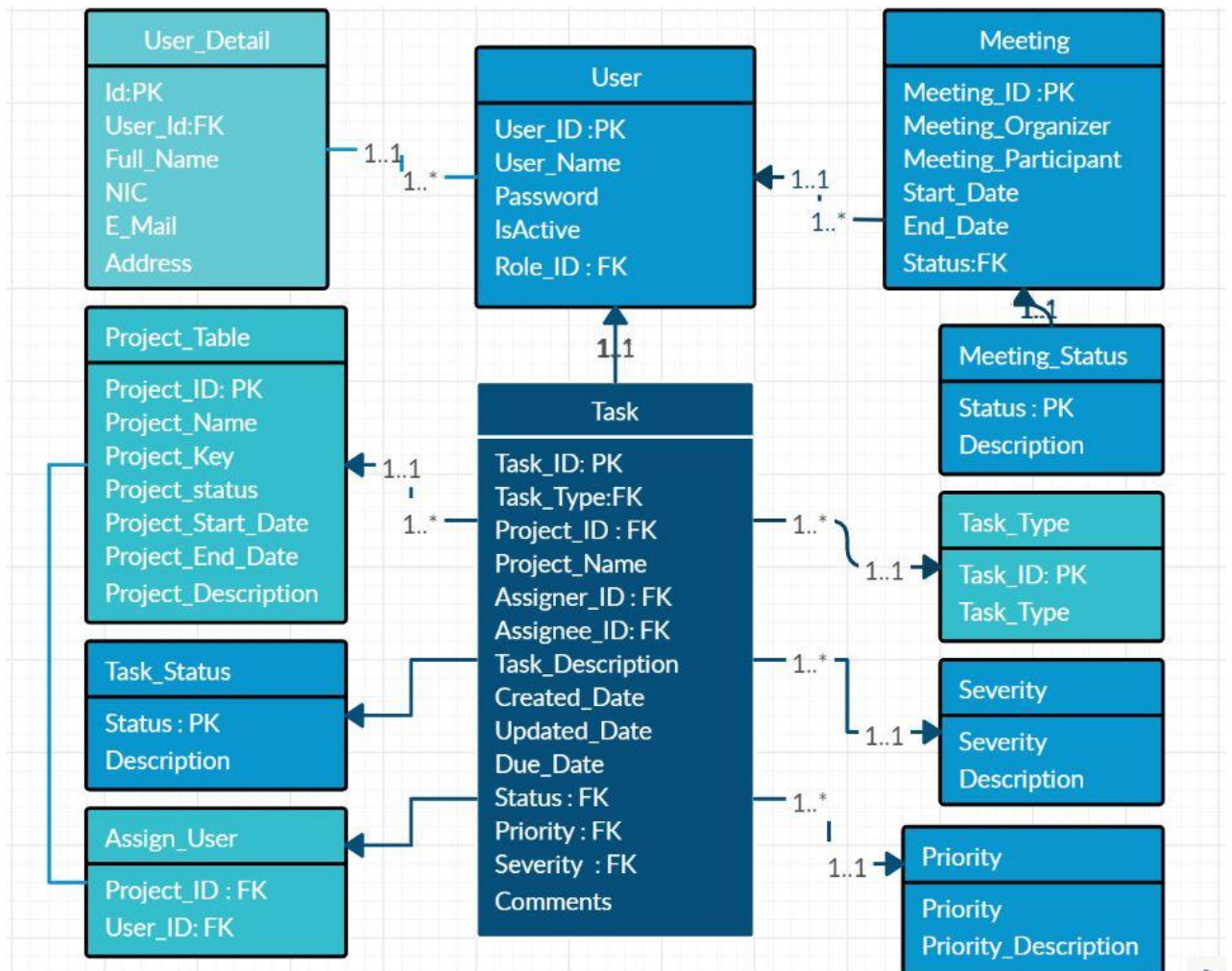


Fig. 3.7. Class diagram of Team collaboration management system

3.11 Data Flow Diagram(DFD)

This DFD diagram depicts the whole work flow of the Defect/CR tracking module.

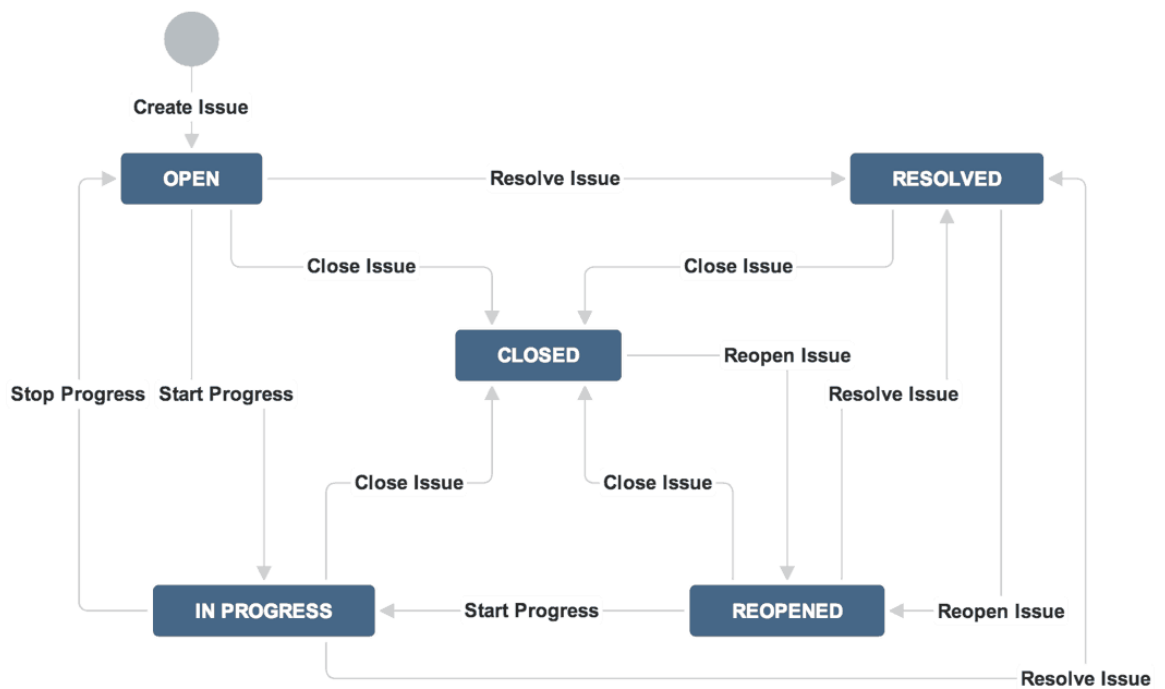


Fig. 3.8. Data flow diagram of defect tracking module

Implementation

4.1 Introduction

This chapter express the way of implementation of the proposed system. This chapter will provide the aid for software development with high level architecture of the system and details of its components. Further more constraints and assumptions will be discussed. The main objective of this phase is transform the software design/model into executable working software application. In implementation process developer should be select the most suitable programming language which comply with system design. After that need to decide the best efficient design patterns, tools and technologies such as IDE, API's.

4.2 Development Tools And Techniques

For implementation purposes there were used several tools and techniques for improve software quality and achieve the software development best practices.

Below described the tools and technologies that were used to develop the team collaboration management system.

4.2.1 Development Tools

□ Eclipse Oxygen.2 (4.7.2) With STS

Eclipse Oxygen.2 with Spring Tool Suite plugin was used as the IDE for this project. The STS(Spring tool suite) plugin is optimized for developing enterprise level Spring framework-based projects. Eclipse IDE make development easy and quicker development with features such as automatically complete system that enables to find and use parameters, classes etc... Also helping automatically generate code in the

Code Editor. As well as it provides debugging facility, code error checking facility and enable configure with external application servers. Such as apache tomcat, glassfish, wildfly, Jboss and etc..

□ **MySQL Workbench 6.3**

MySQL Workbench is a client application used to handle SQL development. Which is visual database design tool that integrates relational data base design, creation and maintenance purposes.

Creately Tool

Visual software to draw and collaborate on ideas, concepts and processes. Used it as a chart and diagram maker purposes. Creately helped to make the use case, data flow and class diagrams. □

4.2.2 Development Technologies

JAVA

Java programming language used for implement the back end code base. It is a open source most popular and object oriented support enterprise level programming language.

Spring Boot framework

Spring Boot is a rich framework for implement the java enterprise level web applications. By using spring boot framework any one able to create production ready stand-alone or dynamic web application within few minutes. It provides comprehensive infrastructure support and several standard plugins, API's for develop enterprise level applications. It gives permits to build and deploy applications quickly. It consists of the common spring compatible library which has set of reusable types. Besides that used Spring Data JPA to handle database transactions and Spring Security with OAUTH2 to handle the user authentication and authorization in the system.

Hibernate Framework

Hibernate is open source java based object relation mapping(ORM) technology. It is a programming technique that maps the java object with tables in relational databases. Through hibernate technologies and annotations developer able to easily manipulate data in relational tables via java entity classes.

Maven

Maven used as build and dependency tool. Maven is open source, stable and feature-rich build tool for java project. In brief, by using maven we can add required libraries from cloud maven repository to our project build path. Developer don't want to add libraries manually to project build path.

Thymeleaf

It is a modern server-side Java template engine for both web and standalone environments. This template engine is better suited for serving XHTML/HTML5 at the view layer of MVC-based web applications. This project used thymeleaf for rendering data from back-end to front-end.

HTML5

HTML5 was used as markup language to build the web pages of team collaboration management system.

Bootstrap 4

Bootstrap 4 is an open source CSS(Cascade style sheet) framework and which provide predefined CSS classes for implement responsive, user friendly and attractive elements on HTML pages. Bootstrap design templates support all the major browsers.

JQuery

JQuery is a light weight javascript library and jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. This system used JQuery validator for form validation purposes and AJAX calls for get data from the server without refreshing web pages. Besides that jquery datepicker plugin used to get dates from calendar and jquery table for represent data in grid

views with pagination options. In addition to that used jquery toaster plugin for give user notifications via bootstrap alerts. □

4.3 Implementation Environment

The implementation environment can be divided into under two categories. Those are server side implementation and client side implementation. The requirements of each implementation can divide as software requirements and hardware requirements. These requirements are described below.

4.3.1 Server Environment

The server environment basically concentrate with how executable application deployed and what are the infrastructural resources need to do it. The server machine should have the power of handle the multiple requests simultaneously. As well as server machine should have multiple cores for load balancing purposes. By considering above requirements, server machine should be at least mid-upper range server and it should powered by 12 GB RAM power and at least 80 GB or above hard disk space. The server should be support JAVA8 (JDK8.0) with maven. The war(web archive) file should be deployed on apache tomcat application server and apache server version should be up-to tomcat 8.0. For get best performance JVM(Java Virtual Machine) should allocate at-least more than 8 GB capacity from RAM capacity. As well as server support MySql for DB.

4.3.2 Client Environment

The client environment means the interface in between user and application. In brief it is the web browser for web client. System was empowered by latest technologies and tools, there for by using latest browser versions user able to avoid the browser compatible issues.

4.4 Network Architecture

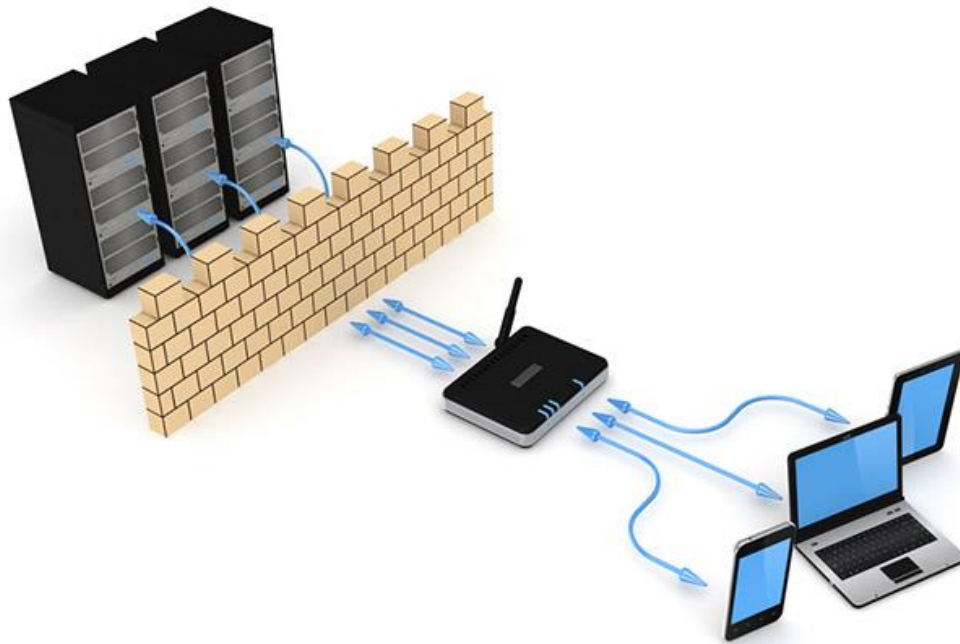


Fig. 4.1. Network architecture of Team collaboration management system

Team collaboration management system comply with the client server architecture. As explain in above diagram team collaboration system has central web server and users connects to the server via internal network(LAN). The (LAN) internal network protected by the firewall.

4.5 Project Structure

This is spring based dynamic web project and project modeling under MVC architecture. Below picture depicts the real project structure of the system.

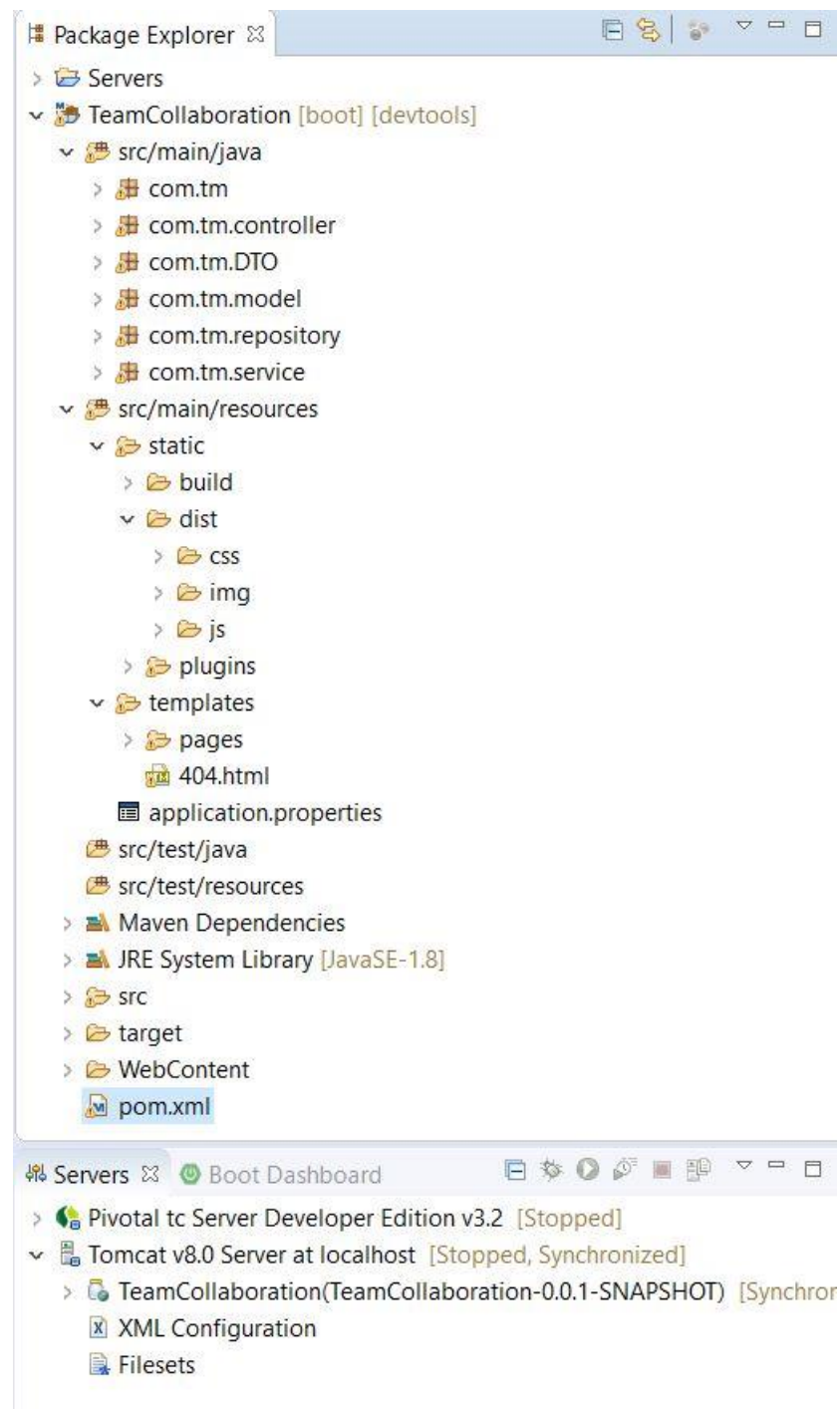


Fig. 4.2. Project Structure

4.6 System User Interfaces

Few of selected user interfaces shown in below section, which describes how interfaces are designed and developed in the team collaboration system.

Login Page

Through this interface, users are allowed to navigate to the home page by providing correct credentials.

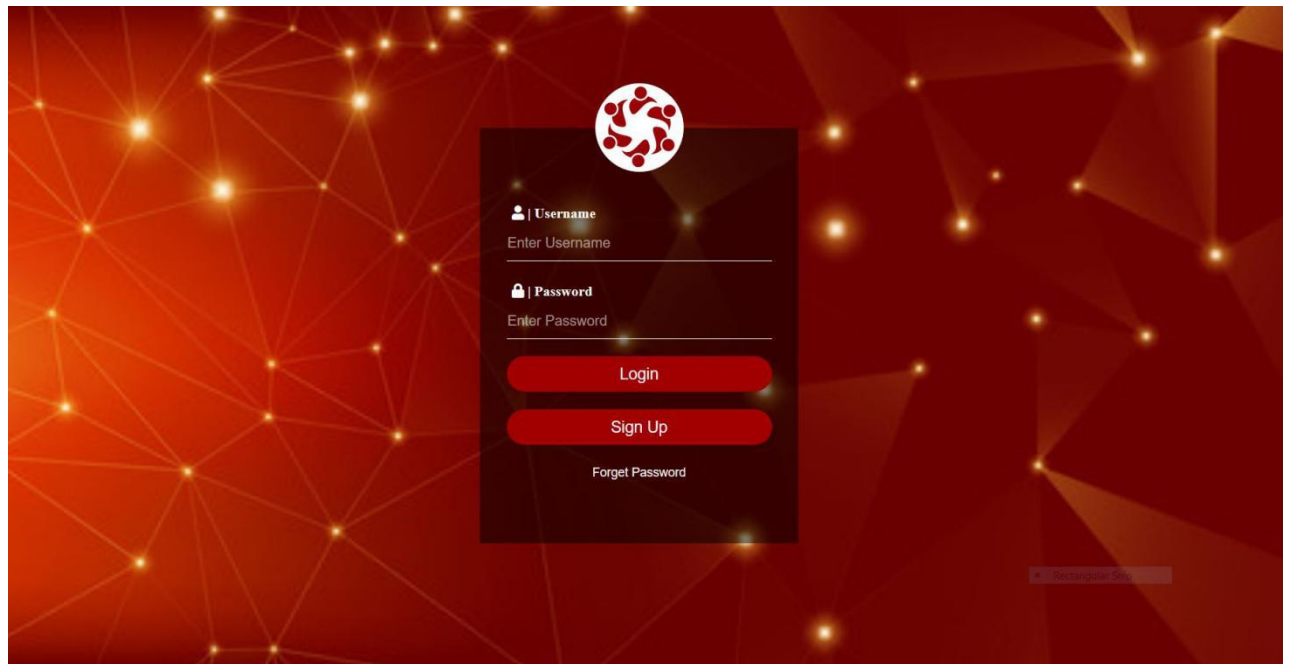


Fig. 4.3. Login interface

Sign-Up Page

The system facilitate the users to sign up with system, who are not on boarded with system.

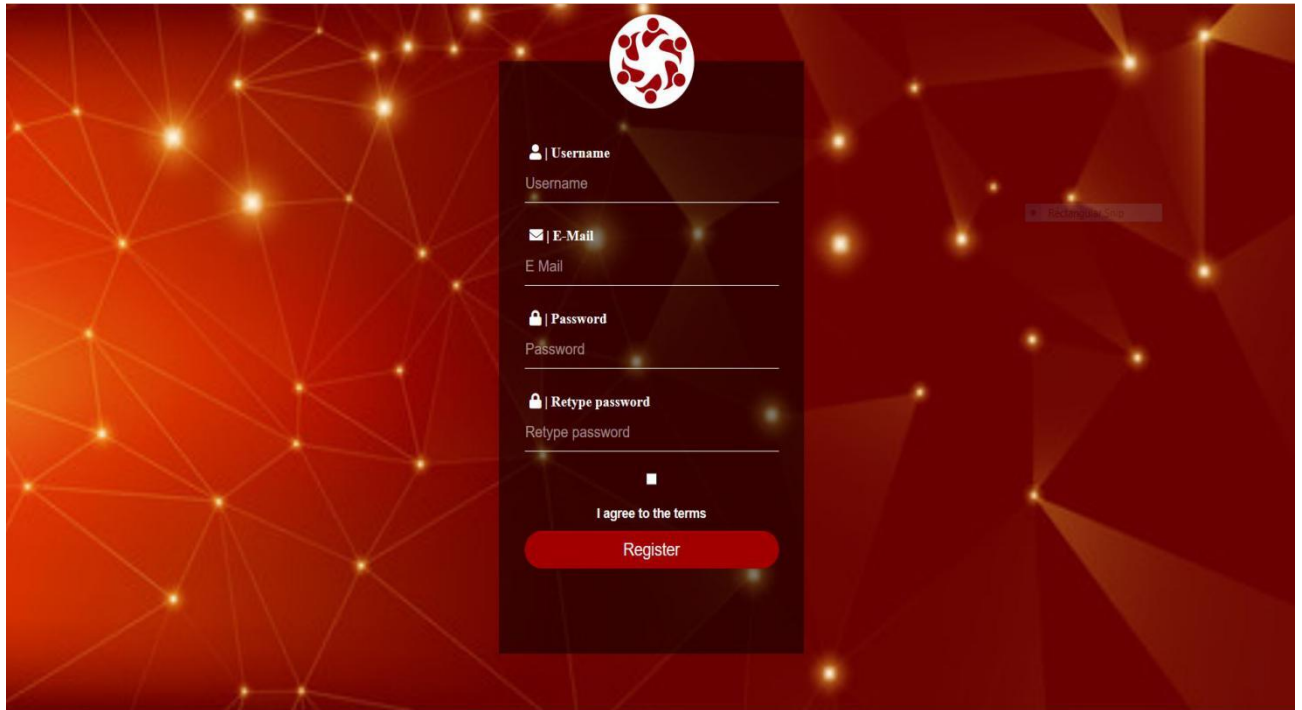


Fig. 4.4. Sign up interface

Home Page

After success login user redirect to home page.

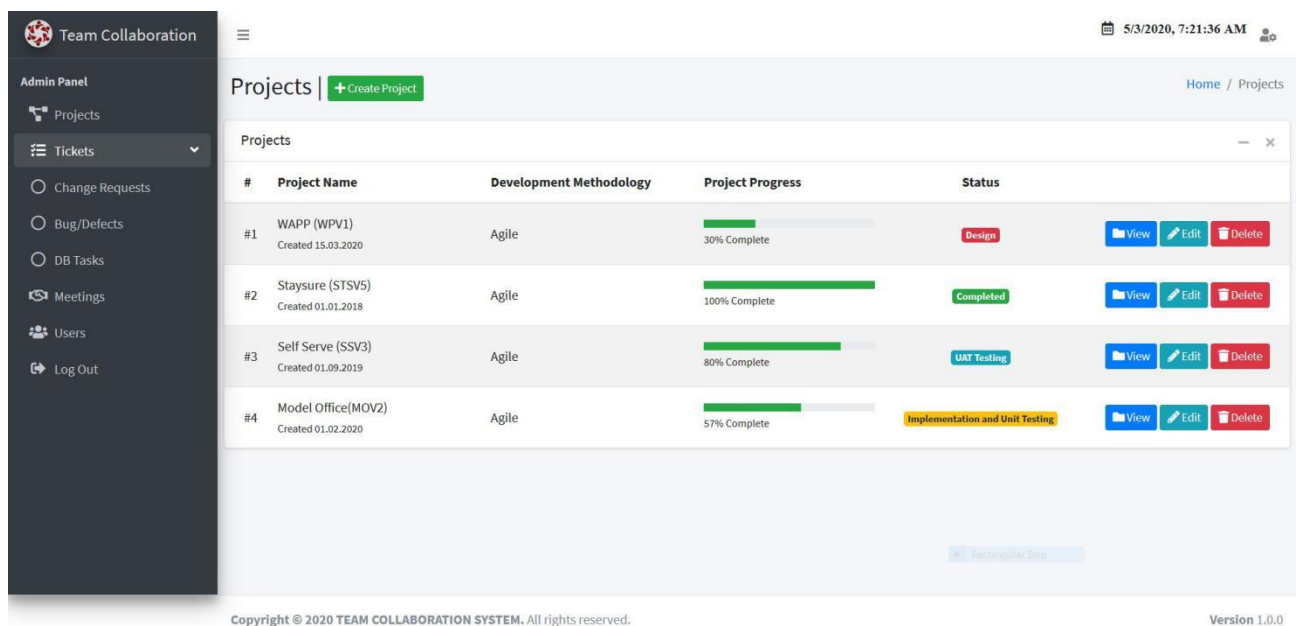


Fig. 4.5. Home interface

User Module

Through this interface system admin able to register users with system and manage user module of Team collaboration system.

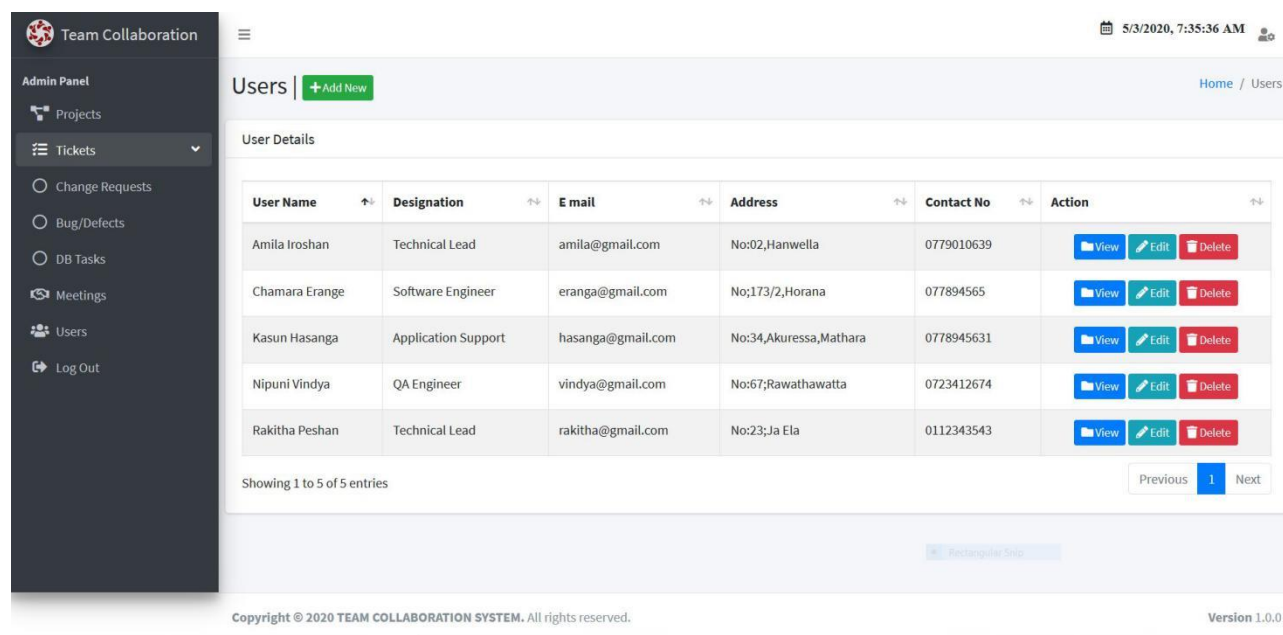


Fig. 4.6. User management interface

Defect/CR Tracking Module

This interface provide CRUD operations for the ticket module.

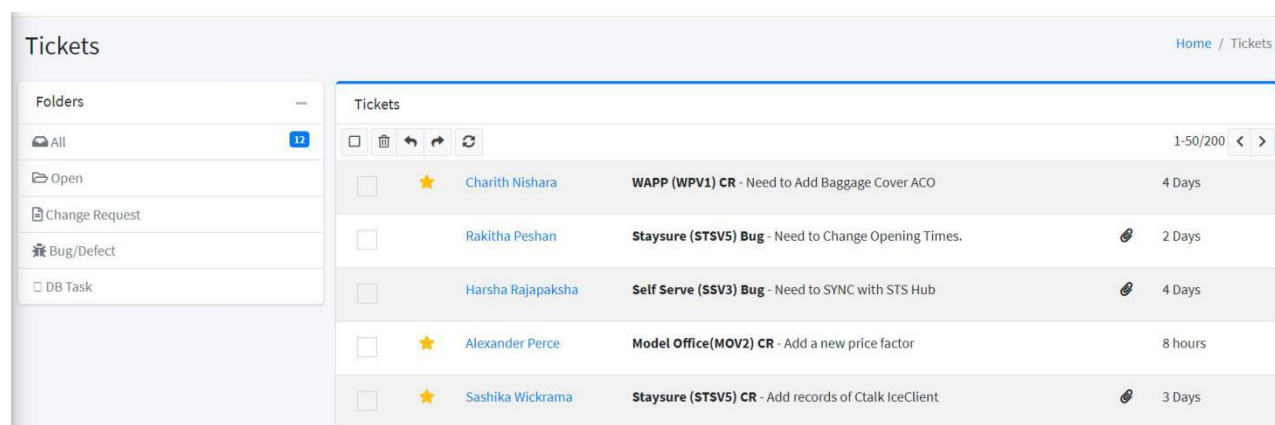


Fig. 4.7. Defect management interface

Meeting Scheduling Module

Meeting scheduling interface provide the functions to the system admin ,which are maintain the information of scheduled meetings.

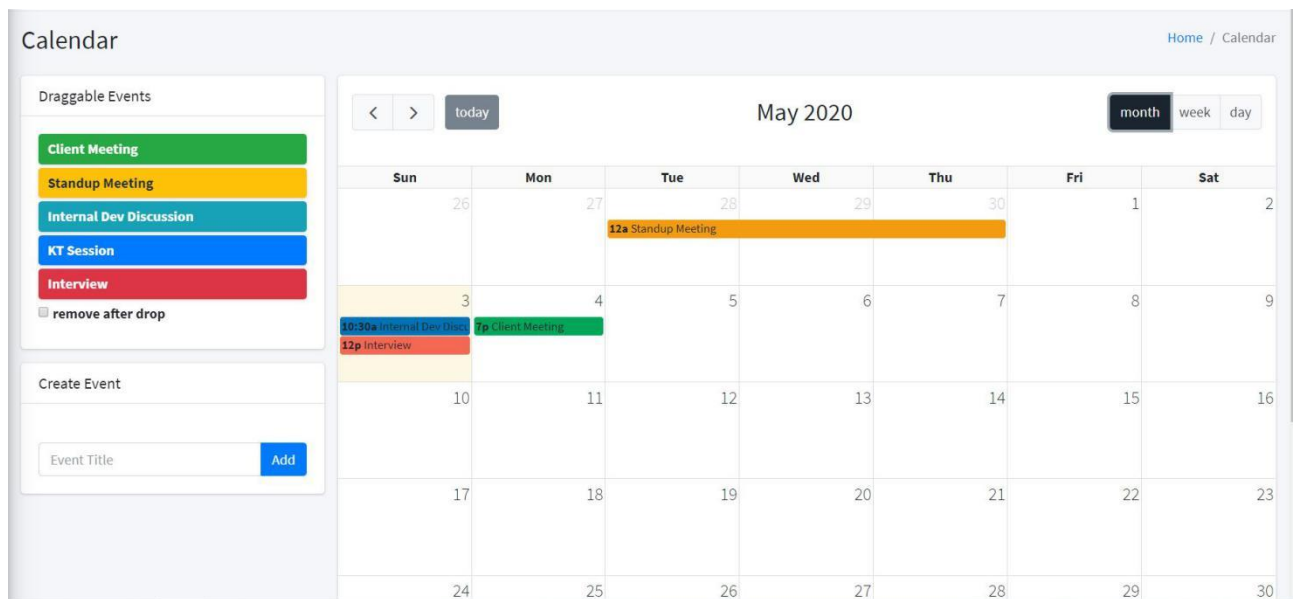


Fig. 4.8. Meeting scheduling interface

User Evaluation And Testing

5.1 Introduction

This chapter describes the evaluation and testing of the Team collaboration system. This chapter will cover the scope of testing, test plans, test cases, test execution and test results. The software testing depends on software verification and validation process. The validation is ensure that the product is satisfied the design specifications and requirements. In briefly check whether developed product meets the specific requirement. In verification ensure that is this product fulfil the user's business requirement and is this right product. The performance testing and acceptance testing will discuss later.

5.2 Testing Procedure

The Team collaboration management system was followed the iterative and incremental processes as software development and testing process. Therefore often these two scenarios working as simultaneously. The each and every task divided as small chunk of components and when development was going on, testing techniques can apply on it. After completed the unit testing ,integrated each component as module vice. Then again proceeded the integration testing for test whole system. User acceptance testing was unable to proceed perfectly because of COVID 19 pandemic.

5.3 Test Plan and Test Cases

This sub chapter describes testing approach for each test type and project works related to quality assurance perspective. The test plan was designed before the start implementation and testing initiated with the begins of implementation. The test plan consists of the test cases, test strategies and test objectives. Then test cases were created by following the test plan and each test case consists of test case id, testing component, actual output and expected output. Each test case divided under module vice for reduce the complexity of the test cases. Following tables specify some major manual testing test cases of team collaboration system.

5.3.1 Login Module


Test Id	01		
Component Name	Login		
Module Name	Login Module		
Test Case	Login Interface		
Expected Output	When user enter invalid credential system gives error message and for valid credentials system redirect to home page.		
Description			
	Test Case	Actual Output	Status
1	Login with valid credentials	Enter correct/valid username and password. System authenticate user and redirect to home page.	PASS
2	Login with invalid login credentials	Error message pop up with reason. 	PASS

Table 5.1. Login test case

5.3.2 User Management Module


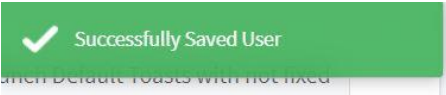
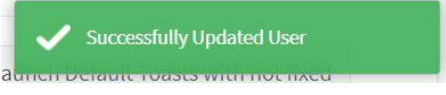
Test Id	02		
Component Name	User Data Management		
Module Name	User module		
Test Case	Add new user/Update existing user		
Expected Output	When system admin going to add a new user to system via user module required field must be highlighted and give a relevant error message.		
Description			
	Test Case	Actual Output	Status
1	Press save button without mandatory fields.	Shows relevant error message. 	PASS
2	Press save button with required information.	Save data to relevant table and show success message. 	PASS
3	Load existing user and update relevant field and press update button.	Update relevant field of table and show relevant message. 	PASS

Table 5.2. User management test case

5.3.3 Project Management Module


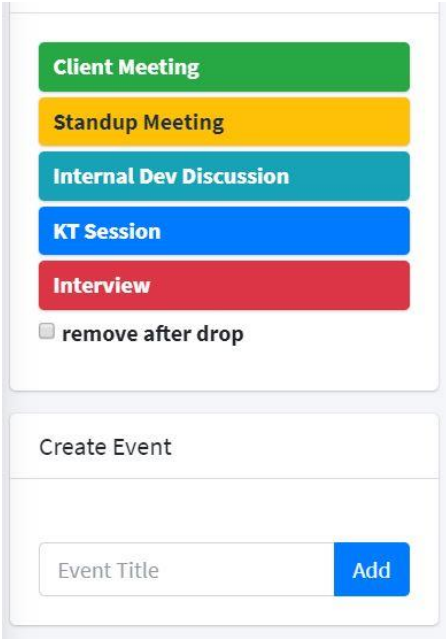
Test Id	03		
Component Name	Project Management Component		
Module Name	Project Management Module		
Test Case	Add/Edit project		
Expected Output	Trigger validations of mandatory fields. When user provide invalid input or exception happens ,system gives proper error message.		
Description			
	Test Case	Actual Output	Status
1	Add new project to system.(Enter correct details to add project form and press save button)	Successfully save data to particular tables according to table relationships.	PASS
2	Press save button with insufficient information.	Does not save data in table and error message display with proper reason. 	PASS

Table 5.3. Project management test case

5.3.4 Meeting Scheduling Module

Test Id	04		
Component Name	Meeting Scheduling Component		
Module Name	Meeting Scheduling Module		
Test Case	Schedule a meeting		
Expected Output	Create a new meeting event.Able to assign team members to event and show it in calender view.		
Description			
	Test Case	Actual Output	Status
1	Add new event to system.Show existing events also.	Successfully save data to particular tables when press the add event button. 	PASS
2	View Added	Show event with event name and time on event calender.	PASS


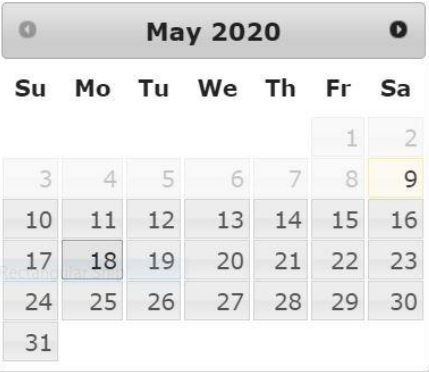
	event on calender.		
3	<p>Select meeting/event date for create meeting disable the past dates on date picker.</p>	<p>Unable to choose the past dates from date picker.</p> 	PASS

Table 5.4. Meeting scheduling test case

5.4 User Acceptance Testing

The user acceptance testing(UAT) was planned to conduct in the actual working environment of Intervest Software Technologies with real data sets and real system users. But unfortunately couldn't proceed it because of COVID 19 epidemic.

6.1 Introduction

This chapter reviews the what are the lesson learned and how to faced challenges which are raised throughout the project.Besides that discussed about things of achievable and the future enhancement of the developed system. Team collaboration web application was developed and deployed on the tomcat8 application server, such that users can access the system via internet.Web application achieved all its objectives in the functional level, by covering different user roles and responsibilities.The success of operational level and reliability of the system can be measure after go live the system.The system was highly compatible with object oriented programming models and built using Java language with Spring boot framework,Thymleaf, HTML5, CSS and JavaScript.

6.2 Challenges Faced During the Project

The Team Collaboration Management System is highly comply with simple work flows and simple and user friendly GUI(Graphical User Interfaces).Application consisted thymleaf java based template with javascript framework for implement AJAX calls.The configuration of thymleaf template and when it configuring with javascript framework took much time and effort consumed than estimation.In addition to that during the implementing some modules, some of major requirements were changed and also database structure was changed according to requirements. Because of the amended works of code base of those specific modules which took lot of time.Could not proceed the user acceptance testing(UAT) because of COVID 19 epidemic situation.

6.3 Lesson Learned

The entire software development life cycle was helped to clearly understand some of the software engineering theory. Specially learned how software design best practices are helpful and apply when implementing software. As well as gained some experience how to apply several design patterns such as MVC, builder, singleton and etc. through out the system. In addition to that I could able to work with many professional individuals and it helped to improve my intellectual skills. Apart from that tight dead lines achieved with proper time management and planning strategies. As well as unified modeling languages(UML) was used to elaborate application structures, system structures and business process. Furthermore lot of new technical stuffs have been learned out during this project and which are listed below,

1. Successfully configured spring security for achieve user authentication and authorization.
2. Implemented restful web services for send response to front end from back end. The system has been provided best response time and throughput via restful web services.
3. Used AJAX(Asynchronous JavaScript and XML) for better data fetching purposes.
4. Successfully configured bootstrap(CSS Framework) with thymleaf template engine.
5. Applied hibernate as object relational mapper for handle the database transactions.

6.4 Future Works

According to feed backs of user acceptance testing modify the system requirements as requested by users. As well as increase the functional performance by eliminate boilerplate codes from code base. Then plan to add email server and report generation module which support to top management when decision making.

References

- [1] Tutorialspoint. (2015,Oct.).Defect logging and tracking [Online]. Available :
https://www.tutorialspoint.com/software_testing_dictionary/defect_logging_and_tracking.htm
- [2] ClickUp,"ClickUp,"ClickUp,[Online]. Available :<https://clickup.com/>.
- [3] Jira,"Jira,"Jira,[Online]. Available :<https://www.atlassian.com/software/jira>.
- [4] Slack.com,"Slack.com,"Slack.com,[Online].Available:<https://slack.com/intl/en-lk>
- [5] Trello.com,"Trello.com,"Trello.com,[Online].Available :<https://trello.com/>
- [6] Freshdesk,"Freshdesk,"Freshdesk,[Online].Available :<https://freshdesk.com/>
- [7] Redbooth,"Redbooth,"Redbooth,[Online].Available :<https://redbooth.com/>
- [8] UKEssays.(2018,Nov.).A Web Based Meeting Scheduler Information
TechnologyEssay.[Online].Available :<https://www.ukessays.com/essays/information-technology/a-web-based-meeting-scheduler-information-technology-essay.php?vref=1>

APPENDIX A - SYSTEM DOCUMENTATION

This section express the software and hardware specification of the system and the guide lines of system setup in development environment.

Hardware

Hardware	Minimum Requirement
Memory	4GB RAM or more.
Processor	Intel Core i3 processor 2.3GHz
Hard disk space	Minimum 10GB free disk space or Higher.
Display	1024 X 768, High colour 32-bit display
Internet	Minimum 512kbps ADSL connection

Table 6.1. Hardware requirements

Software

Software	Minimum Requirement
Operating System	Any Operating System(JDK Support)
Database	My SQL Server 5.5

Web browser	Internet explorer/Fire Fox/Google Chrome
Application Server	Apache TOMCAT 8.0
JDK/JRE Version	Minimum JDK 8.0
Maven Version	Minimum Maven 4.0

Table 6.2. Software requirement

SETUP

- First install above software on server, The server should be compatible with above mentioned hardware requirements.
- The server machine should be configured JAVA HOME and MAVEN HOME in system variables.
- Then need to create a privileged user on MySQL server and should create database called as “teamcollaboration”.
- After that run the provided DB script for create master tables with master data.
- If you use external tomcat server, put provided .war files under webapps folder and then start the tomcat server using .bat or .sh file.

- If you are not using external tomcat server you have to go inside the project folder provided by the system admin and run the the “mvn spring-boot:run” on your CMD.It will run the project on embedded tomcat server.
- Run the system in browser using given URL notations.
- Press login button to access the system, by providing correct credentials.

APPENDIX B - DESIGN DOCUMENTATION

Use case diagrams and descriptions

Use case diagram for BUG/CR tracking module.

Use Case	BUG/CR Tracking
Actors	Project Manager, QA team, Dev team
Overview	
Add new Defect/CR to system	
Pre-Conditions	
System should have QA team members,Dev team members and particular project	
Flow Of Events	
<ol style="list-style-type: none"> 1. System admin or Project manger log into system. 2. Click the create new defect or CR button and then fill out form with required information. 3. Assign project from existing list of project to ticket. 4. If not project first create new project. 5. Then assign particular team members(Dev and QA personal) to ticket. 6. Then click save button. 	
Post Conditions	
Added data to particular tables in data base.	

Table 6.3. Defect/CR tracking use case

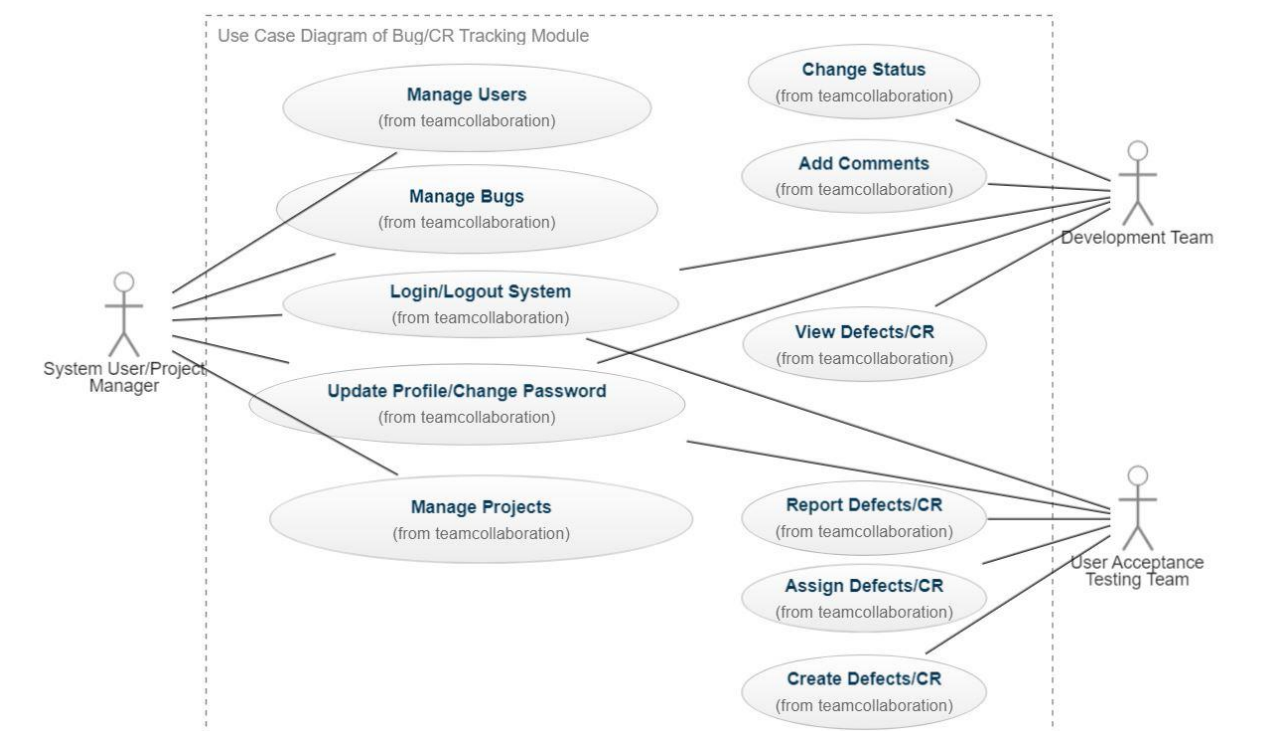


Fig. 6.1. Defect/CR tracking usecase diagram

Use case diagram for meeting schedule module.

Use Case	Meeting schedule
Actors	Project Manager, Project Team, Client
Overview	
Add new meeting(event) to system	
Pre Conditions	
System should have registered team members.	
Flow Of Events	
<ol style="list-style-type: none"> 1. Project manager or client log into system. 2. If event type already exists in system select it and if not create new event type. 3. Then select date time for particular event. 4. Assign participants to particular event. 5. Then click save button. 	
Post Conditions	
New event created with given information.	

Table 6.4. Meeting scheduling use case

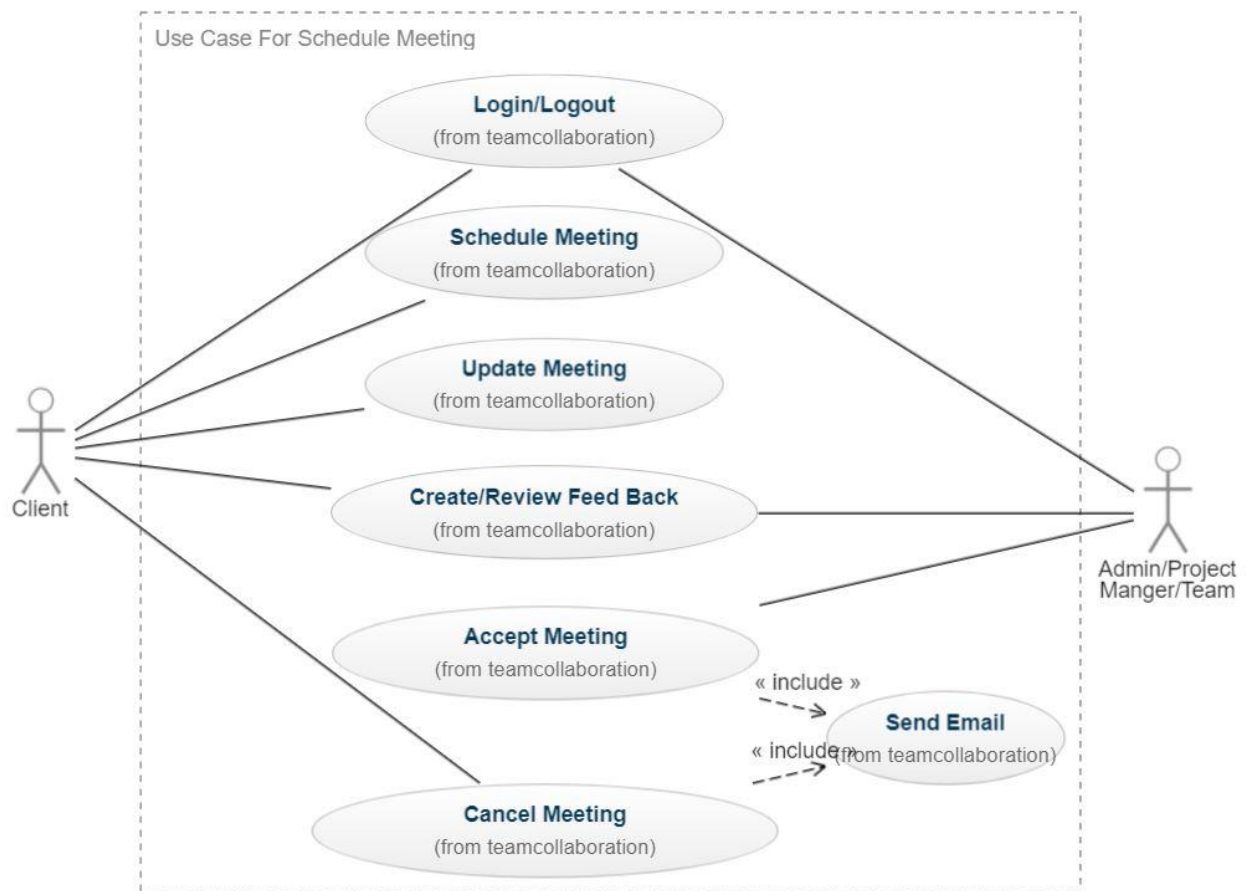


Fig. 6.2. Meeting scheduling use case diagram

APPENDIX C - CODE LISTING

This sub chapter describes the code base structure of the system by using sample code snippets. Team collaboration system modeled under MVC design pattern and code base seperated according to Model ,View and Controller. Some main code segments are described below.

View/Controller/Model

View:

The user interacts with the system through this web pages.

```
<div class="card">
  <div class="card-header">
    <!-- /.card-header -->
  <div class="card-body">
    <table id="userTable" class="table table-bordered table-striped">
      <thead>
        <tr>
          <th>User Name</th>
          <th>Designation</th>
          <th>E mail</th>
          <th>Address</th>
          <th>Contact No</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        <th:block th:each="userlist : ${userlist}">
          <tr >
            <td th:text="${userlist.user_detail.full_name}"></td>
            <td th:text="${userlist.user_detail.designation}"></td>
            <td th:text="${userlist.user_detail.email}"></td>
            <td th:text="${userlist.user_detail.address}"></td>
            <td th:text="${userlist.user_detail.telephone}"></td>
            <td class="project-actions text-center">
              <button class="btn btn-primary btn-sm" id="userView" th:onclick="'userView('+${userlist.id}+ ')">
                <i class="fas fa-folder">
                </i>
                View
              </button>
              <button class="btn btn-info btn-sm" id="userEdit" th:onclick="'userEdit('+${userlist.id}+ ')">
                <i class="fas fa-pencil-alt">
                </i>
            </td>
          </tr>
        </th:block>
      </tbody>
    </table>
  </div>
</div>
```

Fig. 6.3. Code snippet of view page

Controller:

Controller work as a interpreter in between the models and views.

```
@RequestMapping(value = "/userView",method = RequestMethod.POST)
@ResponseBody
public ResponseDTO userView(@RequestBody Long userId, HttpServletRequest request, HttpServletResponse response) {
    ResponseDTO res=new ResponseDTO();
    try {
        UsersDTO usersdto=new UsersDTO();
        Users u=tmUserService.viewUser(userId);
        usersdto.setFullName(u.getUser_detail().getFull_name());
        usersdto.setNic(u.getUser_detail().getNic());
        usersdto.setTpNumber(u.getUser_detail().getTelephone());
        if(u.getUser_role() != null) {
            u.getUser_role()
                .forEach(i -> {
                    if (i.getRole_name().equalsIgnoreCase("Admin")) {
                        usersdto.setUserRole1("true");
                    }else {
                        usersdto.setUserRole2("true");
                    }
                });
        }
        usersdto.setAddress(u.getUser_detail().getAddress());
        usersdto.setEmail(u.getUser_detail().getEmail());
        usersdto.setUserName(u.getUserName());
        usersdto.setPassword(u.getPassword());
        usersdto.setDesignation(u.getUser_detail().getDesignation());
        usersdto.setDetail(u.getUser_detail().getDetail());
        res.setData(usersdto);
        res.setStatus(true);
        return res;
    } catch (Exception e) {
        e.printStackTrace();
        res.setStatus(false);
        return res;
    }
}

@RequestMapping(value = "/updateUser",method = RequestMethod.POST)
```

Fig. 6. 4. Code snippet of controller

Model:

The model is known as entity class. Which is responsible to interact with database transactions. Following code snippet show the sample model class of the system.

```
package com.tm.model;

import java.io.Serializable;

@Entity
@Table(name="users")
public class Users implements Serializable {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="id")
    private Long id;

    @Column(name="username")
    private String userName;

    @Column(name="password")
    private String password;

    @Column(name="isactive")
    private boolean isActive;

    @ManyToMany(fetch = FetchType.LAZY,
        cascade = {
            CascadeType.ALL
        })
    @JoinTable(name = "user_role_map",
        joinColumns = { @JoinColumn(name = "user_id") },
        inverseJoinColumns = { @JoinColumn(name = "role_id") })
    private Set<UserRole> user_role = new HashSet();

    @OneToOne(cascade = {CascadeType.ALL})
    @JoinColumn(name="user_detail")
    private UserDetails user_detail;

    @ManyToMany(mappedBy = "projectManagers")
    Set<Project> projectManagers;
```

Fig. 6.5. Code snippet of model/entity class

Data base configuration:

```
3 spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect
4 spring.jpa.hibernate.ddl-auto=update
5 spring.thymeleaf.cache=false
6 spring.datasource.url=jdbc:mysql://localhost:3306/teamcollaboration
7 spring.datasource.username=root
8 spring.datasource.password=amila
9 spring.datasource.driver-class-name=com.mysql.jdbc.Driver
0 server.error.whitelabel.enabled=false
1 spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.web.servlet.error.ErrorMvcAutoConfiguration
2
```

Fig. 6.6. Code snippet of database configuration

Maven POM repository:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.tm</groupId>
  <artifactId>TeamCollaboration</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>TeamCollaboration</name>

  <parent>
  <dependencies>

    <dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jpa -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
      <version>2.1.6.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.15</version>
    </dependency>

    <dependency>
```

Fig. 6.7. Code snippet of maven repository