



NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA



TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A

PROJECT REPORT

ON

"BUG TRACKING SYSTEM"

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

NITIN KUMAR REDDY K A

1NH15CS157

Under the guidance of

Ms. Vasantha M

Assistant Professor(2),

Dept. of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is hereby certified that the project work entitled "**BUG TRACKING SYSTEM**" is a bonafide work carried out by **NITIN KUMAR REDDY K A (1NH15CS157)** in partial fulfilment for the award of **Bachelor of Engineering** in **COMPUTER SCIENCE AND ENGINEERING** of the New Horizon College of Engineering during the year **2018-2019**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

.....

Signature of Guide
(Ms. Vasantha M)

.....

Signature of HOD
(Dr. B. Rajalakshmi)

.....

Signature of Principal
(Dr. Manjunatha)

External Viva

Name of Examiner

Signature with date

1.

.....

2.

.....

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing my deep sense of gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal, NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Ms. Vasantha M**, Assistant Professor (2), my project guide for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally, a note of thanks to the teaching and non-teaching staff of Dept of Computer Science and Engineering, for their cooperation extended to me, and my friends, who helped me directly or indirectly in the course of the project work.

NITIN KUMAR REDDY K A (1NH15CS157)

ABSTRACT

Bug tracking system is an online project management tool, which will be used to track and manage bugs with the project without any latency. Project Manager will create team members including developers and testing team. Testing team will create and assign bugs to developer in the bug tracker which notifies developer immediately through email notification. Developer will start fixing the bugs which were allocated to him and after completing the fix, will update it as “resolution” in the bug tracker which will notifies the testing team through email notification. Testing team member will immediately verify the fix and close the issue if it is working else testing team will re-open the issue. In this way entire process will be automated which will helps to achieve the quality in the project.

Project success rates can be increased. There will be tracking for all the bugs and quality of the team can be measured based on this application. Every action in this application will send notification to the team members which will notify the team immediately.

The bug report is mailed to the project manager and the developer as soon as the bug is identified. This makes that no error will go unfixed because of poor communication. It makes ensure that anyone who needs to know about a bug can learn of it soon after it is reported. Bug tracking system plays vital role in the testing phase. But it supports assigning projects for the developer, tester by the project manager.

CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENT	II
LIST OF FIGURES	V
1 INTRODUCTION	
1.1 INFORMATION CONSISTENCY	1
1.2 PROBLEM DEFINITION	2
1.3 PROJECT PURPOSE	2
1.4 MODULES DESCRIPTION	2
2 LITERATURE SURVEY	
2.1 EXISTING SYSTEM	4
2.2 PROPOSED SYSTEM	4
2.3 SOFTWARE DESCRIPTION	4
3 REQUIREMENT ANALYSIS	11
3.1 FUNCTIONAL REQUIREMENTS	11
3.2 NON-FUNCTIONAL REQUIREMENTS	11
3.2.1 ACCESSIBILITY	11
3.2.2 MAINTAINABILITY	11
3.2.3 SCALABILITY	12
3.2.4 PORTABILITY	12
3.3 HARDWARE REQUIREMENTS	12
3.4 SOFTWARE REQUIREMENTS	13
4 DESIGN	
4.1 DESIGN GOALS	14
4.1.1 INPUT/OUTPUT PRIVACY	14
4.1.2 EFFICIENCY	14
4.2 SYSTEM ARCHITECTURE	14
4.3 SEQUENCE DIAGRAM	16
4.4 DATA FLOW DIAGRAM	19
4.5 USE CASE DIAGRAM	21

4.6	ER DIAGRAM	23
5	IMPLEMENTATION	
5.1	CODE SNIPPETS	24
6	SOFTWARE INSTALLATION	
6.1	JDK INSTALLATION	27
6.2	Eclipse Installation	31
6.3	Tomcat Installation	33
7	SCREEN SHOTS	
7.1	Login the Bug tracking system web page	35
7.2	Home Page	36
7.3	adding new team member	36
7.4	status of the project	37
7.5	Summary of the bugs	37
7.6	Upload the code	38
8	CONCLUSION	39
	REFERENCES	40

LIST OF FIGURES

Fig no.	Fig.Name	Page no.
2.1	Home page controller	5
2.2	Flow of java program	7
2.3	MySQL database	8
2.4	Clint Application	9
4.1	MVC Architecture	14
4.2	N-tier Architecture	15
4.3	Servlet Engine	15
4.4	Sequence diagram for project manager	16
4.5	Sequence diagram for Developer	17
4.6	Sequence diagram for Tester	18
4.7	Data Flow Diagram	19
4.8	Flow Diagram	20
4.9	Use case diagram for bug tracker	21
4.10	Use case diagram for user and developer	22
4.11	ER Diagram for Bug tracker	23
6.1	Oracle website	27
6.2	JDK download	28
6.3	JDK installation	28
6.4	System window	29
6.5	Environment Variable	29
6.6	Path creation	30
6.7	Command prompt	30
6.8	Eclipse website	31
6.9	Eclipse installer	31
6.10	Terms and Conditions	32
6.11	IDE	32
6.12	Apache website	33
6.13	Downloads	33
6.14	Apache Tomcat	34

CHAPTER 1

INTRODUCTION

1.1 INFORMATION CONSISTENCY

A bug is a common term used to describe an error, mistake, failure or fault in a program. Most bugs are arise from mistakes and errors made by people in either programs source code or its design, and few are caused by compilers. Bug Tracking System is a web-based application that is designed to help quality assurance and programmers keep track of reported software bugs in their work. A major component of bug tracking is a database that records facts and known bugs. The report of database can consist of several information which can also called facts. The project manager assigns projects to the developers. The developer develops the projects as per customer requirements. The project manager itself assigns the developed applications to the Testers for testing. The testers test the application and identify the bugs in the application. When the tester encounters no. of bugs, he generates a unique id number for each individual bug. The bug information along with its id is mailed to the project manager and developer. This is Bug Report. These are stored in the database. This is useful for further reference. Bug information includes the bug id, bug name, bug priority, project name, bug location, bug type. This whole process continues until all the bugs are got fixed in the application. The bug report is mailed to the project manager and the developer as soon as the bug is identified. This makes that no error will go unfixed because of poor communication. It makes ensure that anyone who needs to know about a bug can learn of it soon after it is reported. Bug Tracking System plays an vital role in the testing phase. But it supports assigning projects for the developer, tester by the project manager. The Bug Tracking System maintains the different users separately i.e. it provides separate environments for project manager, developer and tester.

1.2 PROBLEM DEFINITION

Bug Tracking System is to test the application for the bugs and report it to the project manager and developer. The main intention behind the Bug Tracking System is that to track bugs and report them. Store the bug information with a unique id in the database.

The problem in the older system can be defined as the whole project maintenance; user's maintenance and their assignment have to be maintained manually. The Software development companies have to face a lot of problems while maintaining manually all the maintenance of the projects their bugs and their status. This type of problem makes the whole system an inefficient one and thus making a poor and unorganized working. In order to remove this type of problem, so that the paper is planned to develop. Bug tracking software is a "Defect Tracking System" or a set of scripts which maintain a database of problem reports. Bug tracking software allows individuals or groups of developers to keep track of outstanding bugs in the product description etc. in the form of reports from time to time. The paper effectively. Bug tracking software can track bugs and changes, communicate with members, submit and review patches, and manage quality assurance. This web-based application is a great tool for assigning and tracking issues and tasks during software development and any other projects that involve teams of two or more people.

1.3 PROJECT PURPOSE

Bug Tracker is an online application which will be used to track and manage bugs with the project without any latency. Project success rates can be increased. There will be tracking for all the bugs and quality of the team can be measured based on this application. Every action in this application will send notification to the team members which will notify the team immediately

1.4 MODULES DESCRIPTION

Admin: This module has the entire access to all other modules, admin creates the project and assigning the projects to the created manager, adding members to the managers, assigning bugs based on the priority.

Manager: Manager has the full access to the particular project assigned by the admin and controls the team member's access to the bugs assigned.

Developer: Can access the task or bug assigned by the manager, view assigned projects and resolving the assigned bug. Developer can view the bugs list assigned by the manager.

Tester: Tester can access to the projects or bugs assigned by the manager, can view the assigned projects and can add a new bug to the list and send the bug back to the manager. Tester can login to the system and access the assigned projects list.

Reports: Both Admin and Manager can access this module and generate the reports based on the requirements

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

- Most of the companies manually interact with the team by updating status between developers and QA members.
- Project manager has no proper visibility of what is happening in the projects. This leads to failure in the project delivery.

2.2 PROPOSED SYSTEM

- Bug Tracker is an online application which will be used to track and manage bugs with the project without any latency.
- Project success rates can be increased.
- There will be tracking for all the bugs and quality of the team can be measured based on this application.
- Every action in this application will send notification to the team members which will notify the team immediately

2.3 SOFTWARE DESCRIPTION

Spring MVC:

Spring MVC follows MVC design pattern which provides clean separation between Model, View and Controller. In Spring MVC there will be one front controller for the entire project. It is called as 'Dispatcher Servlet'. Every request in this application will be forwarded to the dispatcher servlet with the help of web.xml file. Web.xml is called as deployment descriptor. Dispatcher Servlet should be configured as servlet in web.xml and every request should be routed to dispatcher servlet.

In Spring MVC to handle logic for request, we have to create Controller. To create object for controller we have to use @Controller annotation. To invoke the method against each url pattern, we have to use @RequestMapping("/URI") with GET and POST request.

For every module, we need to create one controller class.

For example: For login module there will be LoginController. For Homepage there will be HomePageController.

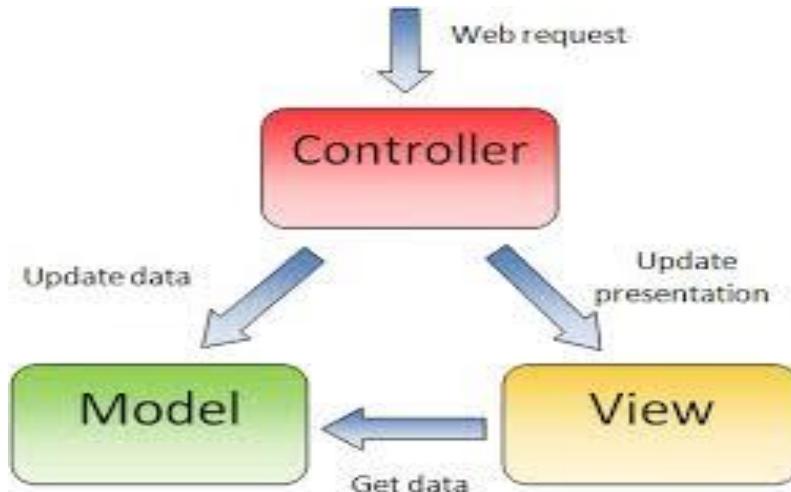


Fig 2.1: Home page controller.

Spring Transactions (commit or rollback): In Business Logic layer, we have to use Spring transactions. Which ensures that every method in business logic layer ensures that transaction is enabled. Transaction maintains unit of work. In one transaction we can call so many queries which belong to one unit of work. If there is any failure in the middle of the transactions all the previous completed work in that transaction will be rolled back. If everything is success within that transaction, all the queries will be committed to database. To enable transactions in business logic layer, we have configured transaction management in spring application Context configuration file. Once after configured, we have to use @Transactional annotation on the top of the method in

The business logic layer.

Transaction contains propagations

- Required (transaction is required. If not it will create new transaction)
- Not required (transaction is not required)
- Requires new (new transaction is required for every method)

- Mandatory (transaction is mandatory. Else it will throw exception)

Hibernate (persistence layer):

Hibernate is an ORM (object relational mapping framework) which maps the java objects with relational (database) tables.

The main advantages of hibernate:

1. No need to write plain SQL queries in java. We can write object oriented programming. Hibernate will generate the sql queries internally
2. For all databases same java logic will be used. This helps java developer to write sample piece of code which will work for any databases(Oracle,MySQL,DB2 etc)
3. by using 'dialect', we can configure which variation of sql queries hibernate will generate.
4. Hibernate provides lot of performance improvement techniques As below
 - Lazy loading
 - Cache mechanisms (primary and secondary)
 - Fetching strategies
 - Object oriented queries.

To configure hibernate we have to use configuration file

Called hibernate.cfg.xml file. For one database we will maintain one hibernate.cfg.xml file. `HibernateConfiguration` contains `sessionFactory` which contains Factory of session object. Session will contain database connection. Session is single threaded.

By using session we will execute queries.

To map java object with database tables, we will use java annotations. They are

- `@Entity`: to register java class with table
- `@Table`: maps the java class with database table. For example to map the java class with user table

```
@Table (name="user")
```

- `@Column`: to map the field in the java class with the columns in the database table
- `@Id` : to map with primary key in the table

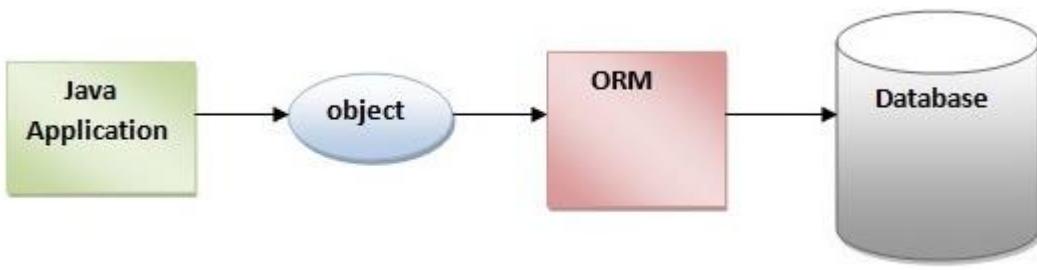


Fig 2.2 Flow of java program

Database (MySQL) :

MySQL is the popular open source database which is used to store the relational data in the database in the form of tables with rows and columns.

MySQL is an RDBMS which follows all the rules as defined

In RDBMS.

For every table in MySQL, we define primary key which is an unique identifier.

To connect with database from java by using hibernate we have to use mainly 4 properties:

1. database driver name
2. database url(type 4 driver)
3. username
4. password

To establish connection between java and database, we have to use above 4 properties. Otherwise database will not allow to connect due to security issues. MySQL follow integrity(data correctness) and normalization rules.



Fig 2.3 MySql database

Server(Tomcat):

Tomcat is an open source web server. Tomcat 7 or 8 support java 7 and above. To run java applications through internet we have to deploy the applications in any one of the servers. Tomcat is one of popular server to use.

Tomcat acts as a container which will manage the life cycle of the objects starting from object creation till destroy.

Tomcat contains mainly 3 directories:

1. webapps : to deploy war file
2. bin : to stop and start tomcat
 - startup.bat
 - shutdown.bat
3. conf: server.xml file for configuration

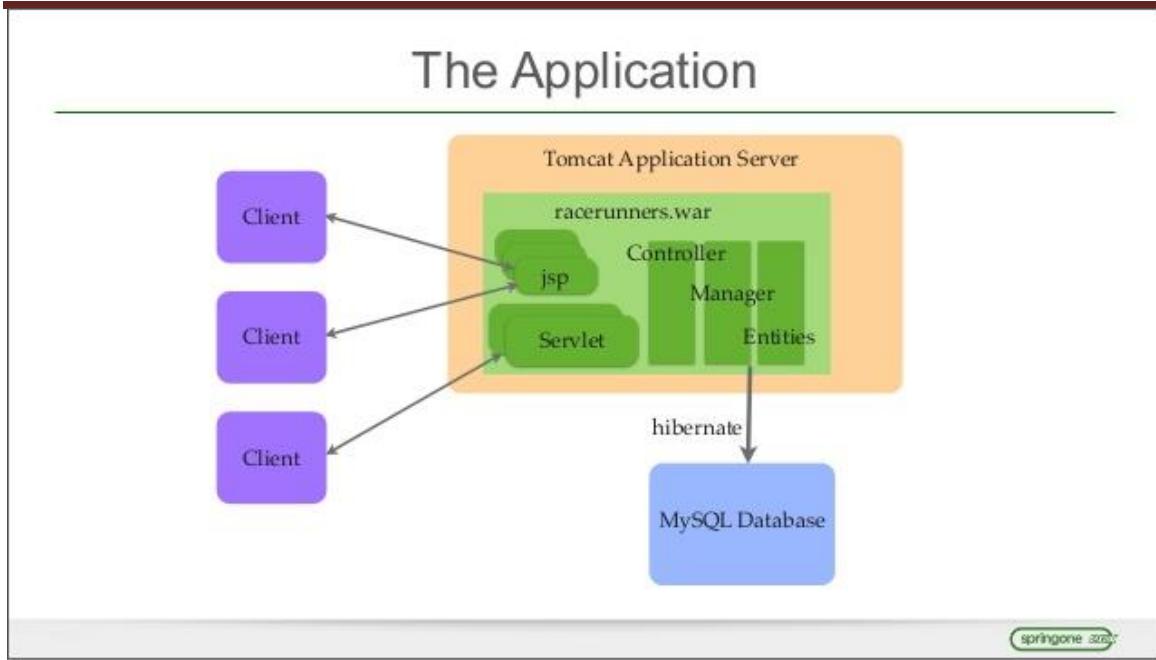


Fig 2.4 Clint Application

JQuery (javascript framework):

JQuery is an open source framework which is implemented on the top of the Java script.

JQuery is mainly implemented on client side(browser) to perform client-side validation.

For example:

- email validation
- password validation
- mandatory field validation

Html5:

For presentation of UI screens, we will use html(hyper text markup language).

For example:

Login screen contains, 2 text fields with Email and Password. Html5 screens are more responsive and support all the popular browsers in the web world

CSS3: (Cascade style sheets):

To apply styles on the top of html screens, we have to use css.

For example: Color, Border, Margin, padding etc.

JSP:

Short for Java Server Page. A server-side technology, Java Server Pages are an extension to the Java servlet technology that was developed by Sun. JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements -- the actual design and display of the page -- to help make the HTML more functional (i.e. dynamic database queries). A JSP is translated into Java servlet before being run, and it processes HTTP requests and generates responses like any servlet. However, JSP technology provides a more convenient way to code a servlet. Translation occurs the first time the application is run. A JSP translator is triggered by the .jsp file name extension in a URL. JSPs are fully interoperable with servlets. You can include output from a servlet or forward the output to a servlet, and a servlet can include output from a JSP or forward output to a JSP. JSPs are not restricted to any specific platform or server. It was originally created as an alternative to Microsoft's ASPs (Active Server Pages). Recently, however, Microsoft has countered JSP technology with its own ASP.NET, part of the .NET initiative.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

3.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

3.2.1 ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. User interface is simple and efficient and easy to use.

3.2.2 MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

New functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project.

Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

3.2.3 SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added.

System can work normally under situations such as low bandwidth and large number of users.

3.2.4 PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another.

Project can be executed under different operation conditions provided it meets its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

3.3 HARDWARE REQUIREMENTS

- Processor : Any Processor above 500 MHz
- RAM : 512Mb
- Hard Disk : 10 GB
- Input device : Standard Keyboard and Mouse
- Output device : VGA and High Resolution Monitor

3.4 SOFTWARE REQUIREMENTS

- Operating system : Windows XP
- Technologies : CoreJava7, Jsp, Servlets, JDBC, Html, Javascript, css
- WebFramework : Spring MVC , Spring ORM , Spring Core , Spring Transactions
- ORM Framework : Hibernate
- Application Server : Tomcat
- Database : MySQL
- IDE Tools : Eclipse

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

4.1.1 INPUT/OUTPUT PRIVACY

No sensitive information from the customer's private data can be derived by the third party. Companies get know the real and fake reviews.

4.1.2 EFFICIENCY

The reviews given on a share has to be transparent to the buyers. The news published on social media, news and decision boards should be real.

4.2 SYSTEM ARCHITECTURE

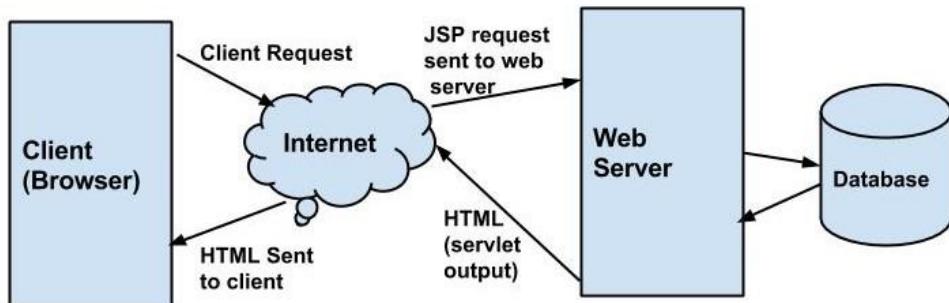


Fig 4.1 MVC Architecture

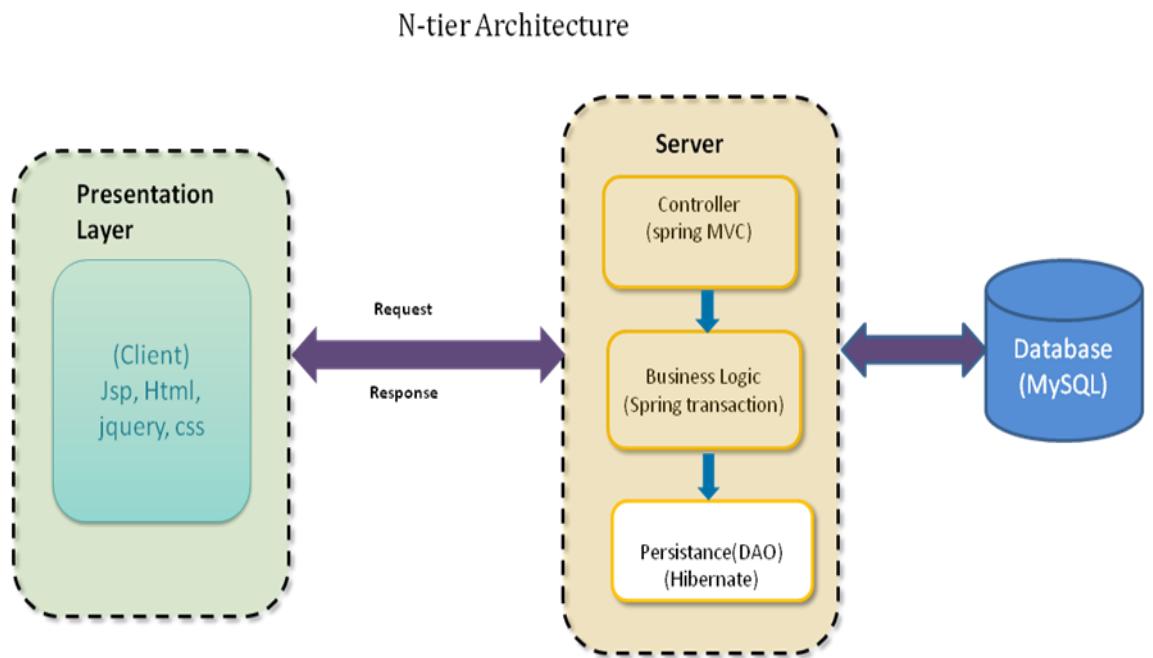


Fig 4.2 N-tier Architecture

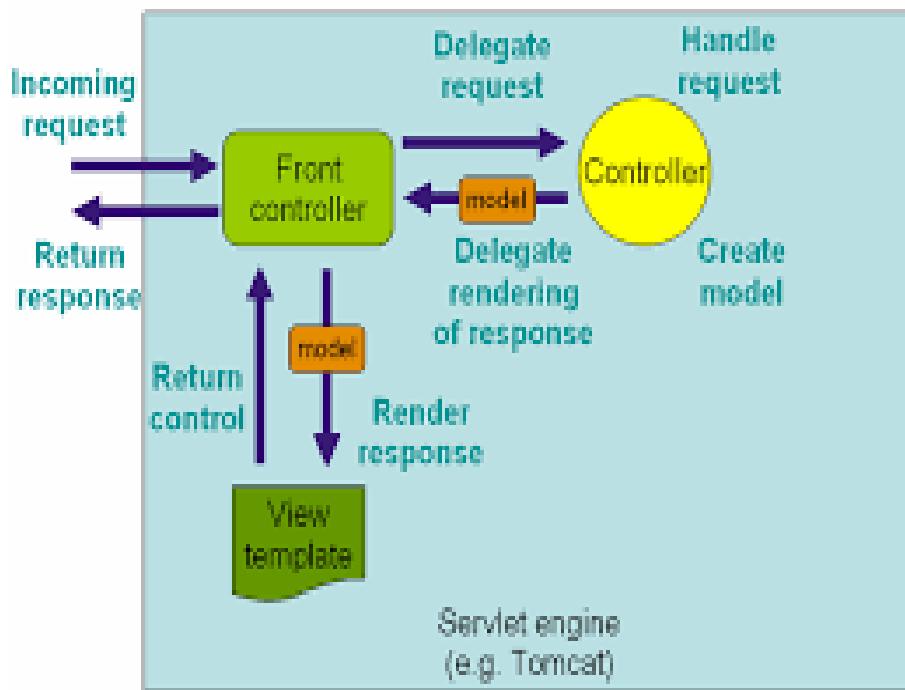


Fig 4.3 Servlet Engine

4.3 SEQUENCE DIAGRAM

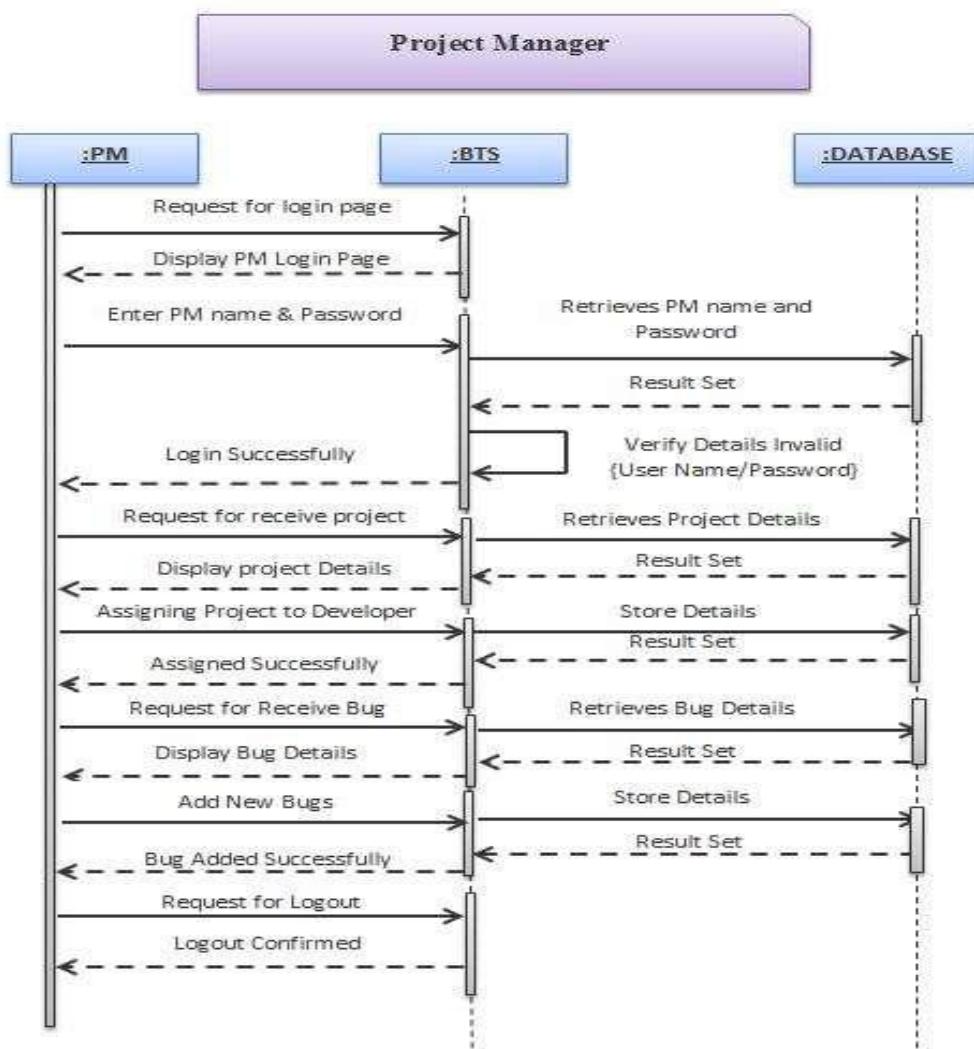


Fig 4.4: Sequence diagram for project manager

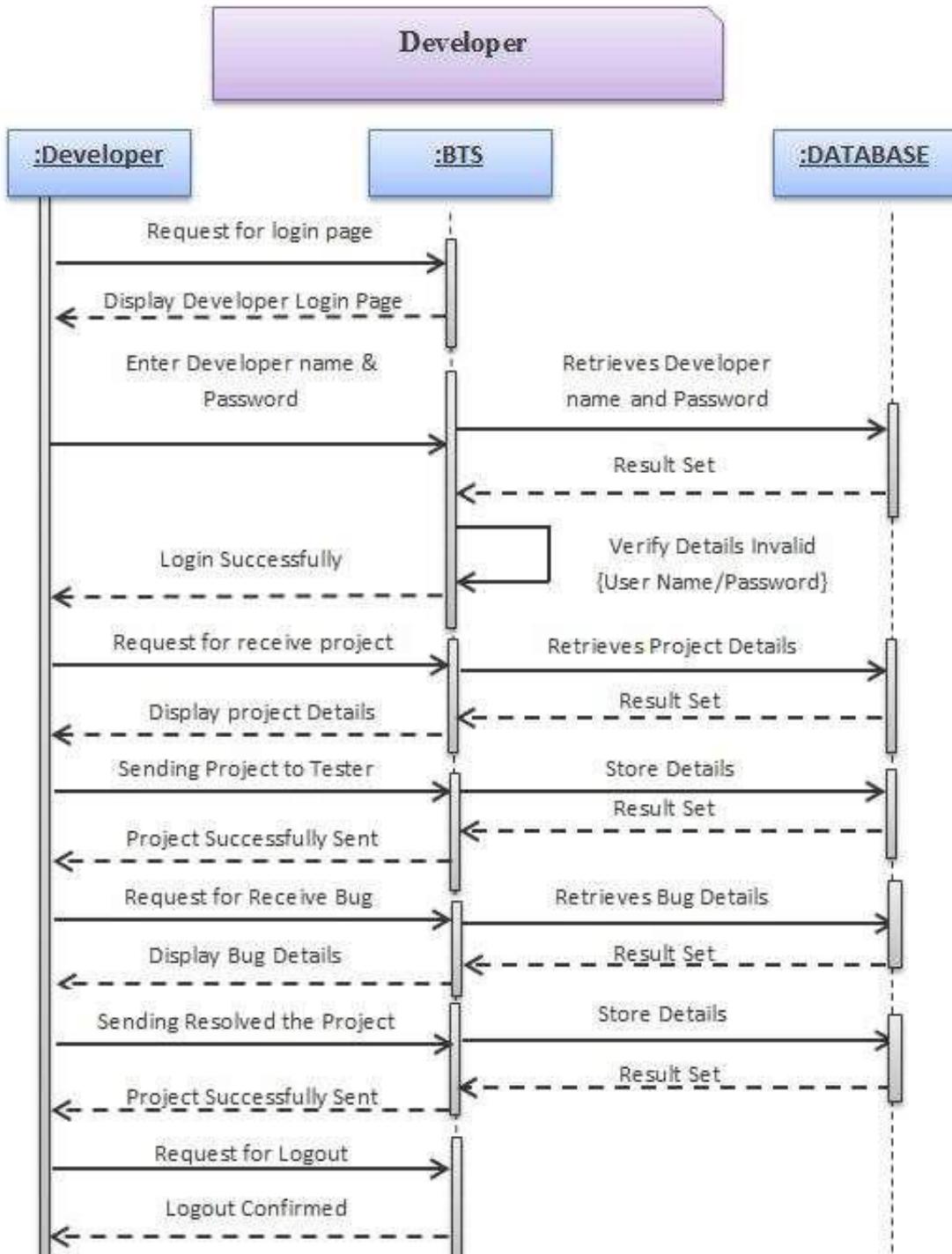


Fig 4.5: Sequence diagram for Developer

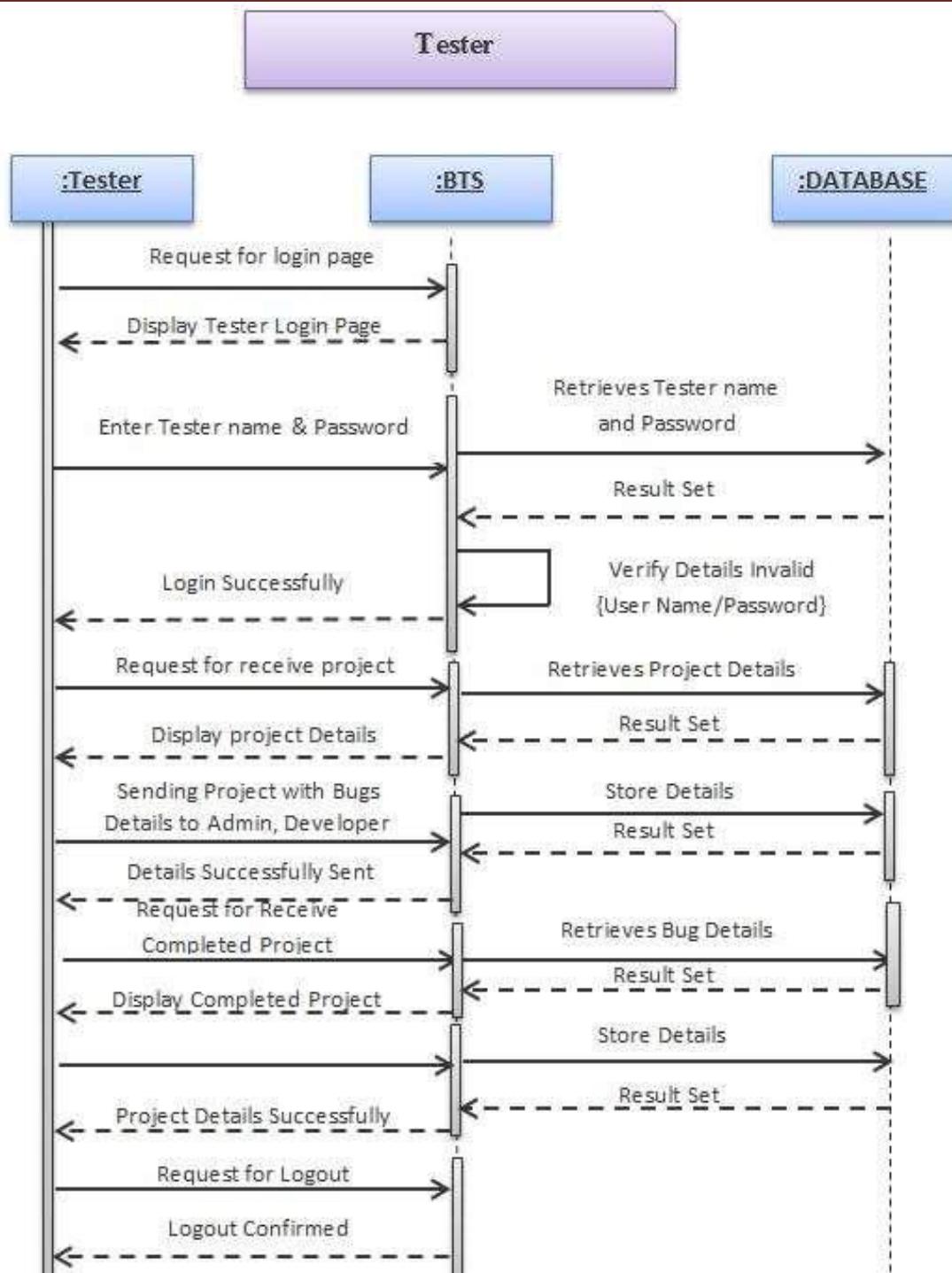


Fig 4.6: Sequence diagram for Tester

4.4 DATA FLOW DIAGRAM

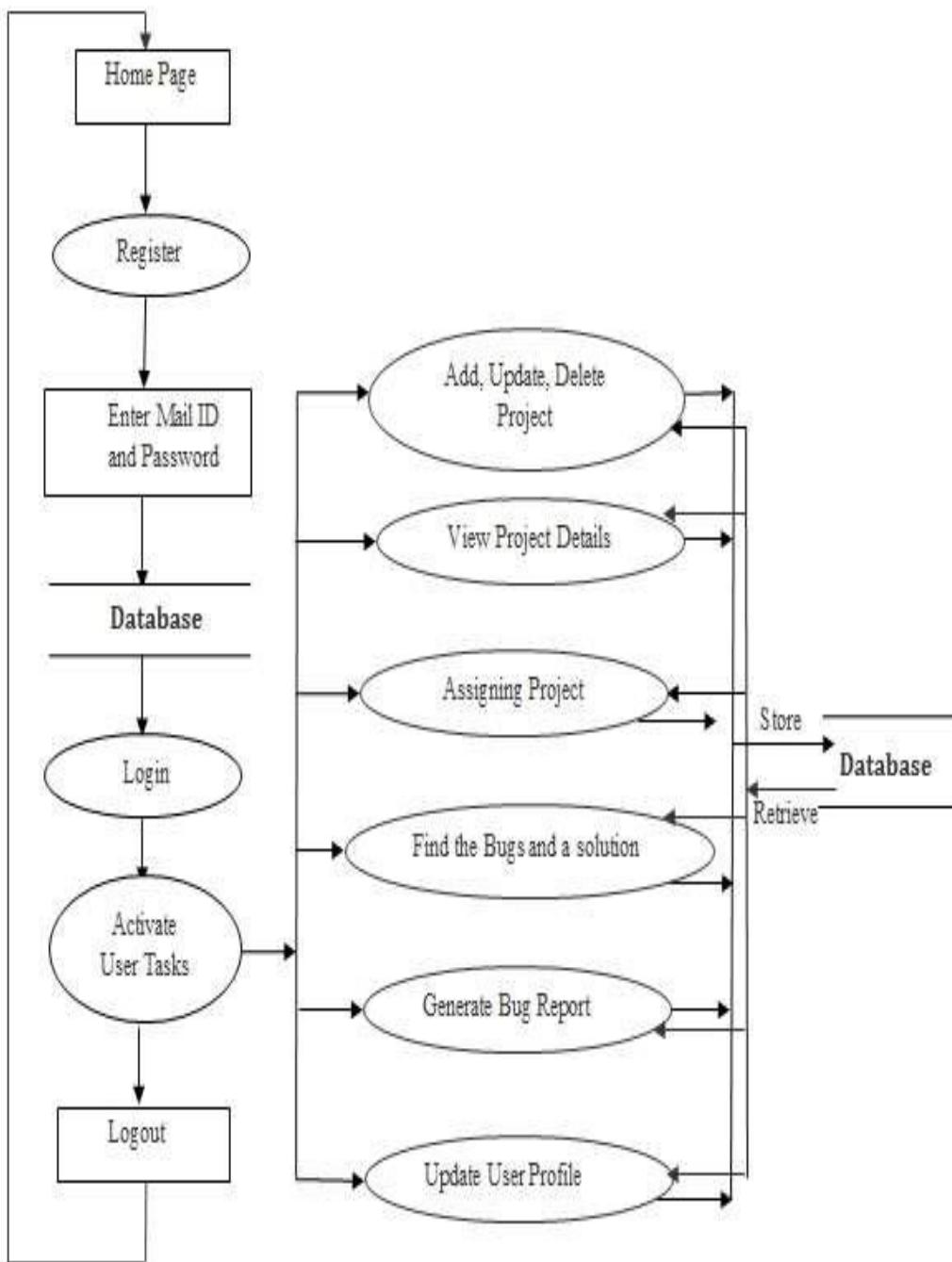


Fig 4.7: DATA FLOW DIAGRAM

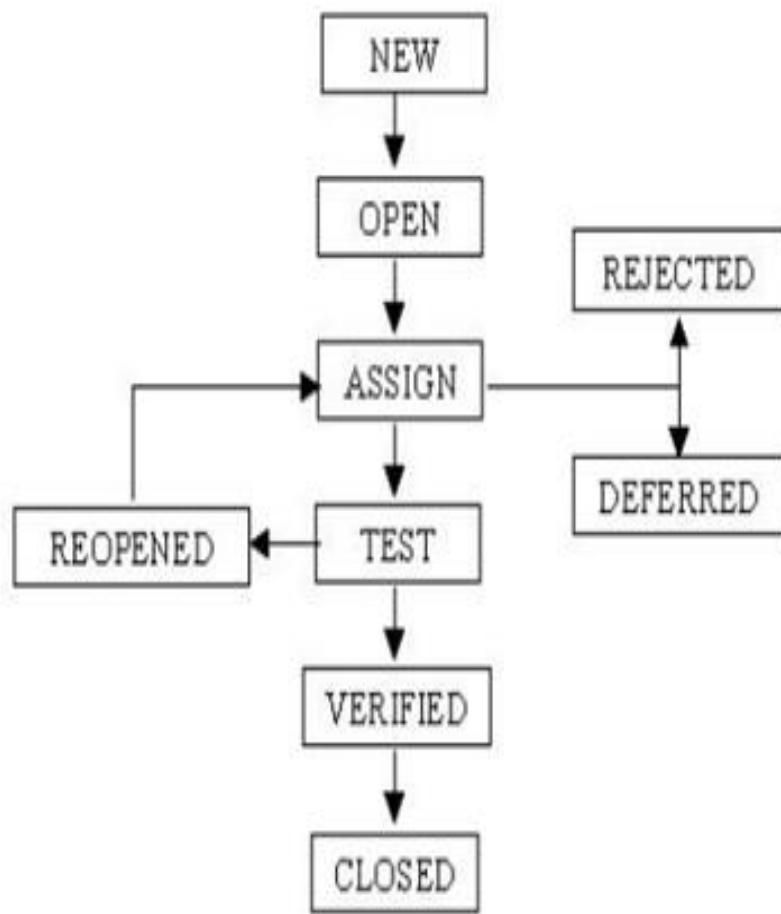


Fig 4.8: FLOW DIAGRAM

4.5 USE CASE DIAGRAM

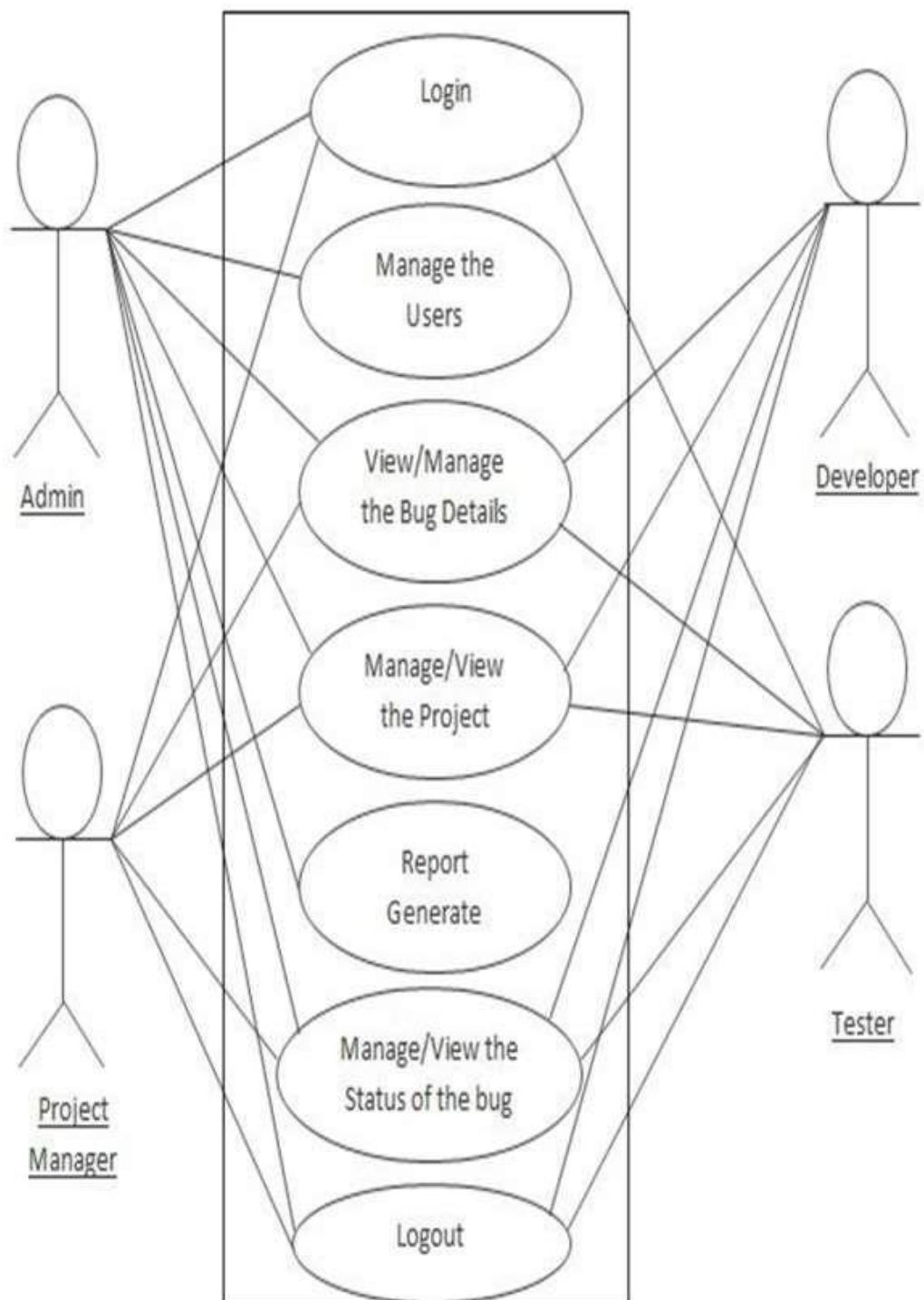


Fig 4.9: Use case diagram for bug tracker

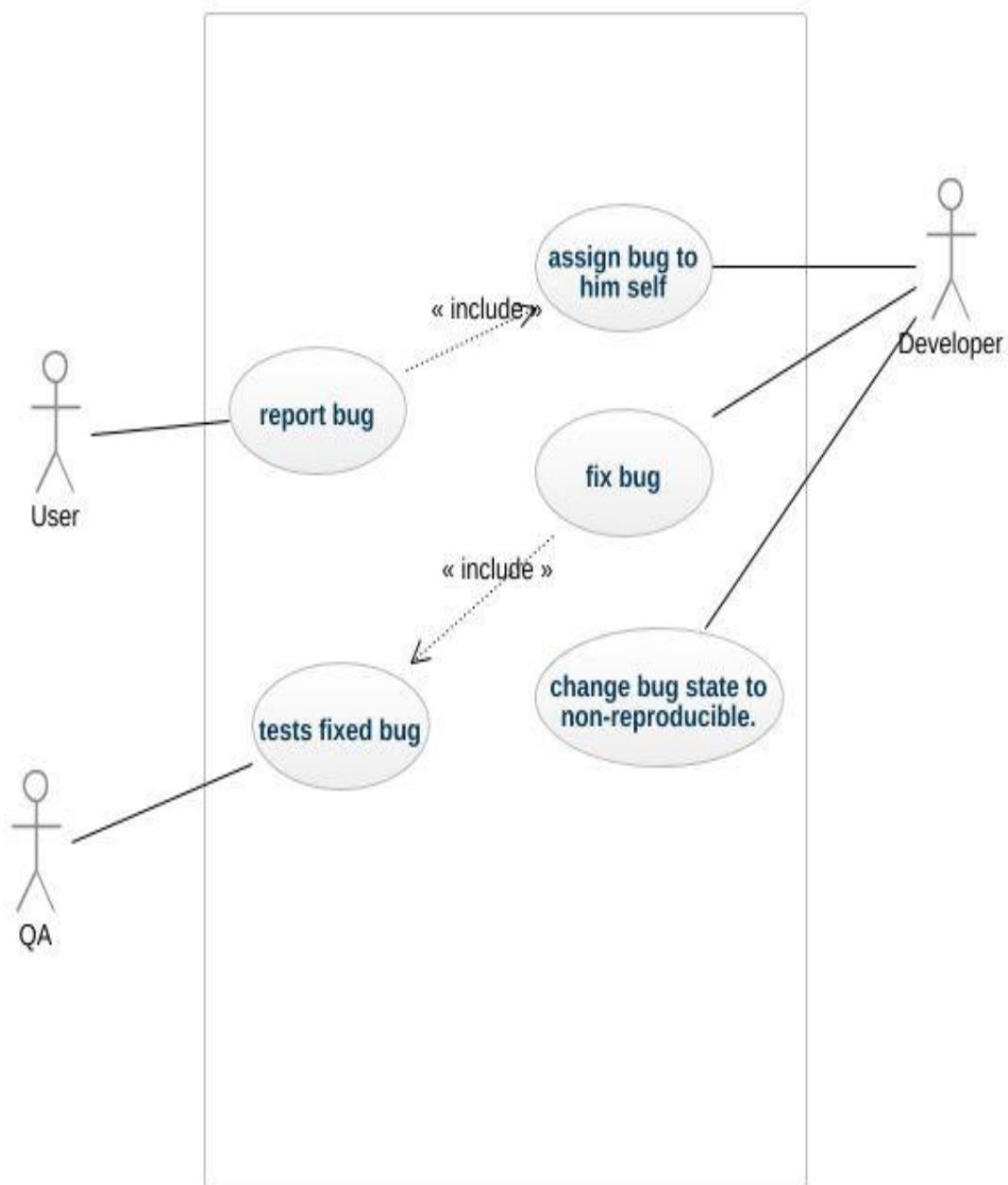


Fig 4.10: Use case diagram for user and developer

4.6 ER DIAGRAM

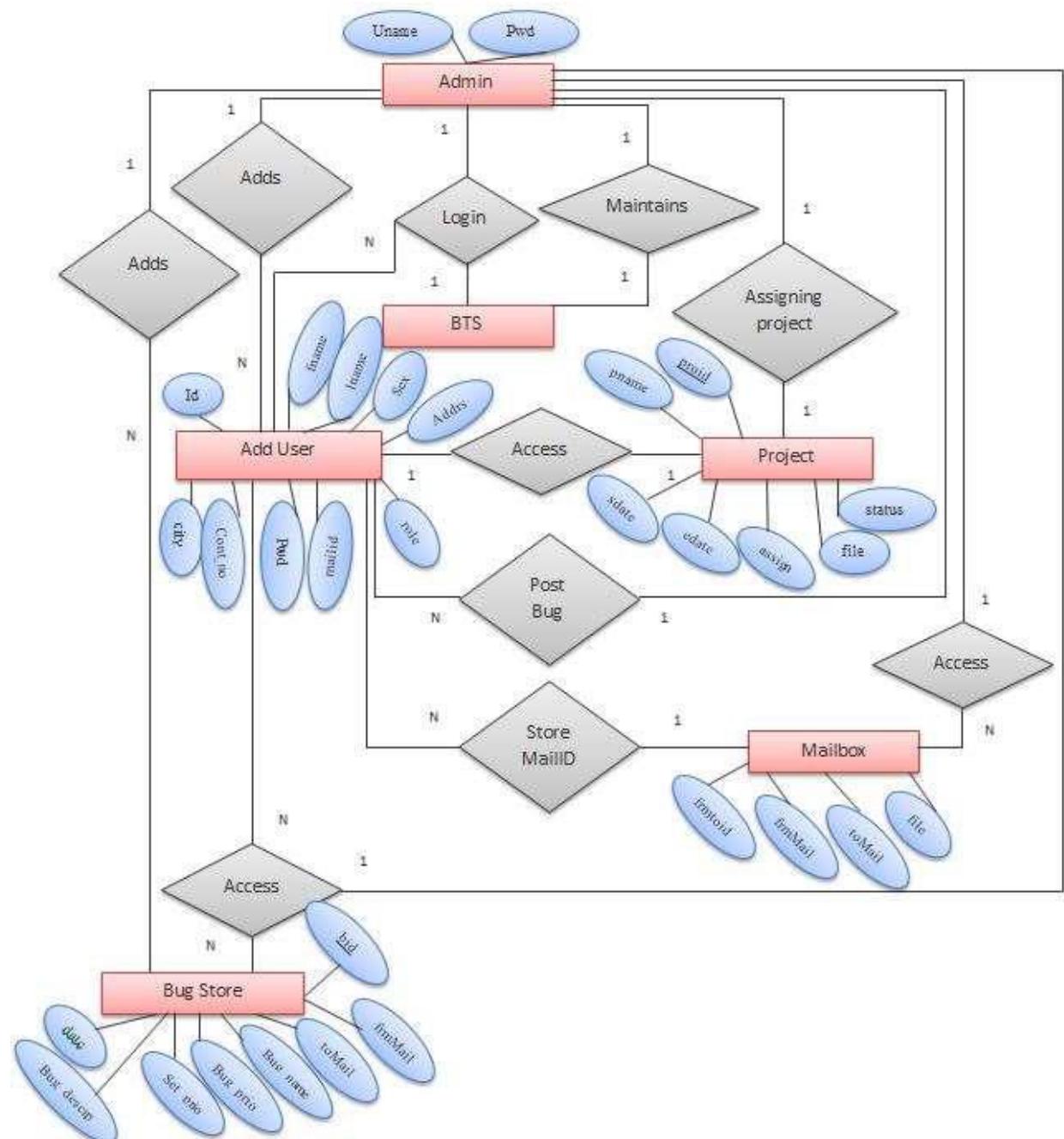
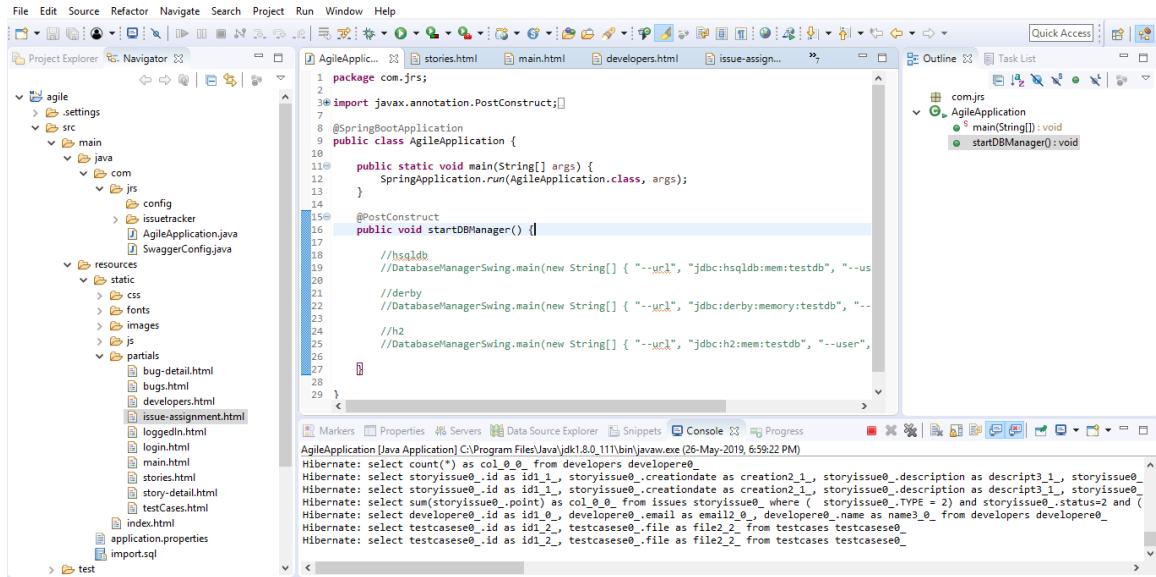


Fig 4.11: ER Diagram for Bug tracker

CHAPTER 5

IMPLEMENTATION

5.1 CODE SNIPPETS

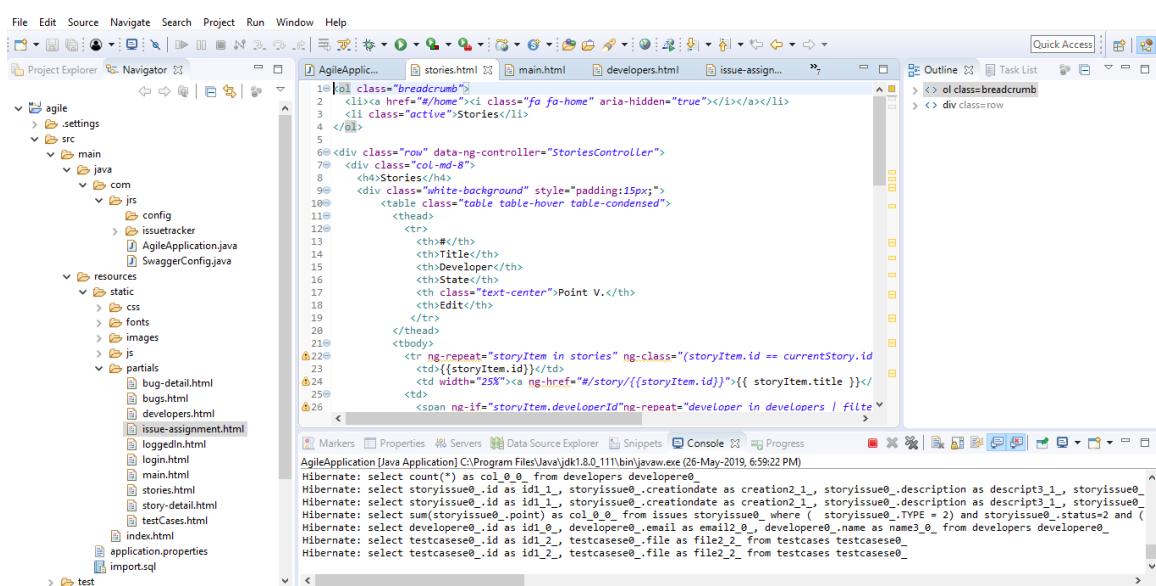


The screenshot shows the Java code for the `AgileApplication` class. The code includes annotations like `@PostConstruct` and `@SpringBootApplication`. It also contains database connection logic using JDBC URLs for MySQL, Derby, and H2 databases.

```

1 package com.jrs;
2
3 import javax.annotation.PostConstruct;
4
5 @SpringBootApplication
6 public class AgileApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(AgileApplication.class, args);
10    }
11
12    @PostConstruct
13    public void startDBManager() {
14
15        //hsqldb
16        //DatabaseManagerSwing.main(new String[] { "--url", "jdbc:hsqldb:mem:testdb", "--user", "sa", "--password" });
17        //derby
18        //DatabaseManagerSwing.main(new String[] { "--url", "jdbc:derby:memory:testdb", "--user", "sa", "--password" });
19        //h2
20        //DatabaseManagerSwing.main(new String[] { "--url", "jdbc:h2:mem:testdb", "--user", "sa", "--password" });
21    }
22}

```



The screenshot shows the HTML code for a story list page. It includes a breadcrumb navigation bar and a table with columns for Title, Developer, State, and Edit. The table uses AngularJS's `ng-repeat` directive to iterate over story items.

```

1<ol class="breadcrumb">
2    <li><a href="#"><i class="fa fa-home" aria-hidden="true"></i></a></li>
3    <li class="active">Stories</li>
4</ol>
5
6<div class="row" data-ng-controller="StoriesController">
7    <div class="col-md-8">
8        <h4>Stories</h4>
9        <div class="white-background" style="padding:15px;">
10            <table class="table table-hover table-condensed">
11                <thead>
12                    <tr>
13                        <th>#</th>
14                        <th>Title</th>
15                        <th>Developer</th>
16                        <th>State</th>
17                        <th class="text-center">Point V.</th>
18                        <th>Edit</th>
19                    </tr>
20                </thead>
21                <tbody>
22                    <tr ng-repeat="storyItem in stories" ng-class="(storyItem.id == currentStory.id ? 'highlight' : '')">
23                        <td>{{storyItem.id}}</td>
24                        <td width="25%"><a href="#/story/{{storyItem.id}}">{{storyItem.title}}</a></td>
25                        <td><span ng-if="storyItem.developerId" ng-repeat="developer in developers | filter: { id: storyItem.developerId }">{{developer.name}}</span></td>
26                        <td><span ng-class="storyItem.state === 'open' ? 'label-warning' : 'label-success'">{{storyItem.state}}</span></td>
27                        <td class="text-center"><button ng-click="edit(storyItem)">Edit</button></td>
28                    </tr>
29                </tbody>
30            </table>
31        </div>
32    </div>
33</div>

```

The screenshot shows an IDE interface with two main panes. The left pane displays the project structure for 'AgileApplication' under 'src/main/java/com/jrs/issuetracker'. The right pane shows the code for 'developer.html'.

```

1<div class="row">
2  <div class="text-left col-sm-6 col-md-3 col-lg-3 col-xl-2 no-space" style="padding: 0px">
3    <div class="thumbnail">
4      <div class="text-center" style="font-size:36pt;">
5        <i class="fa fa-users"></i>
6      </div>
7      <div class="caption" style="padding:20px;">
8        <h5 class="post-caption text-center">
9          <a href="#/developer">Developers</a>
10         </h5>
11         <p style="font-size:14px;">
12           You can add, remove or update developers in this section.
13           <a href="#/developer">Go</a><br><br>
14         </p>
15       </div>
16     </div>
17   </div>
18   <div class="text-left col-sm-6 col-md-3 col-lg-3 col-xl-2 no-space" style="padding: 0px">
19     <div class="thumbnail">
20       <div class="text-center" style="font-size:36pt;">
21         <i class="fa fa-bug"></i>
22       </div>
23       <div class="caption" style="padding:20px;">
24         <h5 class="post-caption text-center">
25           <a href="#/bug">Bugs</a>
26         </h5>

```

The code uses Bootstrap's grid system and includes links to 'Developers' and 'Bugs' sections.

The screenshot shows an IDE interface with two main panes. The left pane displays the project structure for 'AgileApplication' under 'src/main/java/com/jrs/issuetracker'. The right pane shows the code for 'issue-assignment.html'.

```

1<div class="row" data-ng-controller="DevelopersController">
2  <div class="col-md-6">
3    <h4 class="text-center">Team Members</h4>
4    <div class="white-background" style="padding:15px;">
5      <table class="table table-hover table-condensed">
6        <thead>
7          <tr>
8            <th>#</th>
9            <th>Name</th>
10           <th>Actions</th>
11         </tr>
12       </thead>
13       <tbody>
14         <tr ng-repeat="developer in developers" ng-class="(developer.id == currentDeveloper.id) ? 'active' : ''">
15           <td>{{developer.id}}</td>
16           <td width="80%">{{developer.name}}</td>
17           <td width="5%" style="text-align:center;">
18             <a class="btn btn-xs btn-success" ng-click="selectDeveloper(developer)">i</a>
19           </td>
20         </tr>
21       </tbody>
22     </table>
23   </div>
24 </div>
25 </div>
26 </div>

```

The code uses AngularJS's ng-repeat directive to generate a table of team members, with one row highlighted as active.

The screenshot shows an IDE interface with two tabs open: `AgileApplication.java` and `import.sql`. The `AgileApplication.java` tab displays Java code for creating tables and inserting initial data into the database. The `import.sql` tab shows the raw SQL commands used to create the database schema and populate it with sample data.

```

Connection profile: Name: Database: Status: Disconnected, Auto Commit
Type: Name: Database: Status: Disconnected, Auto Commit

1 -- these tables need to be created
2 CREATE TABLE ISSUES (ID INTEGER NOT NULL IDENTITY, TITLE VARCHAR(50) NOT NULL, DESCRIPTION
CREATE TABLE USERS (ID INTEGER NOT NULL IDENTITY, TITLE VARCHAR(50) NOT NULL, EMAIL VARCHAR
4
5 -- create users:
6 -- INSERT INTO USERS (name) VALUES('test@test.com');
7
8
9 -- ass developers
10 INSERT INTO DEVELOPERS (name,email) VALUES('Project Manager','pm@test.com');
11 INSERT INTO USERS (email,password) VALUES('pm@test.com','admin');
12 -- INSERT INTO DEVELOPERS (name) VALUES('Test Developer 2');
13 -- INSERT INTO DEVELOPERS (name) VALUES('Test Developer 3');
14
15
16
17 --add bugs
18 INSERT INTO ISSUES (ID, TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
19 INSERT INTO ISSUES (ID, TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
20 INSERT INTO ISSUES (ID, TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
21 INSERT INTO ISSUES (ID, TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)

-- SQL Import Script (import.sql)
-- This file contains the SQL statements to create the database schema and insert initial data.
-- It includes CREATE TABLE statements for ISSUES and USERS, and INSERT statements for DEVELOPERS and ISSUES.

-- Create ISSUES table
CREATE TABLE ISSUES (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    TITLE NVARCHAR(50) NOT NULL,
    DESCRIPTION NVARCHAR(MAX),
    CREATIONDATE DATETIME,
    DEVELOPERID INT,
    TYPE NVARCHAR(50),
    STATUS NVARCHAR(50),
    PRIORITY NVARCHAR(50)
);

-- Create USERS table
CREATE TABLE USERS (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    TITLE NVARCHAR(50) NOT NULL,
    EMAIL NVARCHAR(100) NOT NULL,
    PASSWORD NVARCHAR(100)
);

-- Insert initial data into DEVELOPERS
INSERT INTO DEVELOPERS (name, email) VALUES ('Project Manager', 'pm@test.com');
INSERT INTO USERS (email, password) VALUES ('pm@test.com', 'admin');

-- Insert initial data into ISSUES
INSERT INTO ISSUES (TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
VALUES ('Feature Request 1', 'Detailed description of Feature Request 1.', GETDATE(), 1, 'Feature', 'Open', 'High');
INSERT INTO ISSUES (TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
VALUES ('Bug Fix 1', 'Detailed description of Bug Fix 1.', GETDATE(), 1, 'Bug', 'Open', 'High');
INSERT INTO ISSUES (TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
VALUES ('Feature Request 2', 'Detailed description of Feature Request 2.', GETDATE(), 2, 'Feature', 'Open', 'Medium');
INSERT INTO ISSUES (TITLE, DESCRIPTION, CREATIONDATE, DEVELOPERID, TYPE, STATUS, PRIORITY)
VALUES ('Bug Fix 2', 'Detailed description of Bug Fix 2.', GETDATE(), 2, 'Bug', 'Open', 'Medium');

-- Insert initial data into TESTCASES
INSERT INTO TESTCASES (NAME, DESCRIPTION, FILE, STATUS, PRIORITY)
VALUES ('Test Case 1', 'Detailed description of Test Case 1.', 'file1.txt', 'Open', 'High');
INSERT INTO TESTCASES (NAME, DESCRIPTION, FILE, STATUS, PRIORITY)
VALUES ('Test Case 2', 'Detailed description of Test Case 2.', 'file2.txt', 'Open', 'High');

```

The screenshot shows an IDE interface with the `issue-assignment.html` file open. The code is an AngularJS template (`.html`) containing HTML, CSS, and JavaScript logic. It includes a breadcrumb navigation bar, a summary section, and a table for displaying assignment results. The table uses `ng-repeat` to iterate over assignment variables and `ng-click` to trigger an assignment function.

```

<div data-ng-controller="IssueAssignmentController">
  <ol class="breadcrumb">
    <li><a href="#">Home</a></li>
    <li class="active">Issue Assignment</li>
  </ol>
  <div class="row" style="margin-top:20px;">
    <div class="col-md-5 white-background" style="padding:30px;">
      <h3>Summary for assignment</h3>
      <table class="table">
        <tr ng-repeat="var in assignmentVariables"><td>{{var}}</td></tr>
      </table>
      <hr>
      <div ng-click="makeAssignment()" class="btn btn-info btn-sm">Make Auto Assignment</a>
    </div>
    <div class="col-md-1"></div>
    <div class="col-md-6 white-background" style="padding:30px;">
      <h3>Assignment Result</h3>
      <div ng-repeat="(key, value) in assignmentResult">
        <div style="color:#A6A4BD">Key #{{key}}</div>
        <table class="table">
          <thead>
            <tr>
              <th>Issue-ID</th>
              <th>Story</th>
              <th>Developer</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>{{value.issueId}}</td>
              <td>{{value.story}}</td>
              <td>{{value.developer}}</td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>

```

CHAPTER 6

SOFTWARE INSTALLATION

6.1 JDK INSTALLATION

Step 1: Go to the official website of oracle using the below link to download jdk and jre(inbuilt).

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

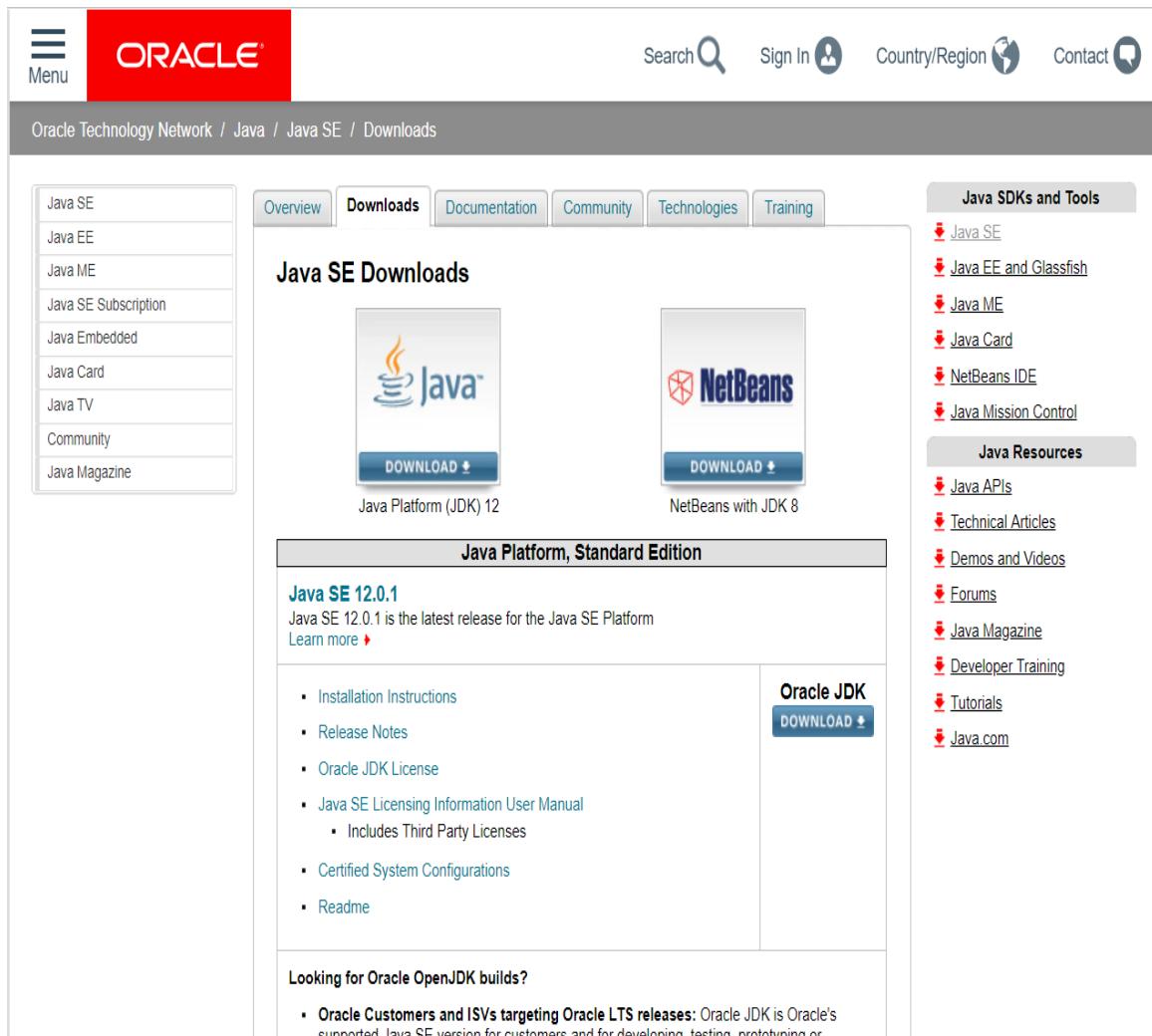


Figure No 6.1: Oracle website

Step 2: click on the latest JDK version

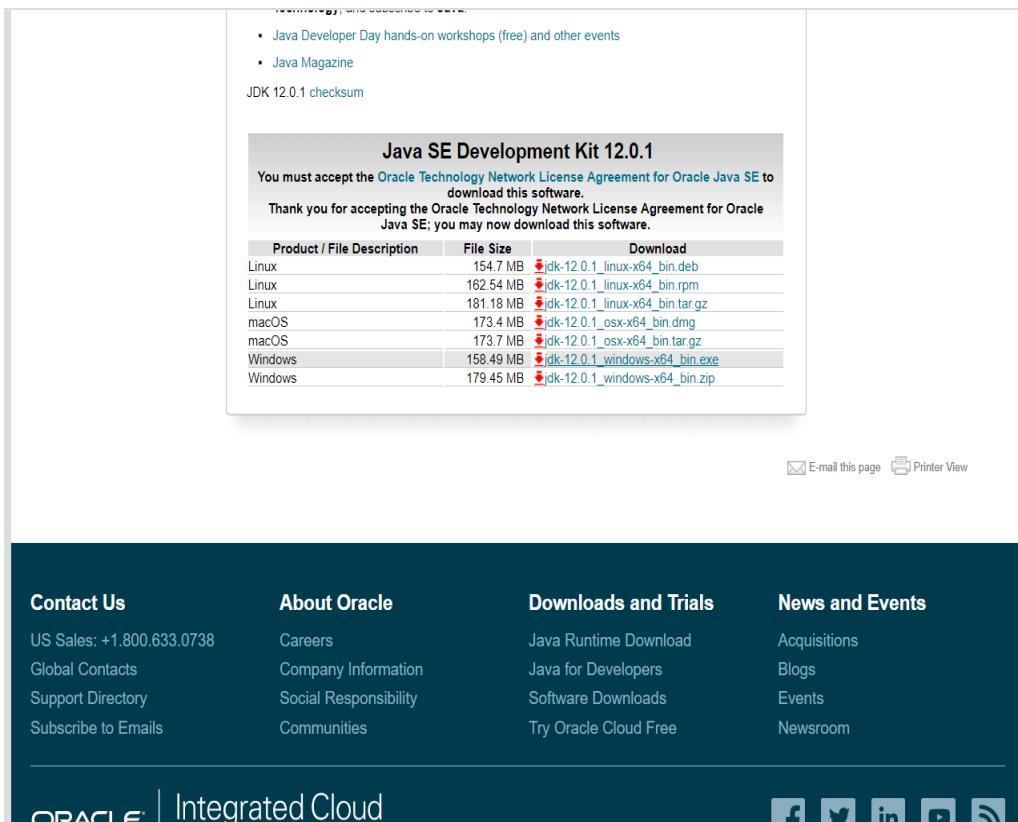


Figure No 6.2:- JDK download

Step 3: Install the Jdk once after the download is complete.

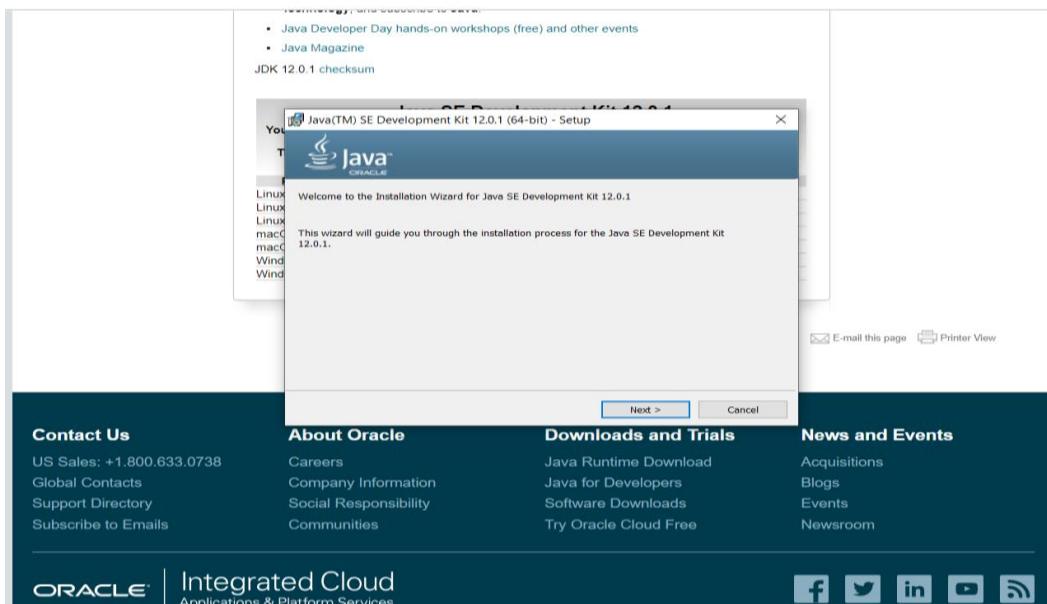


Figure No 6.3:- JDK installation

Step 4: set environment variable

Right click on computer=> properties

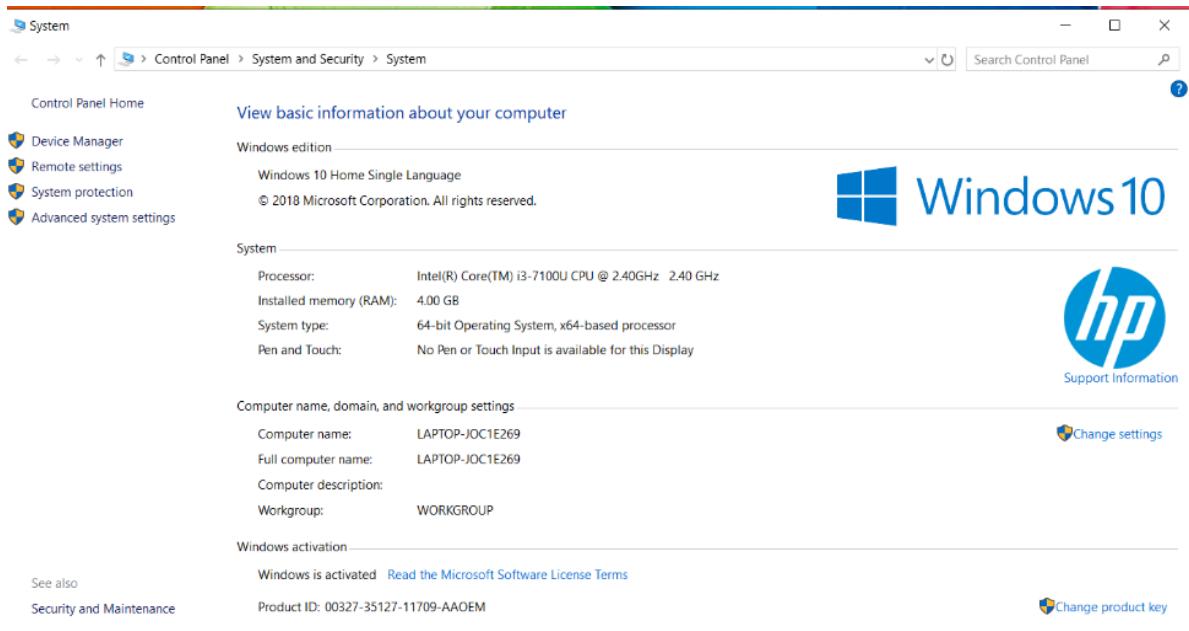


Figure No 6.4:- System window

Select Advanced system setting=> Environment Variables.

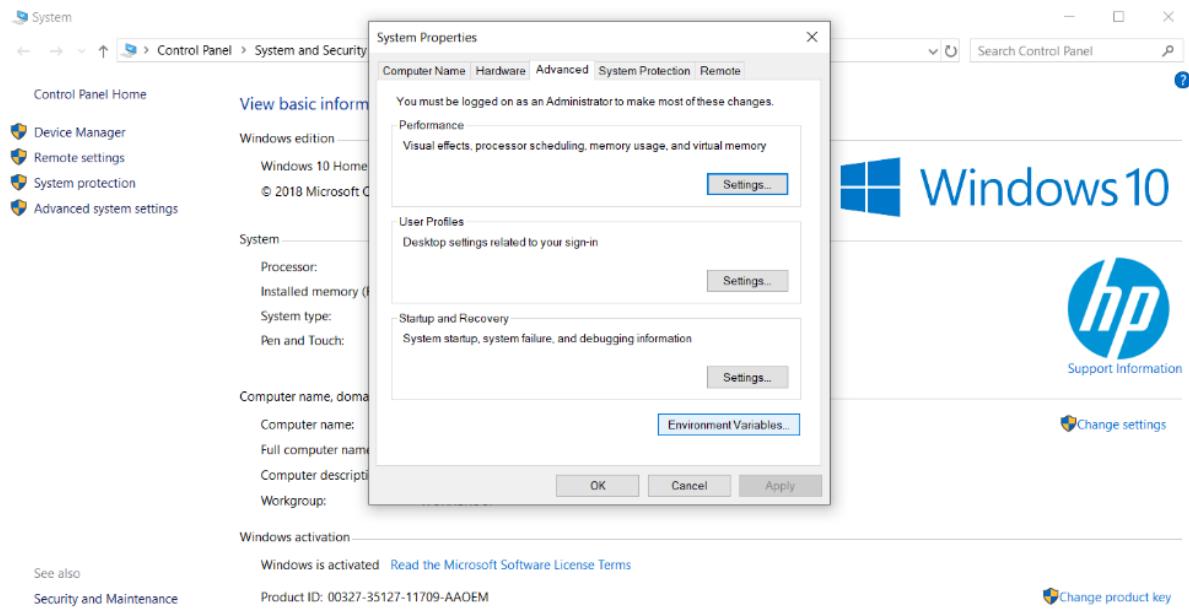


Figure No 6.5:- Environment Variable

Create a new path variable.

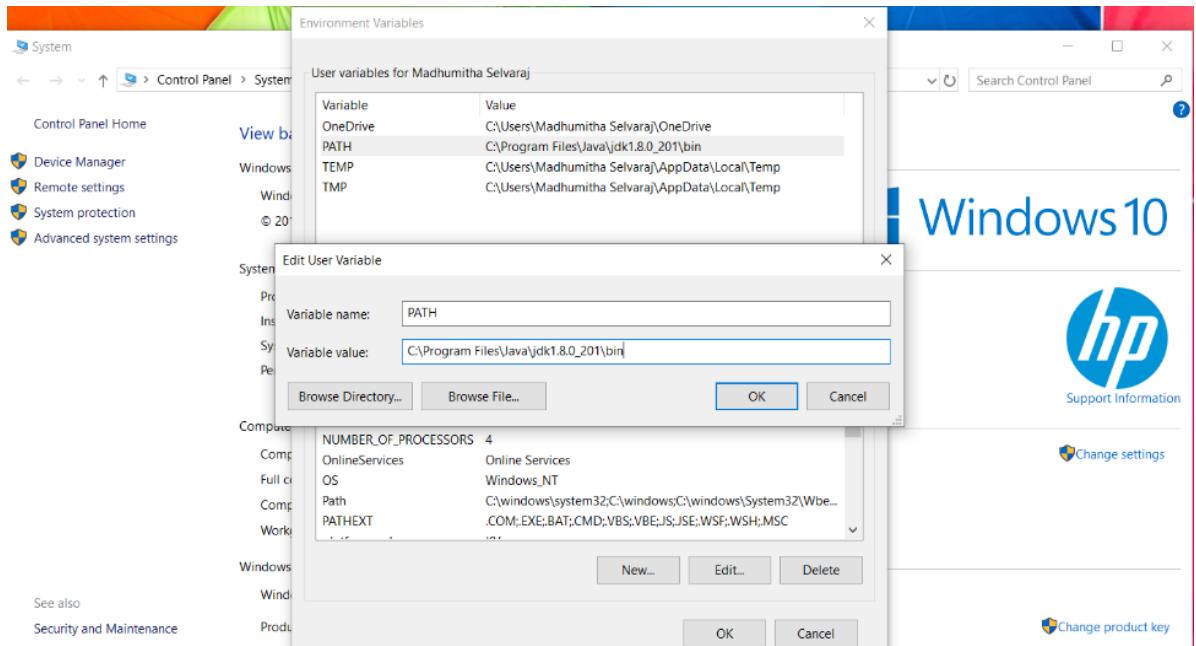


Figure No 6.6:- Path creation

Open command prompt and execute the command JAVAC.

```
C:\> javac
  Usage: javac [options] <mainclass> [args...]
          (to execute a class)
  or  java [options] -jar <jarfile> [args...]
          (to execute a jar file)
  or  java [options] -m <module>/<mainclass> [args...]
      java [options] --module <module>/<mainclass> [args...]
          (to execute the main class in a module)
  or  java [options] <sourcefile> [args]
          (to execute a single source-file program)

  Arguments following the main class, source file, -jar <jarfile>,
  -m or --module <module>/<mainclass> are passed as the arguments to
  main class.

  where options include:

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
  -cclasspath <class search path of directories and zip/jar files>
    A ; separated list of directories, JAR archives,
    and ZIP archives to search for class files.
  -p <module path>
  -module-path <module path>...
    A ; separated list of directories, each directory
    is a directory of modules.
  --upgrade-module-path <module path>...
    A ; separated list of directories, each directory
    is a directory of modules that replace upgradeable
    modules in the runtime image
  --add-modules <module name>[,<module name>...]
    root modules to resolve in addition to the initial module.
    <module name> can also be ALL-DEFAULT, ALL-SYSTEM,
    ALL-MODULE-PATH.
  --list-modules
    list observable modules and exit
  -d <module name>
  --describe-module <module name>
    describe a module and exit
  --dry-run
    create VM and load main class but do not execute main method.
    The --dry-run option may be useful for validating the
    command-line options such as the module system configuration.
  --validate-modules
    validate all modules and exit
    The --validate-modules option may be useful for finding
    conflicts and other errors with modules on the module path.
  -Dname=<value>
```

Figure No 6.7:- Command prompt

6.2 Eclipse Installation

Browse to the below link to download Eclipse Oxygen and click on the download icon.

<https://www.eclipse.org/oxygen/>

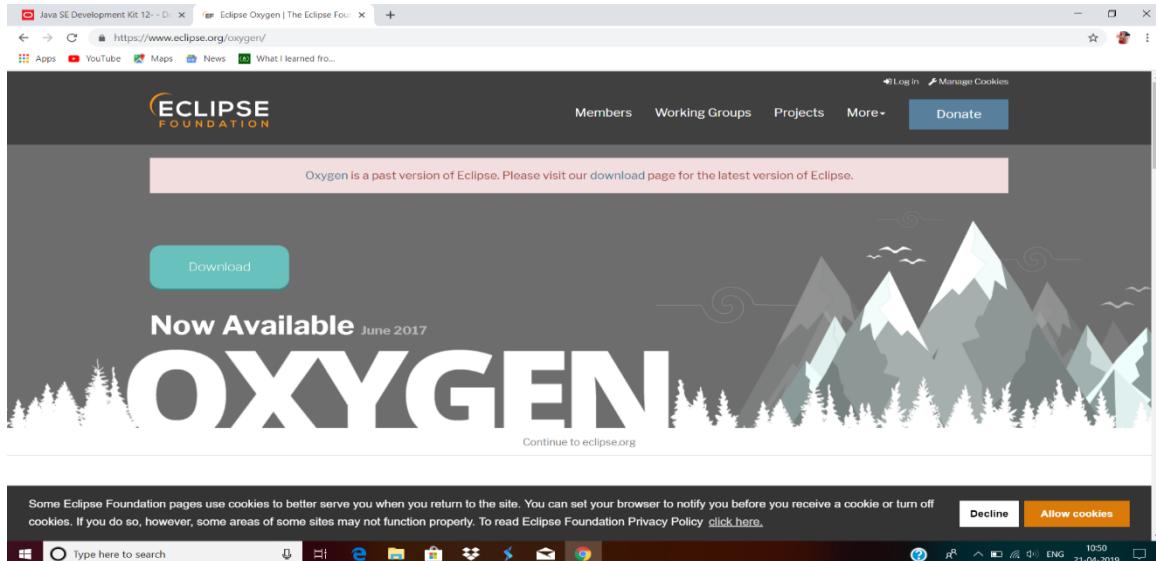


Figure No 6.8:- Eclipse website

Once after the download is complete click on the exe file to install it.

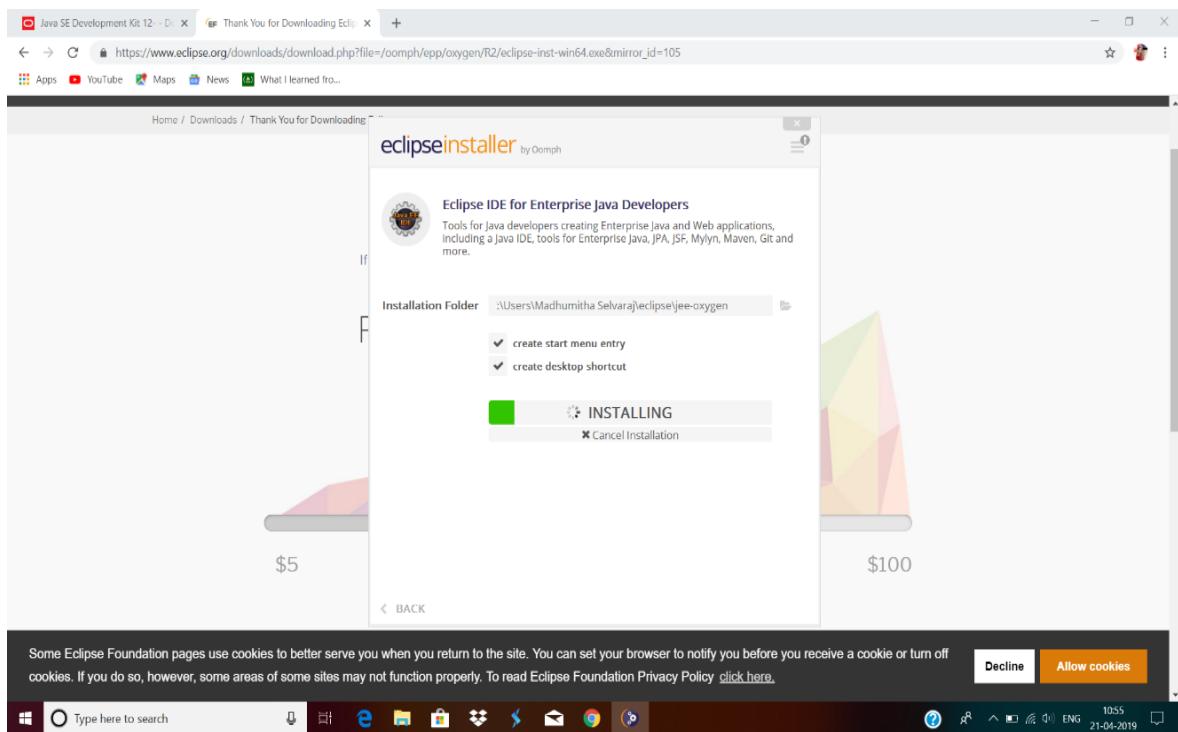


Figure No 6.9:- Eclipse installer

Accept the terms and conditions.

Click on remember accepted license => accept.

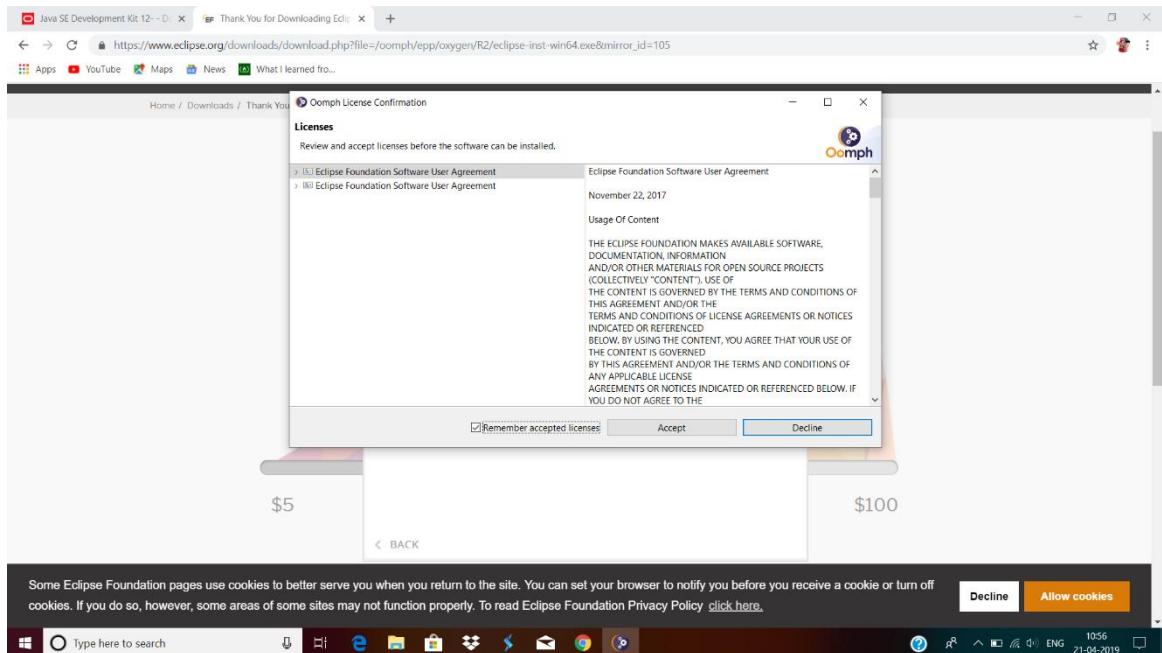


Figure No 6.10 Terms and Conditions

After installation click on launch to open the IDE, select path to save the project files.

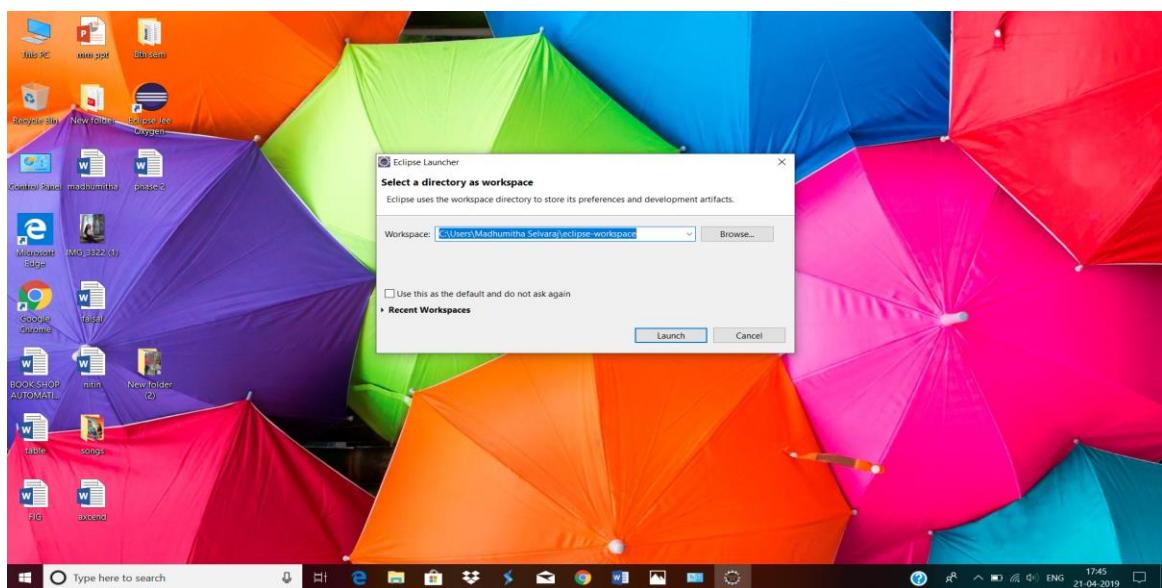


Figure No 6.11:- IDE

6.3 Tomcat Installation

Browse to the official apache website in order to download Tomcat server.

<https://tomcat.apache.org/download-90.cgi>

Click on the desired version of Tomcat from the left panel.

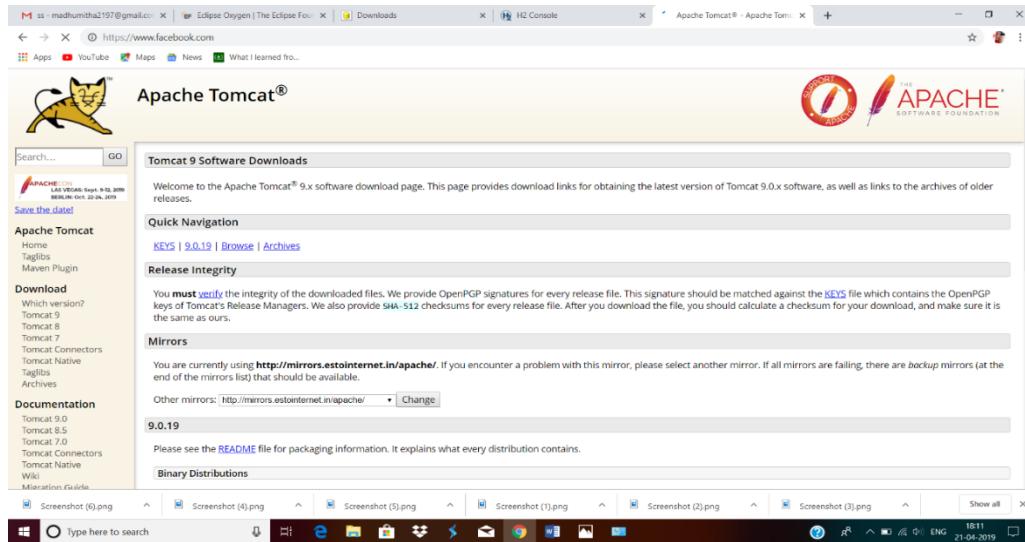


Figure No 6.12:- Apache website

Click on windows service installer, once after the download is complete run it.

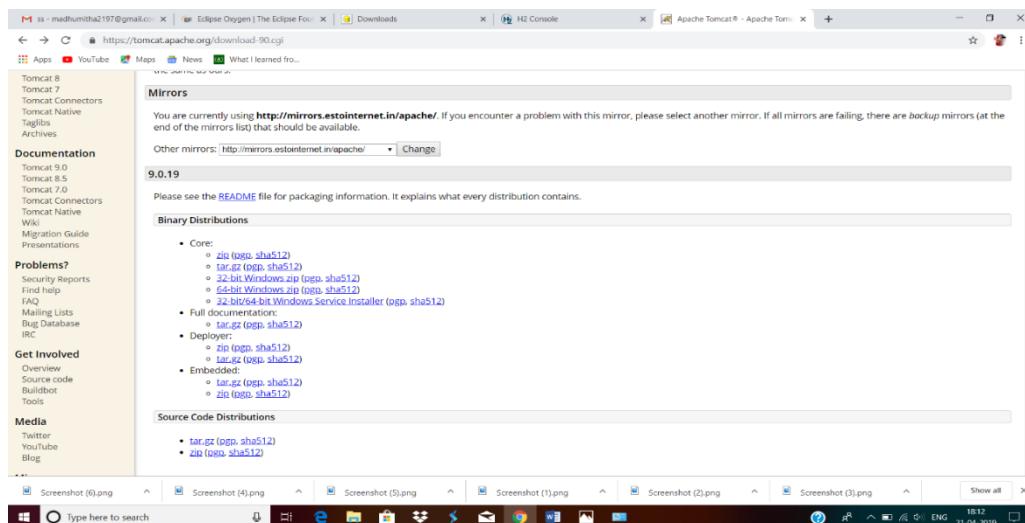


Figure No 6.13:- Downloads

After installation the tomcat server icon will be present in the down right panel.

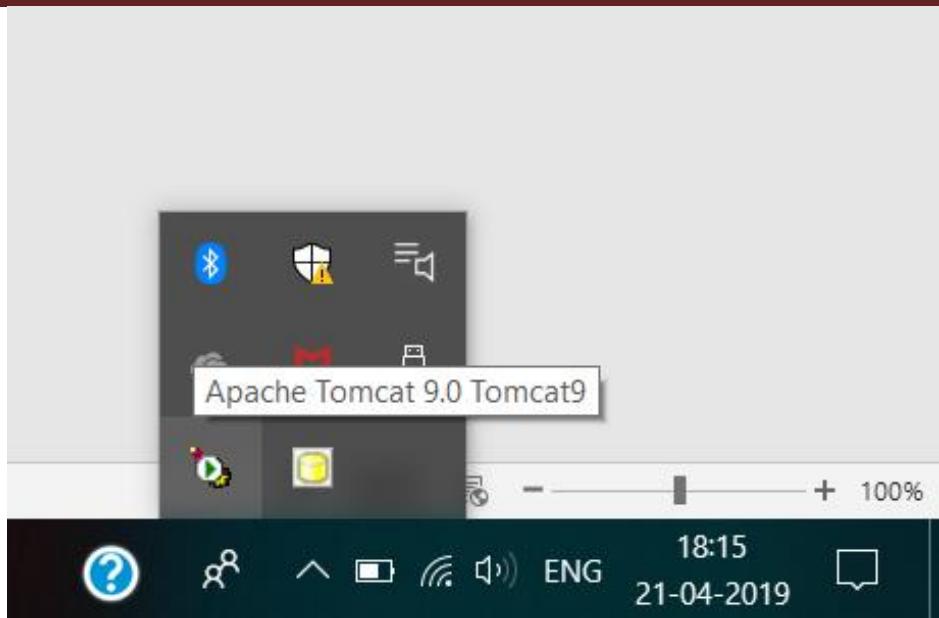


Figure No 6.14:- Apache Tomcat

Open browser and type Localhost:8580

And the below page will be displayed.

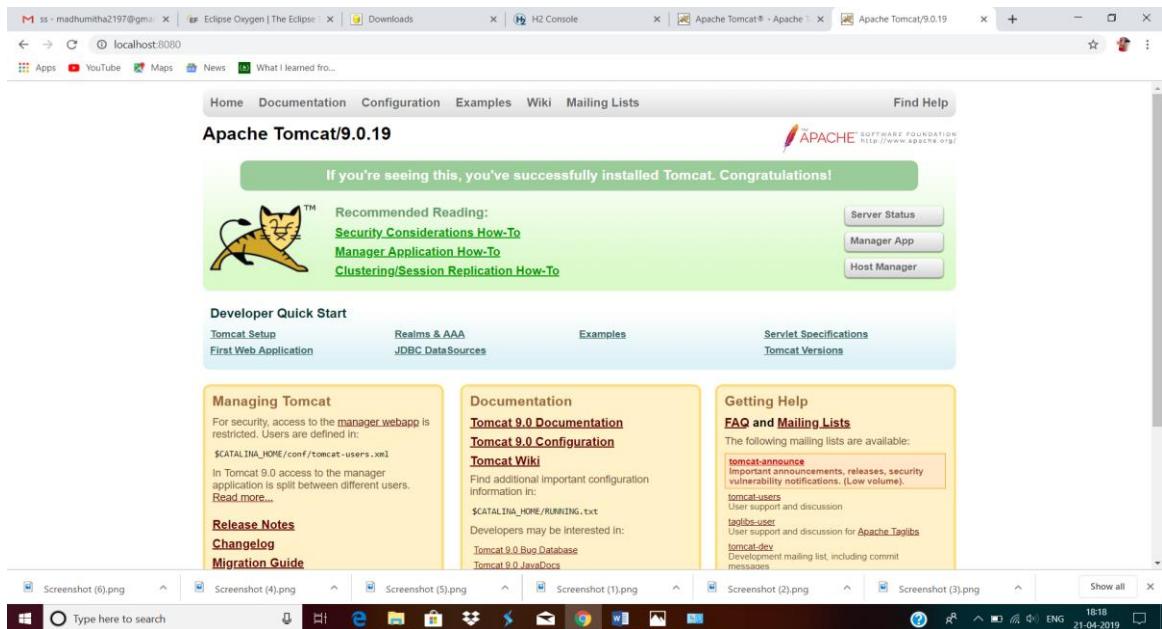
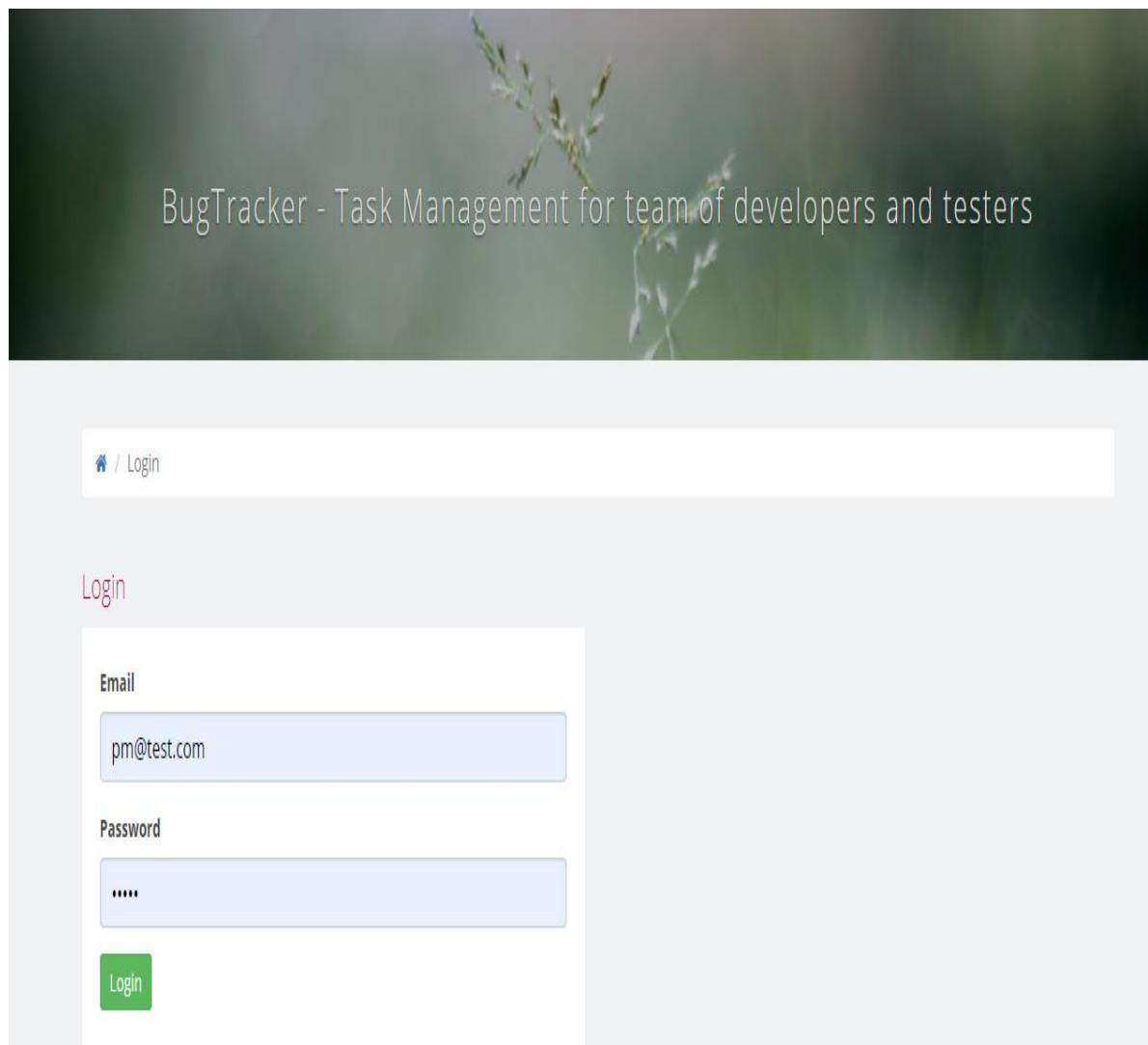


Figure No 6.15:- successful installation

CHAPTER 7

SCREEN SHOTS

7.1 Login the Bug tracking system web page



7.2 Home Page

HOME TEAM TASKS OR BUGS STORIES ASSIGN TEST CASES LOGOUT

BugTracker - Task Management for team of developers and testers

Developers
You can add, remove or update developers in this section. Go

Bugs
You can manage bugs in this section Go

Stories
You can manage stories in this section Go

Assign
Assign Tasks(Bugs) and stories to developers Go

7.3 adding new team member

HOME TEAM TASKS OR BUGS STORIES ASSIGN TEST CASES LOGOUT

BugTracker - Task Management for team of developers and testers

[/ Developers](#)

Team Members

#	Name	Actions
1	Project manager	

New Team Member

Name: Nitin

Email: nitinkumarreddyka@gmail.com

Create

7.4 status of the project

The screenshot shows the 'Stories' page with a table of 18 test stories. A 'New Story' modal is open on the right, containing fields for Title (Login), Description (Issue in login), Developer (Nitin), Estimated Point Value (5), and State (New). An 'Insert' button is at the bottom right.

#	Title	Developer	State	Point V.	Edit
6	test story 1	Project manager	Completed	1	
7	test story 2	Nitin	Estimated	2	
8	test story 3	Rakesh	Estimated	3	
9	test story 4	Not Assigned	Estimated	5	
10	test story 5	Not Assigned	Estimated	8	
11	test story 6	Not Assigned	Estimated	8	
12	test story 7	Not Assigned	Estimated	2	
13	test story 8	Not Assigned	Estimated	3	
14	test story 9	Not Assigned	Estimated	5	
15	test story 10	Not Assigned	Estimated	8	
16	test story 11	Not Assigned	Estimated	1	
17	test story 12	Not Assigned	Estimated	2	
18	test story 13	Not Assigned	Estimated	3	

7.5 Summary of the bugs

The screenshot shows the 'Issue Assignment' summary on the left and a detailed view of assignments on the right.

Summary for assignment:

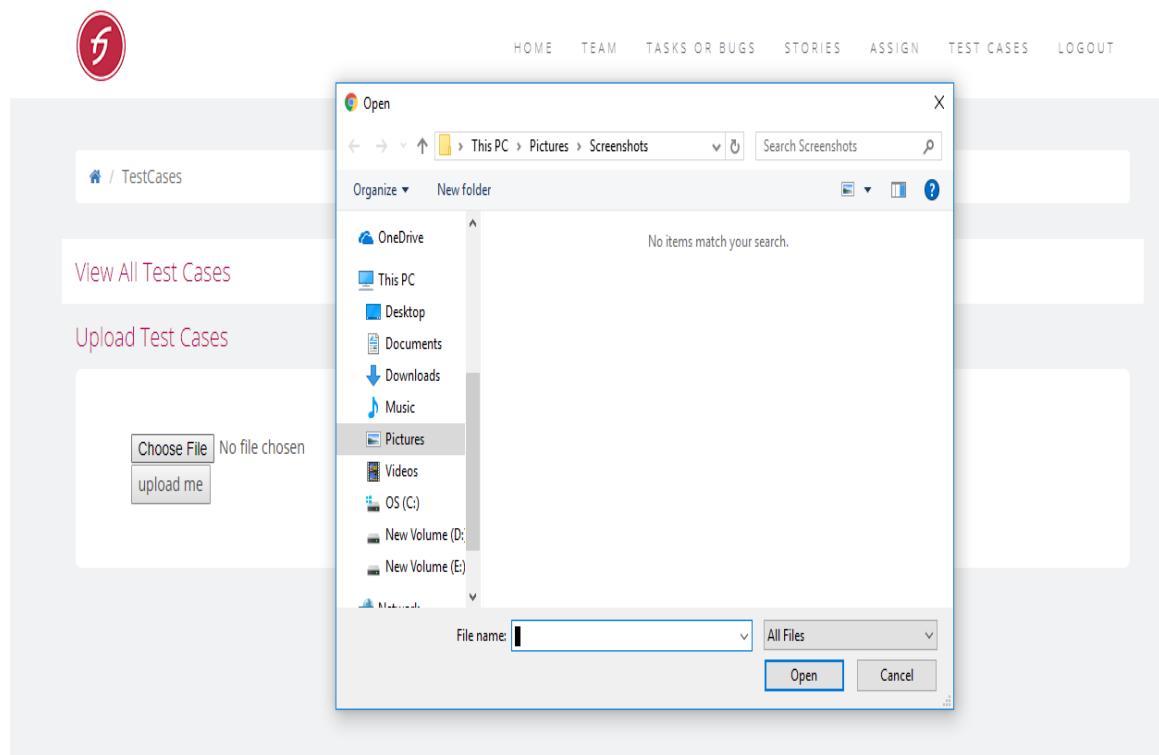
- There are currently 3 developers.
- There are total 18 estimated and unassigned stories.
- Total points of the estimated and unassigned stories are 76.
- There are 10 weeks defined.

Assignments:

Week #1

Issue-ID	Story	Developer	Story Points
25	test story 20	Project manager	8

7.6 Upload the code



CHAPTER 8

CONCLUSION

Bug Tracking System helps to detect and manage the bugs in software products effectively. It is the best project management tool. It can be used to track the bugs in the project modules and assist in troubleshooting errors for testing and development process.

This project highly avoids all sources of delay in bugs reporting level within the project modules in the software industry. As application is deployed in a company server, it is much more secure.

This can be modified and enhanced for performing more complex task related to bug tracking system.

This project has lot of scope to replace existing system followed by software organizations. This application will save time, increase security and provides better performance with well-designed database.

REFERENCES

- [1] Grannell, Craig. *The Essential Guide to CSS and HTML Web Design*, friends of ED, 2007.
- [2] Castro, Elizabeth and Hyslop . *HTML5, and CSS, Eighth Edition: (Visual QuickStart Guide)*, Peachpit Press, 2013.
- [3] Crockford, Douglas. *JavaScript: The Good Parts*, O'Reilly & Associates, 2008.
- [4] Flanagan, David. *Javascript : The Definitive Guide, Fifth Edition*, O'Reilly & Associates, 2006.
- [5] Friedlein, Ashley. *Web Project Management: Delivering Successful Commercial Web Sites* , Morgan Kaufmann, 2000.
- [6] St. Laurent, Simon and DeLong, B.K. *HTML: Moving Toward XML*, Hungry Minds, 2000.
- [7] Mickley Gillenwater, Zoe. *Stunning CSS3: A Project-based Guide to the Latest in CSS*, New Riders 2010.