Bug/Issue Tracker

Bug tracking application for software developers

Submitted by:

Name: Md Jubaer Hossain

ID: 171-35-1904

Section: D

Course title: Final Year Project

Course code: SWE439

Supervised by:

Mr. Md. Mushfiqur Rahman

Lecturer

Department of Software Engineering

Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering

APPROVAL

This **project** titled on “**Bug Tracker**”, submitted by Student Name (ID: **171- 35-1904**) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

-- **Chairman**

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

-- **Internal Examiner 1**

K. M. Imtiaz-Ud-Din
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Fahad

-- **Internal Examiner 2**

Md Fahad Bin Zamal
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

-- **External Examiner**

Professor Dr. Md. Nasim Akhtar
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur

DECLARATION

I hereby declare that I have taken this project named Bug Tracker under the supervision of **Mr. Md. Mushfiqur Rahman, Lecturer, Department of Software Engineering, Daffodil International University.** I also declare that I have submitted neither this project nor any part of it for award of any degree.

Jubaer

Md. Jubaer Hossain
ID: 171-35-1904
Batch: 22th
Department Of Software Engineering
Faculty of science and Information Technology
Daffodil International university

Certified by:

Mr. Md. Mushfiqur Rahman
Lecturer
Department Of Software Engineering
Faculty of science and Information Technology
Daffodil International university

ACKNOWLEDGEMENT

First of all, I am grateful to the Almighty Allah for making me eligible to complete this project. Then I would like to thank my supervisor Mr. Md. Mushfiqur Rahman, Lecturer, Department of software engineering. I am deeply grateful to him for his expert guidance on and encouragement on achieving this project. And I would like to thank everyone who helped me directly or indirectly while building this project.

Table of Contents

LIST OF FIGURES

Chapter 1: Introduction

1.1 Project Overview

Bug/Issue tracker is an application that can be used to track bugs/issues or feature requests while developing a software.

Using this system Admin and Testers can create projects and create new issues. They can assign and track the progress/status of each issue or progress of a project and assign developers to each issue more effectively.

The admin can also manage roles, assign or unassign developer or Testers to a project, issue or feature request.

1.2 Scope

1. Bug/Issue tracker is an application that can be used to manage and track bugs/issues, tasks and feature requests while developing a project
2. This project is developed as a course project of "SWE439: Final Year Project"

1.3 Project Purpose

The purpose of this project is to build a reliable bug tracking system that's efficient and effective to track bugs and issues during software development.

1.4 Stakeholders

1.4.1 Testers

Testers are the primary user of this application. They do most of the activities in this application starting from creating project to creating and tracking each bug and watching the progress of each bug/issues in every project

1.4.2 Developers

Developers are the second primary user who actively work with testers in this application to fix, update and edit the status of each bug

1.4.3 Admin

Admin is the secondary user of this system. They mostly manage, edit, update different user roles and does the overall administrative work within the application

Chapter 2: Software Requirement Specification (SRS)

2.1 Functional requirements

Functional requirements refer to the core functionalities that are mandatory to successfully run a system or application. Here are the functional requirements of Bug Tracker

FR ID	**Description**
FR1	Registration
FR2	Login
FR3	Create New Project
FR4	Edit/Modify a project
FR5	Manage user role
FR6	Create new issue
FR7	Modify/edit and update an issue
FR8	Upload screenshots/images under an issue
FR9	Comment under an issue
FR10	View all projects
FR11	View all issues
FR12	View all users
FR13	View only Assigned issues and projects
FR14	Logout
FR15	Recover password

2.1.1 Registration

Requirement id	FR1
Requirement name	Registration
Description	User must register
Stakeholder	Admins, Testers, Developers

2.1.2 Login

Requirement id	FR2
Requirement name	Login
Description	User must login
Stakeholder	Admins, Testers, Developers

2.1.3 Create new project

Requirement id	FR3
Requirement name	Create New Project
Description	Testers can create new projects
Stakeholder	Testers

2.1.4 Edit/modify a project

Requirement id	FR4
Requirement name	Edit/Modify a project
Description	User can edit, update and modify a project
Stakeholder	Admins, Testers

2.1.5 Manage user role

Requirement id	FR5
Requirement name	Manage user role
Description	Admin can manage user roles and update/edit new user’s role
Stakeholder	Admin

2.1.6 Create new issue

Requirement id	FR6
Requirement name	Create new issue
Description	Users can create a new issue/bug/feature request/task
Stakeholder	Testers

2.1.7 Modify/edit an update issue

Requirement id	FR7
Requirement name	Modify/Edit and update an issue
Description	Assigned users can update and modify an issue
Stakeholder	Testers, Developers

2.1.8 File upload

Requirement id	FR8
Requirement name	File upload feature
Description	User can upload file/attachments while creating an issue ticket
Stakeholder	Admins, Testers, Developers

2.1.9 Comment under an issue

Requirement id	FR9
Requirement name	Comment under an issue
Description	Assigned user can comment under their respective issues/bug tickets
Stakeholder	Testers, Developers

2.1.10 View all projects

Requirement id	FR10
Requirement name	View all projects
Description	Testers should view all the projects created in the system
Stakeholder	Testers

2.1.11 View all issues

Requirement id	FR11
Requirement name	View all issues
Description	Testers should be able to view all the issues created within the system
Stakeholder	Testers

2.1.12 View all users

Requirement id	FR12
Requirement name	View all users
Description	Admin should be able to view all the users and their roles
Stakeholder	Admin

2.1.13 View only assigned issues and projects

Requirement id	FR13
Requirement name	View only assigned issues and projects
Description	Developers can only view those issues and projects that they are assigned
Stakeholder	Developers

2.1.14 Logout

Requirement id	FR14
Requirement name	Logout functionality
Description	All users should be able to logout from the system
Stakeholder	Testers, Developers, Admin

2.1.15 Recover password

Requirement id	FR15
Requirement name	Recover password
Description	All users should be able to recover password through their email
Stakeholder	Testers, Developers, admin

2.2 Non-Functional requirements

a. System must response within 1 second of a request
b. The system must have session management
c. The system must hide sensitive data from the reach of unauthorized person
d. The system should be highly scalable and able to handle large amount of requests

2.3 Security requirements

a. Session management must be implemented
b. User authorization must be implemented
c. User authentication must be implemented
d. The system must be designed to protect itself from common attacks like SQL injection

Chapter 3: System Design

3.1 Use case diagram:

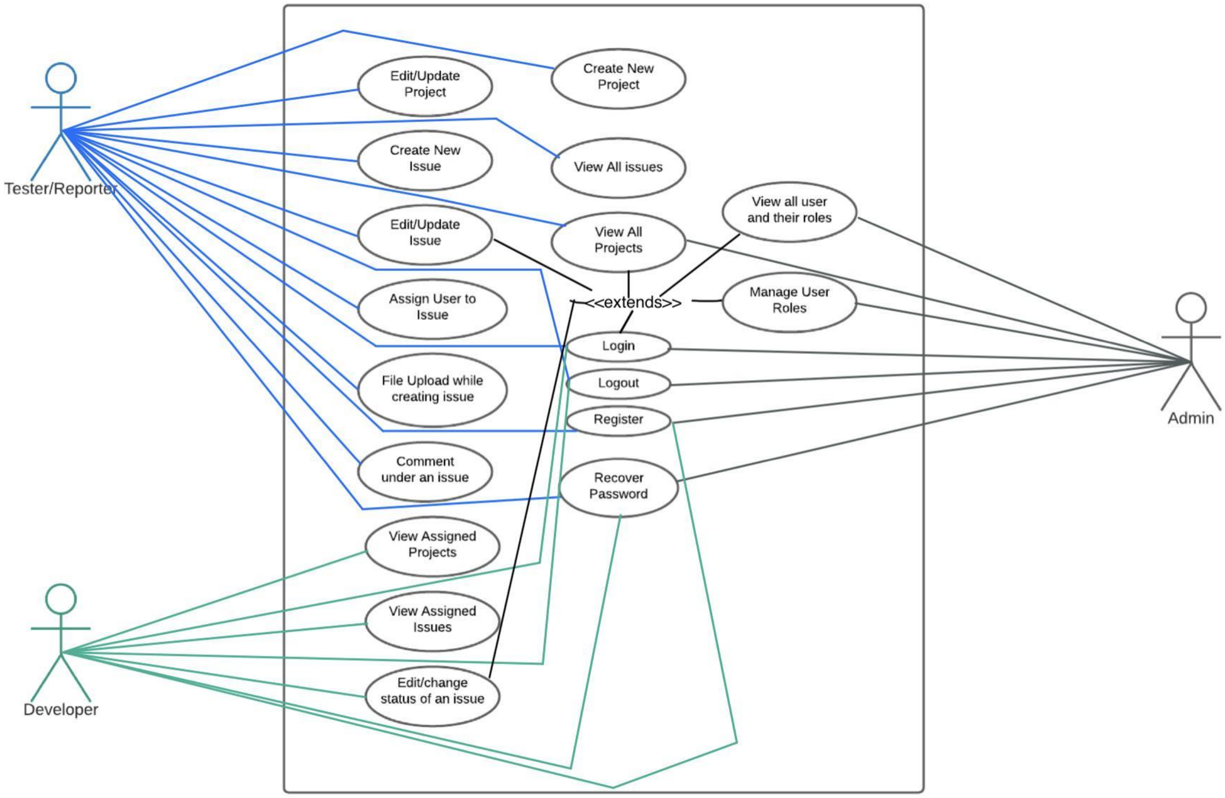


Fig 3.1: Use case diagram

3.2 Use Case Description

3.2.1 Registration

Use Case	User registration
Goal	Users will be able to register into the system
Predictions	None
Success end condition	User successfully registered

Failed end condition	Has to fill up the registration info with a valid email and password	
Primary actors: **Secondary actors:**	Testers, Admin, Developers None	
Trigger	Registration form	
Description	**Step**	**Action**
	1	Fill up the registration form with valid info
	2	User will be successfully registered
Alternative Flows	**Step**	**Branching Action**
	1	Registration form may not fill up properly.
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.2 Login

Use Case	User login
Goal	Users will be able to log into the system
Predictions	User must be registered
Success end condition	User successfully logged in

Failed end condition	Has to try again with valid login info	
Primary actors:	Testers, Admin, Developers	
Secondary actors:	None	
Trigger	Login form	
Description	**Step**	**Action**
	1	Fill up the login form with valid info
	2	User will be successfully logged in
Alternative Flows	**Step**	**Branching Action**
	1	Email or password might not match
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.3 Create a new project

Use Case	Create a new project
Goal	User will be able to create a new project
Predictions	User logged in
Success end condition	Able to create a new project
Failed end condition	Login with a valid tester account

Primary actors: **Secondary actors:**	Tester None	
Trigger	Pressing the create project button	
Description	**Step**	**Action**
	1	Fill up all the project info
	2	The project is successfully created
Alternative Flows	**Step**	**Branching Action**
	1	Project creation failed and user is redirected to the all project list page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.4 Update/edit a project

Use Case	Edit an existing project
Goal	User will be able to update an existing project
Predictions	User logged in
Success end condition	Able to edit an existing project
Failed end condition	Login with a valid tester account
Primary actors:	Tester

Secondary actors:	None	
Trigger	Pressing the edit project button	
Description	**Step**	**Action**
	1	Fill up all the new updated project info
	2	The project is successfully created
Alternative Flows	**Step**	**Branching Action**
	1	Project creation failed and user is redirected to the all project list page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.5 Update user role

Use Case	Update user role
Goal	User will be able to update user’s role
Predictions	User logged in as Admin
Success end condition	Able to edit and update an existing user’s role
Failed end condition	Redirected to the dashboard
Primary actors: **Secondary actors:**	Admin None

Trigger	Pressing the edit user button	
Description	**Step**	**Action**
	1	Change the user’s role
	2	Press save button to update the user’s role
Alternative Flows	**Step**	**Branching Action**
	1	Failed to update user’s role
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	The system must check & authorize that the user is only logged in as an admin

3.2.6 Create a new issue

Use Case	Create a new issue
Goal	User will be able to create a new issue
Predictions	User logged in
Success end condition	Able to create a new issue
Failed end condition	Login with a valid tester account
Primary actors: **Secondary actors:**	Tester None
Trigger	Pressing the create issue button

Description	Step	Action
	1	Fill up all the issue info
	2	The issue is successfully created
Alternative Flows	**Step**	**Branching Action**
	1	issue creation failed and user is redirected to the all issue list page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.7 Update/edit a issue

Use Case	Edit an existing issue
Goal	User will be able to update an existing issue
Predictions	User logged in
Success end condition	Able to edit an existing issue
Failed end condition	Login with a valid tester account
Primary actors: **Secondary actors:**	Tester None
Trigger	Pressing the edit issue button

Description	Step	Action
	1	Fill up all the new updated issue info
	2	The issue is successfully created
Alternative Flows	**Step**	**Branching Action**
	1	issue creation failed and user is redirected to the all issue list page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Fill all the details properly with valid info

3.2.8 Comment under and issue

Use Case	Comment under an issue
Goal	User will be able to Comment under and issue
Predictions	User logged in and has access to that issue
Success end condition	Able to edit an existing project
Failed end condition	Login with a valid tester or developer account
Primary actors: **Secondary actors:**	Tester & developer None
Trigger	Pressing the comment button

Description	Step	Action
	1	Write the comment
	2	Press enter to post the comment
Alternative Flows	**Step**	**Branching Action**
	1	User is redirected to the issue page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	None

3.2.9 Upload screenshot and images

Use Case	Upload screenshot and image
Goal	User will be able to upload screen and image under an issue as reference
Predictions	User logged in as a tester
Success end condition	Able to upload the image
Failed end condition	Asked to try and upload another image
Primary actors: **Secondary actors:**	Tester None
Trigger	While creating an issue

Description	**Step**	**Action**
	1	Press the upload screenshot button
	2	Select image and upload it
Alternative Flows	**Step**	**Branching Action**
	1	Image format not supported
	2	User
Quality Requirements	**Step**	**Requirement**
	1	Only take image formats. No other formats especially scripts or code

3.2.10 View all the issues

Use Case	View a list of all the issues
Goal	User will be able to view all the issues created in the system
Predictions	User logged in
Success end condition	Able to see all the issues created in the system
Failed end condition	Login with a valid tester account
Primary actors: **Secondary actors:**	Tester None
Trigger	Visiting the all issues page

Description	**Step**	**Action**
	1	Press all issues button
	2	View the list of all the issues created
Alternative Flows	**Step**	**Branching Action**
	1	User might not be logged in. In this case take him to the login page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Check whether the user has the necessary permission to view all the issues

3.2.11 View all the projects

Use Case	View a list of all the projects
Goal	User will be able to view all the projects created in the system
Predictions	User logged in
Success end condition	Able to see all the projects created in the system
Failed end condition	Login with a valid tester account
Primary actors: **Secondary actors:**	Tester None

Trigger	Visiting the all projects page	
Description	**Step**	**Action**
	1	Press all projects button
	2	View the list of all the projects created
Alternative Flows	**Step**	**Branching Action**
	1	User might not be logged in. In this case take him to the login page
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	Check whether the user has the necessary permission to view all the project

3.2.12 Recover password

Use Case	Recover password
Goal	User will be able to recover password
Predictions	User already registered
Success end condition	Able to recover his/her password
Failed end condition	System asking to fill up a valid email
Primary actors: **Secondary actors:**	Tester, Developer, Admin None

Trigger	Clicking the forget password button	
Description	**Step**	**Action**
	1	Press the forget password button and request recovery link to the email
	2	Set new password using that recovery link
Alternative Flows	**Step**	**Branching Action**
	1	The recovery email might end up to user’s spam folder
	2	The system my take time to response
Quality Requirements	**Step**	**Requirement**
	1	None

3.3 Activity Diagram

3.3.1 Register

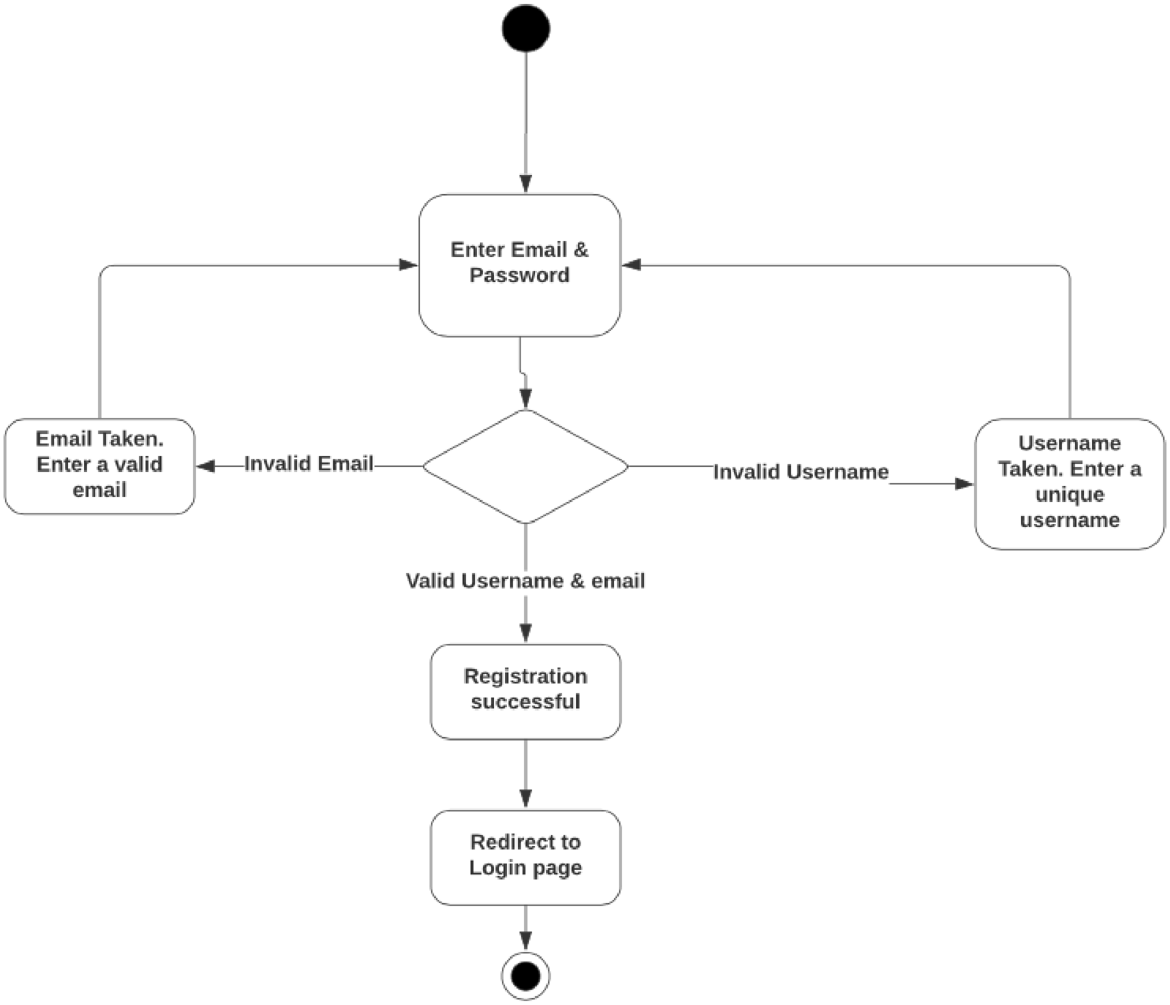


Fig 3.2 Activity diagram: user registration

3.3.2. User login

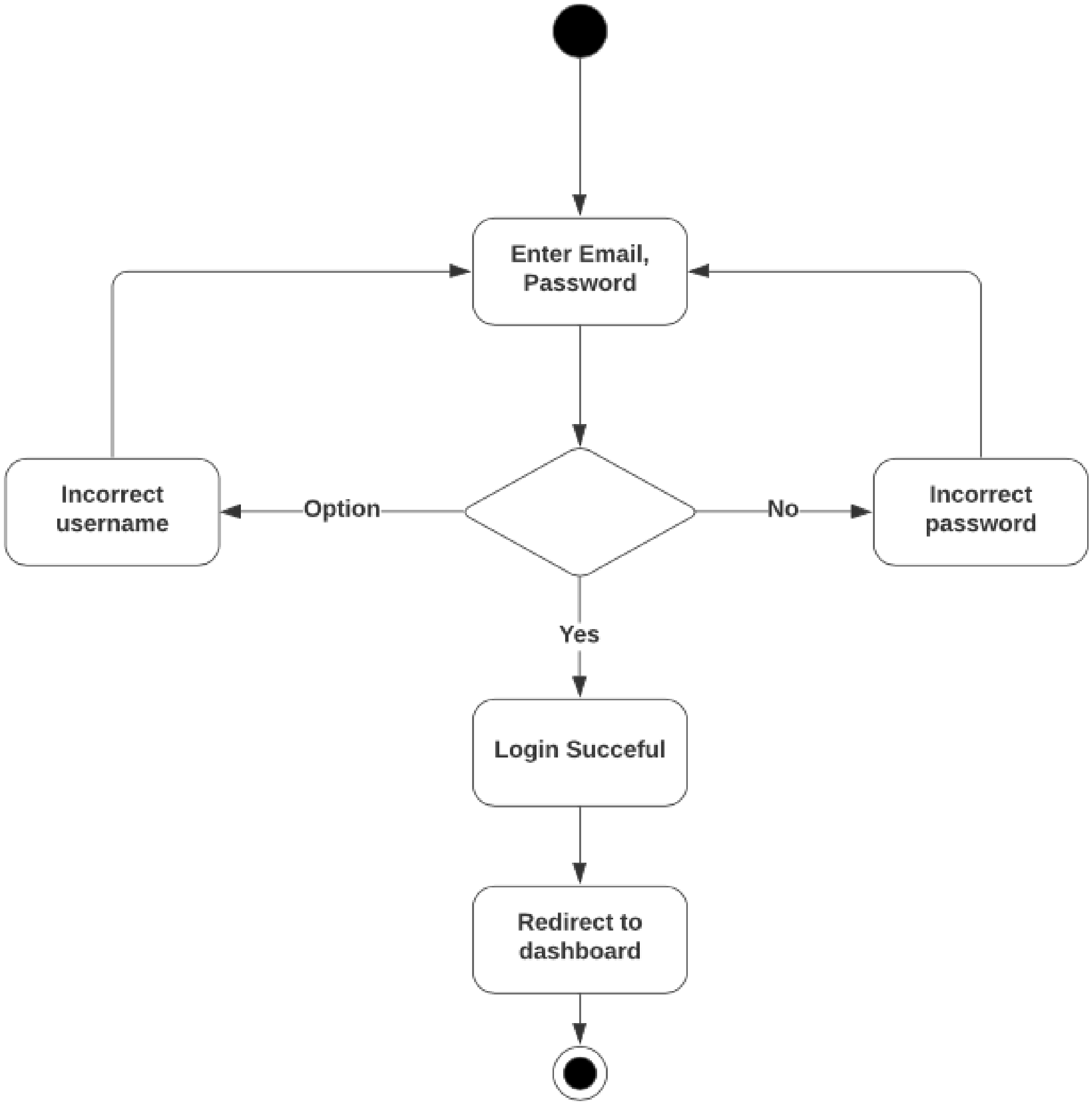


Fig 3.3: Activity diagram: user login

3.3.3 Password recovery

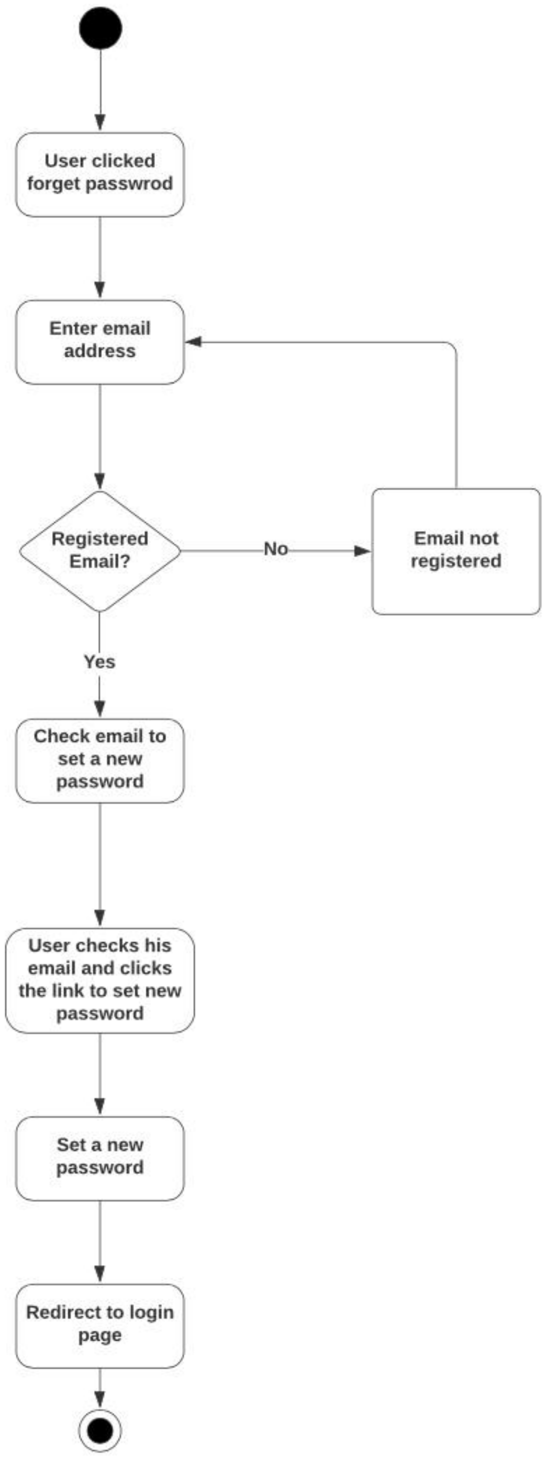

User clicked forget passwrod
Enter email address
Registered Email?
No
Email not registered
Yes
Check email to set a new password
User checks his email and clicks the link to set new password
Set a new password
Redirect to login page

Fig 3.4: Activity diagram: password recovery

3.3.4 Create new issue

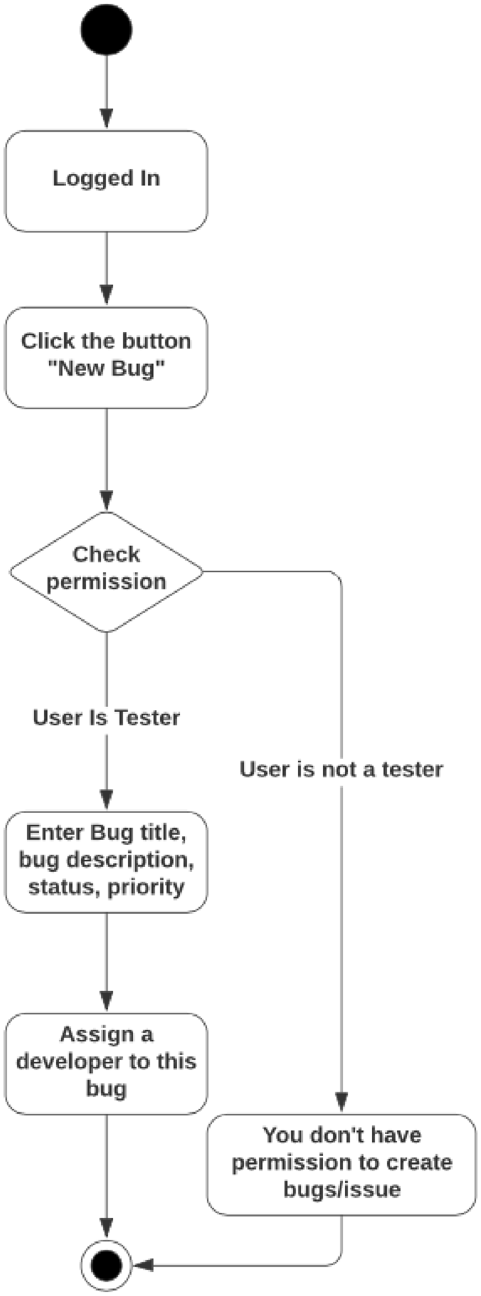

Logged In
Click the button "New Bug"
Check permission
User Is Tester
User is not a tester
Enter Bug title, bug description, status, priority
Assign a developer to this bug
You don't have permission to create bugs/issue

Fig 3.5: Activity diagram: Create new issue

3.3.5 Edit/Update issue

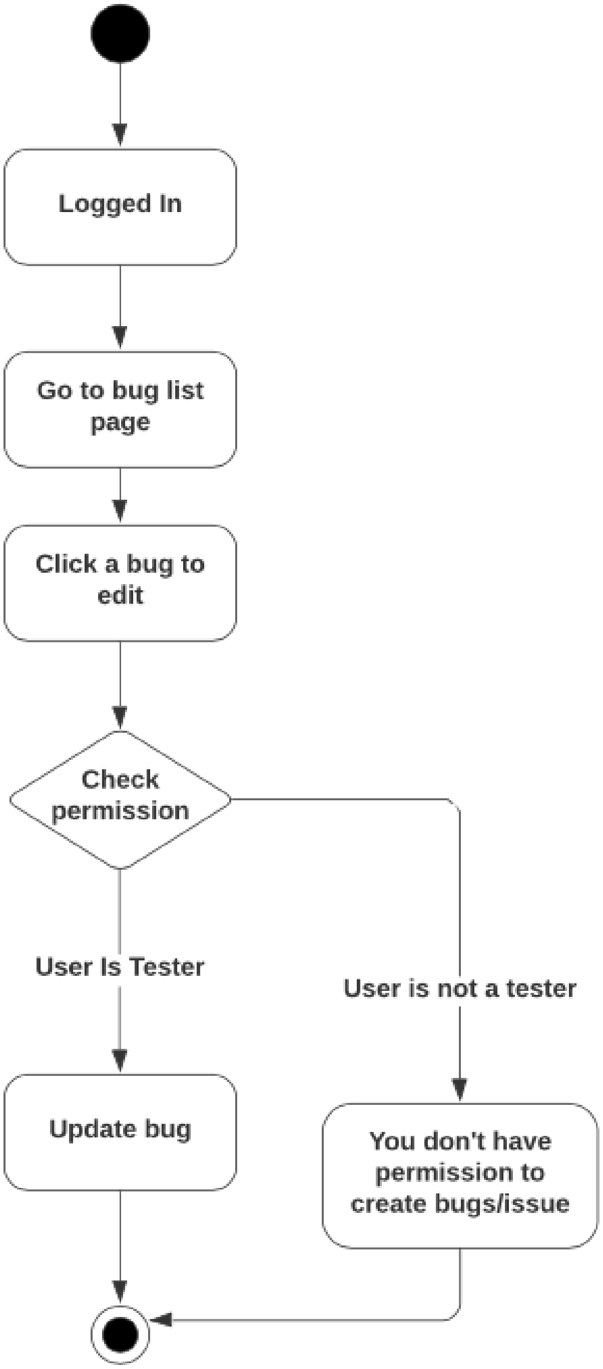


Fig 3.6 Activity diagram: Edit/update an issue

3.3.6 Create new project

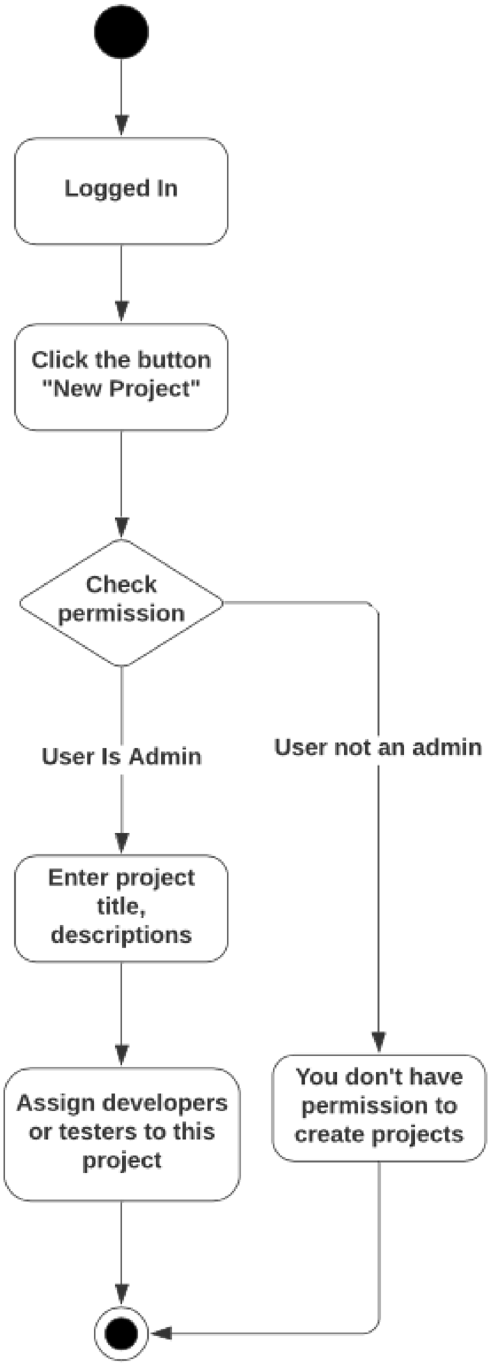

Logged In
Click the button "New Project"
Check permission
User Is Admin
User not an admin
Enter project title, descriptions
Assign developers or testers to this project
You don't have permission to create projects

Fig 3.7: Activity diagram: Create new project

3.3.7 Edit/Update project

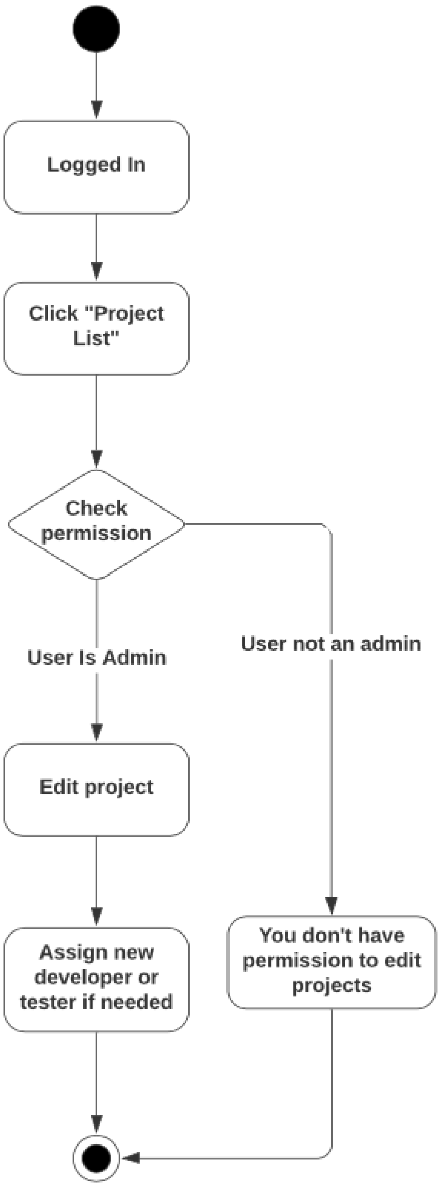


Fig 3.8 Activity diagram: Edit/Update project

3.3.8 Edit/update user role

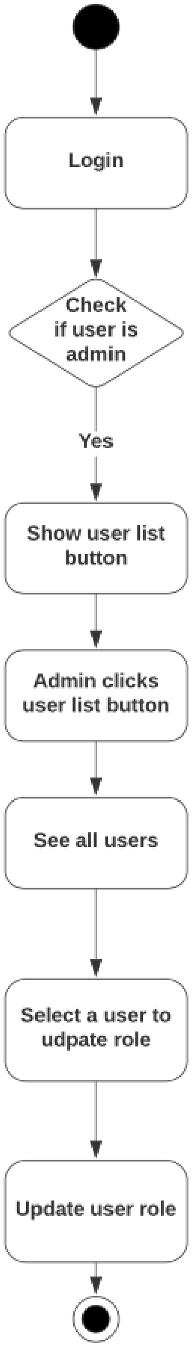


Fig 3.9 Activity diagram: Edit/update user role

3.3.9 Comment under an issue

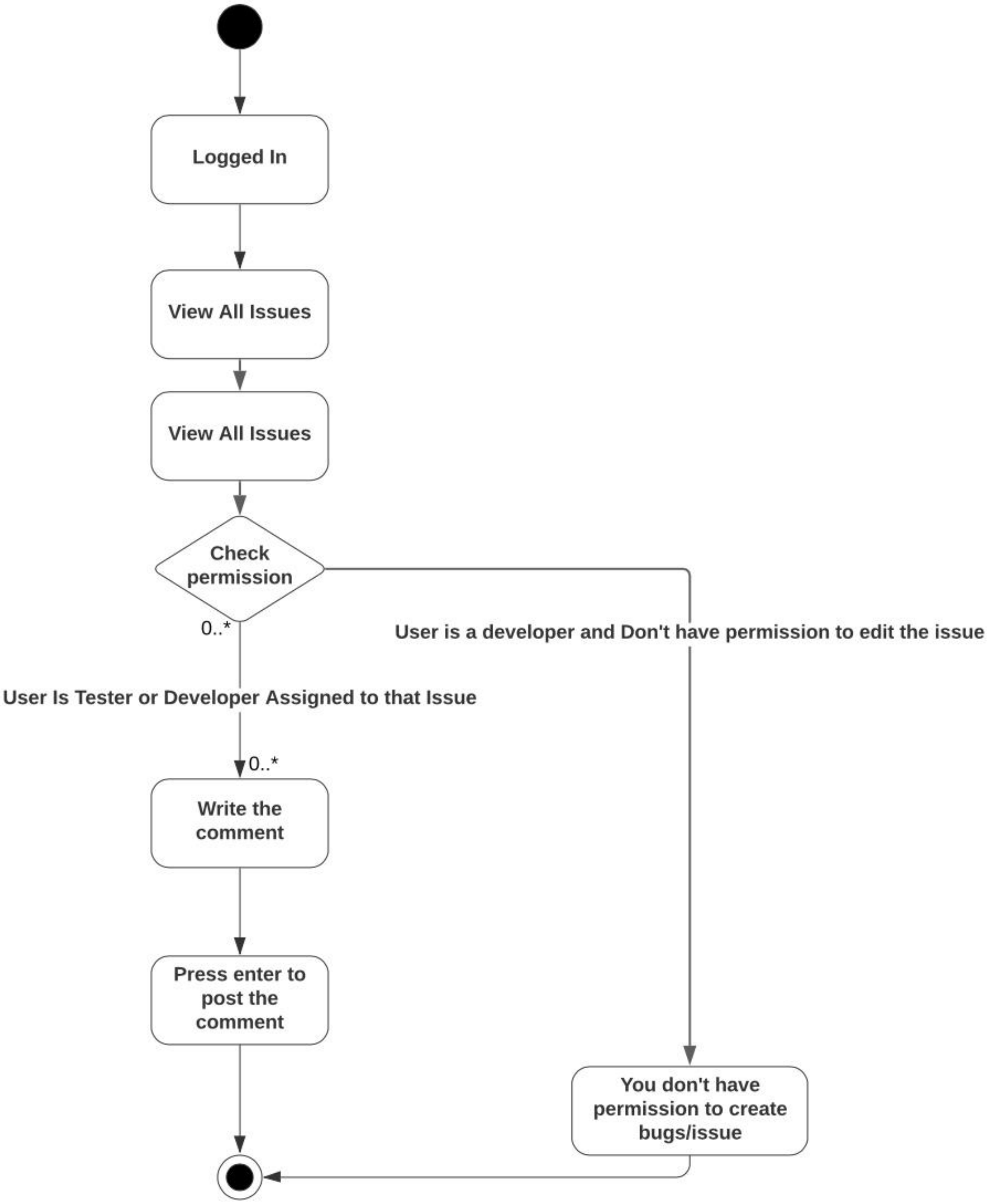

Logged In
View All Issues
View All Issues
Check permission
0..*
User is a developer and Don't have permission to edit the issue
User Is Tester or Developer Assigned to that Issue
0..*
Write the comment
Press enter to post the comment
You don't have permission to create bugs/issue

Fig 3.10 Activity diagram: Comment under an issue

3.4 Sequence diagram

3.4.1 Register

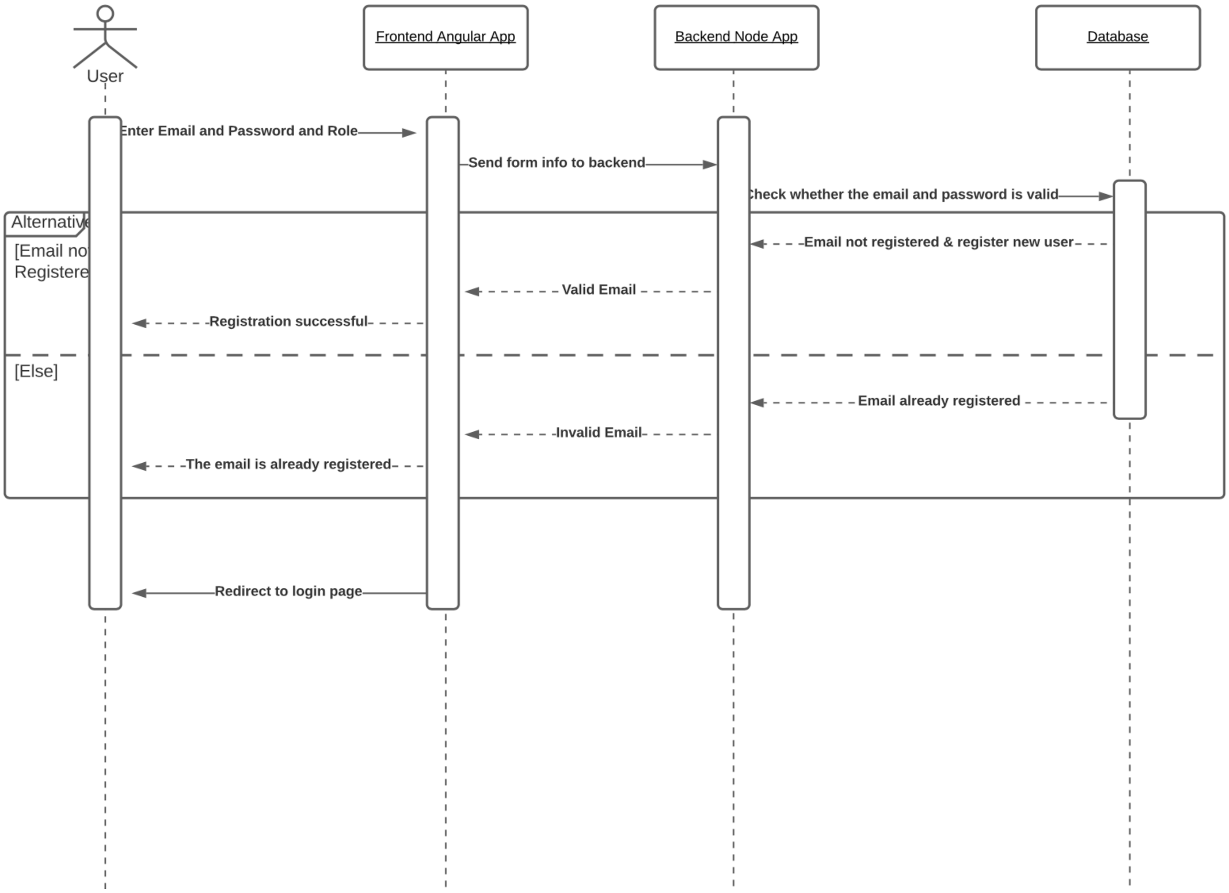


Fig 3.11: Sequence diagram: User registration

3.4.2 Login

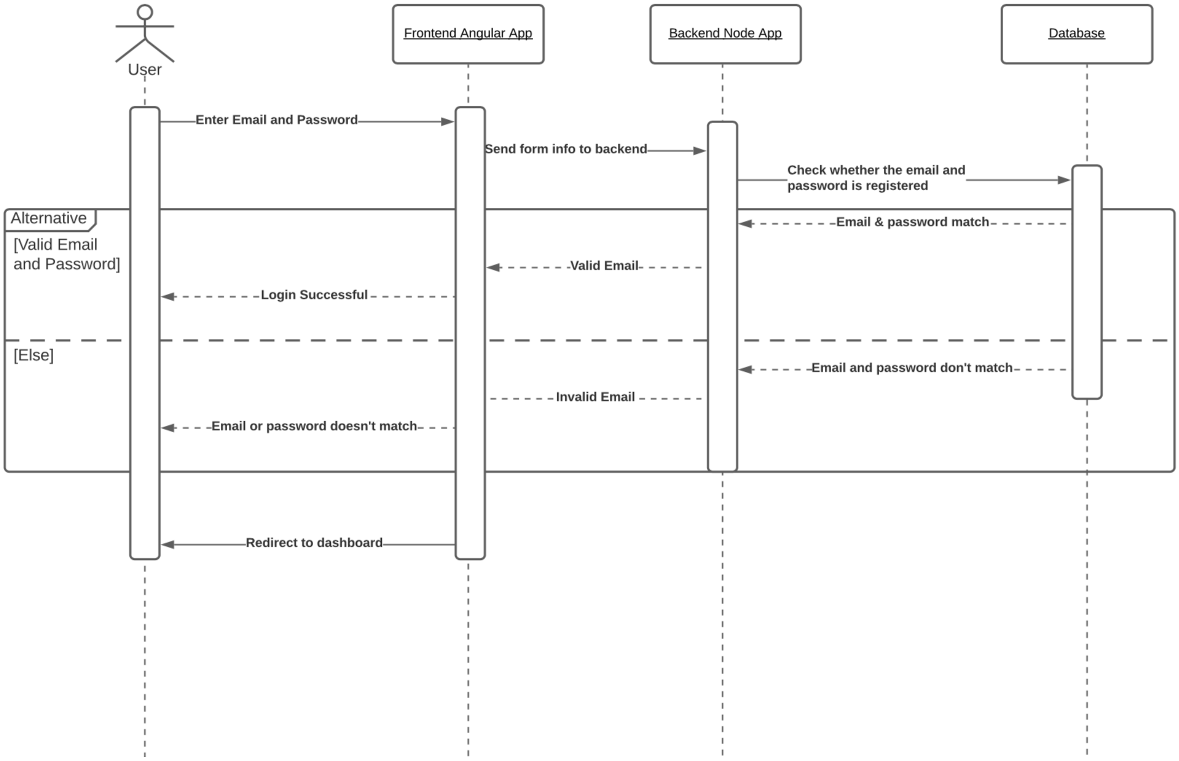


Fig 3.12: Sequence diagram: User login

3.4.3 Recover password

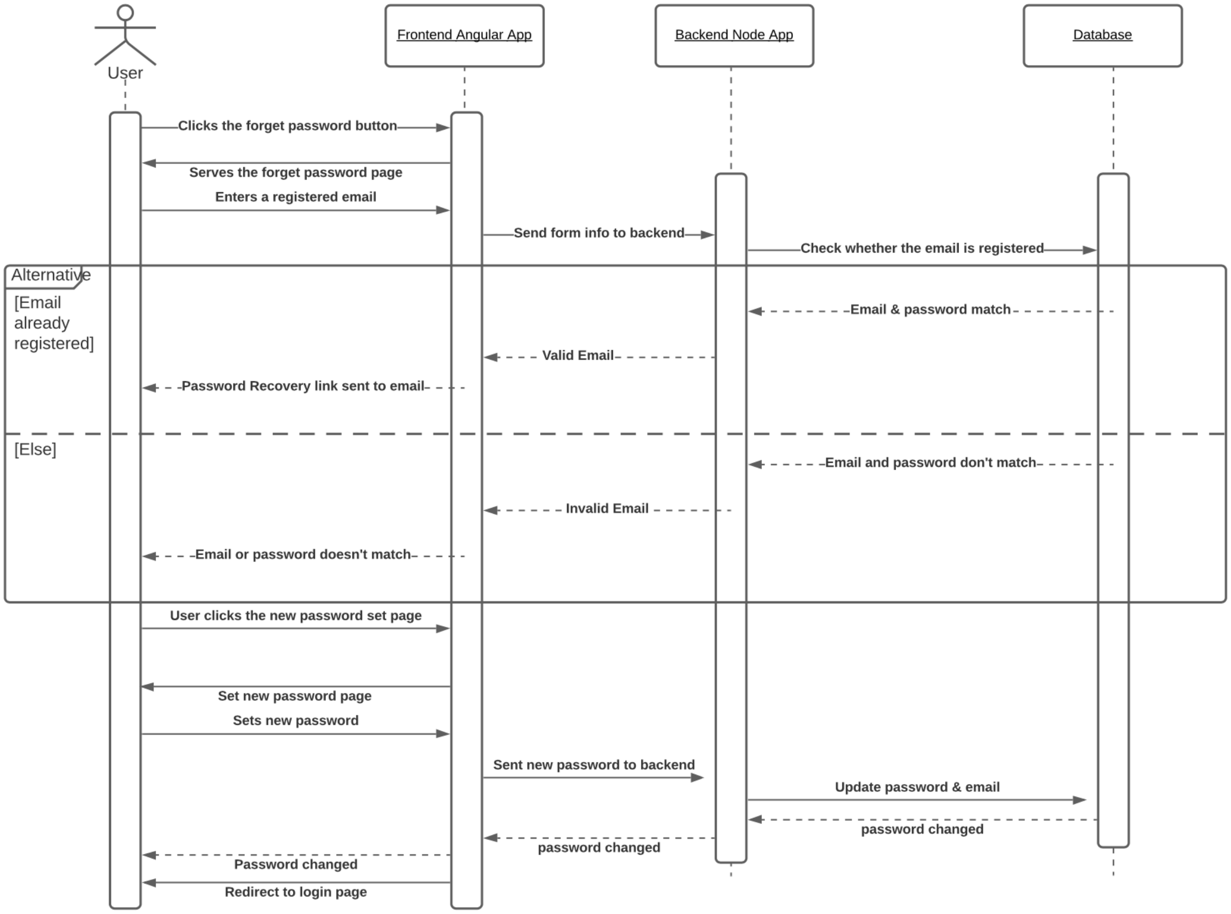

User
Frontend Angular App
Backend Node App
Database
Clicks the forget password button
Serves the forget password page
Enters a registered email
Send form info to backend
Check whether the email is registered
Alternative
[Email already registered]
Email & password match
Valid Email
Password Recovery link sent to email
[Else]
Email and password don't match
Invalid Email
Email or password doesn't match
User clicks the new password set page
Set new password page
Sets new password
Sent new password to backend
Update password & email
password changed
password changed
Password changed
Redirect to login page

Fig 3.13 Sequence diagram: Password recovery

3.4.4 Create an issue

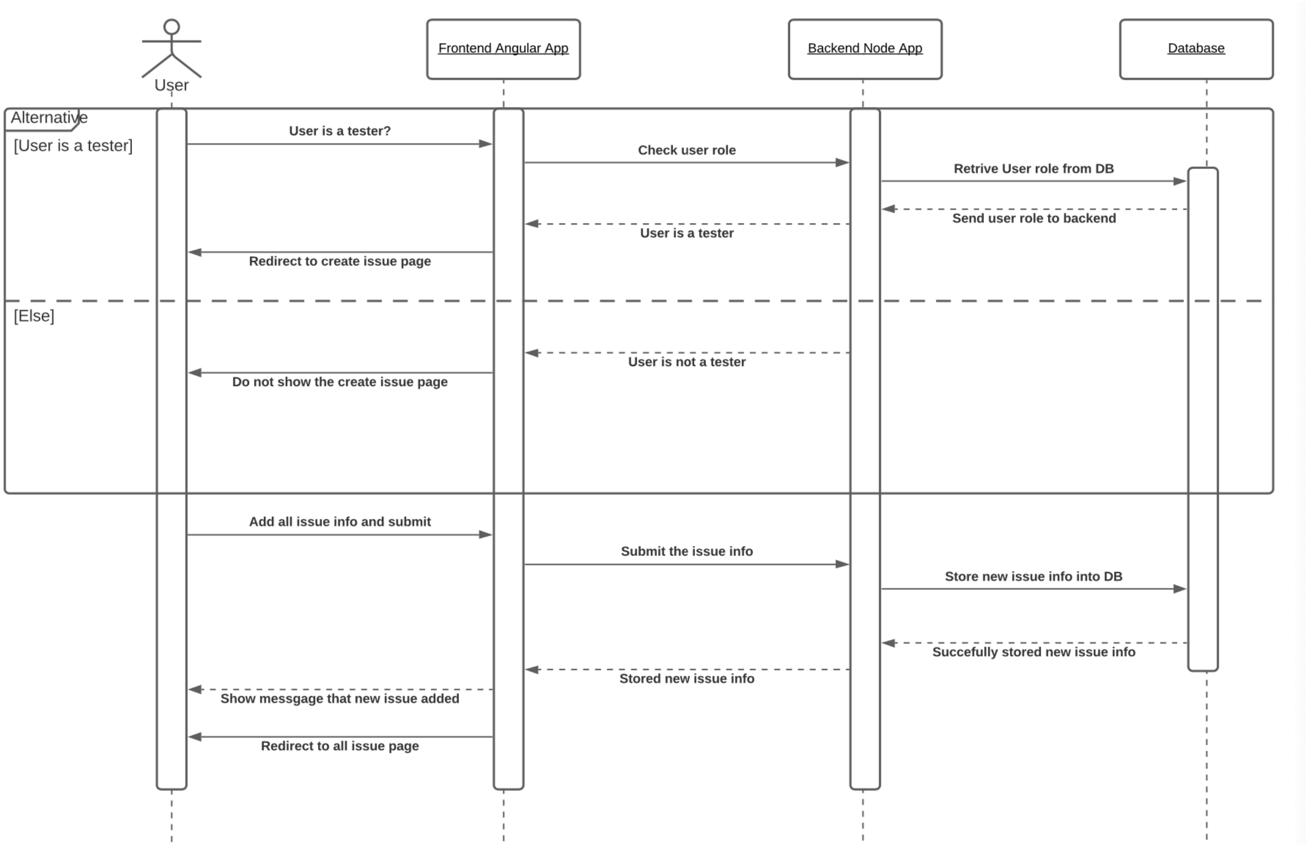

User
Frontend Angular App
Backend Node App
Database
Alternative
[User is a tester]
User is a tester?
Check user role
Retrive User role from DB
Send user role to backend
User is a tester
Redirect to create issue page
[Else]
User is not a tester
Do not show the create issue page
Add all issue info and submit
Submit the issue info
Store new issue info into DB
Succefully stored new issue info
Stored new issue info
Show messgage that new issue added
Redirect to all issue page

Fig 3.14 Sequence diagram: Create an issue

3.4.5 Edit/update an issue

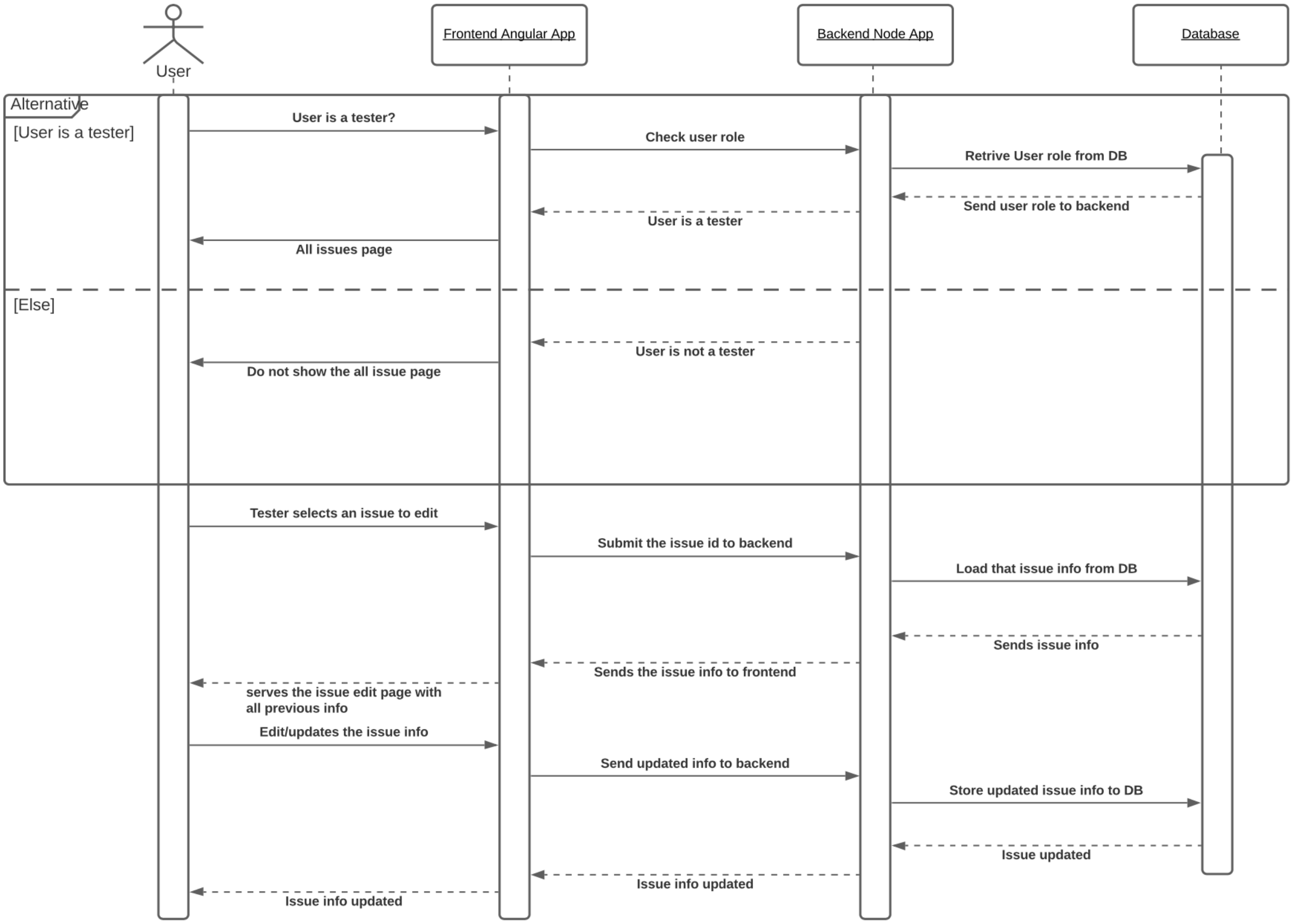

User
Frontend Angular App
Backend Node App
Database
Alternative
[User is a tester]
User is a tester?
Check user role
Retrive User role from DB
Send user role to backend
User is a tester
All issues page
[Else]
User is not a tester
Do not show the all issue page
Tester selects an issue to edit
Submit the issue id to backend
Load that issue info from DB
Sends issue info
Sends the issue info to frontend
serves the issue edit page with all previous info
Edit/updates the issue info
Send updated info to backend
Store updated issue info to DB
Issue updated
Issue info updated
Issue info updated

Fig 3.15 Sequence diagram: Edit/update an issue

3.4.6 Create a project

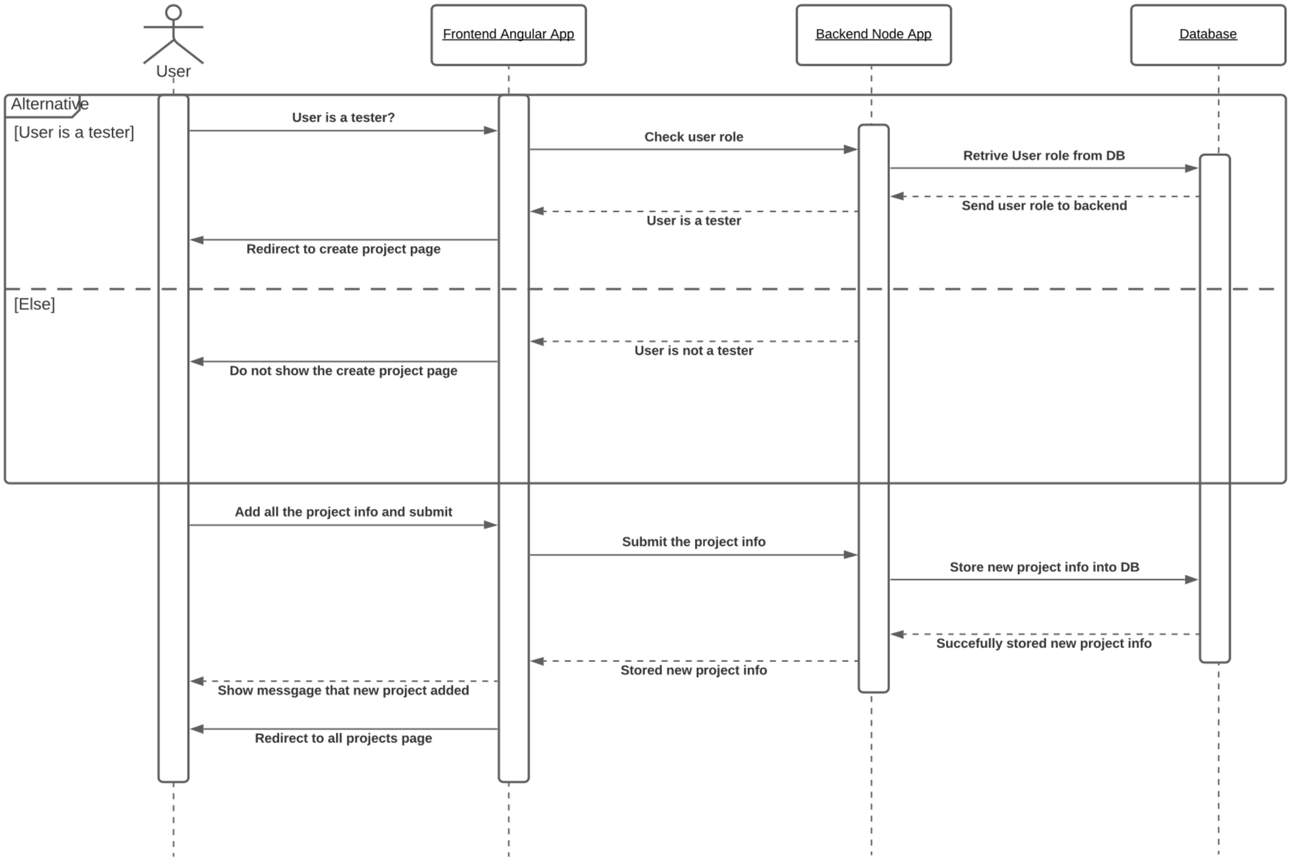

User
Frontend Angular App
Backend Node App
Database
Alternative
[User is a tester]
User is a tester?
Check user role
Retrive User role from DB
Send user role to backend
User is a tester
Redirect to create project page
[Else]
User is not a tester
Do not show the create project page
Add all the project info and submit
Submit the project info
Store new project info into DB
Succefully stored new project info
Stored new project info
Show messgage that new project added
Redirect to all projects page

Fig 3.16: Sequence diagram: Create a project

3.4.7 Edit/update an issue’s status

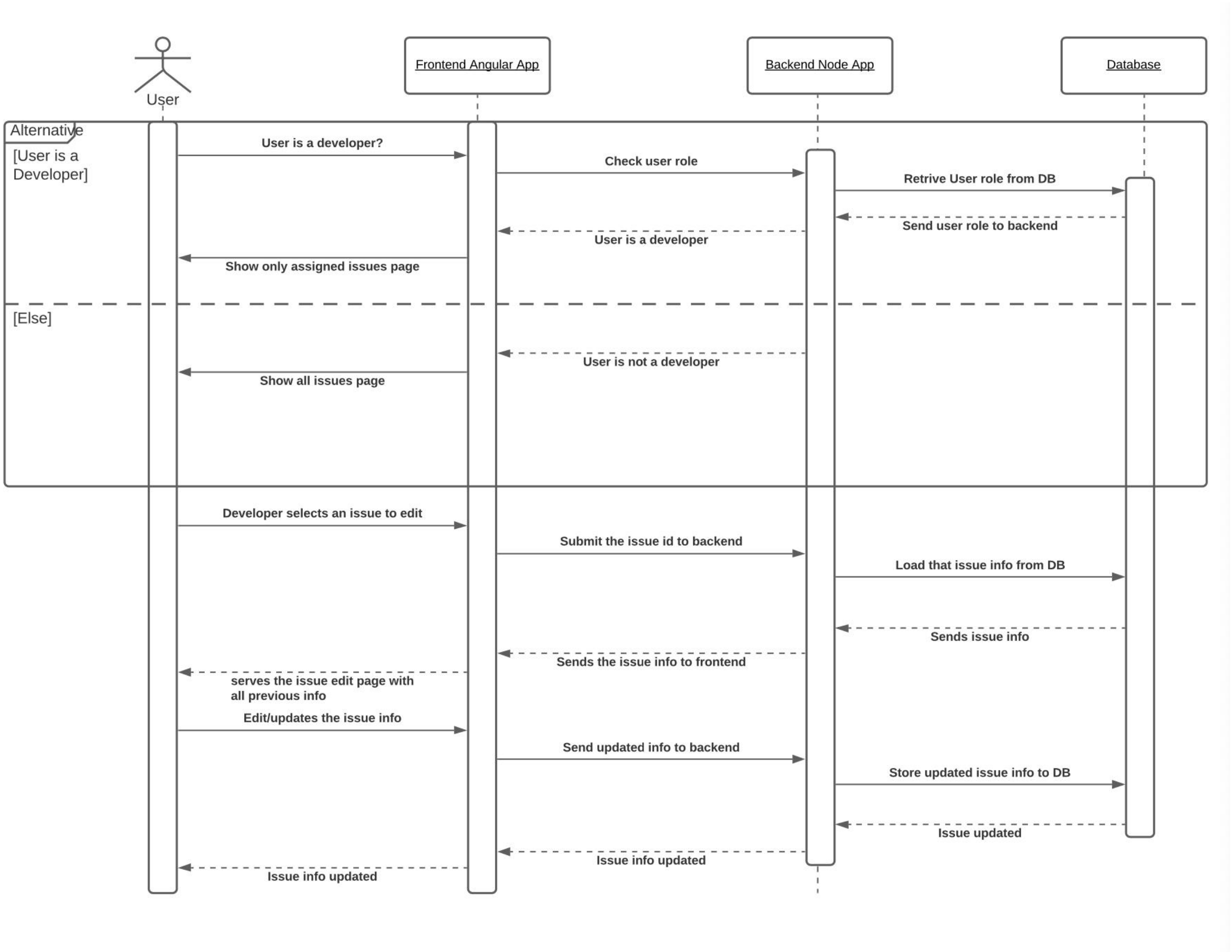

User
Frontend Angular App
Backend Node App
Database
Alternative
[User is a Developer]
User is a developer?
Check user role
Retrive User role from DB
Send user role to backend
User is a developer
Show only assigned issues page
[Else]
User is not a developer
Show all issues page
Developer selects an issue to edit
Submit the issue id to backend
Load that issue info from DB
Sends issue info
Sends the issue info to frontend
serves the issue edit page with all previous info
Edit/updates the issue info
Send updated info to backend
Store updated issue info to DB
Issue updated
Issue info updated
Issue info updated

Fig 3.17 Sequence diagram: Edit/update issue’s status

3.4.8 Edit User's role

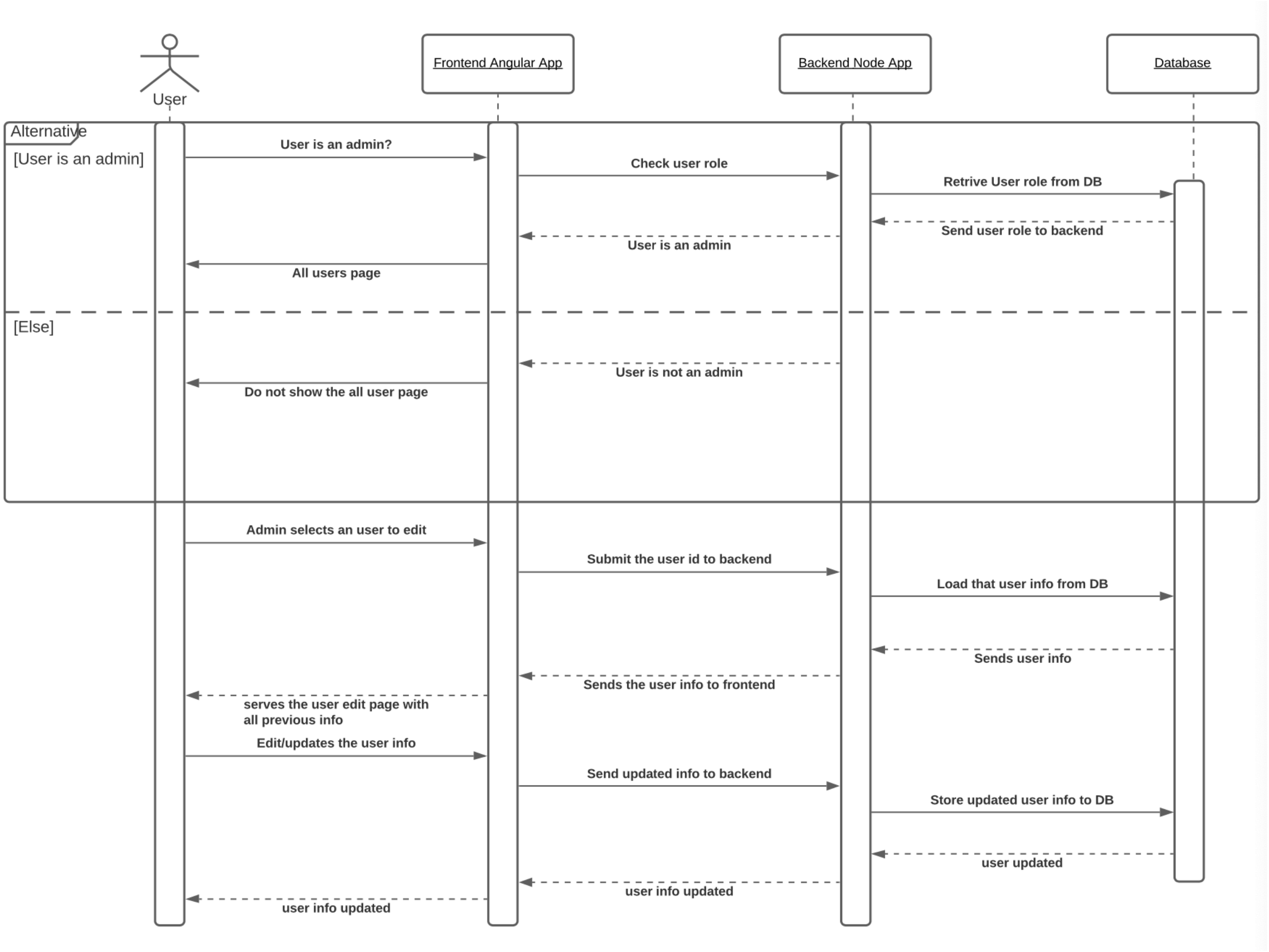

User
Frontend Angular App
Backend Node App
Database
Alternative
[User is an admin]
User is an admin?
Check user role
Retrive User role from DB
Send user role to backend
User is an admin
All users page
[Else]
User is not an admin
Do not show the all user page
Admin selects an user to edit
Submit the user id to backend
Load that user info from DB
Sends user info
Sends the user info to frontend
serves the user edit page with
all previous info
Edit/updates the user info
Send updated info to backend
Store updated user info to DB
user updated
user info updated
user info updated

Fig 3.18 Sequence diagram: Edit user's role

3.4.9 Add comment under an issue

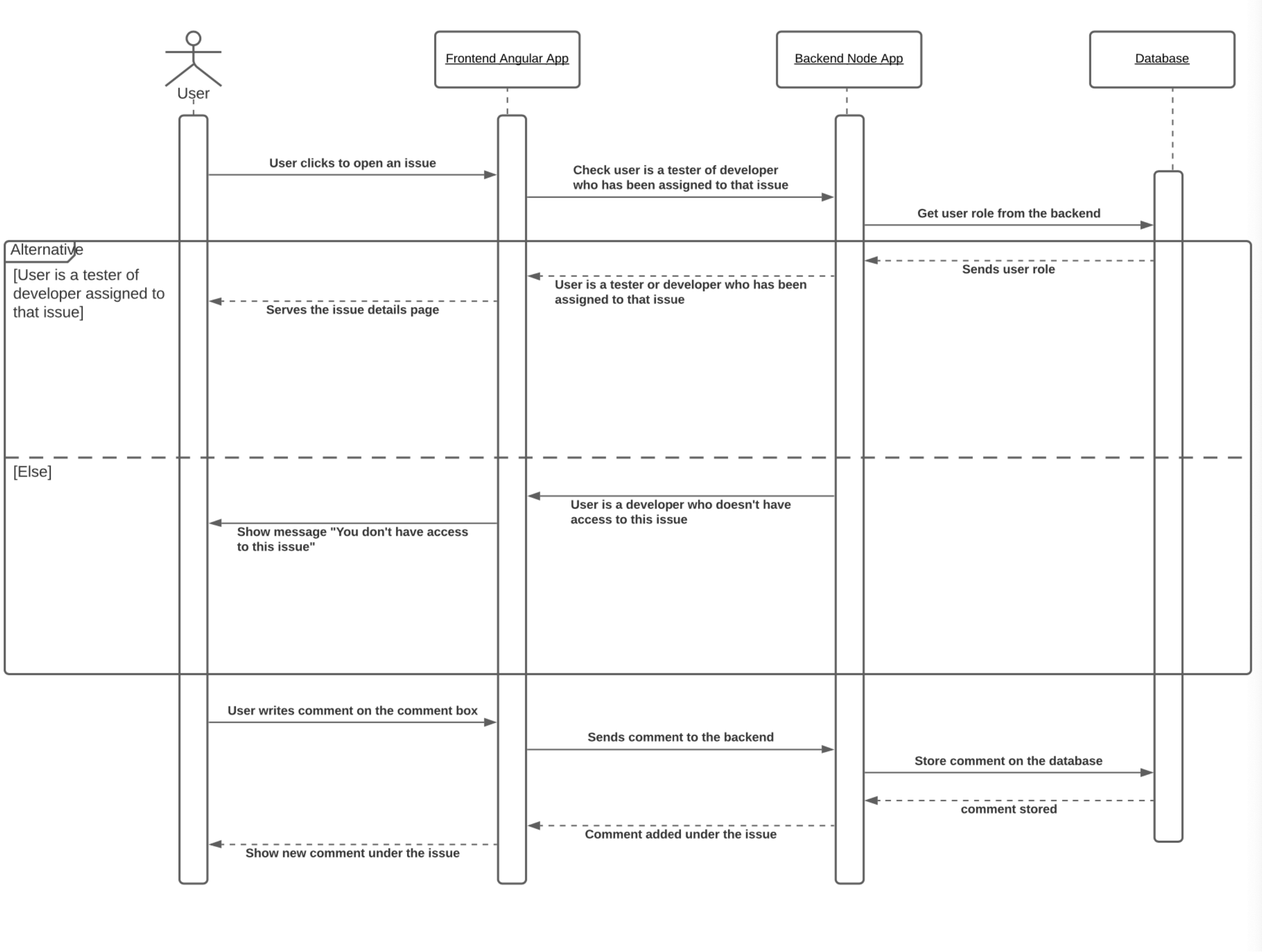
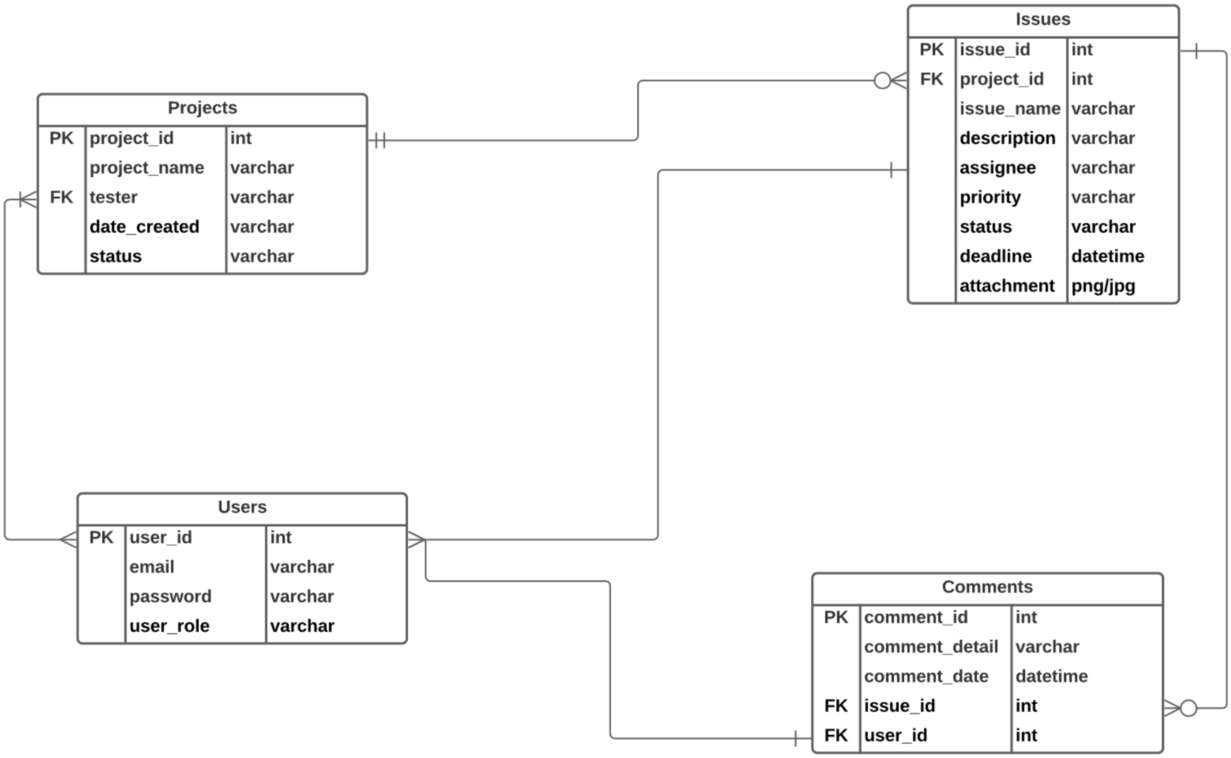

User
Frontend Angular App
Backend Node App
Database
User clicks to open an issue
Check user is a tester of developer who has been assigned to that issue
Get user role from the backend
Alternative
[User is a tester of developer assigned to that issue]
Sends user role
User is a tester or developer who has been assigned to that issue
Serves the issue details page
[Else]
User is a developer who doesn't have access to this issue
Show message "You don't have access to this issue"
User writes comment on the comment box
Sends comment to the backend
Store comment on the database
comment stored
Comment added under the issue
Show new comment under the issue

Fig 3.19 Sequence diagram: Add comment under an issue

3.5 Class diagram



Projects

PK	project_id	int
	project_name	varchar
FK	tester	varchar
	date_created	varchar
	status	varchar

Issues

PK	issue_id	int
FK	project_id	int
	issue_name	varchar
	description	varchar
	assignee	varchar
	priority	varchar
	status	varchar
	deadline	datetime
	attachment	png/jpg

Users

PK	user_id	int
	email	varchar
	password	varchar
	user_role	varchar

Comments

PK	comment_id	int
	comment_detail	varchar
	comment_date	datetime
FK	issue_id	int
FK	user_id	int

Fig 3.20 Class diagram

3.6 Entity Relationship Diagram

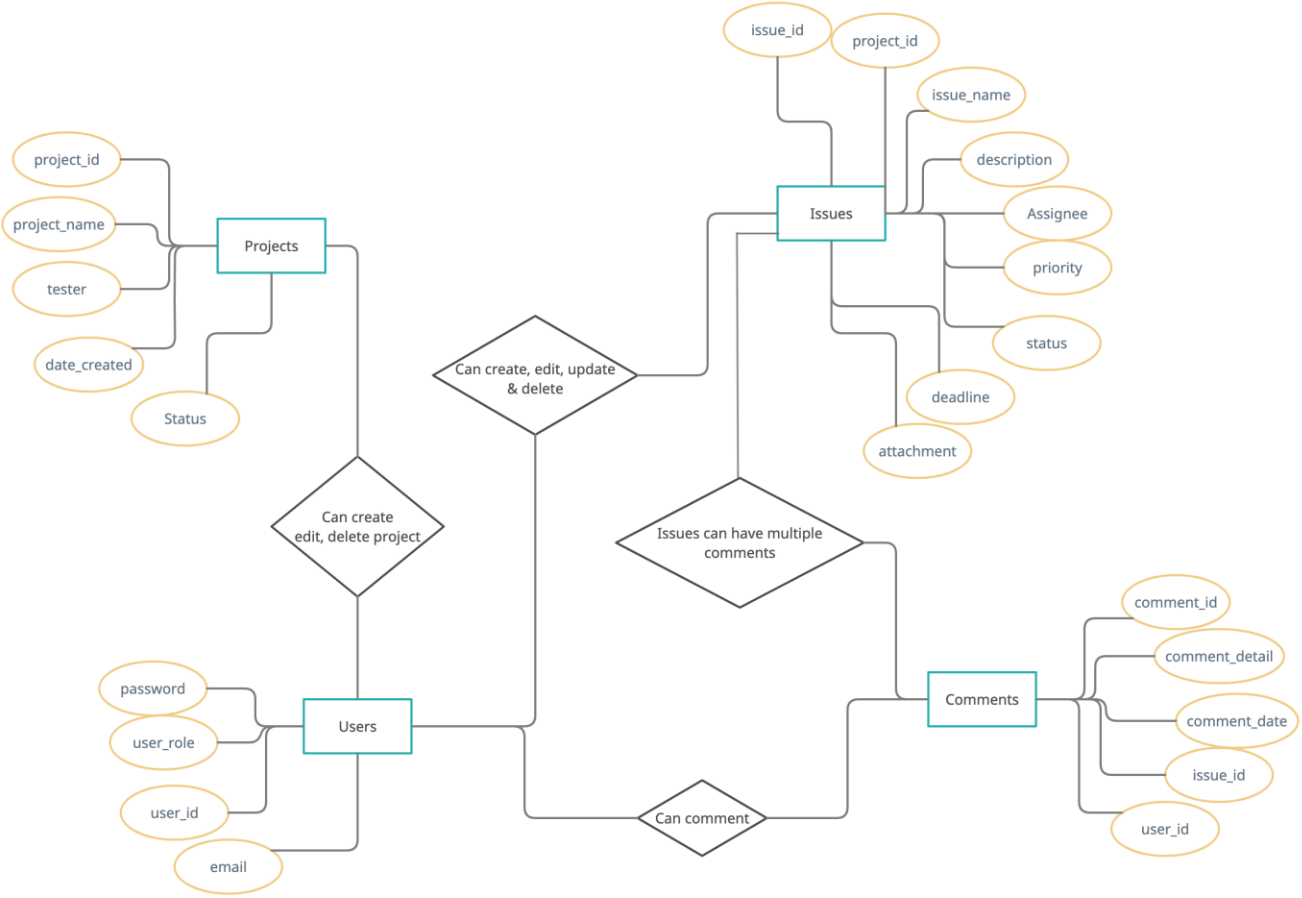


Fig 3.21 Entity Relationship Diagram

3.7 Tools and technologies used

3.7.1 Front-end technologies

a. HTML
b. CSS
c. Javascript(ES6+)
d. Bootstrap
e. JQuery

3.7.2 Front-end frameworks

a. Angular 9

3.7.3 Back-end technologies

a. Node
b. Postman for API testing

3.7.4 Back-end frameworks

a. Express
b. Nodemon

3.7.5 Database

a. MongoDB
b. Mongo Atlas for cloud database
c. Mongo compass

3.7.6 Other technologies

a. Github for version control
b. Visual Studio Code as IDE

Chapter 4: System Test

4.1 Testing features priority levels

Priority Level

- ★ 1 = Low Priority
- ★ 2 = Medium Priority
- ★ 3 = High Priority

4.2 Testing table with priority

Feature	**Priority**	**Description**
Registration	3	System must be able to register new user
Login	3	System must be able to authenticate and login registered users
Create project	2	Testers must be able to create new projects
Edit project	2	Testers must be able to edit projects
Create issues	3	Testers must be able to create new issues
Edit issues	3	Testers must be able to edit issues that were already created
Edit user role	3	Admin must be able to edit user role
Comment under an issue	2	Developers and testers must be able to comment under the assigned issues

Recover password	3	All users must be able to recover passwords
Update assigned issues status	3	Each developer should be able to update issue status to those issues that they are assigned

4.3 Test cases

4.3.1 Registration

<table>
<tr><td colspan="3"><b>Test Case</b>: 4.2.1</td><td></td><td colspan="3"><b>Test Case Name:</b> Registration</td></tr>
<tr><td colspan="3"><b>System:</b> Bug Tracker</td><td></td><td colspan="3"><b>Subsystem:</b></td></tr>
<tr><td colspan="3"><b>Designed by:</b> Jubaer Hossain</td><td></td><td colspan="3"><b>Design Date:</b></td></tr>
<tr><td colspan="3"><b>Executed by:</b></td><td></td><td colspan="3"><b>Execution Date:</b></td></tr>
<tr><td colspan="7"><b>Description:</b> A user must register with a role to access the system</td></tr>
<tr><td colspan="7"><b>Pre-condition:</b>
★ Email address must be unique
★ Passwords must be between 8 to 30 characters.
★ Must contain at least one uppercase letter.
★ Must contain at least one lowercase letter.
★ Must contain at least one number digit.
★ Must contain at least one special character.</td></tr>
<tr><td><b>Step</b></td><td><b>Email address</b></td><td><b>Password</b></td><td><b>Role</b></td><td><b>Response</b></td><td><b>Pass/Fail</b></td><td><b>Comment</b></td></tr>
</table>

1	jubaer1660 @gmail.com	pa$$w0rD	Admin	Registration successfully	Pass	
2	jubaer1660 @gmail.com	pa$$w0rd	Tester	Must contain at least one uppercase letter	Fail	
3	hasan@gmail.com	pa$$wkrD	Developer	Must contain at least one number digit	Fail	
Post-condition: A new user account has been registered successfully						

4.3.2 Login

Test Case: 4.3.2				**Test Case Name:** Login		
System: Bug Tracker				**Subsystem:**		
Designed by: Jubaer Hossain				**Design Date:**		
Executed by:				**Execution Date:**		
Description: A user must login in order to access the system						
Pre-condition: User must be already registered with a valid email and password						
Step	**Email address**	**Password**	**Role**	**Response**	**Pass/Fail**	**Comment**
1	jubaer1660 @gmail.com	pa$$w0rD	Admin	Login successfully	Pass	
2	jubaer1660	pa$$w0rd	Tester	Incorrect	Fail	

	@gmail.com			credential		
Post-condition: User logged in successfully and redirected to the dashboard						

4.3.3 View all user list(Admin)

Test Case: 4.3.3		**Test Case Name:** View all user list(Admin)		
System: Bug Tracker		**Subsystem:**		
Designed by: Jubaer Hossain		**Design Date:**		
Executed by:		**Execution Date:**		
Description: Registered admin can see all user list				
Pre-condition: ★ Database has at least one user registered on the system				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Admin clicks on the “user” button	System will display the user list	Pass	
Post-condition: Display all users registered on the system				

4.3.4 View all project list(Tester)

Test Case: 4.3.4	**Test Case Name:** View all project list(Tester)
System: Bug Tracker	**Subsystem:**

Designed by: Jubaer Hossain		**Design Date:**		
Executed by:		**Execution Date:**		
Description: Registered testers can see all project list				
Pre-condition: ★ Database has at least one project on the system				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Tester clicks on the “Projects” button	System will display the projects list	Pass	
Post-condition: Display all users projects entered on the system				

4.3.5 View all issue list(Tester)

Test Case: 4.3.5		**Test Case Name:** View all issues list(Tester)		
System: Bug Tracker		**Subsystem:**		
Designed by: Jubaer Hossain		**Design Date:**		
Executed by:		**Execution Date:**		
Description: Registered testers can see all issues list				
Pre-condition: ★ Database has at least one issue on the system				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Tester clicks on the “Issues” button	System will display the issues list	Pass	

Post-condition: Display all users issues entered on the system

4.3.6 Create a project

Test Case: 4.3.6		**Test Case Name:** Create a project		
System: Bug tracker		**Subsystem:**		
Designed by: Jubaer Hossain		**Design Date:**		
Executed by:		**Execution Date:**		
Description: Registered testers will be able to create project				
Pre-condition: ★ Must be logged in				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Fill up all the required boxes and clicks on 'Create' button	System create new project	Pass	
2	Tester does not fill up all the required boxes and clicks on 'Create' button	All required boxes must be filled	Fail	
Post-condition: New project created successfully				

4.3.7 Create an issue

Test Case: 4.3.7	**Test Case Name:** Create an issue
System: Bug tracker	**Subsystem:**

Designed by: Jubaer Hossain	**Design Date:**
Executed by:	**Execution Date:**
Description: Registered testers will be able to create new issues	
Pre-condition: ★ Must be logged in	

Step	Action	Response	Pass/Fail	Comment
1	Fill up all the required boxes and clicks on ‘Create issue’ button	System create new issue	Pass	
2	Tester does not fill up all the required boxes or hasn’t assigned a developer to an issue, and clicks on ‘Create’ button	All required boxes must be filled	Fail	

Post-condition: New issue created successfully

4.3.8 Edit a project

Test Case: 4.3.8	**Test Case Name:** Edit a proejct
System: Bug tracker	**Subsystem:**
Designed by: Jubaer Hossain	**Design Date:**
Executed by:	**Execution Date:**
Description: Registered testers can modify and update a project	
Pre-condition: ★ Must be logged in	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click the project that he wants to modify, fill up the updated info and press “save” button	System update the selected project	Pass	
2	Does not fill up all the required boxes and clicks on “Save” button	All required boxes must be filled	Fail	
Post-condition: Project modified successfully				

4.3.9 Edit an issue

Test Case: 4.3.9	**Test Case Name:** Edit an issue
System: Bug tracker	**Subsystem:**
Designed by: Jubaer Hossain	**Design Date:**
Executed by:	**Execution Date:**
Description: Registered testers can modify and update an issue	
Pre-condition: ★ Must be logged in	

Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click the issue that he wants to modify, fill up the updated info and press “save” button	System update the selected project	Pass	
2	Does not fill up all the required boxes and clicks on “Save” button	All required boxes must be filled	Fail	

Post-condition: Issue modified successfully

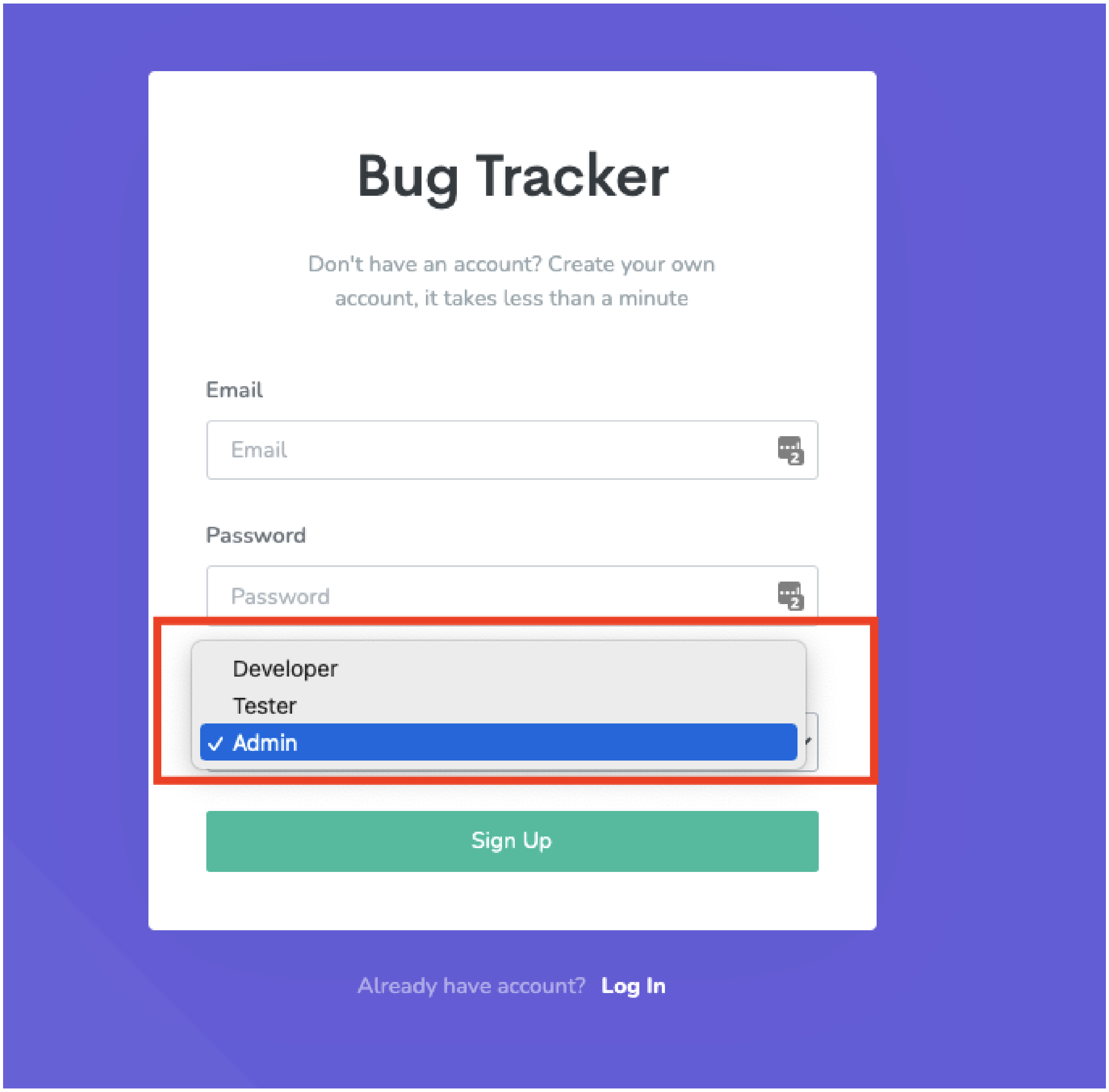
4.3.10 Comment under an issue

Test Case: 4.3.10		**Test Case Name:** Assigned developers and all testers can comment under an issue		
System: Bug tracker		**Subsystem:**		
Designed by: Jubaer Hossain		**Design Date:**		
Executed by:		**Execution Date:**		
Description: Registered testers and assigned developer can comment under an issue				
Pre-condition: ★ Must be logged in				
Step	**Action**	**Response**	**Pass/Fail**	**Comment**
1	Click the issue that he wants to comment, fill up the comment box and press enter button	A new comment has been created under that issue	Pass	
2	User doesn’t fillup the comment box or hasn’t pressed enter button to enter the comment	All required boxes must be filled and user must press enter	Fail	
Post-condition: Comment added under the issue successfully.				

Chapter 5: User Manual

5.1 Admin

5.1.1 Admin Registration



Bug Tracker

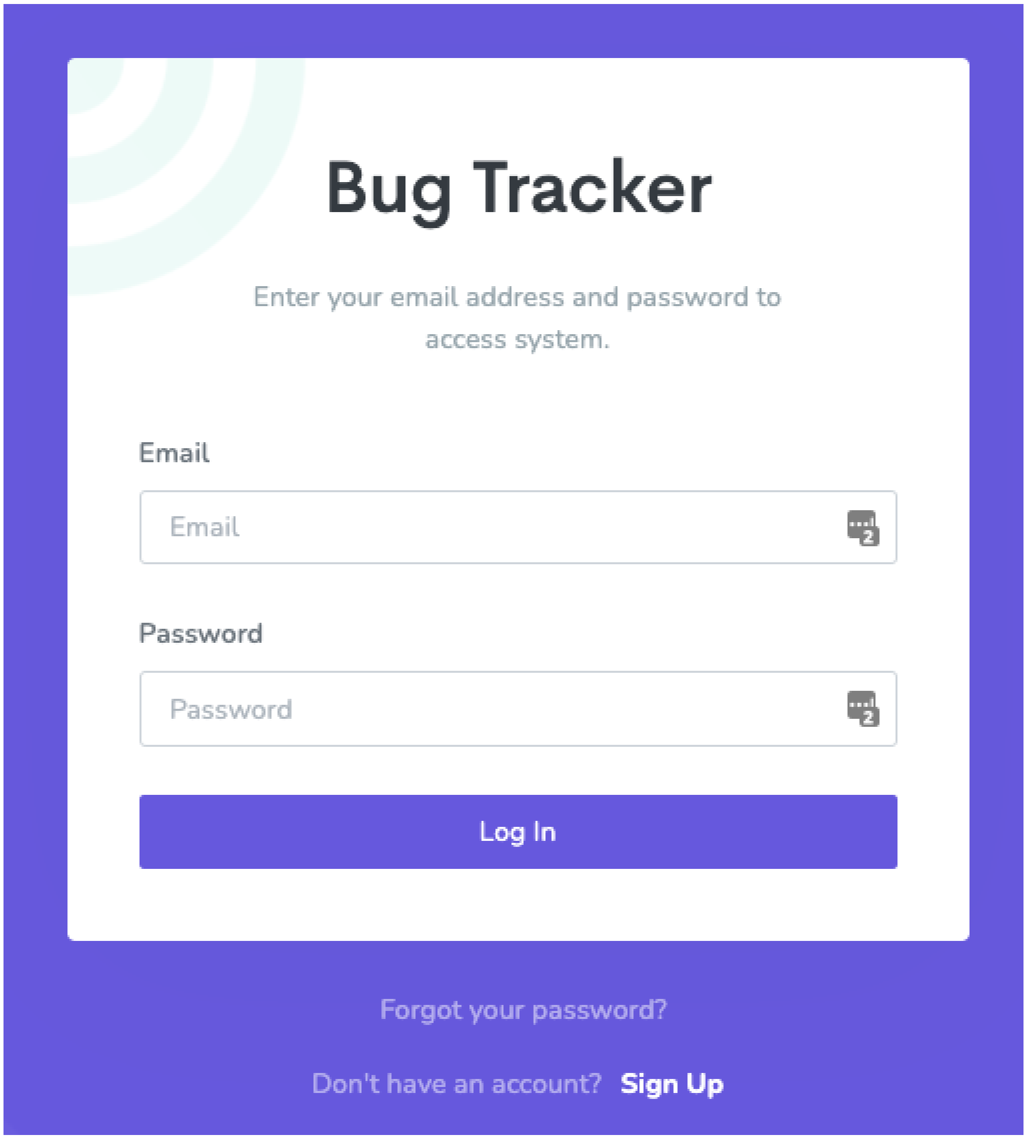
Don't have an account? Create your own account, it takes less than a minute

Email

Password

Fig 5.1: Admin registration

5.1.2 Admin login

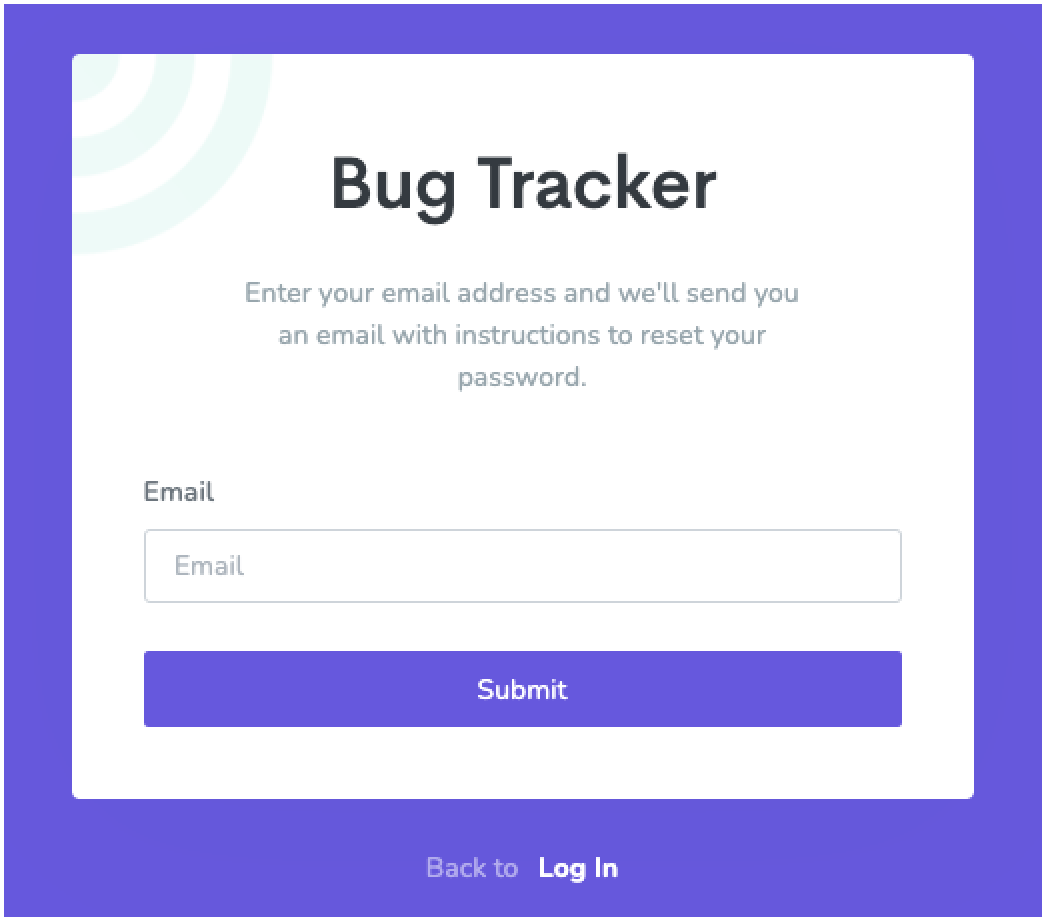


Bug Tracker

Enter your email address and password to access system.

Fig 5.2 Admin login

5.1.3 Password recovery



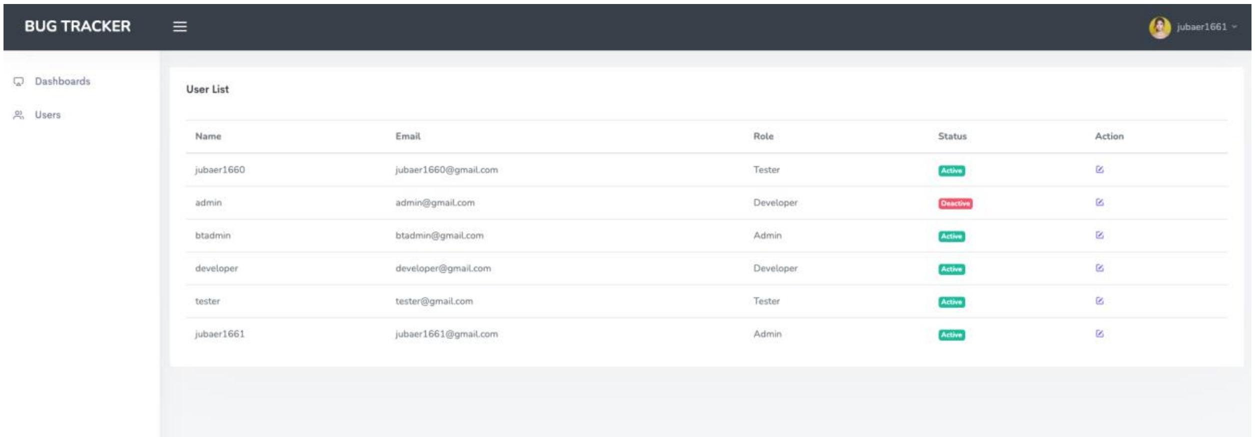
Bug Tracker

Enter your email address and we'll send you an email with instructions to reset your password.

Email: Email

Fig 5.3 Password recovery

5.1.4 Admin Dashboard: User list

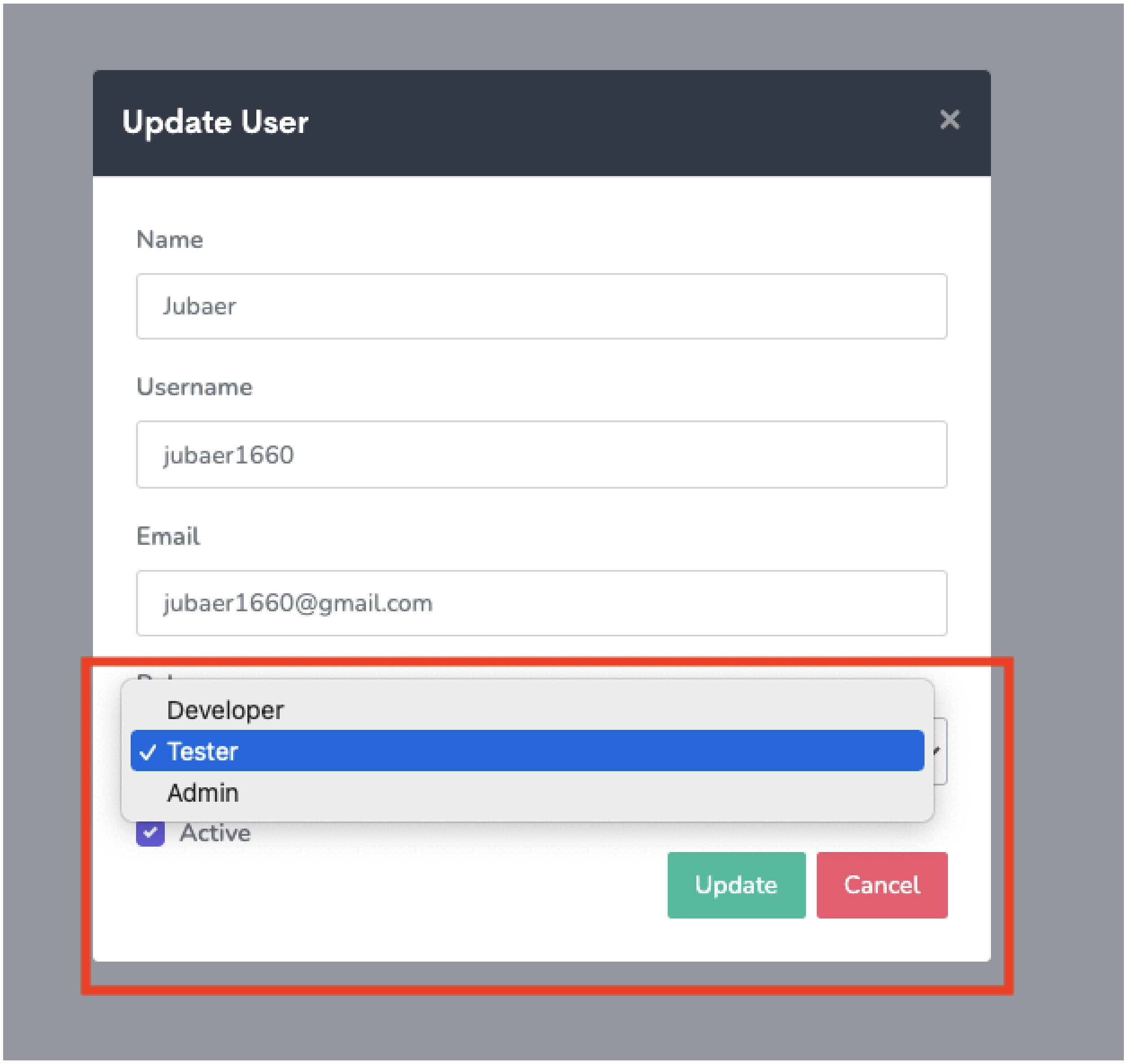


User List

Name	Email	Role	Status	Action
jubaer1660	jubaer1660@gmail.com	Tester	Active	✎
admin	admin@gmail.com	Developer	Deactive	✎
btadmin	btadmin@gmail.com	Admin	Active	✎
developer	developer@gmail.com	Developer	Active	✎
tester	tester@gmail.com	Tester	Active	✎
jubaer1661	jubaer1661@gmail.com	Admin	Active	✎

Fig 5.4 User list

5.1.5 Update user role



Update User

Name

Jubaer

Username

jubaer1660

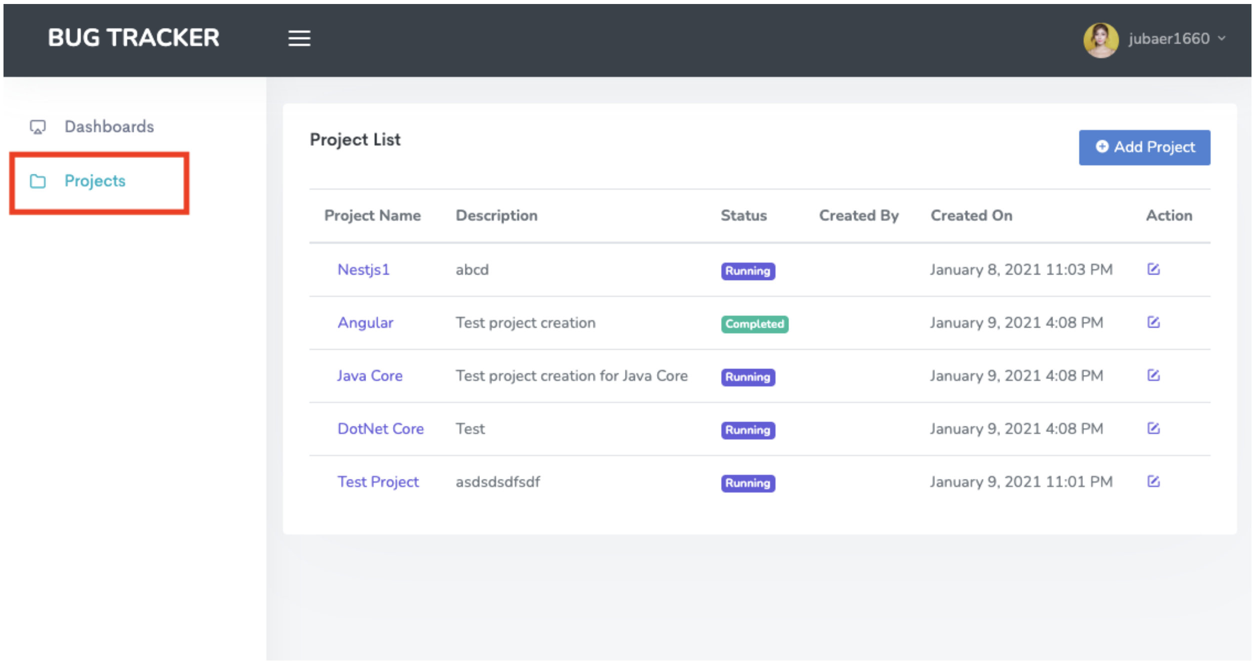
Email

jubaer1660@gmail.com

Fig 5.5 Update user role

5.2 User manual: Tester

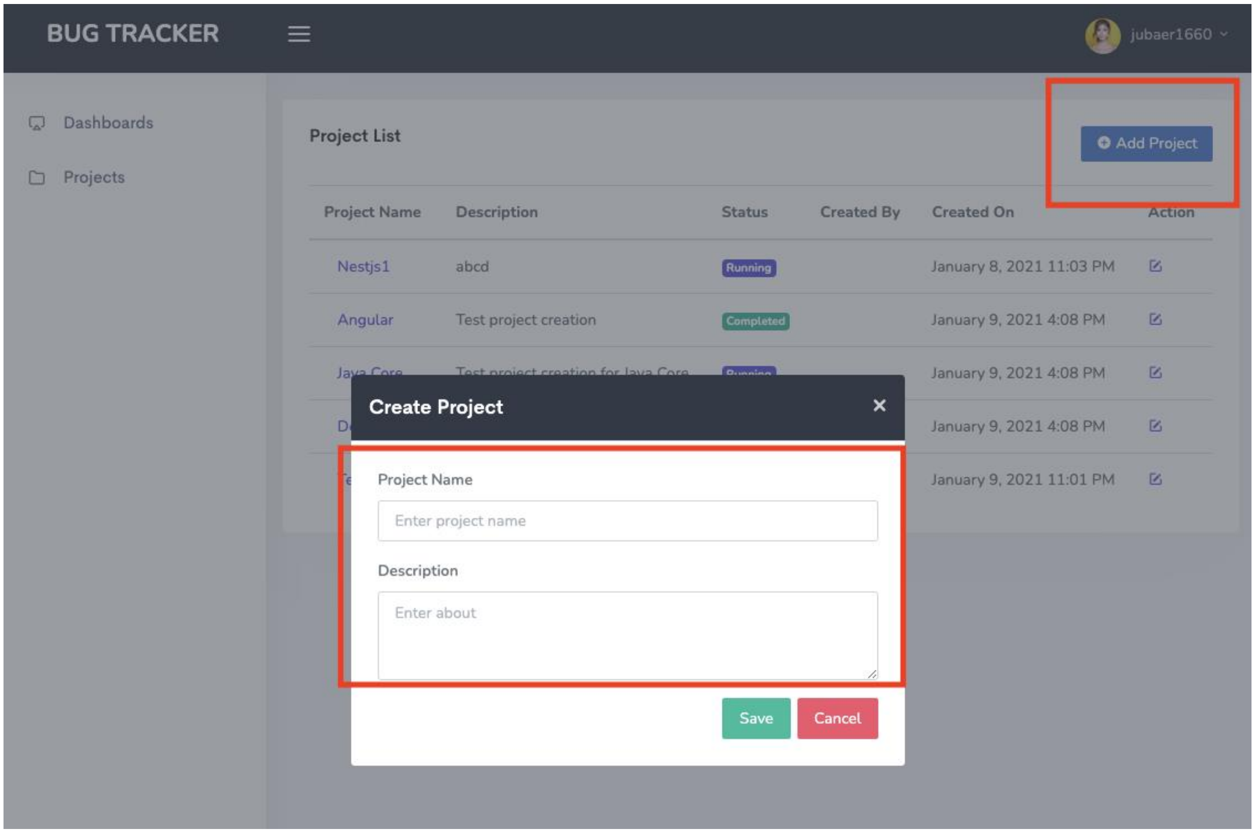
5.2.1 View project list



Project Name	Description	Status	Created By	Created On	Action
Nestjs1	abcd	Running		January 8, 2021 11:03 PM	
Angular	Test project creation	Completed		January 9, 2021 4:08 PM	
Java Core	Test project creation for Java Core	Running		January 9, 2021 4:08 PM	
DotNet Core	Test	Running		January 9, 2021 4:08 PM	
Test Project	asdsdsdfsdf	Running		January 9, 2021 11:01 PM	

Fig 5.6 View all projects

5.2.2 Add new project


BUG TRACKER

jubaer1660

Project List

Project Name	Description	Status	Created By	Created On	Action
Nestjs1	abcd	Running		January 8, 2021 11:03 PM	
Angular	Test project creation	Completed		January 9, 2021 4:08 PM	
Java Core	Test project creation for Java Core	Running		January 9, 2021 4:08 PM	
D				January 9, 2021 4:08 PM	
Te				January 9, 2021 11:01 PM	

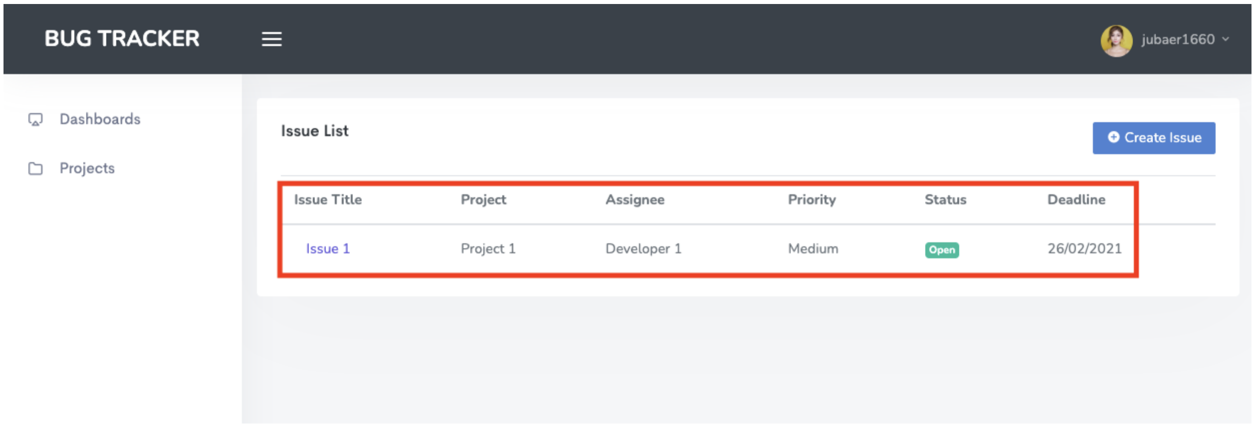
Create Project

Project Name: Enter project name

Description: Enter about

Fig 5.7: Create new project

5.2.3 View all issues under a project


BUG TRACKER

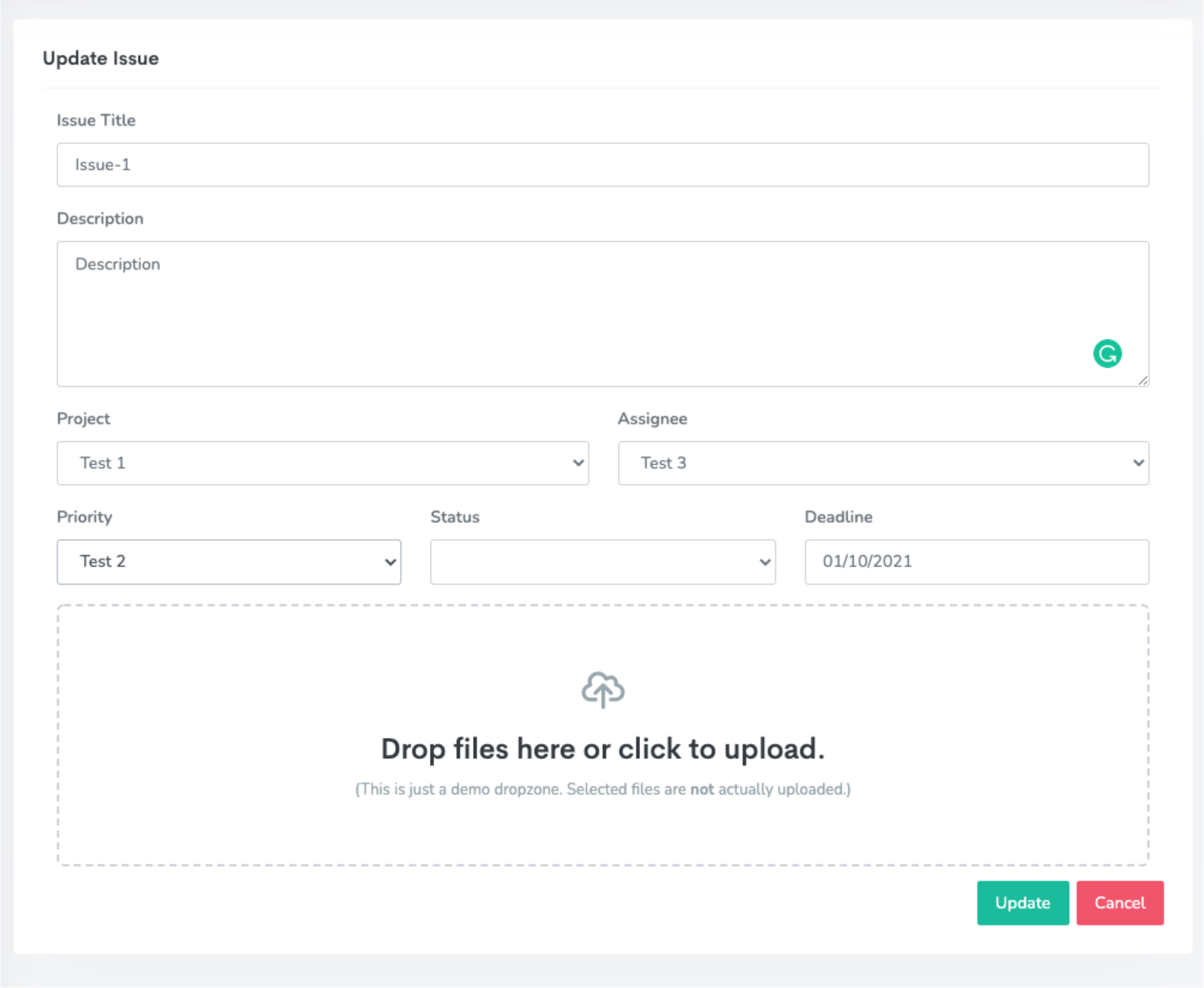
jubaer1660

Issue List

Issue Title	Project	Assignee	Priority	Status	Deadline
Issue 1	Project 1	Developer 1	Medium	Open	26/02/2021

Fig 5.8 View all issues under a project

5.2.3 Create a new issue



Update Issue

Issue Title

Issue-1

Description

Description

Project

Test 1

Assignee

Test 3

Priority

Test 2

Status

Deadline

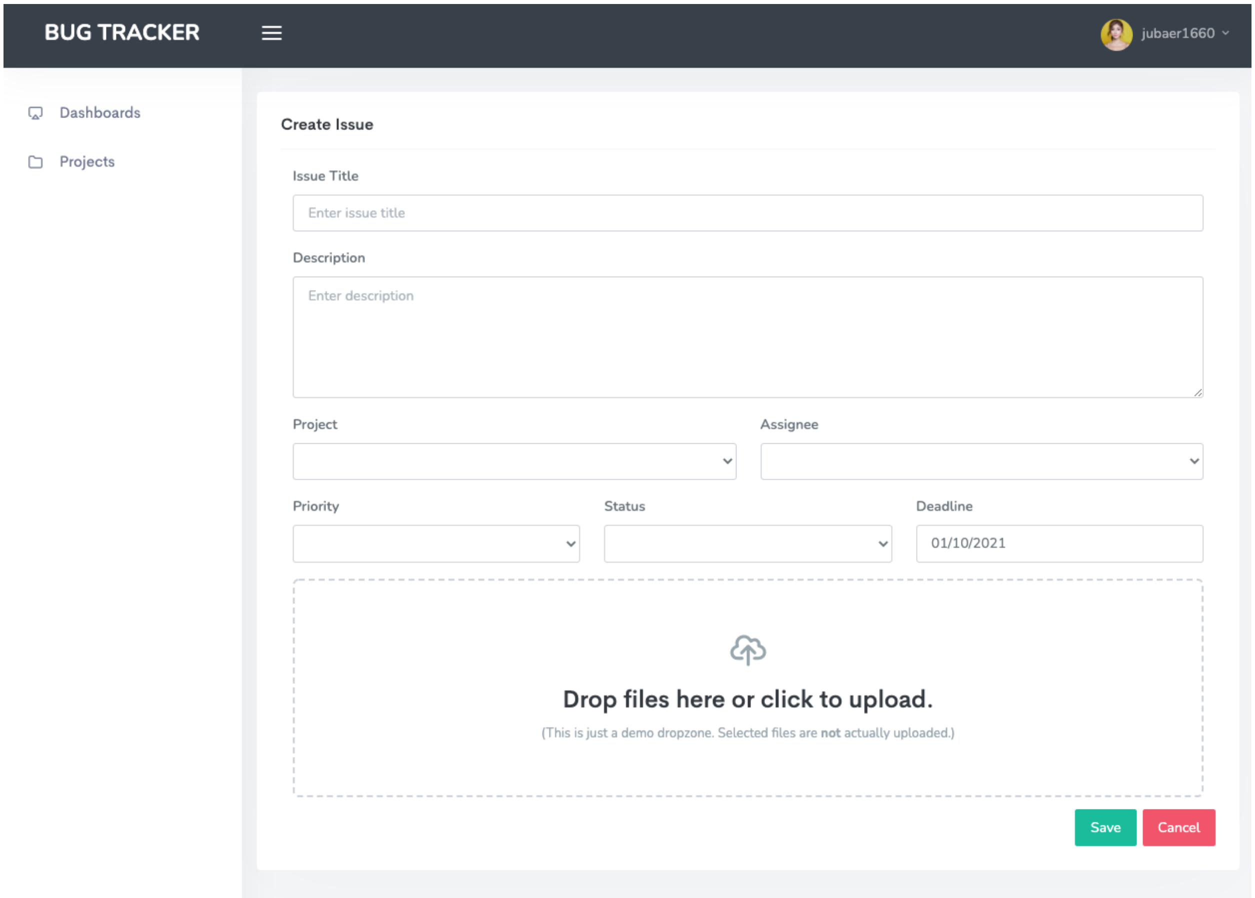
01/10/2021

Drop files here or click to upload.

(This is just a demo dropzone. Selected files are **not** actually uploaded.)

Fig 5.9 Create a new issue

5.2.4 Edit an issue



Create Issue

Issue Title

Enter issue title

Description

Enter description

Project

Assignee

Priority

Status

Deadline

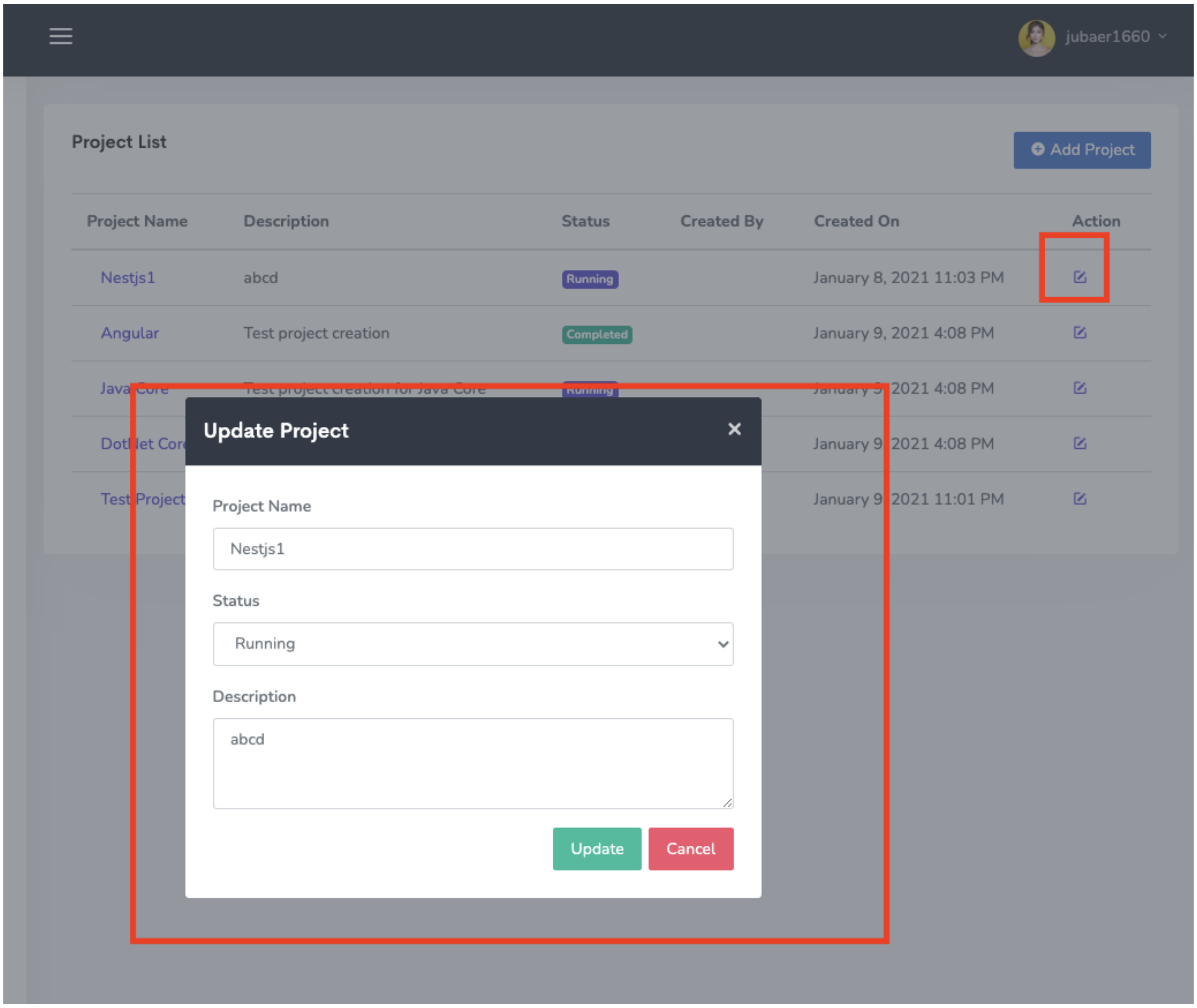
01/10/2021

Drop files here or click to upload.

(This is just a demo dropzone. Selected files are **not** actually uploaded.)

Fig 5.10 Edit an issue

5.2.5 Edit a project



Project List

Project Name	Description	Status	Created By	Created On	Action
Nestjs1	abcd	Running		January 8, 2021 11:03 PM	
Angular	Test project creation	Completed		January 9, 2021 4:08 PM	
Java Core	Test project creation for Java Core	Running		January 9, 2021 4:08 PM	
DotNet Cor				January 9, 2021 4:08 PM	
Test Project				January 9, 2021 11:01 PM	

Update Project

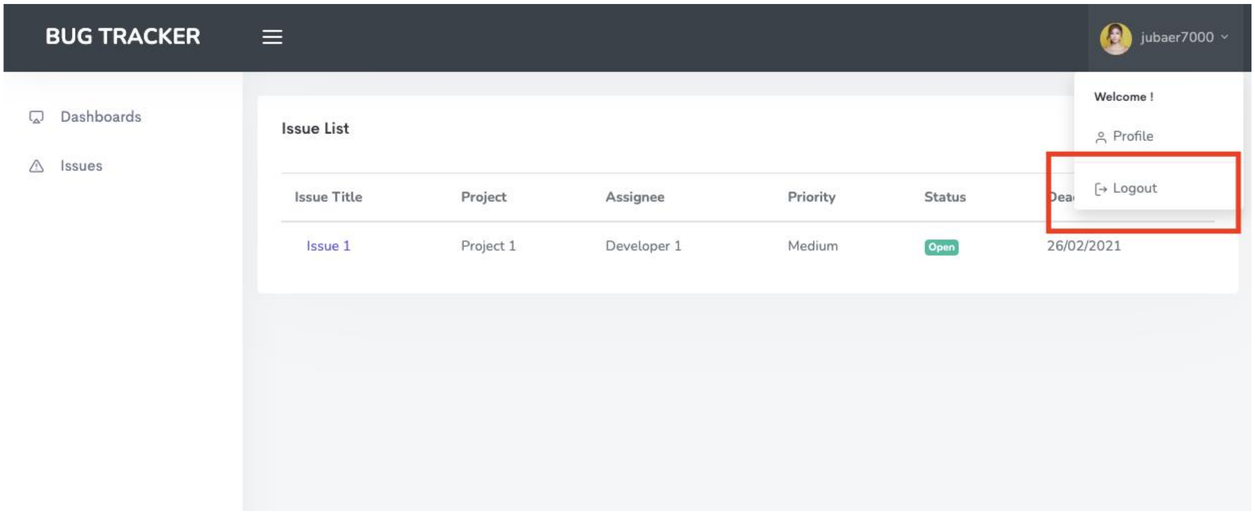
Project Name: Nestjs1

Status: Running

Description: abcd

Fig 5.11 Edit a project

5.2.6 Logout



Issue List

Issue Title	Project	Assignee	Priority	Status	Dea
Issue 1	Project 1	Developer 1	Medium	Open	26/02/2021

Fig 5.12 Logout

Chapter 6: Conclusion

6.1 Github link

The final version of the project bug-tracker has been deployed to this github link: https://github.com/jubaer-hossain/Bug-Tracker

6.2 Future scope

a. Add archive issue & project feature
b. More clear user role and more restricted functionalities for each user
c. Deploy the project in a live server

THE END