

E-Tutor

Supervisor: Lai Manh Dzung

May 2020

Enterprise Web Software Development (Group Report)

Git Repository: <https://bit.ly/e-tutor-github>

Presentation (Screencast): <https://bit.ly/e-tutor-screencast>

Presentation (Powerpoint): <https://bit.ly/e-tutor-slide>

Website: <https://bit.ly/e-tutor-website>

Login Details: Admin (Email: draknoel@gmail.com, Password: password)

Tutor (Email: leroy.riley@example.com, Password: password)

Student (Email: pedro.ortiz@example.com, Password: password)

Backlog: <https://bit.ly/e-tutor-backlog>

Team Members: Israel Obanjesu (Scrum Master, Server-side Developer),

Trương Quốc Cường (Server-side Developer),

Idris Usman Fadil (Tester),

Ngô Vĩnh Thái (Tester),

Trịnh Anh Tuấn (Client-side Developer),

Phạm Trần Hồng Sơn (Client-side Developer)

A dissertation submitted in partial fulfilment of the University
of Greenwich's BSc(Hons). Computing

TABLE OF CONTENTS

Table of Contents	ii
List of Figures	vii
List of Tables	vii
1 Introduction	1
2 Project Summary	2
3 Requirements Specification	3
3.1 Functional Requirements	3
3.2 Non-Functional Requirements	3
4 Database Design.....	4
4.1 Entity Relationship Diagram	4
4.2 Data Security.....	5
4.2.1 Role-Based Access Control Model	5
4.2.2 Limit Network Exposure	6
4.2.3 Authentication and Authorization	6
4.2.4 Transport Encryption	6
4.3 Data Validation	6
5 Site Design.....	8
5.1 Sitemap.....	8
5.2 Wireframe	9
5.2.1 Login	9
5.2.2 Homepage	11
5.2.3 Student	13
5.2.4 Admin Dashboard	16
5.2.5 Issue	18

5.2.6	Blog	24
5.2.7	Allocate student to tutor.....	27
5.3	Interface Design	28
5.3.1	Login – Forgot password – Change password	28
5.3.2	Homepage	30
5.3.3	Dashboard	32
5.3.4	Student – Student Profile	33
5.3.5	Issue - Issue Detail – Create New Issue.....	35
5.3.6	Blog – Post Detail – Create Blog	38
5.3.7	Meeting	42
5.3.8	Admin Dashboard	43
5.3.9	All Group – Add New Group	45
5.3.10	All User – Add User	47
5.3.11	Allocate student to tutor.....	49
5.3.12	Meeting – Create New Meeting	50
5.3.13	Admin Profile – Edit profile	52
6	Functionalities	54
6.1	Login	54
6.1.1	Login with Email / Password.....	54
6.1.2	Login with Google	54
6.2	Allocate Students to Tutor	55
6.3	Blogging	56
6.3.1	Create Post	56
6.3.2	View Posts	56
6.3.3	Like Post	57
6.3.4	Dislike Post	58

6.4	Sending Emails	59
6.4.1	Sending Email for Allocation of Students to Tutor	59
6.4.2	Sending Email for Resetting Password.....	60
6.4.3	Sending Email for Meeting Request	61
6.4.4	Sending Email for Meeting Reminder	61
6.5	Messaging.....	62
6.5.1	Send Message.....	62
6.5.2	View Messages	63
6.5.3	Delete Message	63
6.6	Issues	64
6.6.1	Create Issue	64
6.6.2	View Issues	65
6.6.3	Update Issue	65
6.6.4	Delete Issue	66
6.7	Flowchart for Login with Email / Password.....	66
6.8	Flowchart for Login with Google	67
6.9	Flowchart for Allocating Students to Tutor	68
6.10	Flowchart for Creating Post	69
6.11	Flowchart for Liking Post	70
6.12	Flowchart for Disliking Post	71
6.13	Flowchart for Sending Message	72
6.14	Flowchart for Deleting Message	73
6.15	Flowchart for Creating Issue	74
6.16	Flowchart for Updating Issue	75
6.17	Flowchart for Deleting Issue	76
7	Testing	77

7.1	Test plan	77
7.1.1	Test plan identifier	77
7.1.2	Introduction.....	77
7.1.3	References	77
7.1.4	Test artefacts	77
7.1.5	Test features.....	77
7.1.6	Test Approach.....	78
7.1.7	Features failed	78
7.1.8	Deliverables	78
7.1.9	Estimates	78
7.1.10	Schedule	78
7.1.11	Responsibilities	79
7.1.12	Approvals.....	79
7.2	Test Cases and Test Logs	79
7.3	Test Evidence	84
7.3.1	Test for sending messages	84
7.3.2	Test for deleting messages.....	85
7.3.3	Test for login with correct input data	85
7.3.4	Test for login with wrong input data (validation).....	86
7.3.5	Test for logout	86
7.3.6	Test for reset password.....	86
7.3.7	Test for reset password invalid mail (validation)	87
7.3.8	Test for allocation of students to tutor.....	87
7.3.9	Test for display all allocated students	88
7.3.10	Test for display all allocated tutors.....	88
7.3.11	Test for displaying unallocated tutor.....	89

7.3.12	Test for creating students	89
7.3.13	Test for creating students(validation).....	89
7.3.14	Test for deactivating user.....	90
7.3.15	Test for activating user.....	90
7.3.16	Test for creating and requesting for a meeting	91
7.3.17	Test for creating a issue	91
7.3.18	Test for viewing issue details.....	91
7.3.19	Test for editing issue	92
7.3.20	Test for deleting issue	92
7.3.21	Test for creating comment.....	93
7.3.22	Test for viewing comment.....	93
7.3.23	Test for deleting comment.....	94
8	Development Process	95
8.1	User Stories.....	95
8.2	Product Backlog	95
8.3	Sprints.....	97
8.3.1	Sprint 1	97
8.3.2	Sprint 2	98
8.3.3	Sprint 3	99
8.3.4	Sprint 4	101
8.3.5	Sprint 5	103
8.4	Release Burndown	104
8.5	Team Challenges	105
8.6	Team Meetings	105
9	Presentation.....	108
10	Conclusion.....	111

LIST OF FIGURES

Figure 1: Entity Relationship Diagram (ERD)	4
Figure 2: Configuring MongoDB Database Access.....	5
Figure 3: Configuring Roles for MongoDB Users	5
Figure 4: Configuring Network Access for MongoDB	6
Figure 5: Data Validation	7
Figure 6.sitemap	8
Figure 7.wireframe of login	9
Figure 8.wireframe of login on mobile.....	10
Figure 9.wireframe of homepage	11
Figure 10.wireframe of homepage on mobile.....	12
Figure 11.wireframe of student page	13
Figure 12.wireframe of student profile.....	14
Figure 13.wireframe of student page - student profile page.....	15
Figure 14.wireframe of admin dashboard	16
Figure 15. wireframe of admin dashboard on mobile	17
Figure 16. wireframe of issue	18
Figure 17. wireframe of issue detail	19
Figure 18. wireframe of creating issue.....	20
Figure 19.wireframe of editing issue	21
Figure 20. wireframe of issue detail - edit issue.....	22
Figure 21. wireframe of creating issue - issue page	23
Figure 22. wireframe of blog page.....	24
Figure 23.wireframe of post detail	25
Figure 24.wireframe of creating post page	26
Figure 25.wireframe of blog - post detail - creating post on mobile.....	27
Figure 26. wireframe of step 1 of allocating student to tutor	27
Figure 27. wireframe of step 2 of allocating student to tutor	28
Figure 28.Image of Login page.....	29

Figure 29. Image of Forgot password page	29
Figure 30.Image of Change password page	30
Figure 31.Image of Login - Forgot Password - Change Password mobile	30
Figure 32.Image of homepage	31
Figure 33.Image of Homepage mobile	32
Figure 34.Image of dashboard for user	32
Figure 35.Image of dashboard for user mobile	33
Figure 36.Image of student page	33
Figure 37. Image of Student profile page.....	34
Figure 38. Image of Student - Student profile page	35
Figure 39.Issue page.....	35
Figure 40.Issue detail page	36
Figure 41.Create Issue page	37
Figure 42.Image of Issue - Issue Detail - Create Issue page mobile	38
Figure 43. Image of blog page.	39
Figure 44. Image of post detail page.....	40
Figure 45. Image of creating post page.....	41
Figure 46.image of blog page - post detail - creating post page on mobile.....	41
Figure 47.Meeting page for user	42
Figure 48.Meeting page for user on mobile	43
Figure 49.Image of admin dashboard	44
Figure 50.Image of admin dashboard on mobile.....	45
Figure 51. Image of list of group	45
Figure 52.Step 1 of adding new group	46
Figure 53.Step 2 of adding new group	46
Figure 54.Image of list of group and steps of adding new group	47
Figure 55. All user page	47
Figure 56.Image of Add new user page.....	48
Figure 57. All user - Add New User page on mobile	48
Figure 58.Step 1 of allocate student to tutor	49
Figure 59.Step 2 of allocating student to tutor	49
Figure 60.Steps of allocating student to tutor.....	50

Figure 61.All meeting for admin	50
Figure 62.Image of adding new meeting.....	51
Figure 63.Image of all meeting - add new meeting page for admin on mobile.....	51
Figure 64.Image of admin profile	52
Figure 65.Image of editing profile.....	52
Figure 66.Image o admin profile - edit profile page on mobile	53
Figure 67: Code for Logging In with Email / Password	54
Figure 68: Code for Logging in with Google	55
Figure 69: Code for allocating students to tutor	55
Figure 70: Code for creating posts.....	56
Figure 71: Code for viewing posts.....	57
Figure 72: Code for liking a post	58
Figure 73: Code for disliking a post.....	59
Figure 74: Code for sending email for allocation of students to tutor	60
Figure 75: Code for sending email for resetting password	61
Figure 76: Code for sending email for meeting request	61
Figure 77: Code for sending email for meeting reminder.....	61
Figure 78: Code for sending messages.....	62
Figure 79: Code for viewing messages.....	63
Figure 80: Code for deleting messages	63
Figure 81: Code for creating issues.....	64
Figure 82: Code for viewing issues.....	65
Figure 83: Code for updating issues.....	65
Figure 84: Code for deleting issues	66
Figure 85: Flowchart for Login with Email / Password	66
Figure 86: Flowchart for Login with Google	67
Figure 87: Flowchart for Allocating Students to Tutor	68
Figure 88: Flowchart for Creating Post	69
Figure 89: Flowchart for liking post	70
Figure 90: Flowchart for disliking post	71
Figure 91: Flowchart for sending messages	72
Figure 92: Flowchart for deleting message	73

Figure 93: Flowchart for creating issue	74
Figure 94: Flowchart for updating issue.....	75
Figure 95: Flowchart for deleting issue.....	76
Figure 96: Test Cases for Sprint 1	80
Figure 97: Test Cases for Sprint 3	81
Figure 98: Test Cases for Sprint 4	82
Figure 99: Test Cases for Sprint 5	83
Figure 100 Test for sending messages	84
Figure 101 test for delete messages	85
Figure 102 Test for login.....	85
Figure 103 Test for login.....	86
Figure 104 Test for logout	86
Figure 105 Test for forget password	86
Figure 106 test for rest password	87
Figure 107 Allocation student to tutor	87
Figure 108 All Allocated students	88
Figure 109 all allocated tutors	88
Figure 110 all unallocated tutors	89
Figure 111: Test for creating students	89
Figure 112: Test for creating students (validation)	89
Figure 113 deactivate user	90
Figure 114 Test for activate user	90
Figure 115: Test for creating and requesting for a meeting	91
Figure 116: Test for creating an issue	91
Figure 117: Test for viewing issue details	92
Figure 118: Test for editing issue.....	92
Figure 119: Test for deleting issue.....	92
Figure 120: Test for creating comment.....	93
Figure 121: Test for viewing comment	93
Figure 122: Test for deleting comment.....	94
Figure 123: User Stories	95
Figure 124: Product Backlog.....	96

Figure 125: Sprint 1 Backlog	97
Figure 126: Burndown Chart Sprint 1	98
Figure 127: Sprint 2 Backlog	98
Figure 128: Burndown Chart For Sprint 2	99
Figure 129: Sprint 3 Backlog	100
Figure 130: Burndown Chart For Sprint 3	101
Figure 131: Sprint 4 Backlog	102
Figure 132: Burndown Chart For Sprint 4	103
Figure 133: Sprint 5	104
Figure 134: Burndown Chart For Sprint 5	104
Figure 135: Release Burndown	104
Figure 136: E-Tutor Presentation.....	111

LIST OF TABLES

Table 1: Functional Requirements	3
Table 2: Team Meeting Details	107

1 INTRODUCTION

This report provides the summary and key highlights of the project's journey. From requirement specification which gave a detailed information of both functional and non-functional requirements to the project design. The project design which includes both the database and site design implemented and the reason behind this was shown. Subsequently, a variety of different software methodologies available was briefly talked about and how the project team arrived at a given software development methodology which was used for this project.

2 PROJECT SUMMARY

The project started on 5th of March and lasted for almost two months. Throughout the development course, we had a total of 5 sprints. At the early state of the project's development, the team planned to have a physical meeting twice (Sundays and Thursdays) in a week. We were able to hold a physical meeting twice but after that we couldn't meet anymore due to the global crisis of Covid-19. Since then, we decided to move our meeting discussions to an online messaging service (Messenger).

The first two physical meetings were really helpful. On the first meeting, we introduced ourselves and discussed about our team roles and the technologies to use for the project. On the second meeting, we designed the prerequisites for the project (database, and wireframes). After the first two meetings, we discussed about the first sprint online and the Team Leader (Israel) documented it. Throughout the sprints, we had some challenges and bugs, but we brainstorm and help each other out to fix the bug / challenge. Overall, it can be said that the journey of this project's development was insightful as it helped many of us to learn new technologies and how to work efficiently with Agile Scrum.

3 REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

S.N	Requirements
1	The system shall enable user to login
2	The system shall enable student send messages to tutor
3	The system shall enable tutor create group chat
4	The system shall enable tutor create request for meetings
5	The system shall enable student create Issues
6	The system shall enable tutor receive email notifications
7	The system shall enable tutor comment on students' issues
8	The system shall enable student view personal dashboard
9	The system shall enable tutor view personal dashboard
10	The system shall enable administrator restrict user access to certain pages and features
11	The system shall enable administrator deactivate/delete a student/staff
12	The system shall enable administrator receive notifications from students
13	The system shall enable administrator create students and tutor accounts
14	The system shall enable user to uploading documents and comment on them, for blogging

Table 1: Functional Requirements

3.2 Non-Functional Requirements

The interface must be suitable for all devices (phones, tables, desktops)

Each user will have their own personal dashboard base on their role. Student will have dashboard summarizing their interaction with their person tutor and tutor will have a dashboard of their personal tutees that can be sorted and filter appropriately.

The website must have a place for messaging.

Homepage for tutor, student and admins must be different.

4 DATABASE DESIGN

4.1 Entity Relationship Diagram

This project using MongoDB which is simply a database, it is non-relational database and no sequel database (NoSQL).

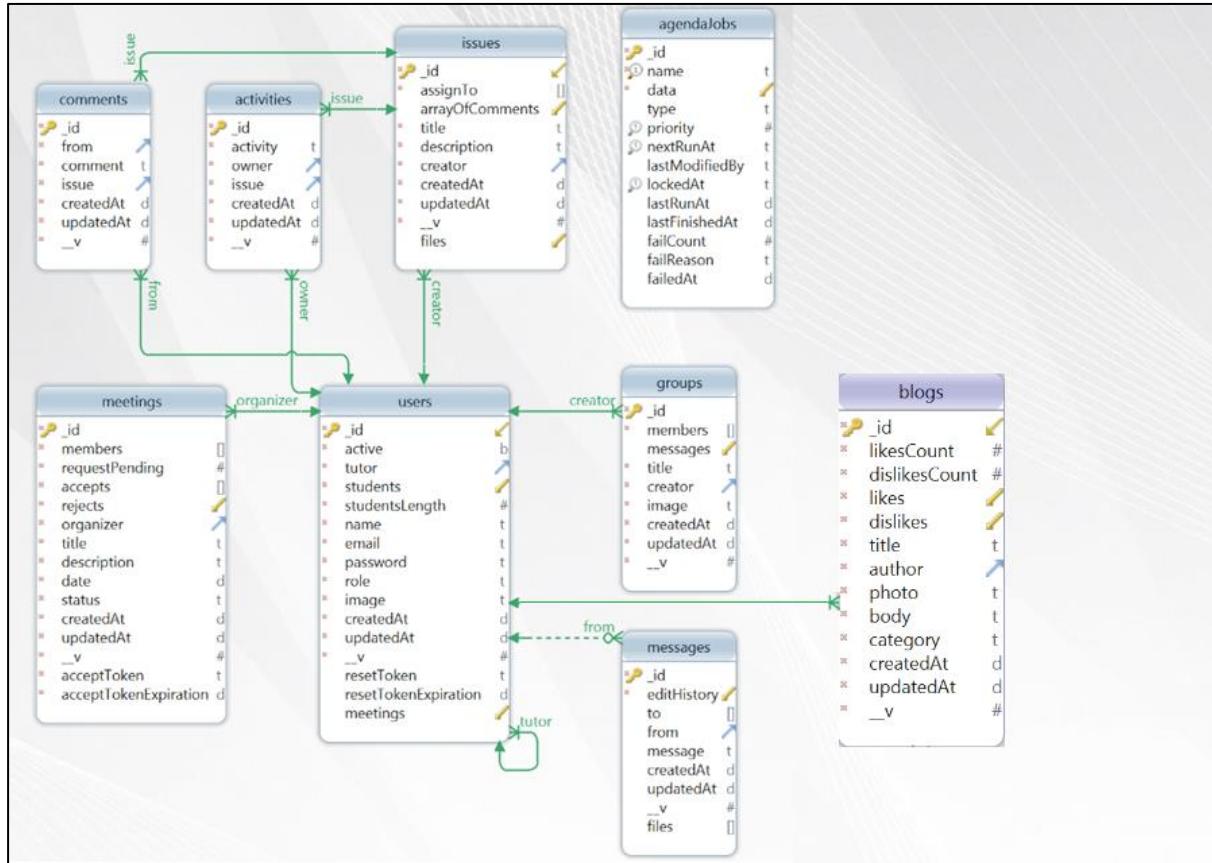


Figure 1: Entity Relationship Diagram (ERD)

As shown in the image above, we have a total of eight (8) entities, and some of them are connected to one another. Although, there are no relationship constraints among them because NoSQL does not have relationship rules.

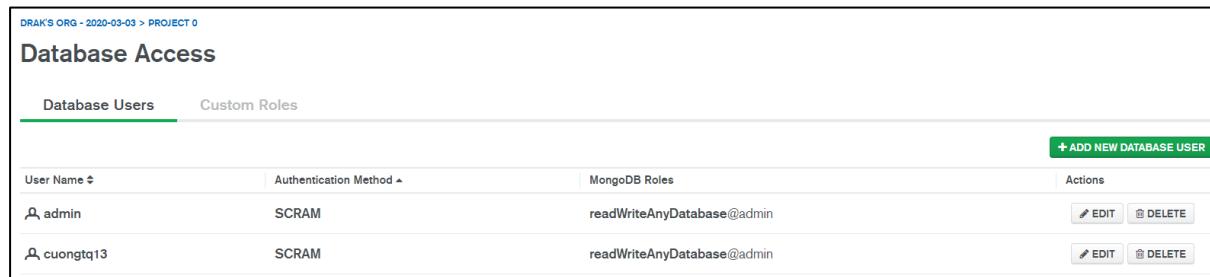
The reasons for using MongoDB in this project because it stores data in flexible; the document model maps to the object in application code, making data easy to work with. MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.

4.2 Data Security

MongoDB provides a lot of features such as access control, authentication, and encryption, to secure MongoDB deployments. Below is some features this product implement to protect and secure this project's database.

4.2.1 Role-Based Access Control Model

MongoDB require to create a user administrator first, then create additional users. Create a unique MongoDB user for each application/person that accesses the system.

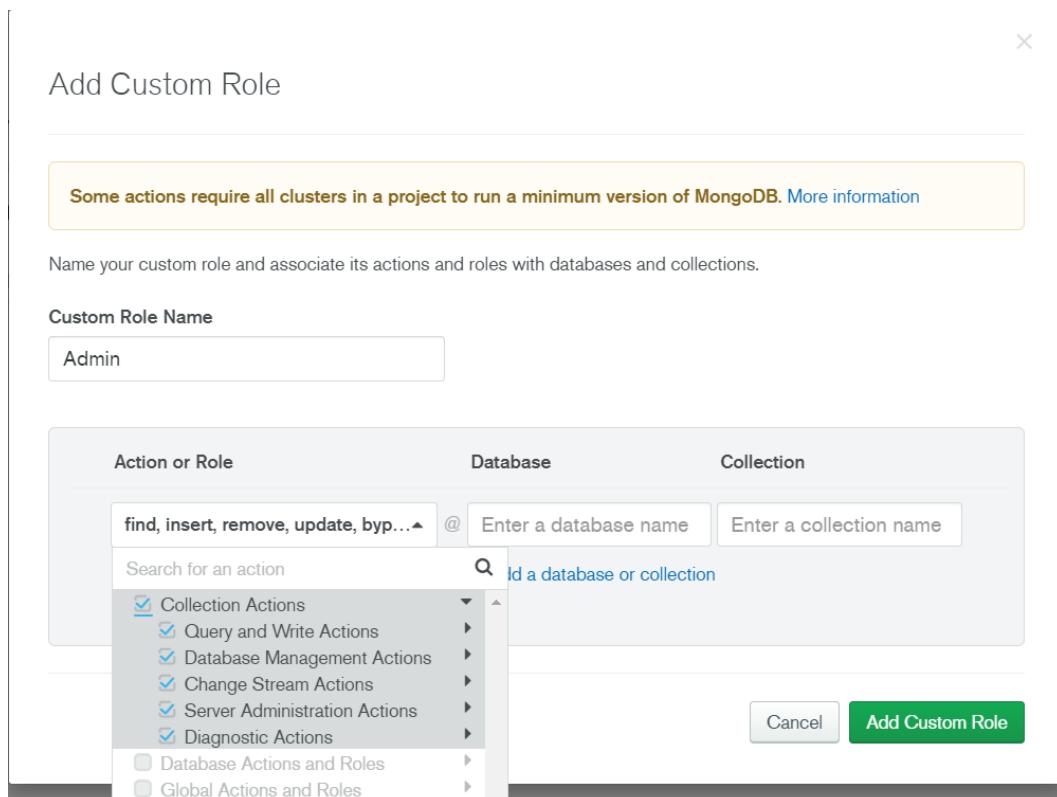


The screenshot shows the 'Database Access' section of the MongoDB interface. It lists two database users:

User Name	Authentication Method	MongoDB Roles	Actions
admin	SCRAM	readWriteAnyDatabase@admin	<button>EDIT</button> <button>DELETE</button>
cuongtq13	SCRAM	readWriteAnyDatabase@admin	<button>EDIT</button> <button>DELETE</button>

A green button at the top right says '+ ADD NEW DATABASE USER'.

Figure 2: Configuring MongoDB Database Access



The screenshot shows the 'Add Custom Role' dialog box. It includes a note: 'Some actions require all clusters in a project to run a minimum version of MongoDB. More information'. The 'Custom Role Name' field contains 'Admin'. The 'Action or Role' dropdown shows 'find, insert, remove, update, bypass...'. The 'Database' and 'Collection' fields both have placeholder text 'Enter a database name' and 'Enter a collection name'. A sidebar lists actions under 'Collection Actions' (checked) and other categories like 'Database Actions and Roles' (unchecked). Buttons at the bottom are 'Cancel' and 'Add Custom Role'.

Figure 3: Configuring Roles for MongoDB Users

4.2.2 Limit Network Exposure

Ensure that MongoDB runs in trusted network environment and configure security groups or firewall to control traffic for MongoDB instances. Only allow trusted clients to access the network interfaces and ports on this MongoDB instances are available. For instance, this project use IP whitelisting to allow access from trusted IP addresses.

The screenshot shows the 'Network Access' section of the MongoDB configuration interface. The 'IP Whitelist' tab is selected. A green button labeled '+ ADD IP ADDRESS' is visible. Below it, a message states: 'You will only be able to connect to your cluster from the following list of IP Addresses:'. A table lists four IP addresses, all marked as 'Active':

IP Address	Comment	Status	Actions
118.71.210.56/32		Active	<button>EDIT</button> <button>DELETE</button>
118.71.206.58/32		Active	<button>EDIT</button> <button>DELETE</button>
113.164.224.226/32		Active	<button>EDIT</button> <button>DELETE</button>
0.0.0.0/0 (includes your current IP address)		Active	<button>EDIT</button> <button>DELETE</button>

Figure 4: Configuring Network Access for MongoDB

4.2.3 Authentication and Authorization

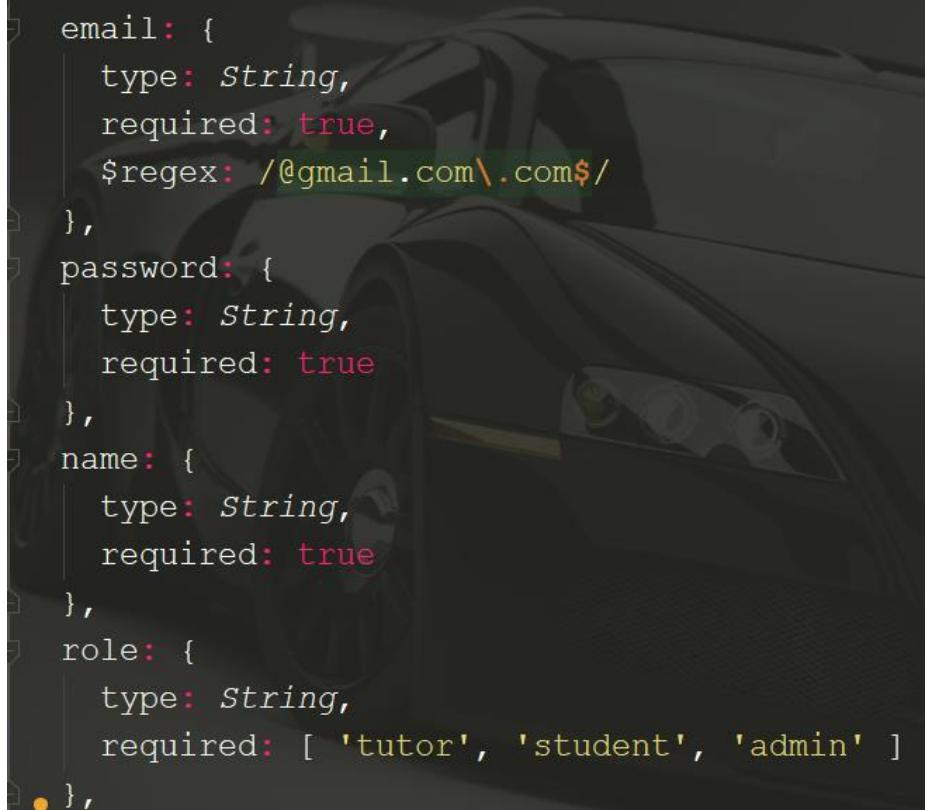
The database is password-protected so anyone cannot easily access it, thereby making the chance of data leak to be very low. Additionally, different users are given different roles in the database so anyone cannot just decide to modify the database data without permission.

4.2.4 Transport Encryption

MongoDB uses TLS/SSL (Transport Encryption) to encrypt all MongoDB's network traffic. SSL/TLS ensure that MongoDB network traffic is only readable by the intended client.

4.3 Data Validation

MongoDB provides the capability to perform schema validation during updates and insertions.



```
email: {
    type: String,
    required: true,
    $regex: /@gmail.com\.com$/
},
password: {
    type: String,
    required: true
},
name: {
    type: String,
    required: true
},
role: {
    type: String,
    required: [ 'tutor', 'student', 'admin' ]
},
```

Figure 5: Data Validation

As shown in the image above, data validation can be done in MongoDB with the use of some specific keyword such as (“type”, “required”, “\$regex” etc.);

“type” – This is used for ensuring that a field accepts data in the format specified. Some examples of data format includes: String, Boolean, Number etc.

“required” – This is used for ensuring that a field is not empty.

“regex” – This is used for validating a field based on the regular expression pattern provided.

5 SITE DESIGN

About design front-end, the Website uses CSS Bootstrap library with Bootstrap version 3, they are combined with JQuery JavaScript library, a responsive UI is achieved across the sites.

This website is not built on website templates available, so things to do first is to create a framework for the similar pages on the website. Based on this framework, we can design other pages quickly, we just changed content that pages display. The main color of the website is green, from the header, footer, logo to buttons, this thing makes up the consistency of the website, users will focus on using the website without being distracted by the color factor. This website is design simple; it focuses on information content displayed to the user. The content is displayed clearly; headings are displayed larger to the attention of the user.

The library has assisted in designing the sections of the site horizontally with different sizes in the different types of screens (example: phone, tablet, laptop...)

About button or hyperlink, besides the title by character, the familiar icon also uses to the user which can use the website easily.

5.1 Sitemap

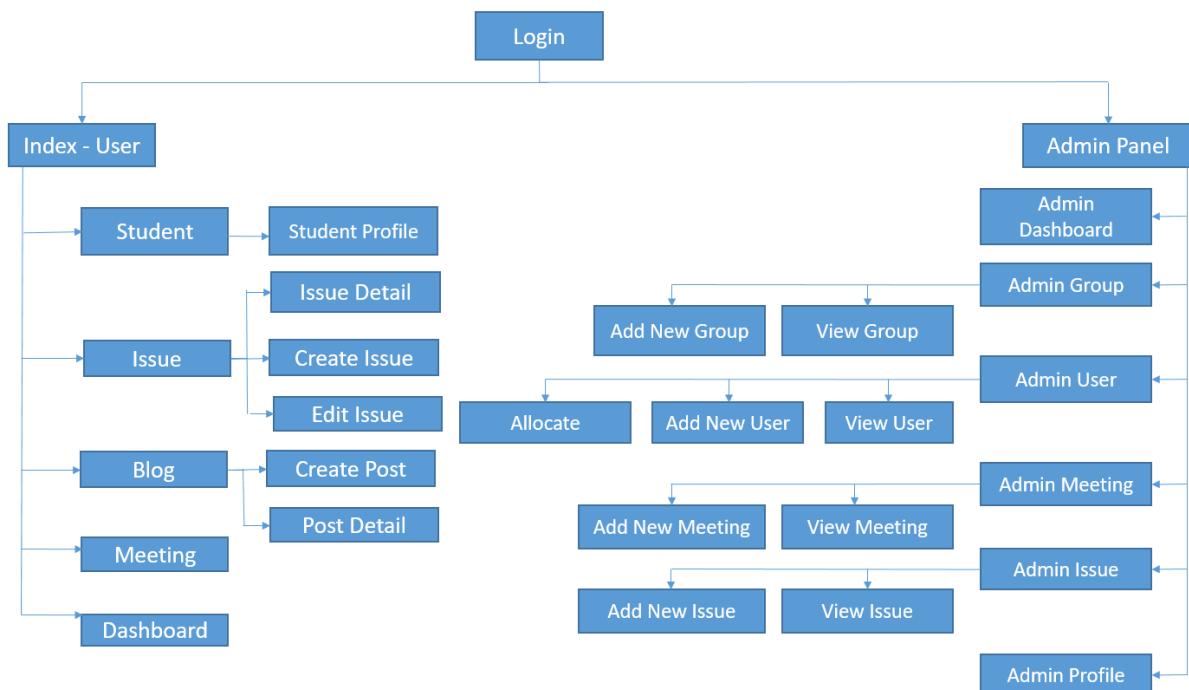


Figure 6.sitemap

The above image shows an overview of the system; it shows all available functions of the system. The hierarchy shows the process of performing the functions in it, the users will easily remember and can quickly learn how to use the system. The system has two parts that are the user side and the admin side. For the user side, there are pages such as homepage, student page, meeting page, issue page, dashboard. For the admin side, there are pages such as admin dashboard, admin-group, admin-user, admin-meeting, admin-issue, admin-profile, admin-comment.

5.2 Wireframe

5.2.1 Login

Enter email address and password to login

The wireframe shows a window titled 'E-Tutor' with a 'Login' sub-header. It contains fields for 'Email' and 'Password', each with an associated input box, and a 'Login' button at the bottom.

Login	
Email	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

Figure 7.wireframe of login

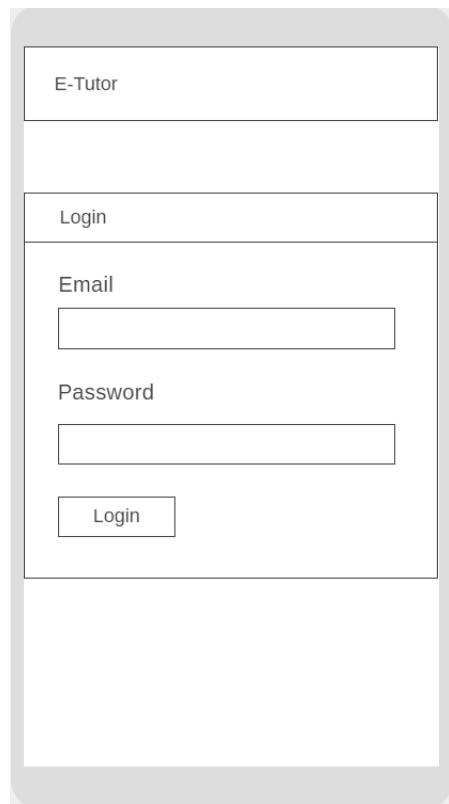


Figure 8.wireframe of login on mobile

5.2.2 Homepage

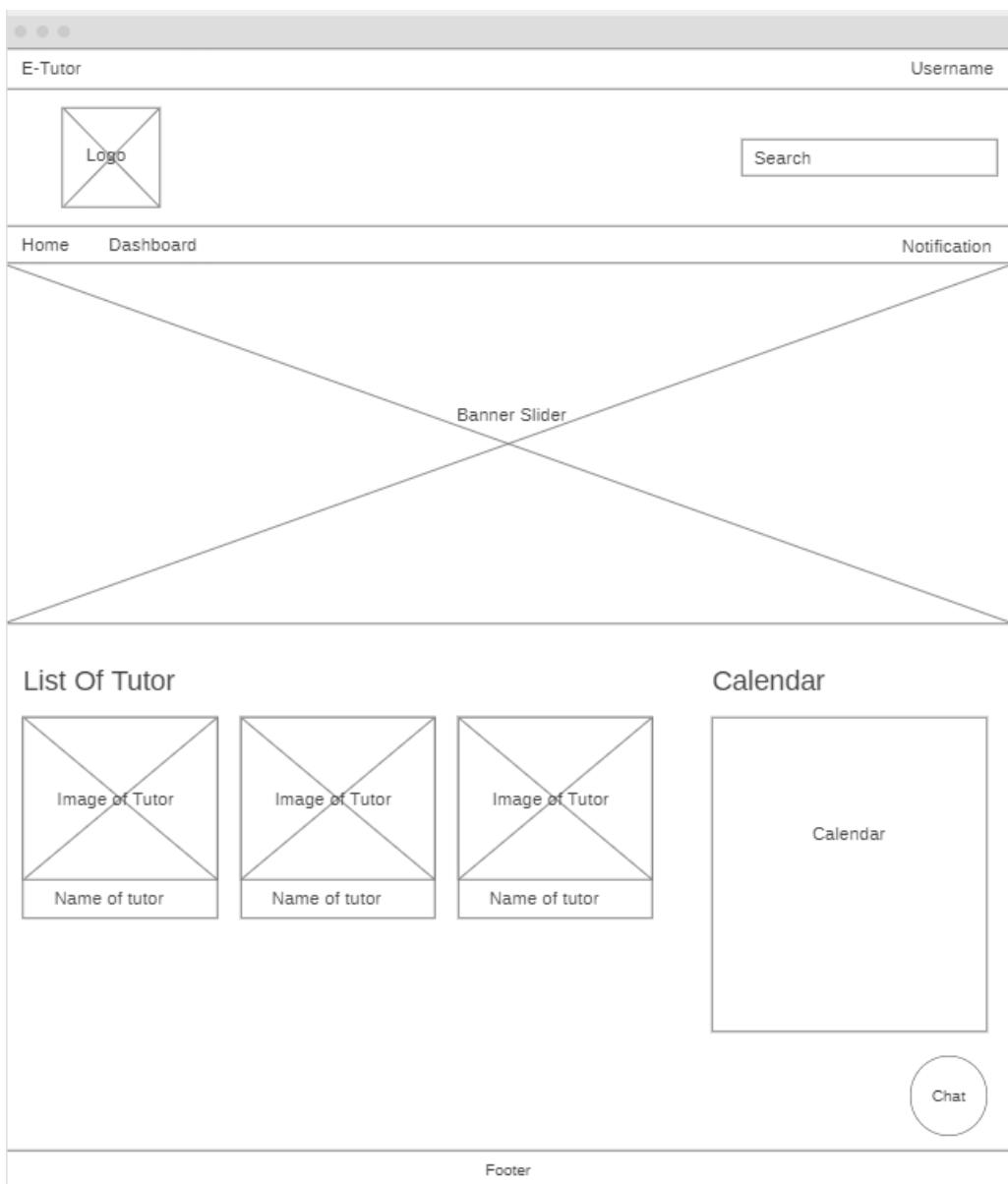


Figure 9.wireframe of homepage

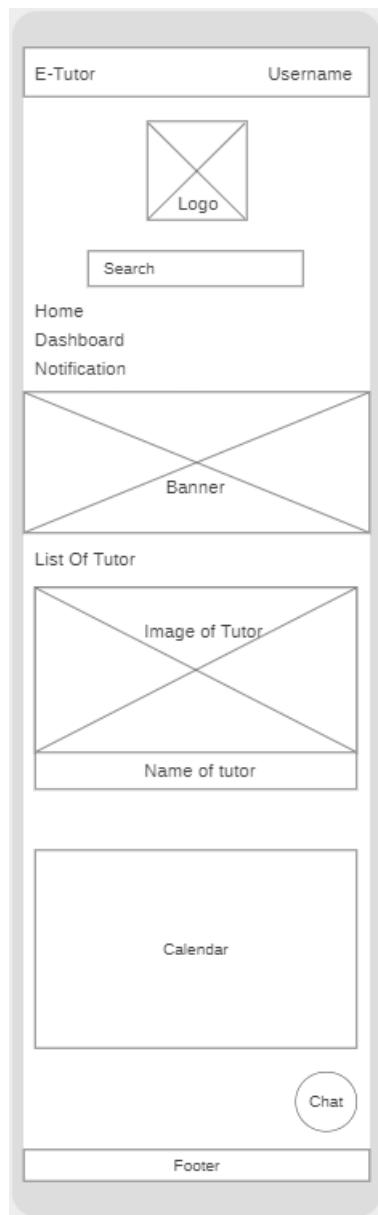


Figure 10.wireframe of homepage on mobile

5.2.3 Student

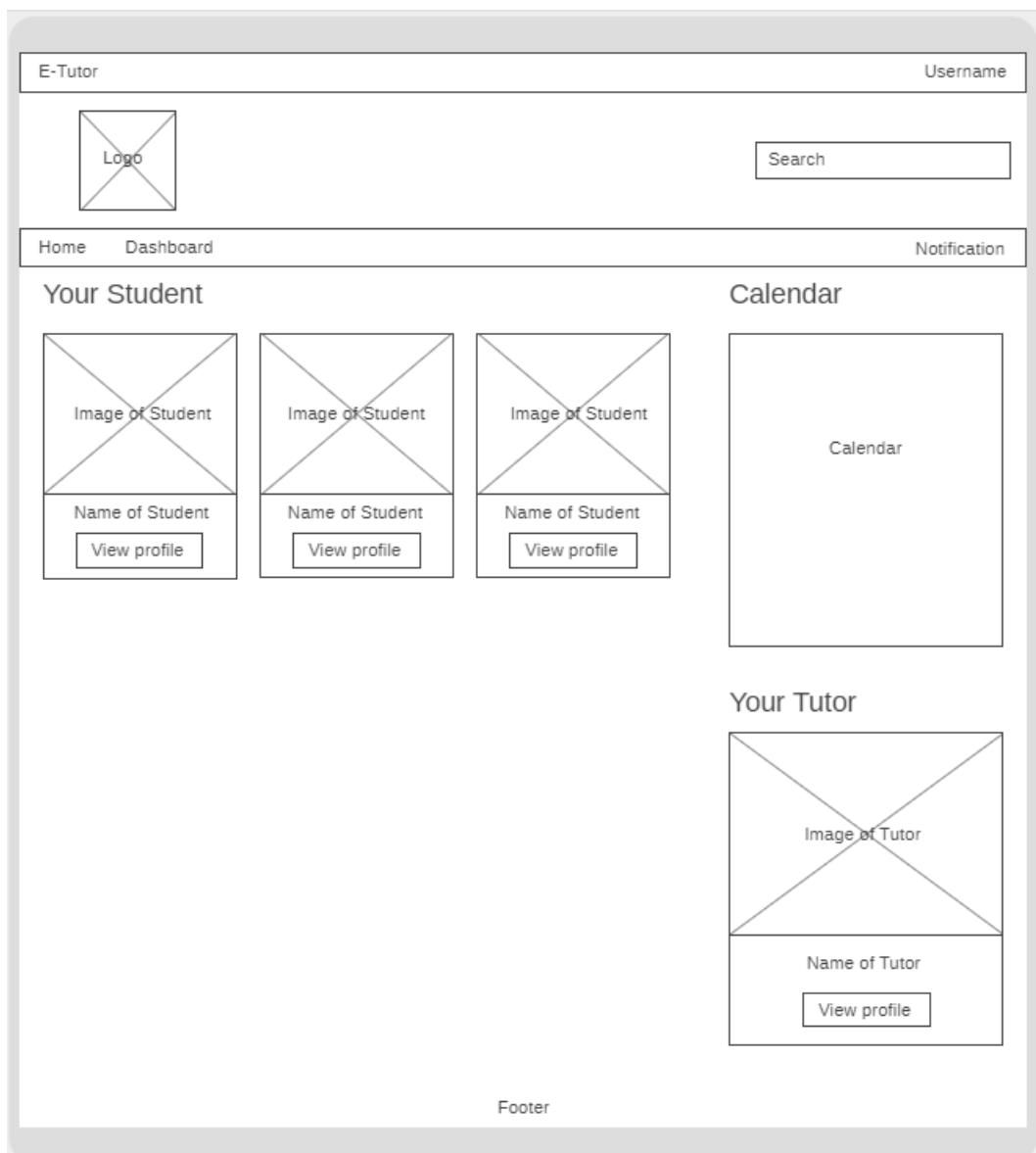


Figure 11.wireframe of student page

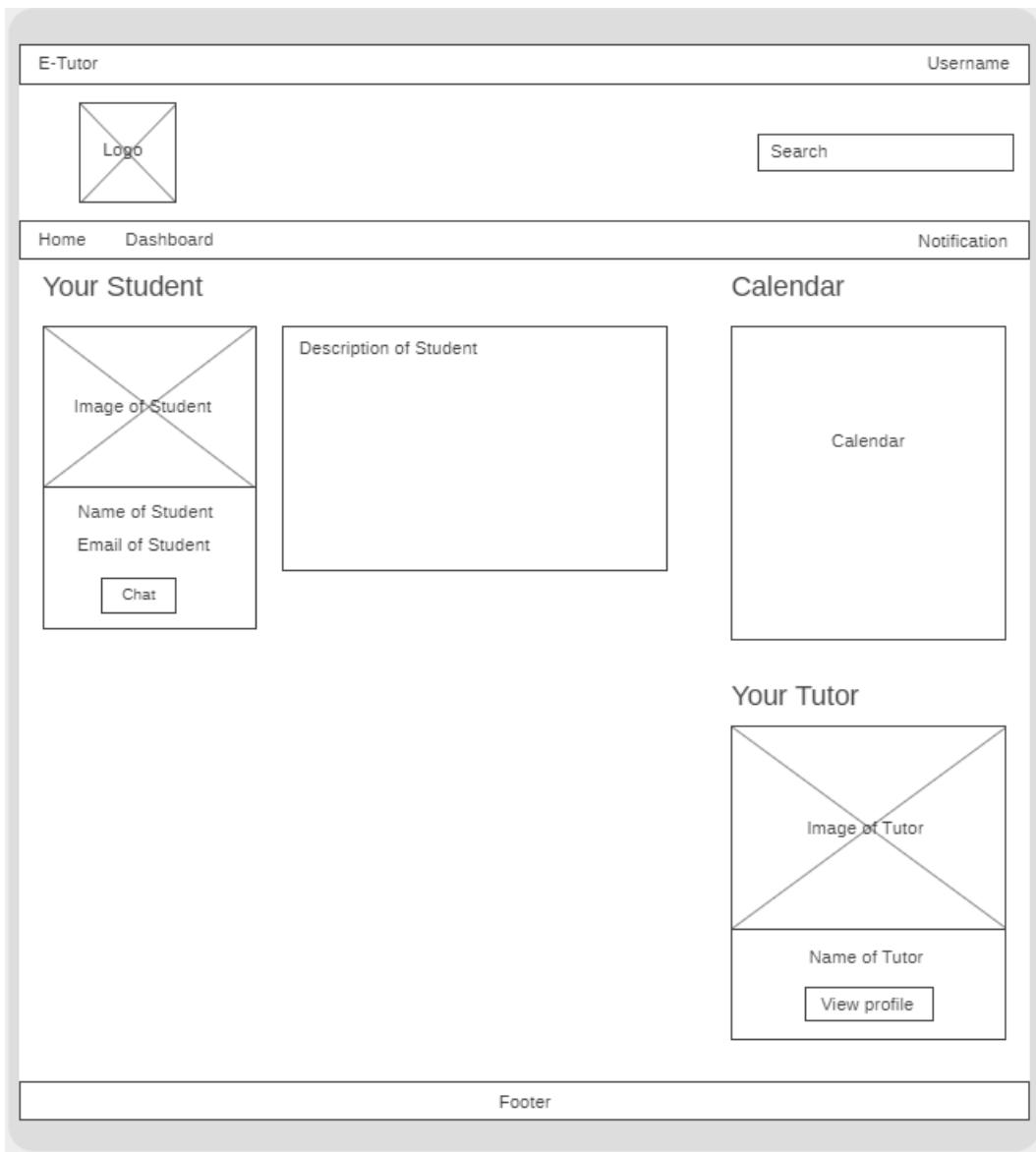


Figure 12.wireframe of student profile

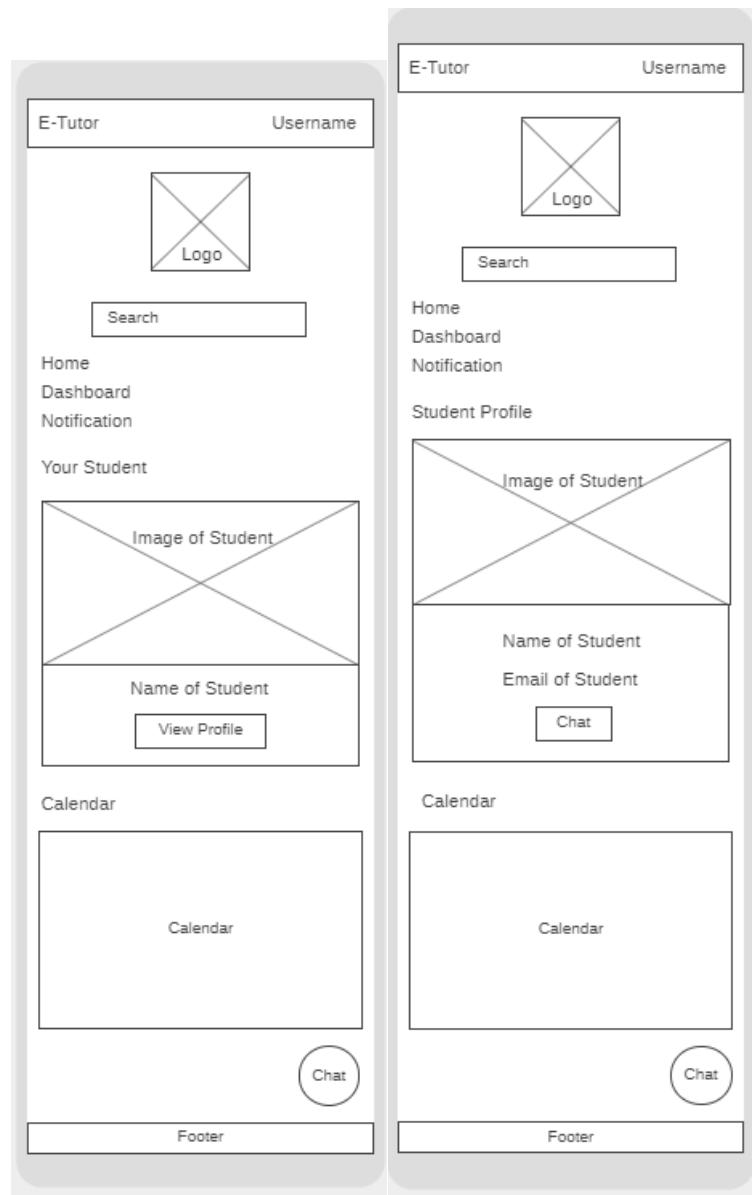


Figure 13.wireframe of student page - student profile page

5.2.4 Admin Dashboard

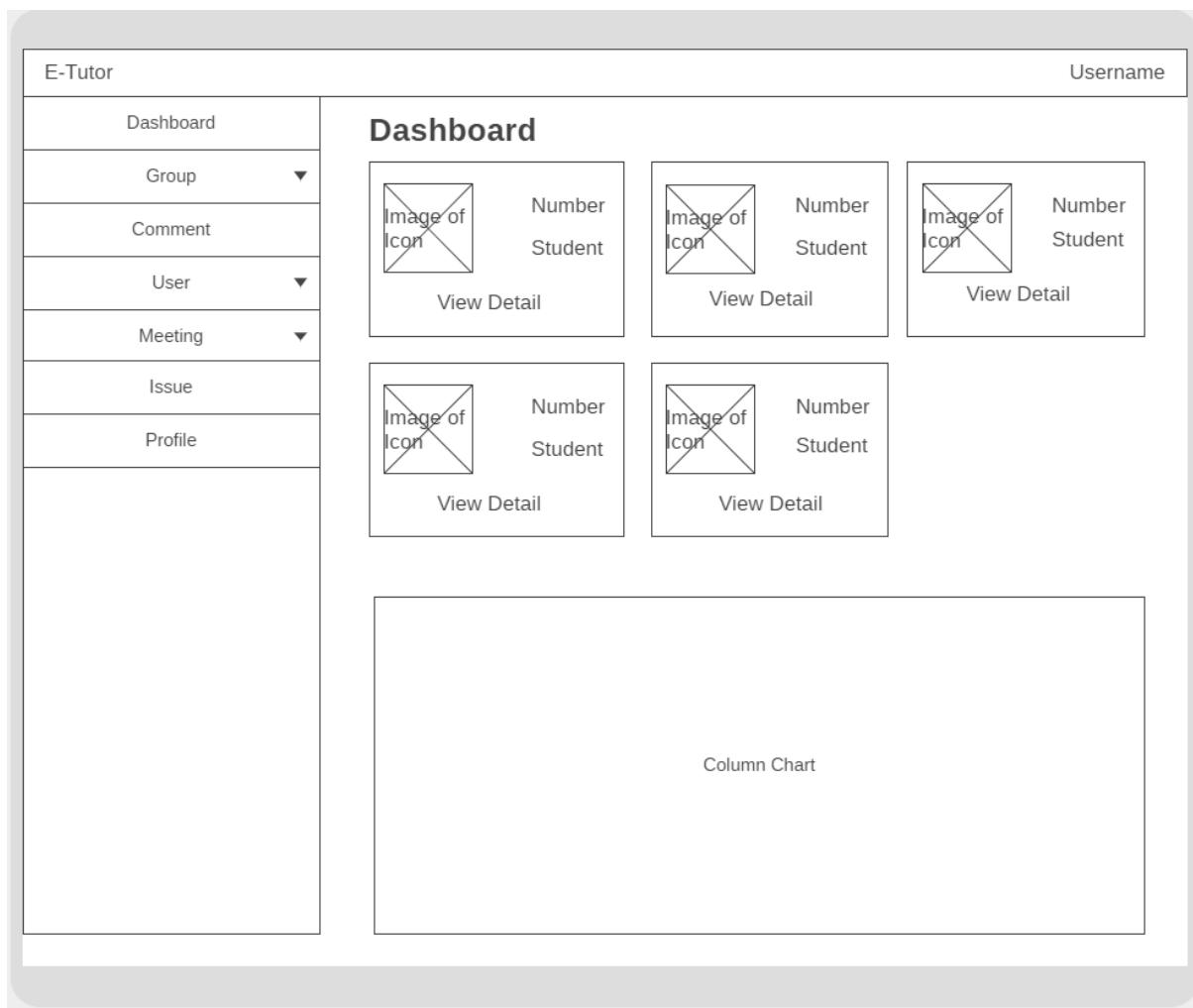


Figure 14. wireframe of admin dashboard

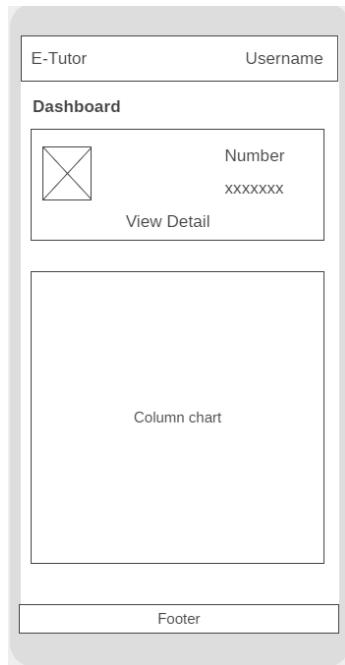


Figure 15. wireframe of admin dashboard on mobile

5.2.5 Issue

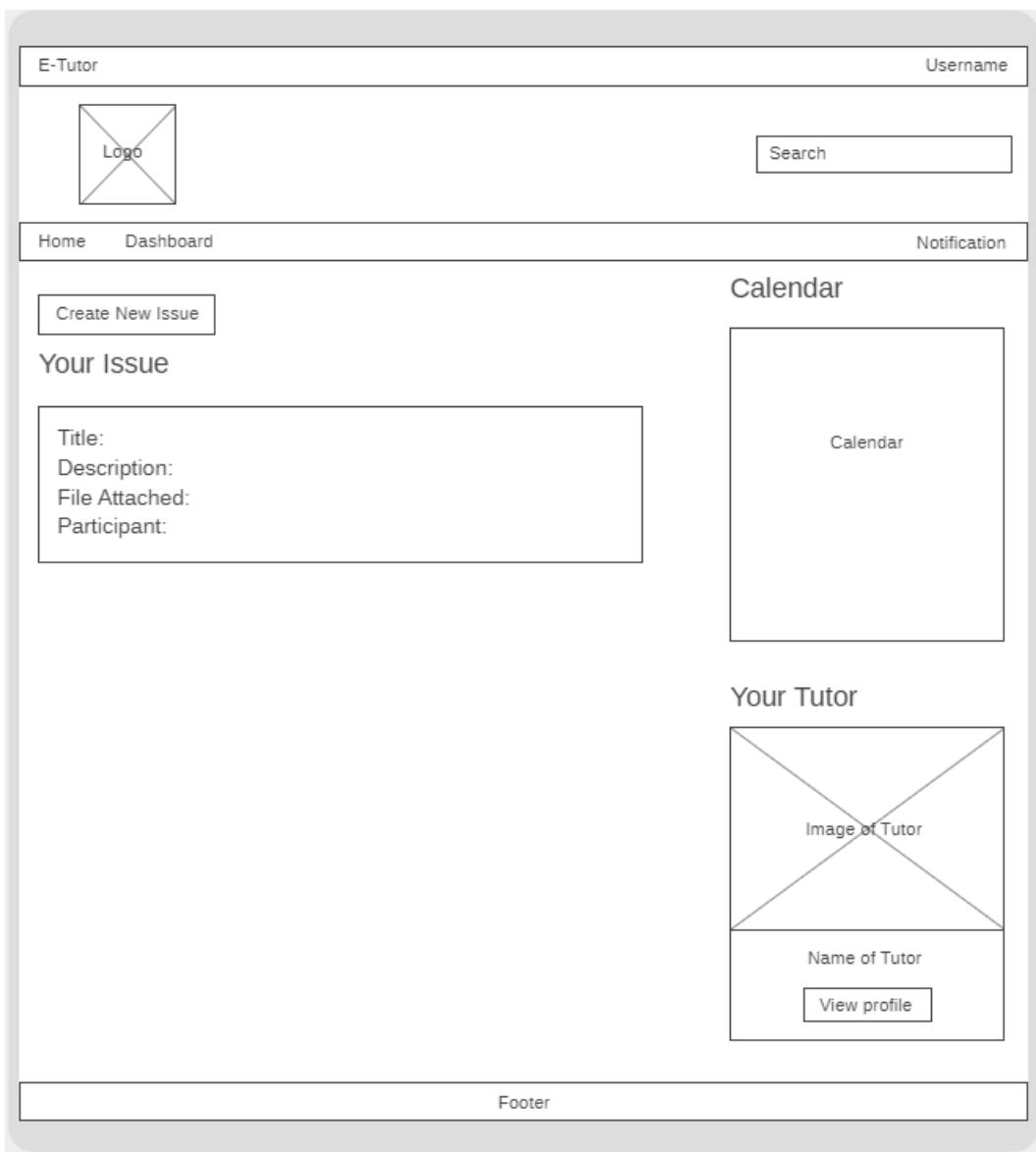


Figure 16. wireframe of issue

E-Tutor

Username

Logo

Search

Home Dashboard Notification

Issue Detail

Title

Author

Date

Description

File Attached

Comment

Image of user

Image of user

Leave Comment

Submit

Calendar

Calendar

Your Tutor

Image of Tutor

Name of Tutor

View profile

Footer

The wireframe illustrates the layout of the 'issue detail' page. At the top, there's a header bar with 'E-Tutor' on the left and 'Username' on the right. Below it is a logo placeholder and a search input field. The main content area has a navigation bar with 'Home', 'Dashboard', and 'Notification'. The 'Issue Detail' section contains fields for 'Title', 'Author', 'Date', and 'Description', each represented by a redacted text area. A 'File Attached' section follows. The 'Comment' section shows two comment entries, each with a user icon and a redacted text area. Below this is a 'Leave Comment' section with a text input field and a 'Submit' button. To the right, there's a 'Calendar' section and a 'Your Tutor' section featuring a placeholder for the tutor's image, the tutor's name, and a 'View profile' button. A footer bar at the bottom spans the width of the page.

Figure 17. wireframe of issue detail

The wireframe illustrates the 'Create New Issue' interface. At the top left is the 'E-Tutor' logo, and at the top right is a 'Username' field. Below the logo is a placeholder box labeled 'Logo'. To the right of the logo is a 'Search' input field. The navigation bar includes 'Home', 'Dashboard', and 'Notification' buttons. The main content area is titled 'Create New Issue' and contains fields for 'Title' (with a placeholder box) and 'Description' (with a larger text area). A 'Select File' button is positioned below the description field. To the right of the main content is a 'Calendar' section, which contains a placeholder box labeled 'Calendar'. Below the main content area are two buttons: 'Select Participant' and 'Create New Issue'. To the right of the 'Create New Issue' button is a 'Your Tutor' section, featuring a placeholder box labeled 'Image of Tutor' with a large 'X' over it. Inside this box is the 'Name of Tutor' and a 'View profile' button. At the bottom of the page is a 'Footer' section.

Figure 18. wireframe of creating issue

The wireframe illustrates the layout of the 'Edit Issue' page. At the top, there is a header bar with the 'E-Tutor' logo on the left and a 'Username' field on the right. Below the header is a search bar. The main content area is titled 'Edit Issue' and contains fields for 'Title' (with a file input field) and 'Description' (with a file input field). To the right of these fields is a 'Calendar' section. Below the main content area are two buttons: 'Select File' and 'Edit Issue'. To the right of the 'Edit Issue' button is a 'Your Tutor' section, which includes a placeholder for 'Image of Tutor', the 'Name of Tutor', and a 'View profile' button. At the bottom of the page is a 'Footer' section.

E-Tutor	Username
	Search
Home Dashboard	Notification
Edit Issue	
Title	
Description	
Select File	
Edit Issue	
Calendar	
Your Tutor	
Name of Tutor	
View profile	
Footer	

Figure 19.wireframe of editing issue



The wireframe illustrates the layout of an 'issue detail - edit issue' page, divided into two main sections: a left panel for viewing issue details and a right panel for editing the issue.

Left Panel (Issue Detail):

- E-Tutor:** [Text input field]
- Username:** [Text input field]
- Issue Detail:**
 - Title:** [Text input field]
 - Author:** [Text input field]
 - Date:** [Text input field]
 - Description:** [Text input field]
- File Attached:** [Text input field]
- Comment:**
 - Image of user:** [Image placeholder]
 - Leave Comment:** [Text input field]
- Submit:** [Text input field]
- Footer:** [Text input field]

Right Panel (Edit Issue):

- E-Tutor:** [Text input field]
- Username:** [Text input field]
- Edit Issue:** [Text input field]
- Title:** [Text input field]
- Description:** [Text input field]
- Select File:** [Text input field]
- Select Participant:** [Text input field]
- Edit Issue:** [Text input field]
- Footer:** [Text input field]

Figure 20. wireframe of issue detail - edit issue

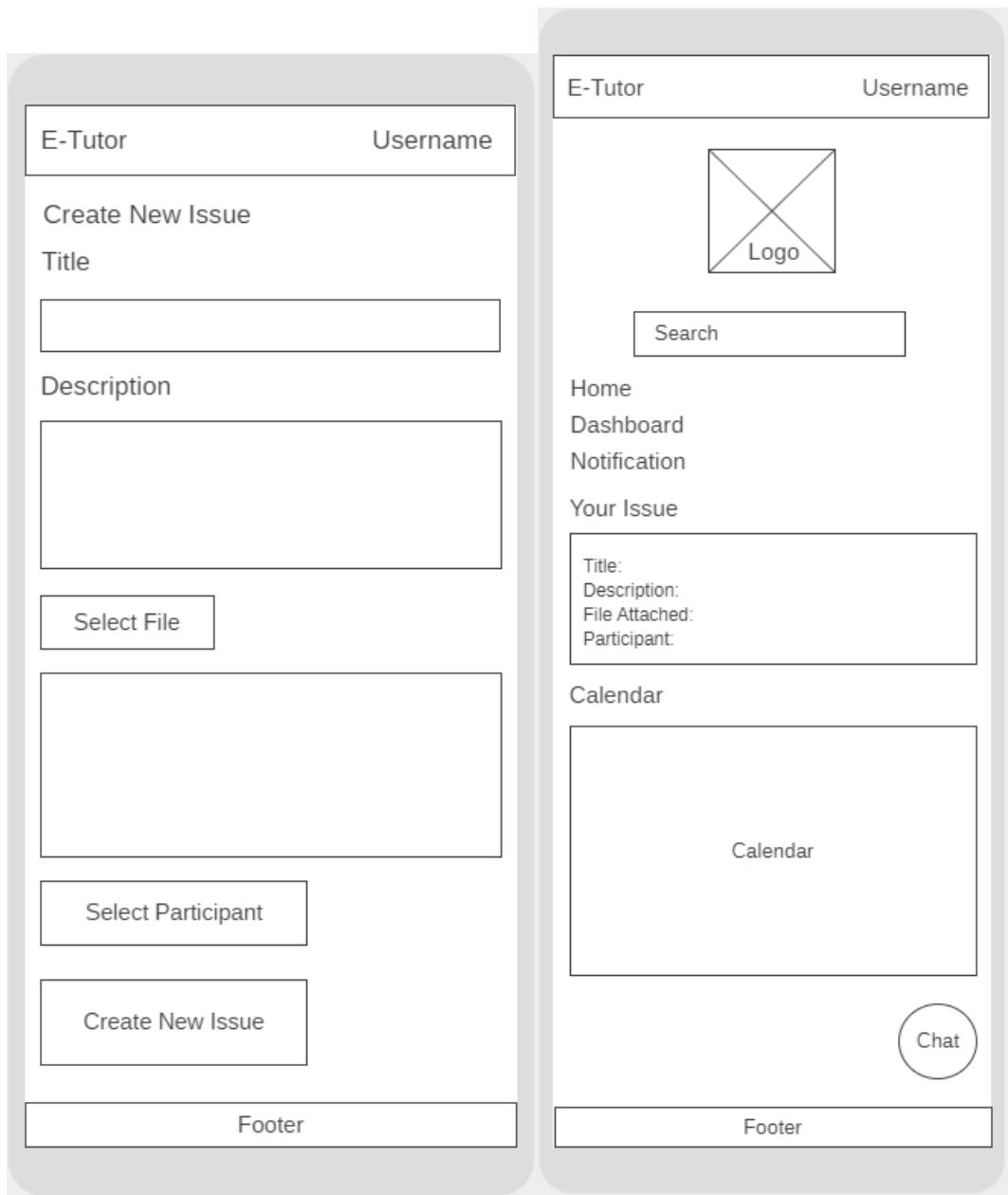


Figure 21. wireframe of creating issue - issue page

5.2.6 Blog

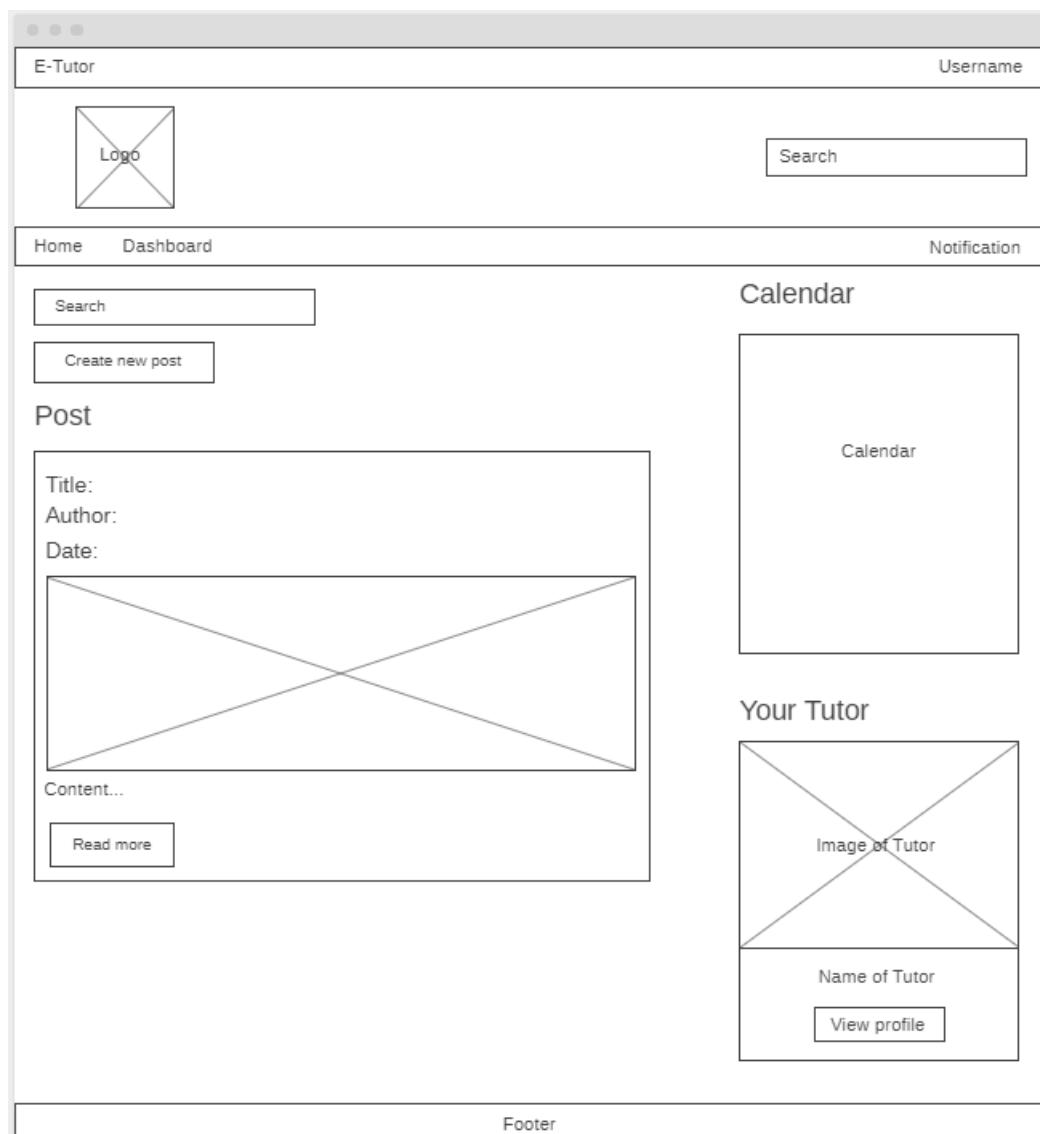


Figure 22. wireframe of blog page

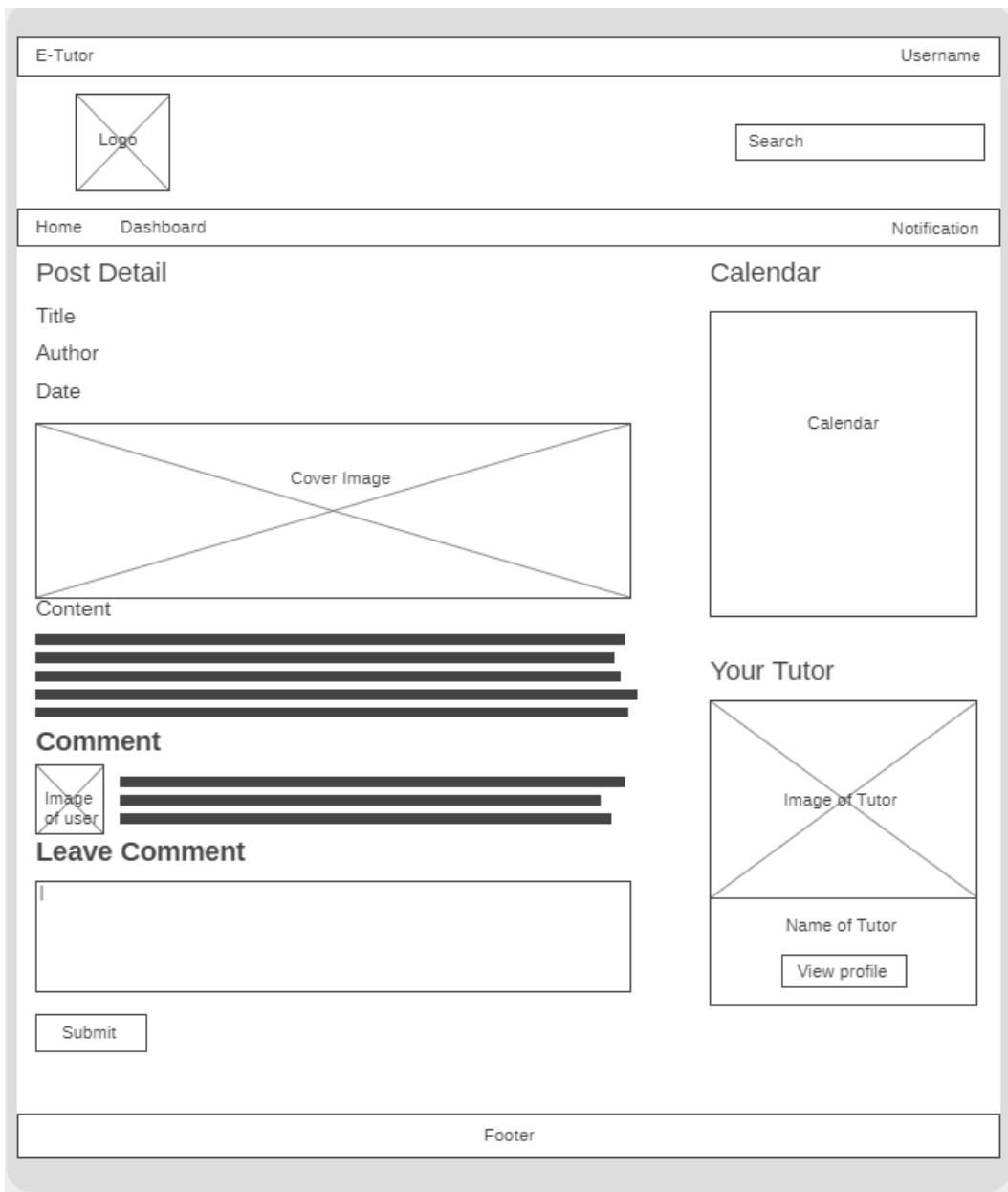


Figure 23.wireframe of post detail

The wireframe illustrates the layout of the 'Create New Post' page. At the top, there is a header bar with the 'E-Tutor' logo, a 'Username' field, and a search bar. Below the header, a navigation bar includes 'Home' and 'Dashboard' links and a 'Notification' section. The main content area is titled 'Create New Post' and contains fields for 'Title' (with a text input box), 'Category' (with a text input box), and 'Content' (with a large text area). To the right of the content area is a 'Calendar' section, which is currently empty. Below the content area is a 'Select Cover Image' button and a placeholder box for the cover image. At the bottom left is a 'Create New Post' button. On the right side, there is a 'Your Tutor' section featuring a placeholder box for the tutor's image, labeled 'Image of Tutor'. Below the image placeholder is a 'Name of Tutor' field and a 'View profile' button. A footer section at the bottom contains a single text input box.

Figure 24.wireframe of creating post page

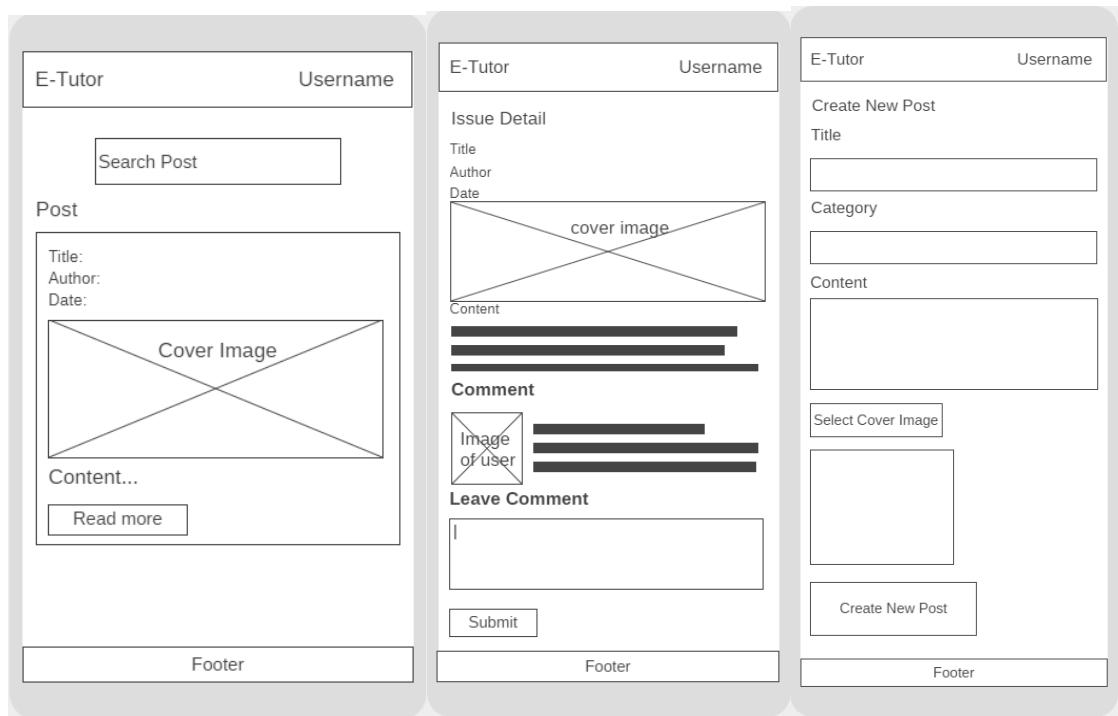


Figure 25. wireframe of blog - post detail - creating post on mobile

5.2.7 Allocate student to tutor

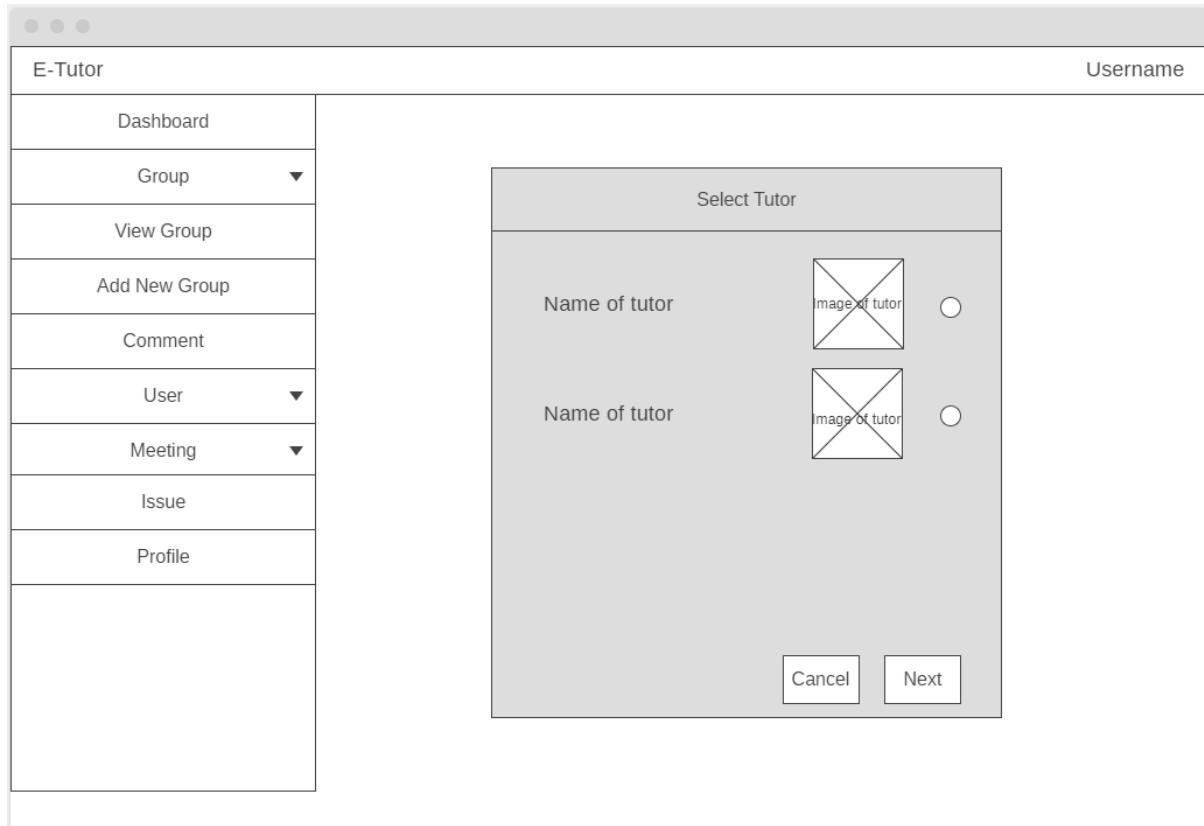


Figure 26. wireframe of step 1 of allocating student to tutor

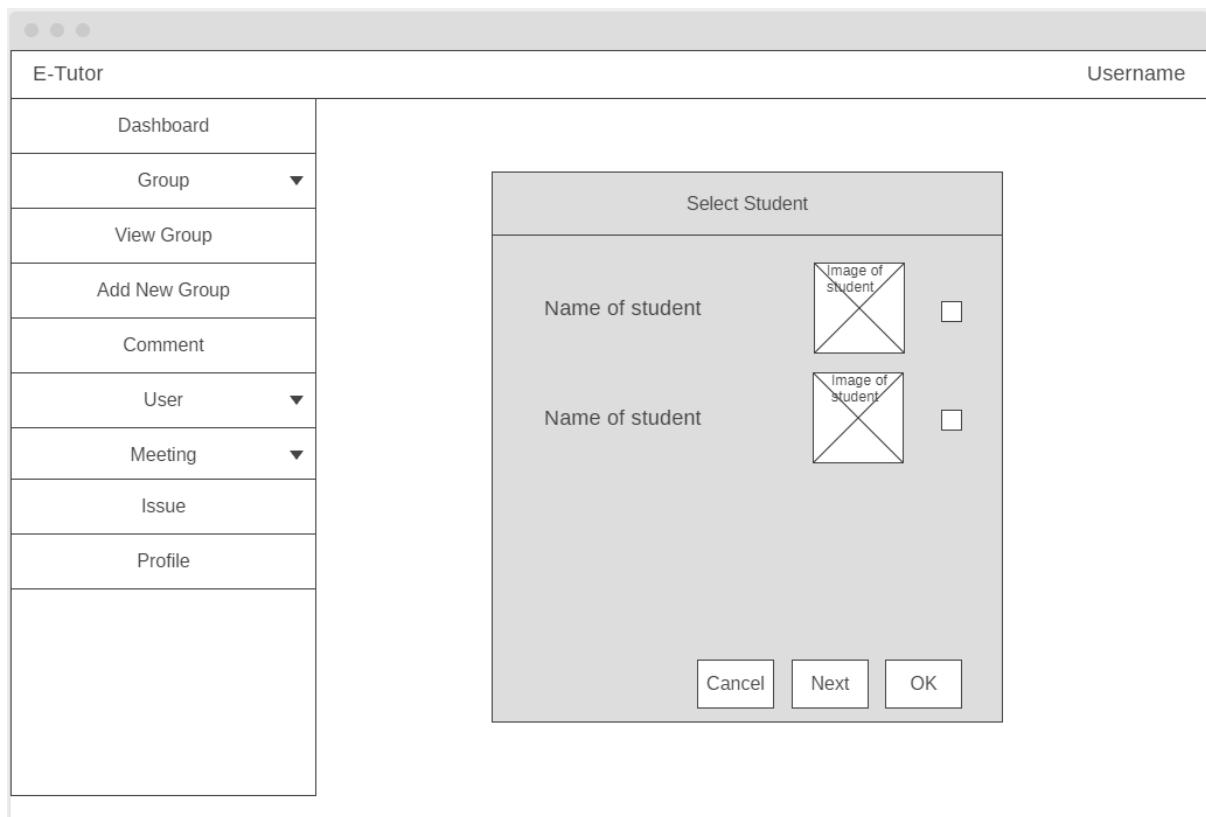


Figure 27. wireframe of step 2 of allocating student to tutor

5.3 Interface Design

Here are all images of the front end design of site

5.3.1 Login – Forgot password – Change password

User will be provided email and password to login to the website, depending on the role of the user in the site from which the user has been provided, the web page will display the content that corresponds to the user's role.

The screenshot shows the E-Tutor login interface. At the top, there's a teal header bar with the text "E-Tutor" on the left and "Login" on the right. Below this is a white login form with a teal header labeled "Login". The form contains fields for "E-Mail Address" (with the value "draknoel@gmail.com") and "Password" (with the value "*****"). There's also a "Remember Me" checkbox, which is unchecked. At the bottom of the form are three buttons: "Login" (in a teal box), "OR" (in a grey box), and "Login with Google" (in a red box with a "g" icon). Below these buttons is a link "Forgot Your Password?".

Figure 28. Image of Login page

When the user does not remember their password, they select forgot password, the site will move to a new page that is Forgot Password. The user enters the email previously registered, then the website will send an email with a link to a new password change page.

The screenshot shows the E-Tutor forgot password interface. At the top, there's a teal header bar with the text "E-Tutor". Below it is a white form with a teal header labeled "Forgot Password". The form has a single field for "E-Mail Address" with a placeholder "Email" and a "Submit" button below it. To the right of the "Submit" button is a "Back To Login" link.

Figure 29. Image of Forgot password page

In the change password page, the user enters a new password then they have to confirm password entered to change.

Figure 30.Image of Change password page

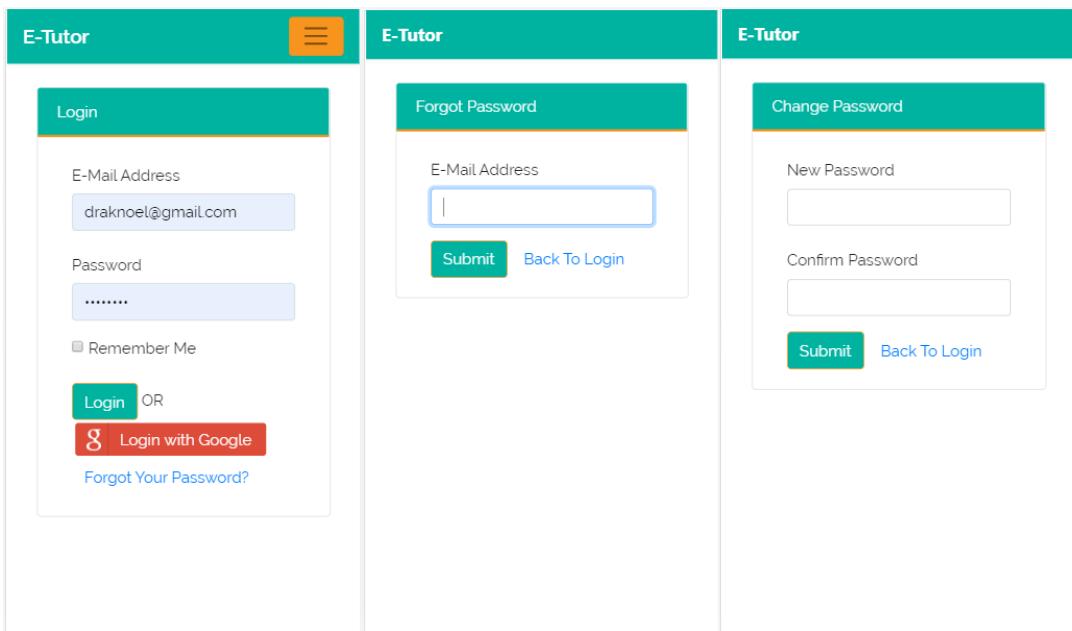


Figure 31.Image of Login - Forgot Password - Change Password mobile

5.3.2 Homepage

Student can see all their tutor, they can chat with a tutor, creating events by the calendar, they also can receive notification from the system.

The screenshot displays the homepage of the E-Tutor application. At the top, there is a navigation bar with the title "E-Tutor" on the left and a "Log Out" button on the right. Below the navigation bar is a search bar with the placeholder "Search..." and a magnifying glass icon. On the far left, there is a circular profile picture of a person wearing glasses and a green shirt. To the right of the profile picture is a "Notifications" icon with a bell symbol.

The main content area features a large, blurred photograph of a classroom where students are working on laptops. Below the photograph, there are two sections: "List Of Tutors" and "Calendar".

List Of Tutors: This section contains six tutor profiles, each with a small portrait photo and the tutor's name below it. The profiles are arranged in two rows of three.

- Leroy Riley
- Lucille Graves
- Eileen Kuhn

- Lucy Garcia
- Brian Tucker
- Lydia Allen

Calendar: This section shows a monthly calendar for April 2020. The days of the week are labeled from Wednesday to Tuesday. The date 26 is highlighted with a red circle, indicating it is the current day. A green speech bubble icon with a plus sign is located in the bottom right corner of the calendar area.

Figure 32.Image of homepage

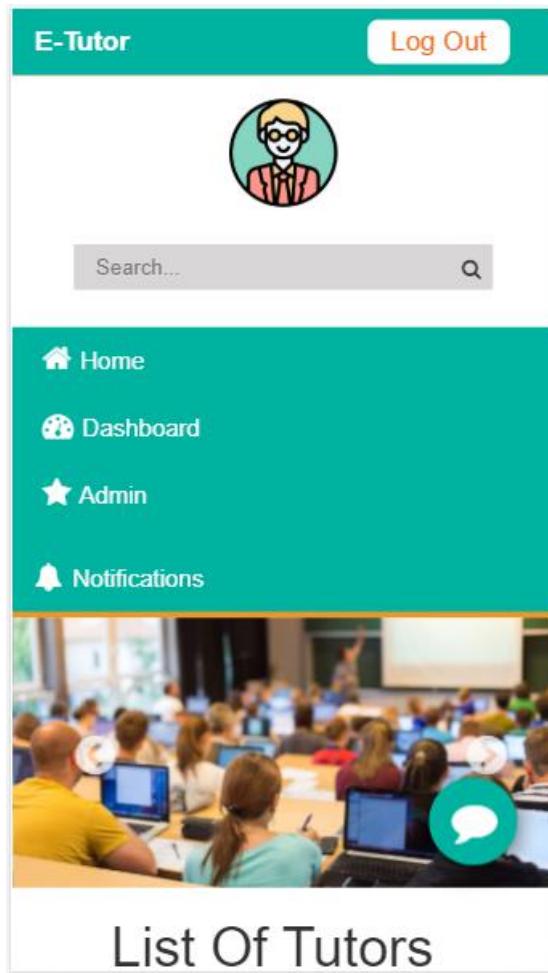


Figure 33. Image of Homepage mobile

5.3.3 Dashboard

This site will display an activity summary of student, issue, and meeting, besides that is recent activities of user

The dashboard has a teal header with 'Home', 'Dashboard', and 'Notifications'. The main area includes an 'Activity Summary' section with three cards: 'Meeting' (4), 'Issue' (12), and 'Student' (12). Below is a 'Recent Activities' section showing a comment on an issue. To the right is a 'Calendar' for April 2020, with the 27th highlighted. A 'Your Tutor' section with a message icon is at the bottom right.

Figure 34. Image of dashboard for user

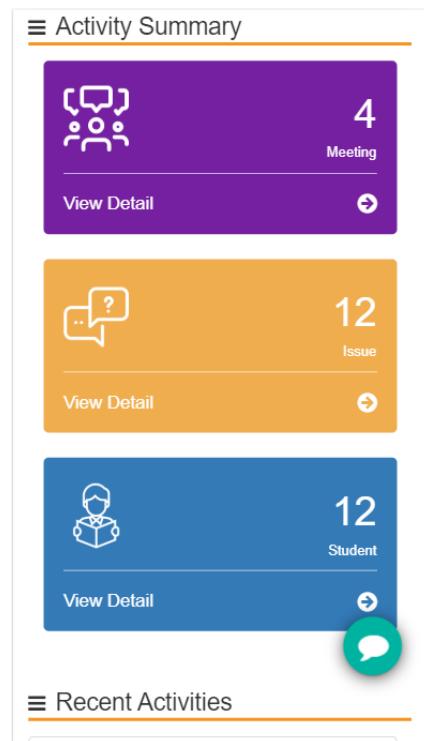


Figure 35. Image of dashboard for user mobile

5.3.4 Student – Student Profile

This page will display all student of a tutor, when the tutor clicks on "view profile" button, the website will move to a new page which is "Student Profile".

The figure shows a student page interface. At the top, there is a circular profile picture of a student and a search bar. Below the search bar, a navigation bar includes links for "Home" and "Dashboard", and a "Notifications" section with a bell icon.

The main content area is titled "Your Student" and displays three student profiles, each featuring a photo of a boy with glasses and a black hoodie. The profiles are labeled "Harry Potter" and include a "View Profile" button. To the right of the student list is a "Calendar" section for April 2020, showing dates from 1 to 30. A green speech bubble icon is located at the bottom right of the calendar area.

Figure 36. Image of student page

In the student profile page, the information of students will display, for example, name, email, description of student and role of student in website, the tutor can chat with the student when clicks on the "Chat" button.

The screenshot shows the E-Tutor application interface. At the top, there's a green header bar with the title 'E-Tutor' on the left and a user profile picture of Severus Snape on the right. Below the header is a search bar with a magnifying glass icon. The main content area has a teal header with 'Home' and 'Dashboard' buttons on the left and a 'Notifications' button on the right. A sidebar on the left is titled 'Your Student' and features a large photo of Harry Potter, his name, email (harrypotter@gmail.com), and a 'Chat' button. To the right of the sidebar is a 'Calendar' section for April 2020, showing dates from 1 to 30, with April 27th highlighted. Below the calendar is a section titled 'Your Tutor' featuring a photo of Donald Trump.

Figure 37. Image of Student profile page

The screenshot shows a student profile page. At the top left is a navigation bar with 'HOME' (highlighted), 'Dashboard', and 'Notifications'. Below it, a section titled 'Your Student' displays a portrait of Harry Potter. The profile card includes his name 'Harry Potter' and a 'View Profile' button. To the right is a large image of Harry Potter in a black hoodie. Below the image is his name 'Harry Potter' and email 'harrypotter@gmail.com', followed by a 'Chat' button.

About me
It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.
Role: Role of User

Calendar

Figure 38. Image of Student - Student profile page

5.3.5 Issue - Issue Detail – Create New Issue

The issue page will show your issue. One issue includes the title of the issue, issue description, file attached, participant. Users can view detailed content of the issue in the Detail Issue page.

The screenshot shows an issue creation page. At the top left is a navigation bar with 'Home' (highlighted), 'Dashboard', and 'Notifications'. Below it, a section titled 'Your Issue' contains a 'Create New Issue' button. A form for creating a new issue is shown, with fields for 'Title: Issue Title', 'Description: Issue Description', 'Files Attached: Image.png, issue.docx, issue.pdf', and 'Participants: Tutor 1, Tutor 2'. To the right is a calendar for April 2020, with the 27th highlighted. A 'Chat' button is located at the bottom right of the calendar.

Figure 39.Issue page

Detail Issue is the page to the user discuss the issue, user can view the comment and leave their comment.

The screenshot shows the E-Tutor application interface. At the top, there is a navigation bar with the title "E-Tutor" and a user profile "Severus Snape". Below the navigation bar is a search bar with placeholder text "Search..." and a magnifying glass icon. On the left side, there is a sidebar with icons for "Home" and "Dashboard". The main content area is titled "Issue Detail". It displays a post by "Donald Trump" from "24/04/2020" at "20:00". The post content is: "It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English". Below the post are two attachments: a small thumbnail of Donald Trump and a PDF icon. To the right of the post is a "Calendar" section showing the month of April 2020. The calendar highlights the 27th of April. Below the calendar is a "Your Tutor" section featuring a large portrait of Donald Trump, his name, and a "View profile" button. At the bottom of the main content area is a "Leave Comment" section with a text input field labeled "Comment..." and a speech bubble icon.

Figure 40.Issue detail page

To create an issue, the user accesses the "Create Issue" page from the "Create New Issue" button on the Issue page. In creating issue form, user need enter issue title, description of the issue, upload file attachments, and selecting participants.

E-Tutor

Severus Snape ▾



Search... Q

Home Dashboard Notifications

Create New Issue

Title

Description

Select File

Chọn file Không có tệp nào được chọn

Select Participants

CREATE NEW ISSUE

Calendar

April 2020 27

Wed	Thu	Fri	Sat	Sun	Mon	Tue
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Your Tutor



Donald Trump

View profile

Figure 41.Create Issue page

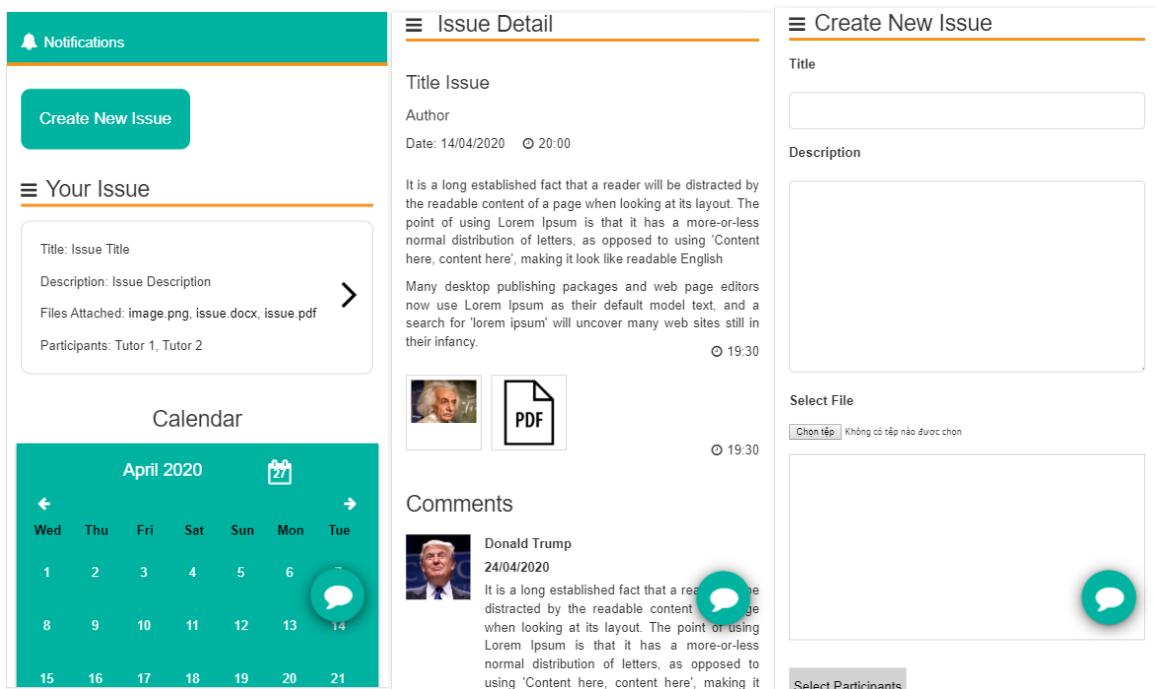


Figure 42. Image of Issue - Issue Detail - Create Issue page mobile

5.3.6 Blog – Post Detail – Create Blog

The blog page will show all posts. One post includes the title of the post, content of post, image cover of post, category. Users can view detailed content of the post in the Blog Detail page.

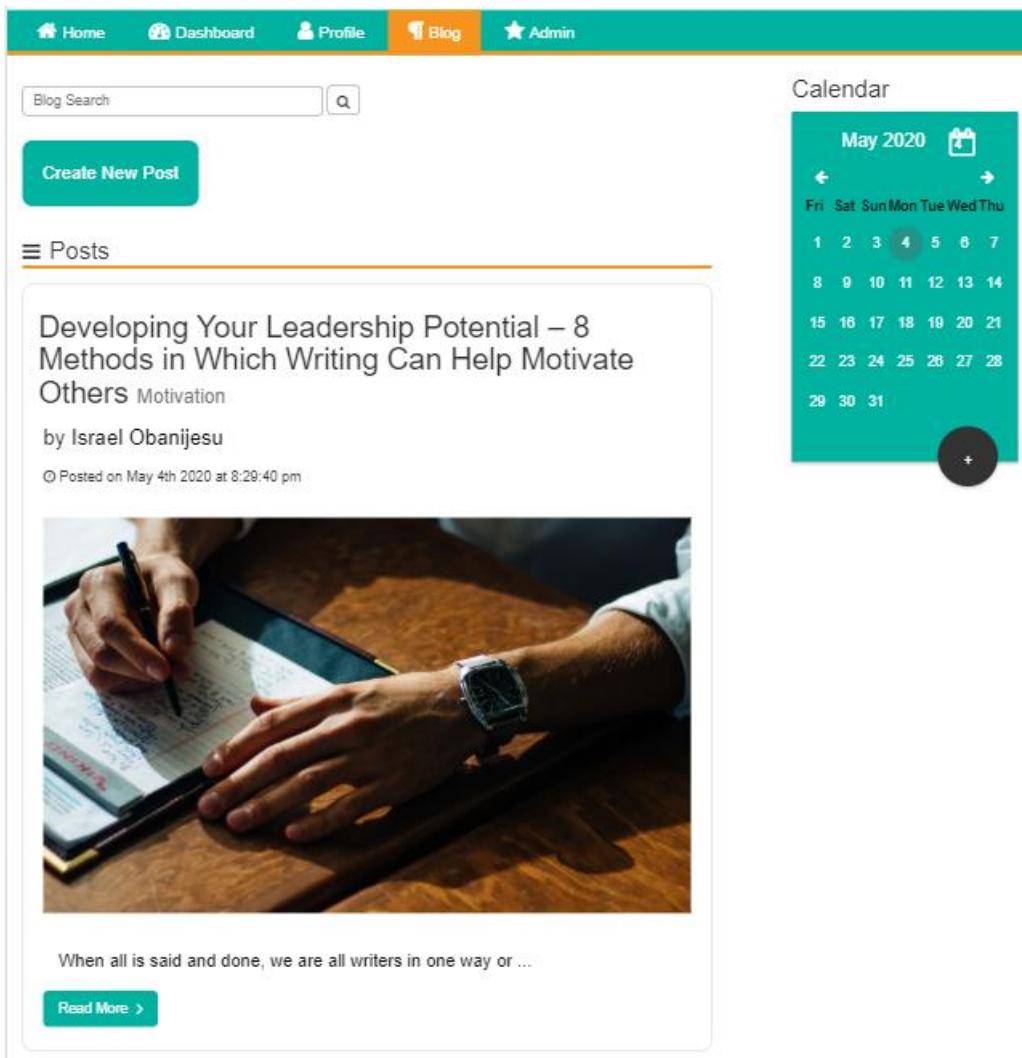


Figure 43. Image of blog page.

Post Detail is the page to the user discuss the post, user can view the comment and leave their comment.

Home
Dashboard
Profile
Blog
Admin

≡ Developing Your Leadership Potential – 8 Methods in Which Writing Can Help Motivate Others

Motivation

by Israel Obanijesu

Posted on May 4th 2020 at 8:29:40 pm



+

+
New Post

<div style="position: absolute; top: 10px; right: 10px; background-color: #009640; color: white; padding: 5px; border-radius: 5px; width: 1

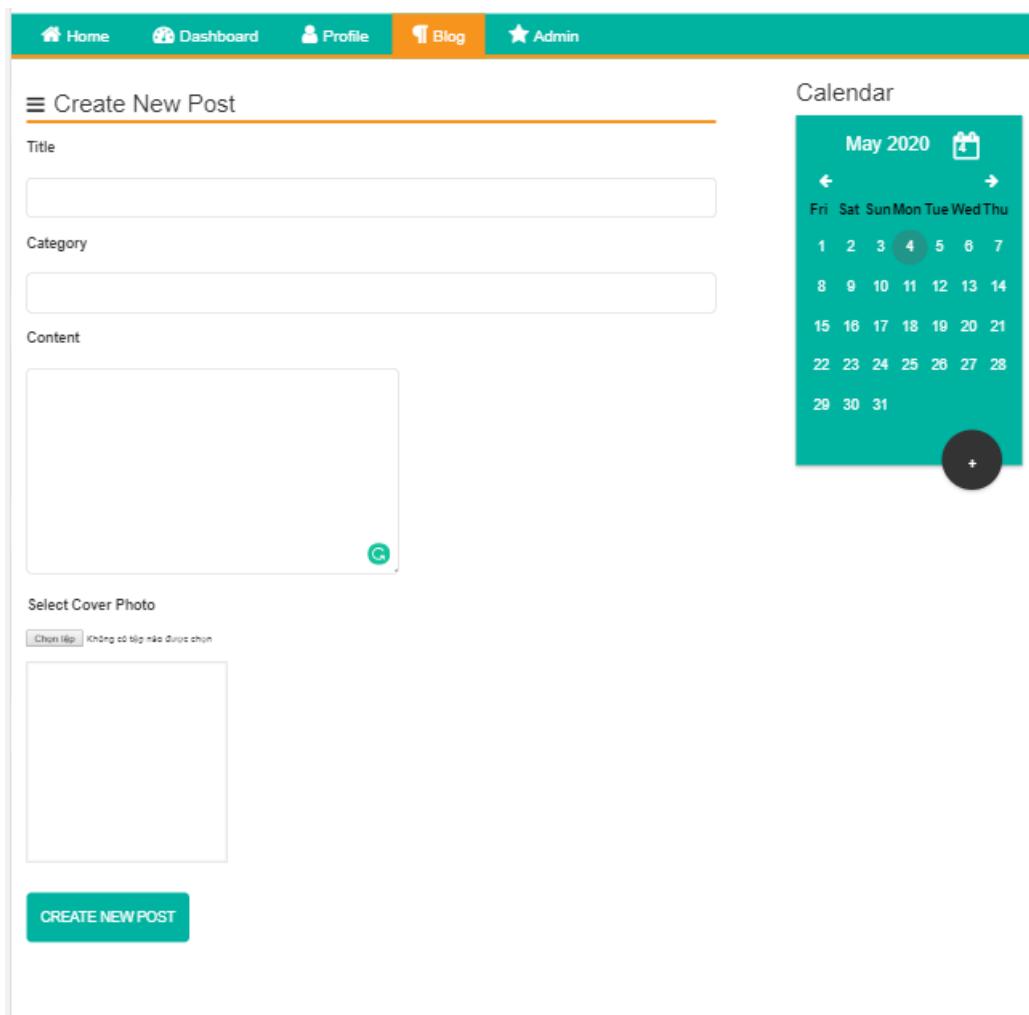


Figure 45. Image of creating post page

Figure 46.image of blog page - post detail - creating post page on mobile

5.3.7 Meeting

This page will display the content of the meeting such as the title of the meeting, time, name of the organizer, meeting description, member of the meeting. The meeting page has 3 sections Past meeting, Ongoing meeting, and Future meeting.

The figure shows a user interface for a meeting application. On the left, there are two sections: "Past Meetings" and "Ongoing Meetings", each containing a list of meeting details. In the center is a "Calendar" for April 2020, with the 27th highlighted. On the right is a "Your Tutor" section featuring a photo of Donald Trump and a speech bubble icon.

Past Meetings

- Title: Meeting Title
- Date: 22/04/2020
- Organizer: Israel
- Description: Meeting Description
- Member: Israel, Cuong, Son

Ongoing Meetings

- Title: Meeting Title
- Date: 22/04/2020
- Organizer: Israel
- Description: Meeting Description
- ...
...

Calendar

Wed	Thu	Fri	Sat	Sun	Mon	Tue
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Your Tutor

A small circular icon with a white speech bubble inside is positioned next to the photo of Donald Trump.

Figure 47.Meeting page for user

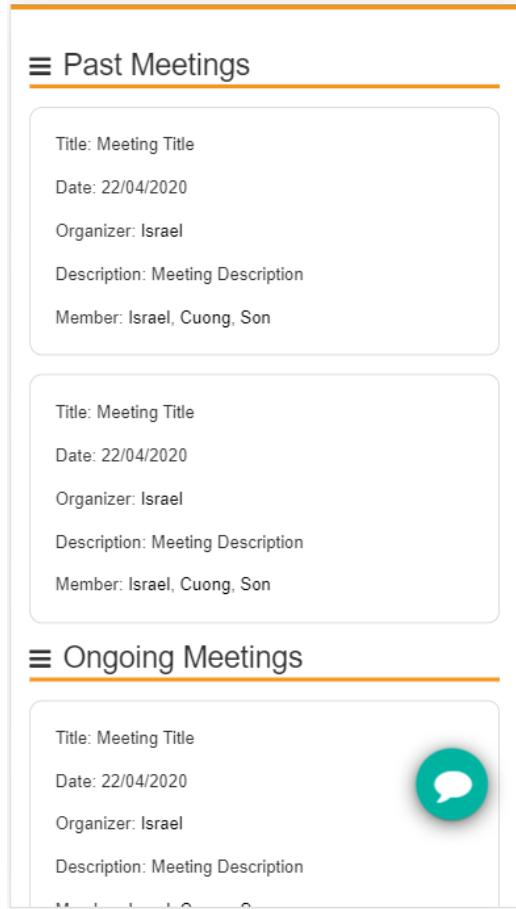


Figure 48.Meeting page for user on mobile

5.3.8 Admin Dashboard

The site allows users to manage user activities within the site, it will display stats of components in the site as student, meeting, issue, tutor, and group. This information is displayed by the column chart.



Figure 49. Image of admin dashboard

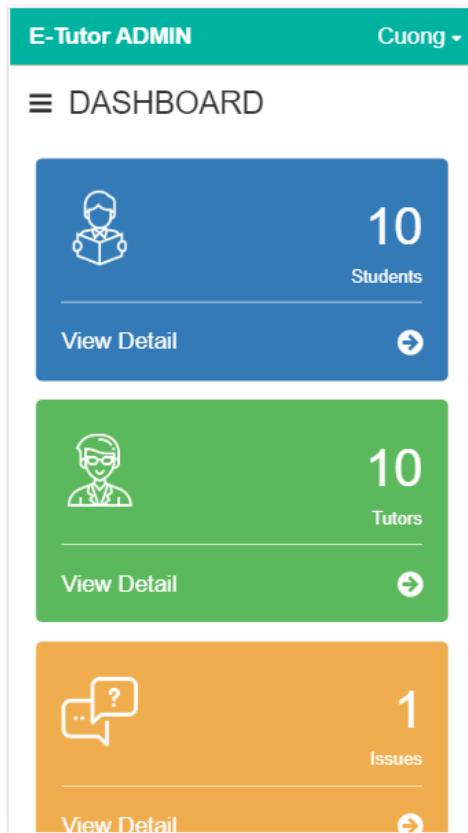


Figure 50. Image of admin dashboard on mobile

5.3.9 All Group – Add New Group

This site will show all group

The 'LIST OF GROUP' page shows two groups:

- Group 1:** 3 members (View Members)
- Group 2:** 5 members (View Members)

Figure 51. Image of list of group

To create a new group, user select "Add new group" from the side menu, one modal is displayed, user enters the group name and image of the group and then selecting member of the group.

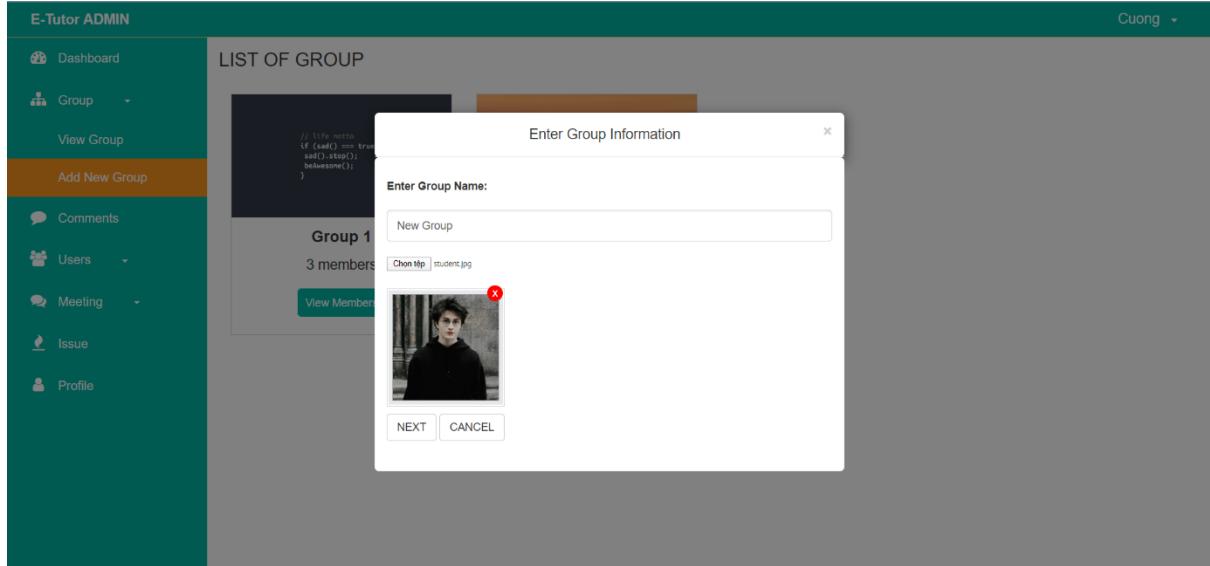


Figure 52.Step 1 of adding new group

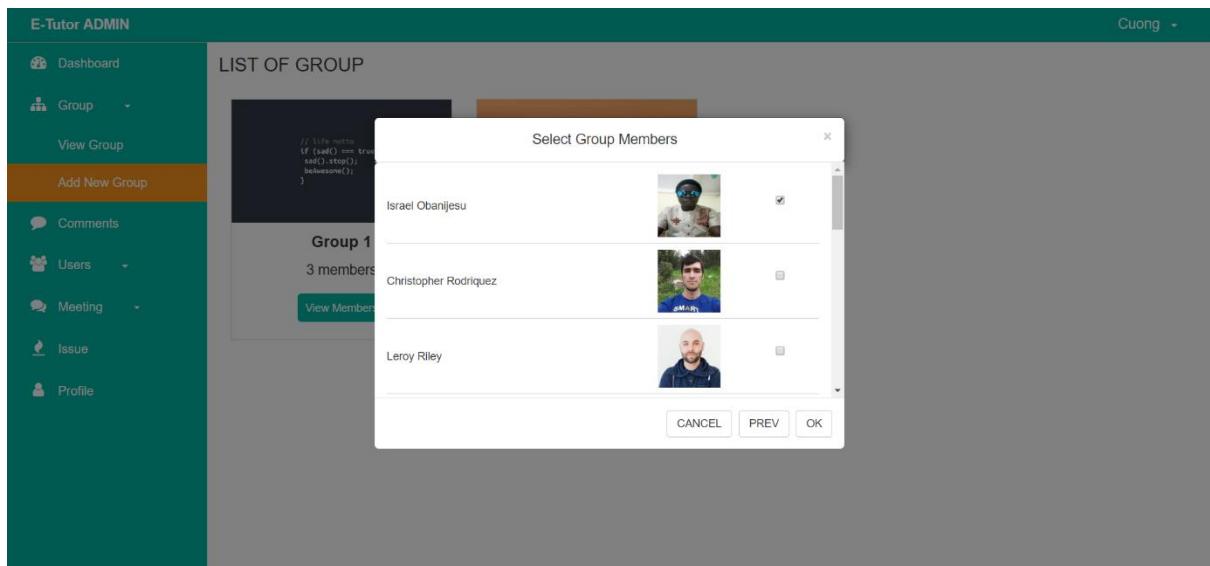


Figure 53.Step 2 of adding new group

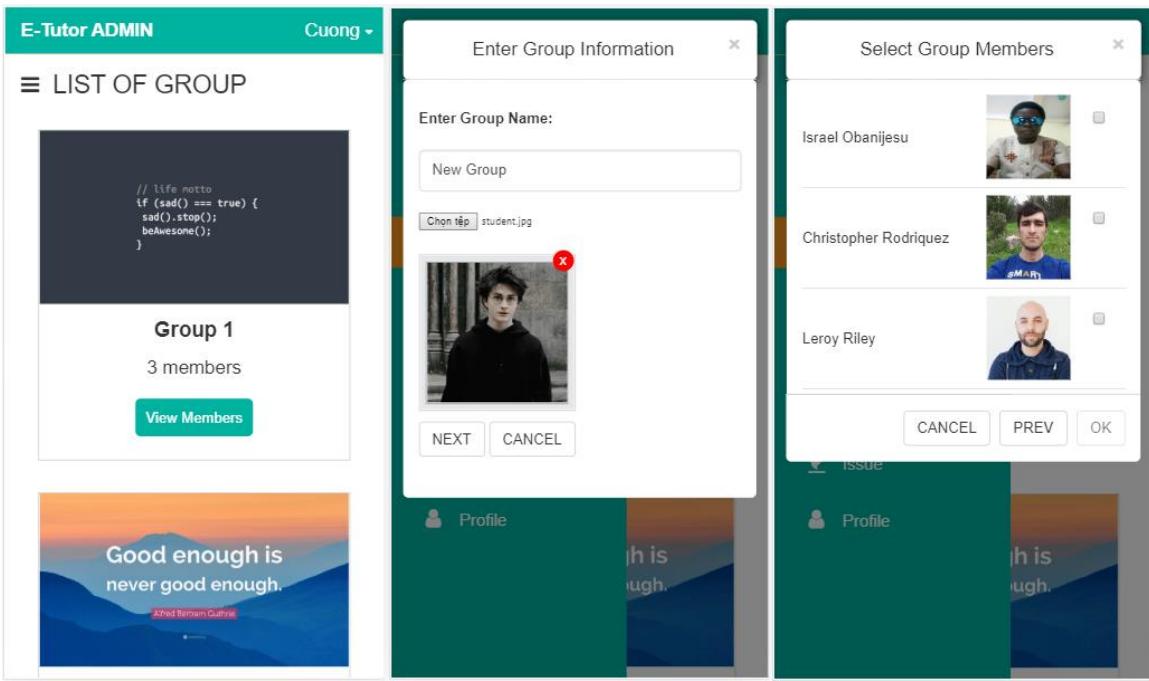


Figure 54. Image of list of group and steps of adding new group

5.3.10 All User – Add User

This site displays all user, admin can view with the user filter as All User, All Tutor...

E-Tutor ADMIN		Cuong																																																																																	
Dashboard Group Comments Users Meeting Issue Profile		Users <input type="button" value="Show All Users"/> <table border="1"> <thead> <tr> <th>No</th><th>Name</th><th>Email</th><th>Tutor</th><th>Role</th><th>Action</th><th>Action</th></tr> </thead> <tbody> <tr><td>1</td><td>Israel Obanjesu</td><td>israeloogch17619@fpt.edu.vn</td><td>N/A</td><td>admin</td><td>DEACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>2</td><td>Christopher Rodriguez</td><td>christopher.rodriguez@example.com</td><td>N/A</td><td>admin</td><td>DEACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>3</td><td>Lois Ellis</td><td>lois.ellis@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>4</td><td>Morris Alexander</td><td>morris.alexander@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>5</td><td>Cassandra Elliott</td><td>cassandra.elliott@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>6</td><td>Terra Ruiz</td><td>terra.ruiz@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>7</td><td>Stacey Cook</td><td>stacey.cook@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>8</td><td>Andrew Obrien</td><td>andrew.obrien@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>9</td><td>Caleb Anderson</td><td>caleb.anderson@example.com</td><td>N/A</td><td>admin</td><td>ACTIVATE</td><td>VIEW USER CARD</td></tr> <tr><td>10</td><td>Leroy Riley</td><td>leroy.riley@example.com</td><td>N/A</td><td>tutor</td><td>DEACTIVATE</td><td>VIEW USER CARD</td></tr> </tbody> </table>					No	Name	Email	Tutor	Role	Action	Action	1	Israel Obanjesu	israeloogch17619@fpt.edu.vn	N/A	admin	DEACTIVATE	VIEW USER CARD	2	Christopher Rodriguez	christopher.rodriguez@example.com	N/A	admin	DEACTIVATE	VIEW USER CARD	3	Lois Ellis	lois.ellis@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	4	Morris Alexander	morris.alexander@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	5	Cassandra Elliott	cassandra.elliott@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	6	Terra Ruiz	terra.ruiz@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	7	Stacey Cook	stacey.cook@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	8	Andrew Obrien	andrew.obrien@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	9	Caleb Anderson	caleb.anderson@example.com	N/A	admin	ACTIVATE	VIEW USER CARD	10	Leroy Riley	leroy.riley@example.com	N/A	tutor	DEACTIVATE	VIEW USER CARD
No	Name	Email	Tutor	Role	Action	Action																																																																													
1	Israel Obanjesu	israeloogch17619@fpt.edu.vn	N/A	admin	DEACTIVATE	VIEW USER CARD																																																																													
2	Christopher Rodriguez	christopher.rodriguez@example.com	N/A	admin	DEACTIVATE	VIEW USER CARD																																																																													
3	Lois Ellis	lois.ellis@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
4	Morris Alexander	morris.alexander@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
5	Cassandra Elliott	cassandra.elliott@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
6	Terra Ruiz	terra.ruiz@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
7	Stacey Cook	stacey.cook@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
8	Andrew Obrien	andrew.obrien@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
9	Caleb Anderson	caleb.anderson@example.com	N/A	admin	ACTIVATE	VIEW USER CARD																																																																													
10	Leroy Riley	leroy.riley@example.com	N/A	tutor	DEACTIVATE	VIEW USER CARD																																																																													
First 1 2 3 Last																																																																																			

Figure 55. All user page

To add a new user, the user accesses the "Add New User" page from the side menu. In creating issue form, the admin needs to enter the full name of the user, email of the user, uploading image of user, selecting the role of the user.

E-Tutor ADMIN Cuong ▾

- [Dashboard](#)
- [Group](#) ▾
- [Comments](#)
- [Users](#) ▾
- [Meeting](#) ▾
- [Issue](#)
- [Profile](#)

Full Name

Email

Role

Select a Role

Picture

Select a File



[ADD NEW USER](#)

Figure 56. Image of Add new user page

E-Tutor ADMIN Cuong ▾

E-Tutor ADMIN Cuong ▾

All Meeting

No	Title	Organizer	Date	Status
1	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 4:51:37 am	done
2	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 9:54:50 am	done
3	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 10:17:47 am	done

[Previous](#) 1 [2](#) [3](#) [Next](#)

Add User

Full Name

Email

Role

Select a Role

Picture

Select a File



Figure 57. All user - Add New User page on mobile

5.3.11 Allocate student to tutor

To allocate student to tutor, admin select "Allocate student to tutor" from the side menu, one modal is displayed, admin choose 1 tutor of the first modal then select "Next" button, in the second modal, admin choose a student, finally select "Done" button.

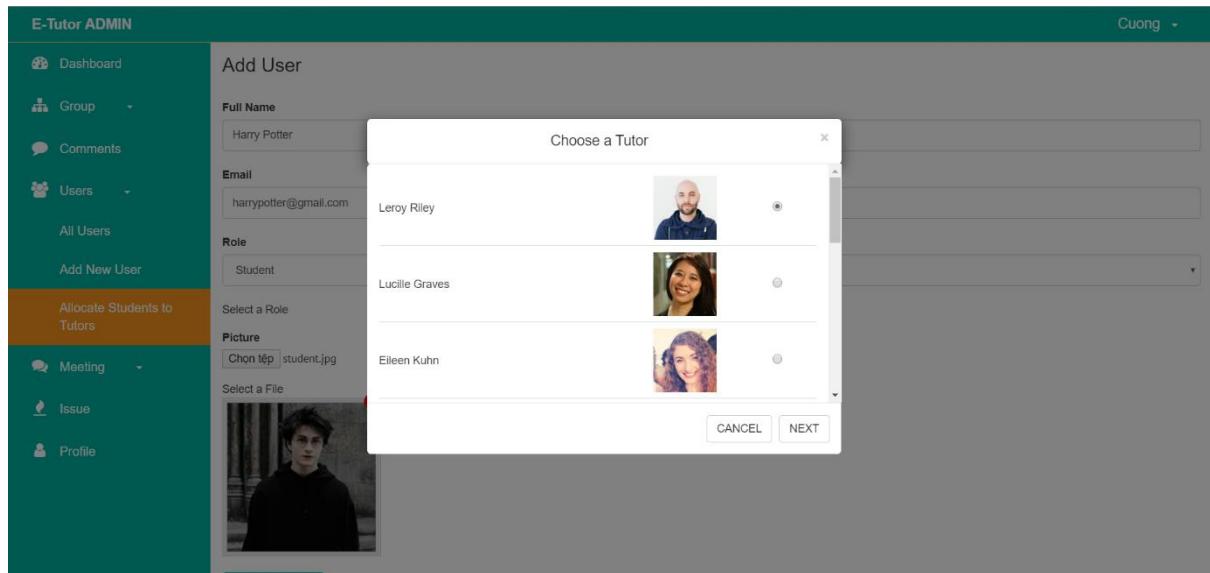


Figure 58. Step 1 of allocate student to tutor

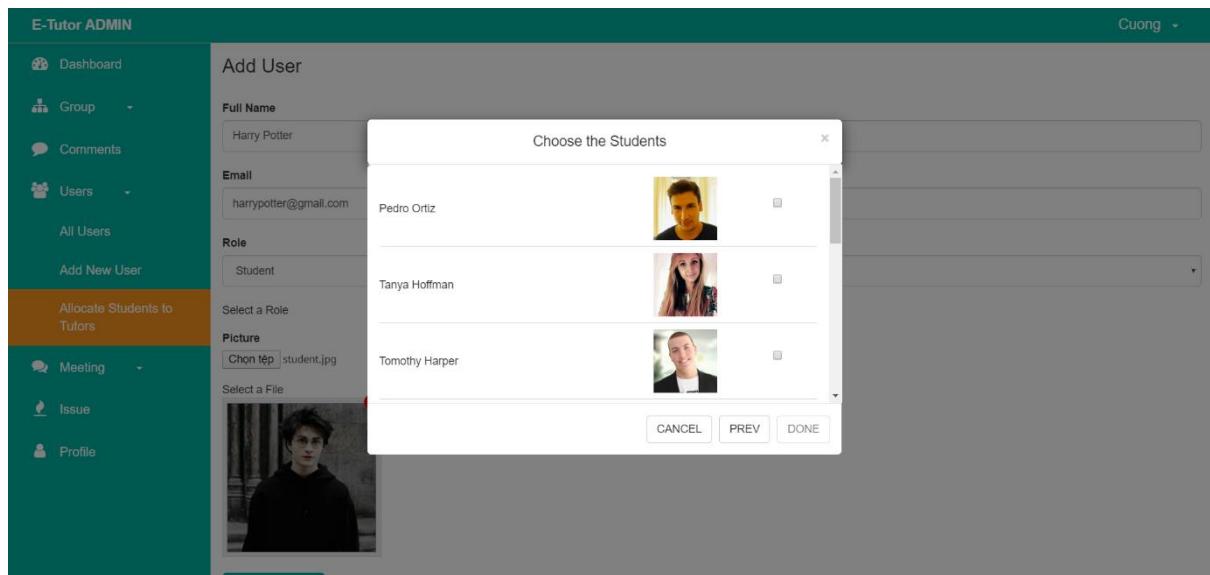


Figure 59. Step 2 of allocating student to tutor

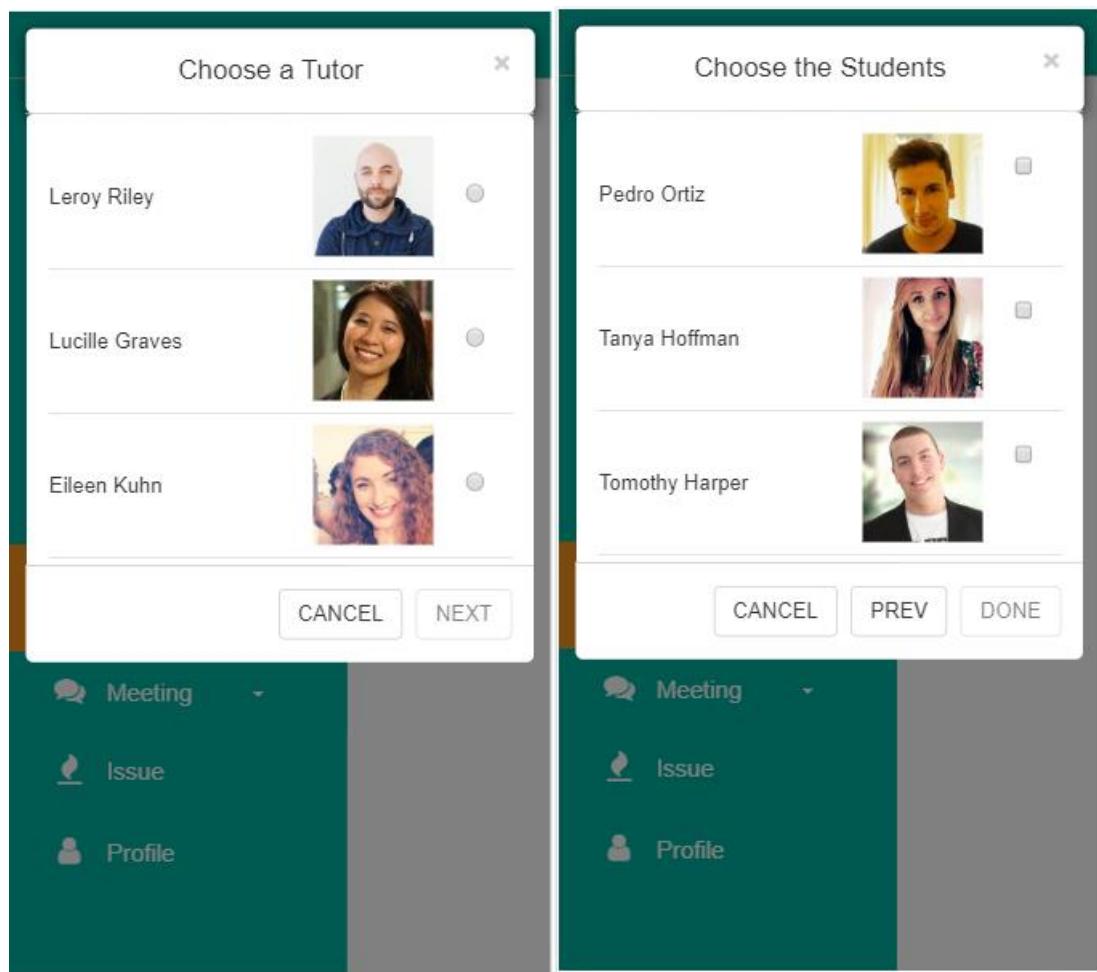


Figure 60. Steps of allocating student to tutor

5.3.12 Meeting – Create New Meeting

This site will display all meeting and status of meeting.

E-Tutor ADMIN					
Cuong					
All Meeting					
No	Title	Organizer	Date	Status	
1	Group 1 Meeting	Israel Obanjesu	April 17th 2020, 4:51:37 am	done	
2	Group 1 Meeting	Israel Obanjesu	April 17th 2020, 9:54:50 am	done	
3	Group 1 Meeting	Israel Obanjesu	April 17th 2020, 10:17:47 am	done	

Figure 61. All meeting for admin

To add a new meeting, the user accesses the "Add New Meeting" page from the side menu. In creating issue form, the admin needs to enter the title of the meeting, a description of the meeting, and selecting the member for a meeting.

Figure 62.Image of adding new meeting

No	Title	Organizer	Date	Status
1	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 4:51:37 am	done
2	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 9:54:50 am	done
3	Group 1 Meeting	Israel Obanijesu	April 17th 2020, 10:17:47 am	done

Figure 63.Image of all meeting - add new meeting page for admin on mobile

5.3.13 Admin Profile – Edit profile

Admin can view the admin profile when they click on Profile from the side menu. This is modal, it will show image, name, email, and description of the admin. To edit profile, admin select the "Edit" button.

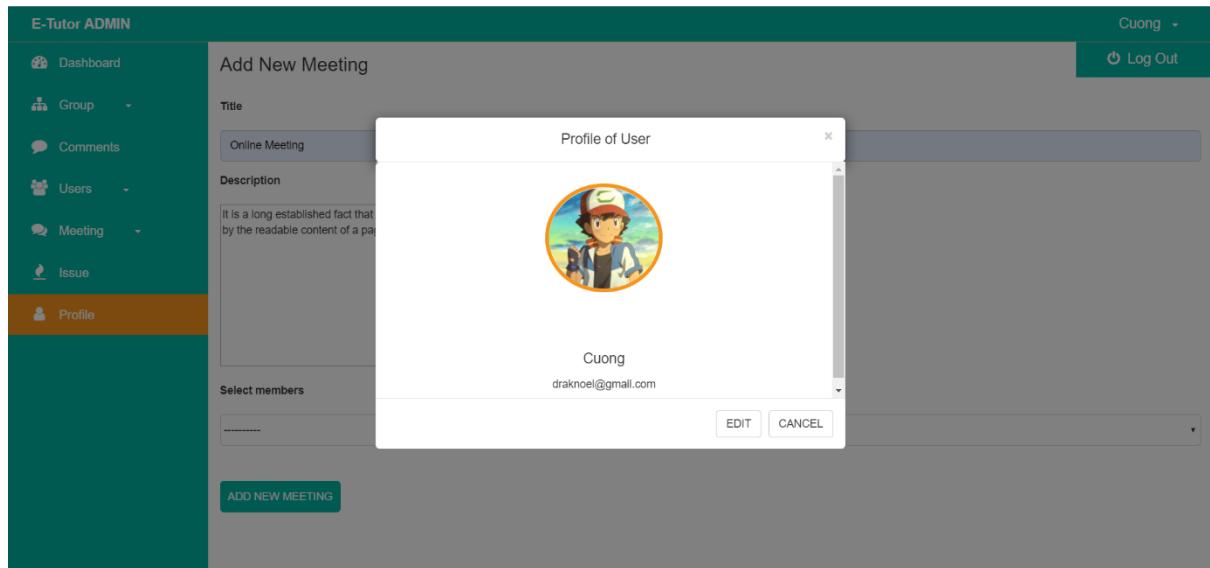


Figure 64.Image of admin profile

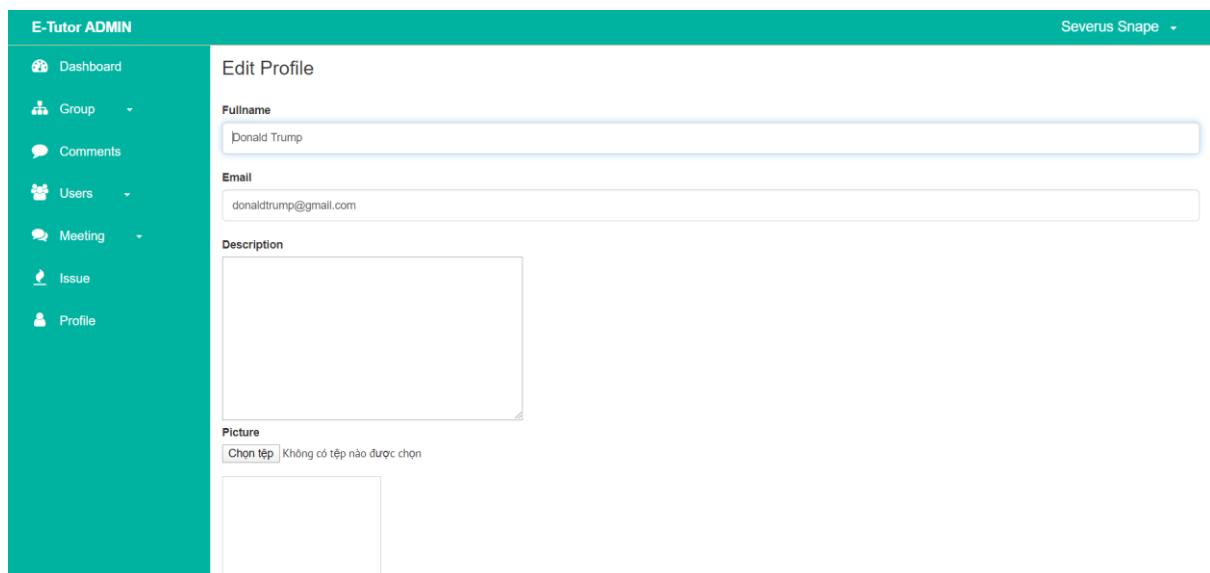


Figure 65.Image of editing profile

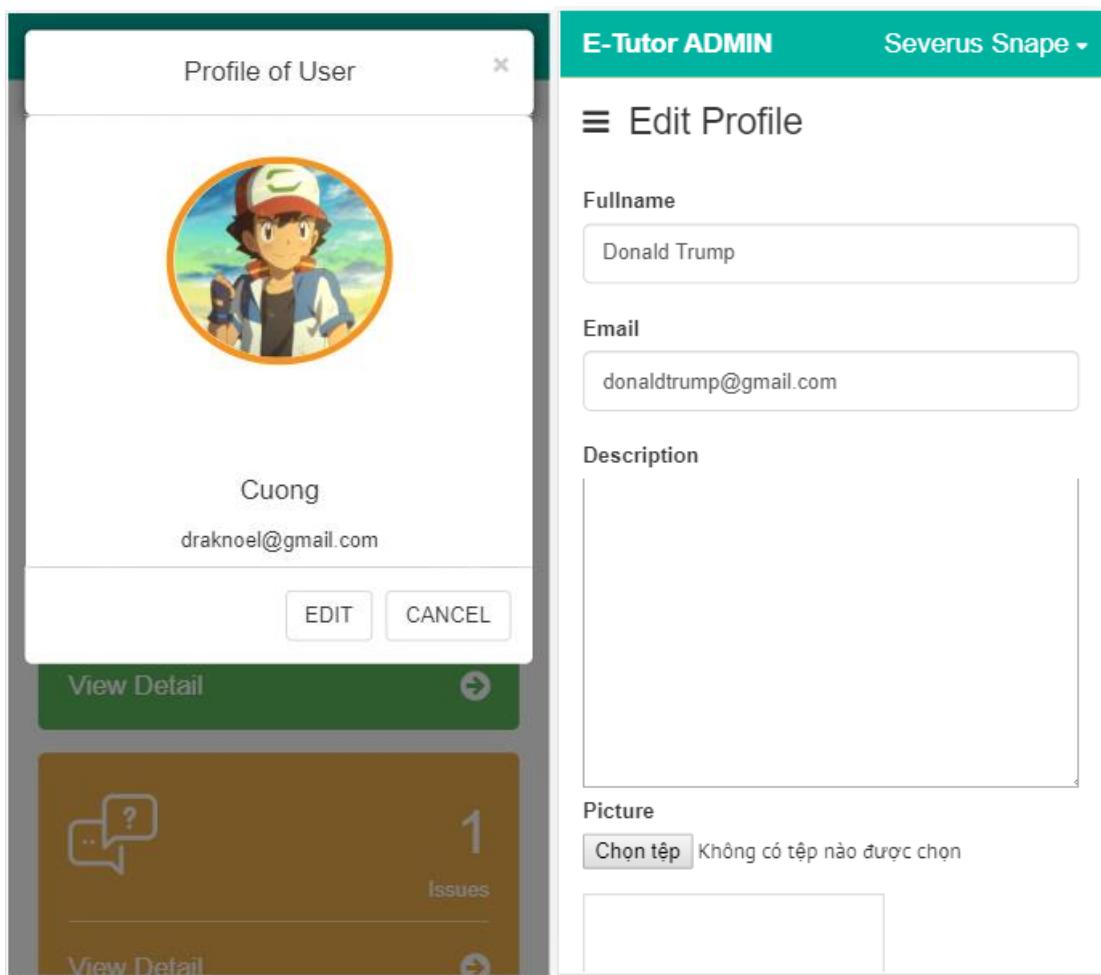


Figure 66. Image o admin profile - edit profile page on mobile

6 FUNCTIONALITIES

In this section, the key functionalities of the project will be pointed out with a short explanation of how they work.

6.1 Login

6.1.1 Login with Email / Password

We implemented two methods for logging in (Email / Password, and Login with Google). For the email / password login method, we simply used Node.js to process user's data and if it's valid, a timed token (JWT) is returned back to the client, which will expire in 1 hour. This token will serve as evidence that the user is authenticated so it must be safe and secured by the client. Additionally, in order to reduce the chance of the user being compromised, the token is made to expire in 1 hour so that the user can login again and require for a new one. So, if an attacker manages to get a hold of the token from the user's browser, it won't last for long.

```
const token = jwt.sign({
  userId: user._id.toString(),
  userRole: user.role
}, `${process.env.JWT_PASSWORD}`, {expiresIn: '1h'});
res.status(200).json({
  message: 'User Logged In Successfully!',
  data: {
    token: token,
    userId: user._id.toString(),
    userRole: user.role,
    userName: user.name,
    userPhoto: user.image
  }
});
```

Figure 67: Code for Logging In with Email / Password

The code written above, generates a new token and sign the users in.

6.1.2 Login with Google

In order to allow users to login with their Google account, we used “googleapis” package to handle the login process. Here is the code for logging in with google:

```

signInWithGoogle: (req, res) => {
  const url = OAuth2Client.generateAuthUrl({
    access_type: 'offline',
    scope: 'https://www.googleapis.com/auth/userinfo.email'
  });
  res.status(200).json({ url });
},

```

Figure 68: Code for Logging in with Google

6.2 Allocate Students to Tutor

Students are allocated to tutor with the use of node.js, and agenda.js was used to trigger the job that will send emails to the specified students and tutor via nodemailer. Here is the code for allocating students to tutor:

```

const studentsName = [];
for (let i = 0; i < students.length; i++) {
  tutor.students.push(students[i]._id);
  studentsName.push(students[i].name);
}
for (let i = 0; i < students.length; i++) {
  students[i].tutor = tutor._id;
  students[i].save();
  await agenda.now('allocation notification to student', {
    email: students[i].email,
    name: students[i].name,
    tutor: tutor.name
  });
}
await agenda.now('allocation notification to tutor', {
  email: tutor.email,
  name: tutor.name,
  students: studentsName
});
tutor.studentsLength = tutor.students.length;
const updatedTutor = await tutor.save();
res.status(201).json({
  message: 'Assign successfully',
  data: { updatedTutor }
});

```

Figure 69: Code for allocating students to tutor

6.3 Blogging

6.3.1 Create Post

In order to create post, users must submit the title, photo, body, and category of the post. An activity is also recorded after user creates a post. Here is the code for creating post:

```
createPost: async (req, res, next) => {
  try {
    const author = req.userId;
    const title = req.body.title;
    const photo = req.body.photo;
    const body = req.body.body;
    const category = req.body.category;
    const post = new Blog({title, author, photo, body, category});
    await post.save();
    const activity = new Activity({
      activity: 'createPost',
      owner: req.userId,
      post: post._id
    });
    await activity.save();
    res.status(201).json({
      message: 'Post Created Successfully!',
      data: post
    });
  } catch (err) {
    if (!err.statusCode) err.statusCode = 500;
    next(err);
  }
},
```

Figure 70: Code for creating posts

6.3.2 View Posts

Below is the code for viewing all the posts in the system. The posts are sorted in a descending order based on their likes count.

```
getPosts: async (req, res, next) => {
  try {
    const currentPage = req.query.page || 1;
    const perPage = req.query.perPage || 10;
    const posts = await Blog.find()
      .skip((currentPage - 1) * perPage)
      .limit(perPage)
      .populate('author')
      .sort({ likesCount: -1 });
    res.status(200).json({
      message: 'Posts Retrieved Successfully!',
      data: posts
    });
  } catch (err) {
    if (!err.statusCode) err.statusCode = 500;
    next(err);
  }
},
```

Figure 71: Code for viewing posts

6.3.3 Like Post

Below is the code for liking a post. An activity is created each time, a user likes a post.

```
if (post.dislikes.includes(req.userId)) {
    post.dislikesCount -= 1;
    post.dislikes = post.dislikes.filter(el => el.toString() !== req.userId);
}

if (!post.likes.includes(req.userId)) {
    post.likesCount += 1;
    post.likes.push(req.userId);
} else {
    post.likesCount -= 1;
    post.likes = post.likes.filter(el => el.toString() !== req.userId)
}

await post.save();
const activity = new Activity({
    activity: 'likePost',
    owner: req.userId,
    post: post._id
});
await activity.save();
res.status(201).json({
    message: 'Liked Post Successfully!',
    data: post
});
```

Figure 72: Code for liking a post

6.3.4 Dislike Post

Below is the code for disliking a post. An activity is also created each time, a user dislikes a post. The code for disliking a post is a direct inversion of the code for liking a post.

```

if (post.likes.includes(req.userId)) {
  post.likesCount -= 1;
  post.likes = post.likes.filter(el => el.toString() !== req.userId);
}

if (!post.dislikes.includes(req.userId)) {
  post.dislikesCount += 1;
  post.dislikes.push(req.userId);
} else {
  post.dislikesCount -= 1;
  post.dislikes = post.dislikes.filter(el => el.toString() !== req.userId);
}

const activity = new Activity({
  activity: 'dislikePost',
  owner: req.userId,
  post: post._id
});
await activity.save();
await post.save();
res.status(201).json({
  message: 'Liked Post Successfully!',
  data: post
});

```

Figure 73: Code for disliking a post

6.4 Sending Emails

In order to send emails, we used nodemailer with SendGrid API service, which are wrapped in jobs that are triggered with agenda.js. We have a total of 9 email jobs. Here are some important ones.

6.4.1 Sending Email for Allocation of Students to Tutor

```

agenda.define('allocation notification to tutor', async job => {
  const email = job.attrs.data.email;
  const name = job.attrs.data.name;
  const students = job.attrs.data.students;
  transporter.sendMail({
    to: email,
    from: 'admin@e-tutor.com',
    subject: 'Allocation Notification',
    html: `
      <p>Hello ${name},</p>
      <p>You have been allocated to ${students.length} Students: ${students.map(el => ` ${el}`)}</p>
    `,
  });
});

agenda.define('allocation notification to student', async job => {
  const email = job.attrs.data.email;
  const name = job.attrs.data.name;
  const tutor = job.attrs.data.tutor;
  transporter.sendMail({
    to: email,
    from: 'admin@e-tutor.com',
    subject: 'Allocation Notification',
    html: `
      <p>Hello ${name},</p>
      <p>You have been allocated to ${tutor}</p>
    `,
  });
});

```

Figure 74: Code for sending email for allocation of students to tutor

6.4.2 Sending Email for Resetting Password

```

agenda.define('reset password email', async job => {
  const email = job.attrs.data.email;
  const url = job.attrs.data.url;
  const token = job.attrs.data.token;
  transporter.sendMail({
    to: email,
    from: 'admin@e-tutor.com',
    subject: 'Password Reset',
    html: `
      <p>You requested a password reset</p>
      <p>Click this
      <a href="${url}/change-password.html?token=${token}">
        link</a> to set a new password.</p>
    `,
  });
});

```

Figure 75: Code for sending email for resetting password

6.4.3 Sending Email for Meeting Request

```
agenda.define('meeting invitation email', async job => {
  const email = job.attrs.data.email;
  const url = job.attrs.data.url;
  const token = job.attrs.data.token;
  const organizer = job.attrs.data.organizer;
  const date = job.attrs.data.date;
  const title = job.attrs.data.title;
  transporter.sendMail({
    to: email,
    from: 'admin@e-tutor.com',
    subject: 'Meeting Invitation',
    html: `
      <p>You have been invited to join a meeting</p>
      <p>Organizer: ${organizer}</p>
      <p>Date: ${date}</p>
      <p>Title: ${title}</p>
      <p>Click this <a href="${url}/accept-meeting.html?token=${token}">link</a> to accept or reject the meeting</p>
    `;
  });
});
```

Figure 76: Code for sending email for meeting request

6.4.4 Sending Email for Meeting Reminder

```
agenda.define('meeting reminder email', async job => {
  // if (await checkCancelled(job)) return;
  await mongoose.connect(MONGODB_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  });
  const email = job.attrs.data.email;
  const name = job.attrs.data.name;
  const title = job.attrs.data.title;
  const id = job.attrs.data.id;
  const meeting = await Meeting.findById(id);
  meeting.status = 'done';
  transporter.sendMail({
    to: email,
    from: 'admin@e-tutor.com',
    subject: 'Meeting Reminder',
    html: `
      <p>Hello ${name}</p>
      <p>Your Meeting "${title}" starts now</p>
    `;
  });
});
```

Figure 77: Code for sending email for meeting reminder

6.5 Messaging

6.5.1 Send Message

In order to send messages to users in real-time, we used Socket.IO to broadcast and listen to message broadcasts. Below is the code for used to sending message:

```
// Broadcast "send message" event
if (groupId) {
    io.getIO().to(groupId).emit('message', {
        action: 'send',
        message: {...message._doc}
    });
} else {
    io.getIO().to(from + to[0]).emit('message', {
        action: 'send',
        message: {...message._doc}
    });
}
// Return a successful response
res.status(201).json({
    message: 'Message sent successfully!',
    data: { message }
});
```

Figure 78: Code for sending messages

6.5.2 View Messages

```
const messages = await Message
  .find({
    $or: [
      {
        $and: [
          { from: req.userId },
          { to: { $in: otherUsers } }
        ]
      },
      {
        $and: [
          { from: { $in: otherUsers } },
          { to: req.userId }
        ]
      }
    ]
  })
  .populate('from').populate('to')
  .skip((currentPage - 1) * perPage)
  .limit(perPage);
res.status(200).json({
  message: 'Messages Fetched Successfully!',
  data: { messages }
});
```

Figure 79: Code for viewing messages

6.5.3 Delete Message

```
await Message.findByIdAndRemove(messageId, { useFindAndModify: false });
const group = await Group.findById(message.to[0]);
if (group) {
  io.getIO().to(group._id).emit(
    'message',
    { action: 'delete', message: messageId }
  );
} else {
  io.getIO().to(userId + message.to[0]).emit(
    'message',
    {action: 'delete', message: messageId}
  );
}
res.status(200).json({message: 'Message Deleted Successfully'});
```

Figure 80: Code for deleting messages

6.6 Issues

6.6.1 Create Issue

```
const issue = new Issue({
  title,
  description,
  creator,
  assignTo,
  files
});
try {
  await issue.save();
  const activity = new Activity({
    activity: 'createIssue',
    owner: req.userId,
    issue: issue._id
  });
  await activity.save();
  res.status(201).json({
    message: 'Issue created successfully!',
    data: {issue}
  });
}
```

Figure 81: Code for creating issues

6.6.2 View Issues

```
let issues;
if (pagination === 'true') {
  issues = await Issue.find({
    $or: [
      {creator: userId},
      {assignTo: {$in: userId}}
    ]
  }).skip((currentPage - 1) * perPage)
    .limit(perPage)
    .sort({createdAt: -1}).populate('assignTo').populate('creator');
} else {
  issues = await Issue.find({
    $or: [
      {creator: userId},
      {assignTo: {$in: userId}}
    ]
  }).sort({createdAt: -1}).populate('assignTo').populate('creator');
}
res.status(200).json({
  message: 'User Issues Retrieved Successfully!',
  issues,
  totalItems
});
```

Figure 82: Code for viewing issues

6.6.3 Update Issue

```
issue.title = title;
issue.description = description;
issue.files = files;
issue.assignTo = assignTo;
await issue.save();
const activity = new Activity({
  activity: 'editIssue',
  owner: req.userId,
  issue: issue._id
});
await activity.save();
res.status(201).json({
  message: 'Issue Updated Successfully!',
  data: issue
});
```

Figure 83: Code for updating issues

6.6.4 Delete Issue

```
await Comment.deleteMany({ issue: issue._id });
await Issue.findByIdAndRemove(issueId, { useFindAndModify: false });
const activity = new Activity({
  activity: 'deleteIssue',
  owner: req.userId,
  issue: issue._id
});
await activity.save();
res.status(200).json({message: 'Issue Deleted Successfully'});
```

Figure 84: Code for deleting issues

6.7 Flowchart for Login with Email / Password

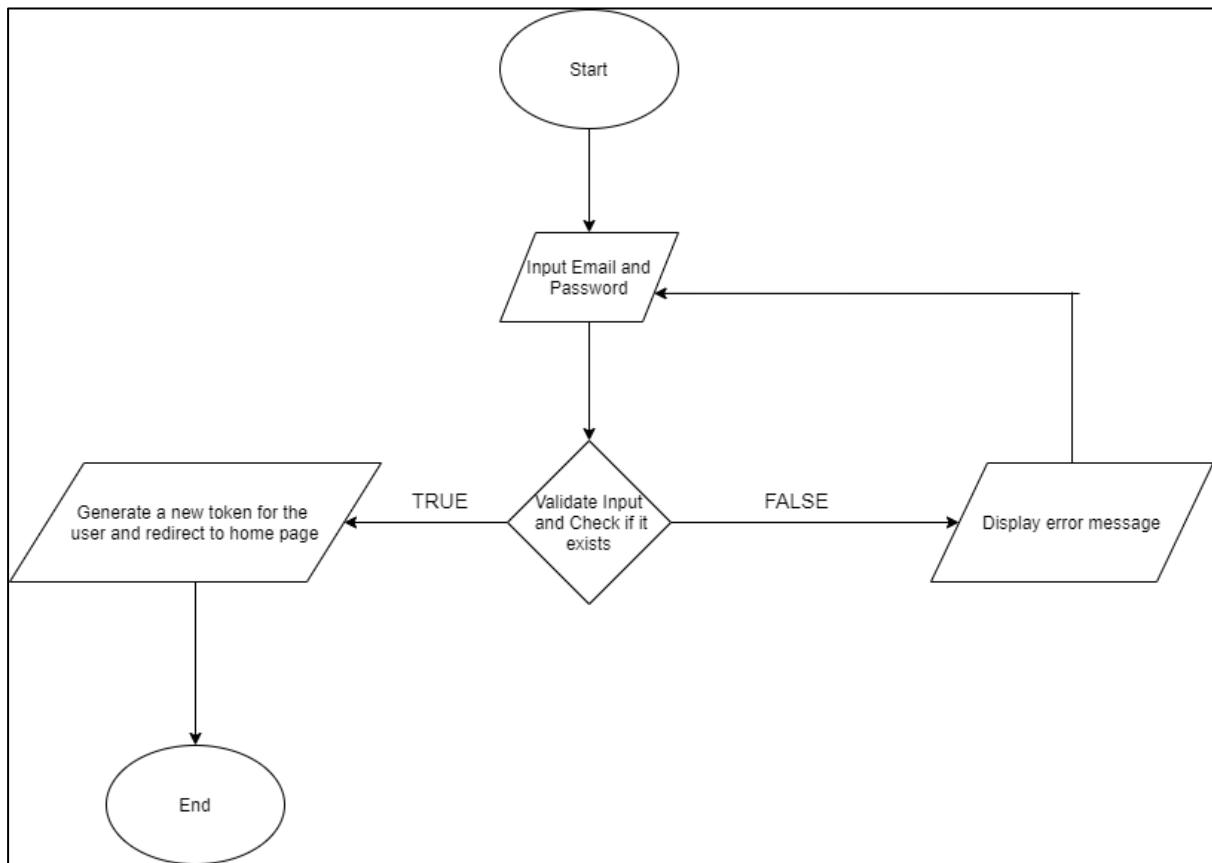


Figure 85: Flowchart for Login with Email / Password

6.8 Flowchart for Login with Google

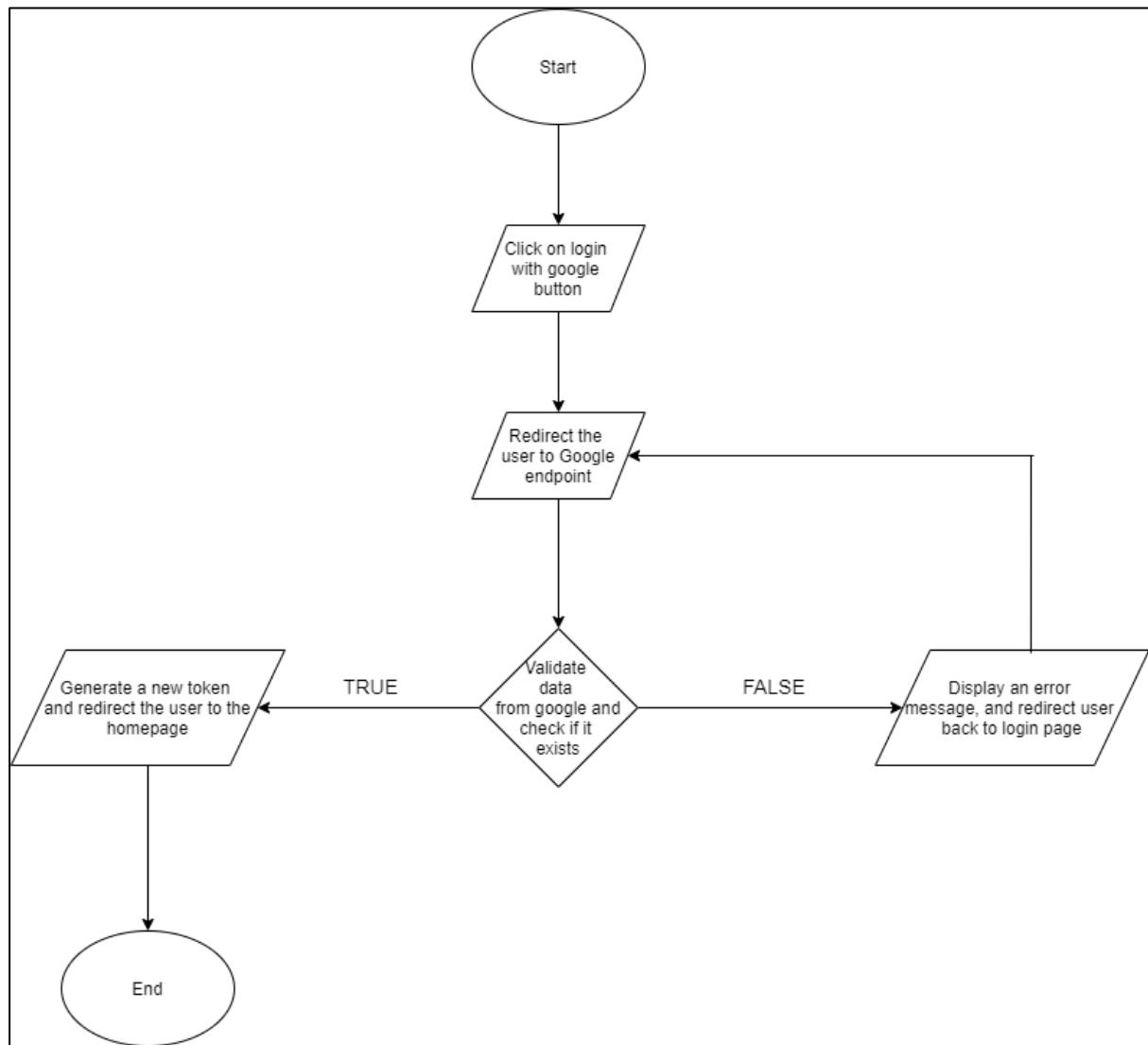


Figure 86: Flowchart for Login with Google

6.9 Flowchart for Allocating Students to Tutor

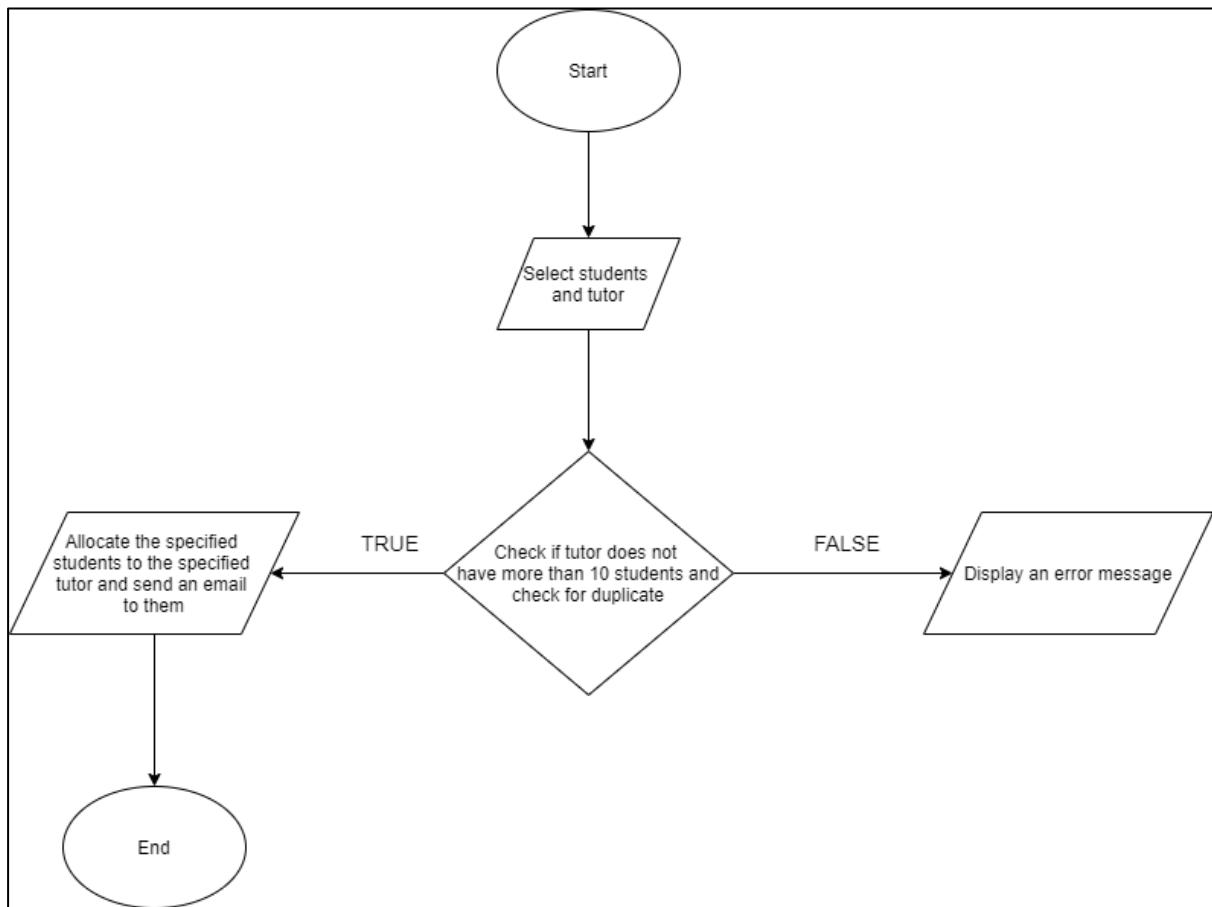


Figure 87: Flowchart for Allocating Students to Tutor

6.10 Flowchart for Creating Post

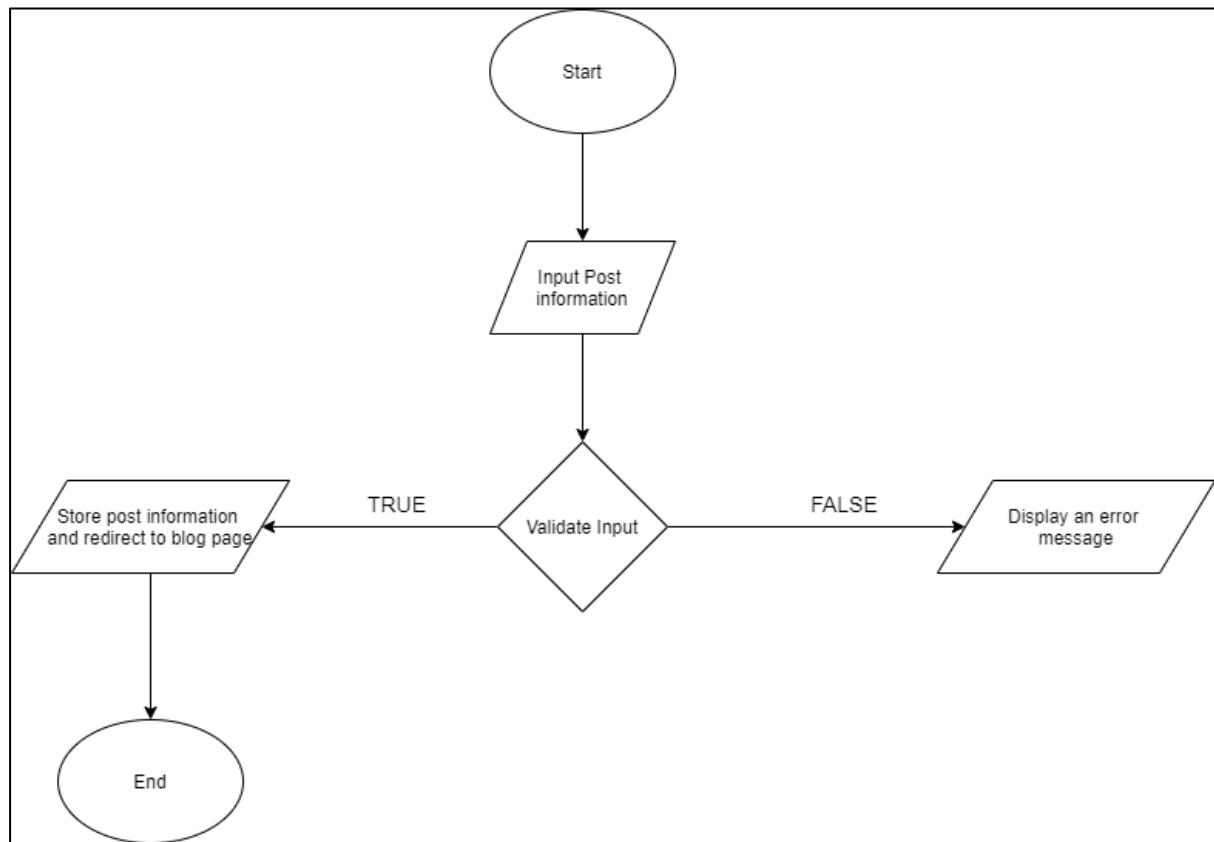


Figure 88: Flowchart for Creating Post

6.11 Flowchart for Liking Post

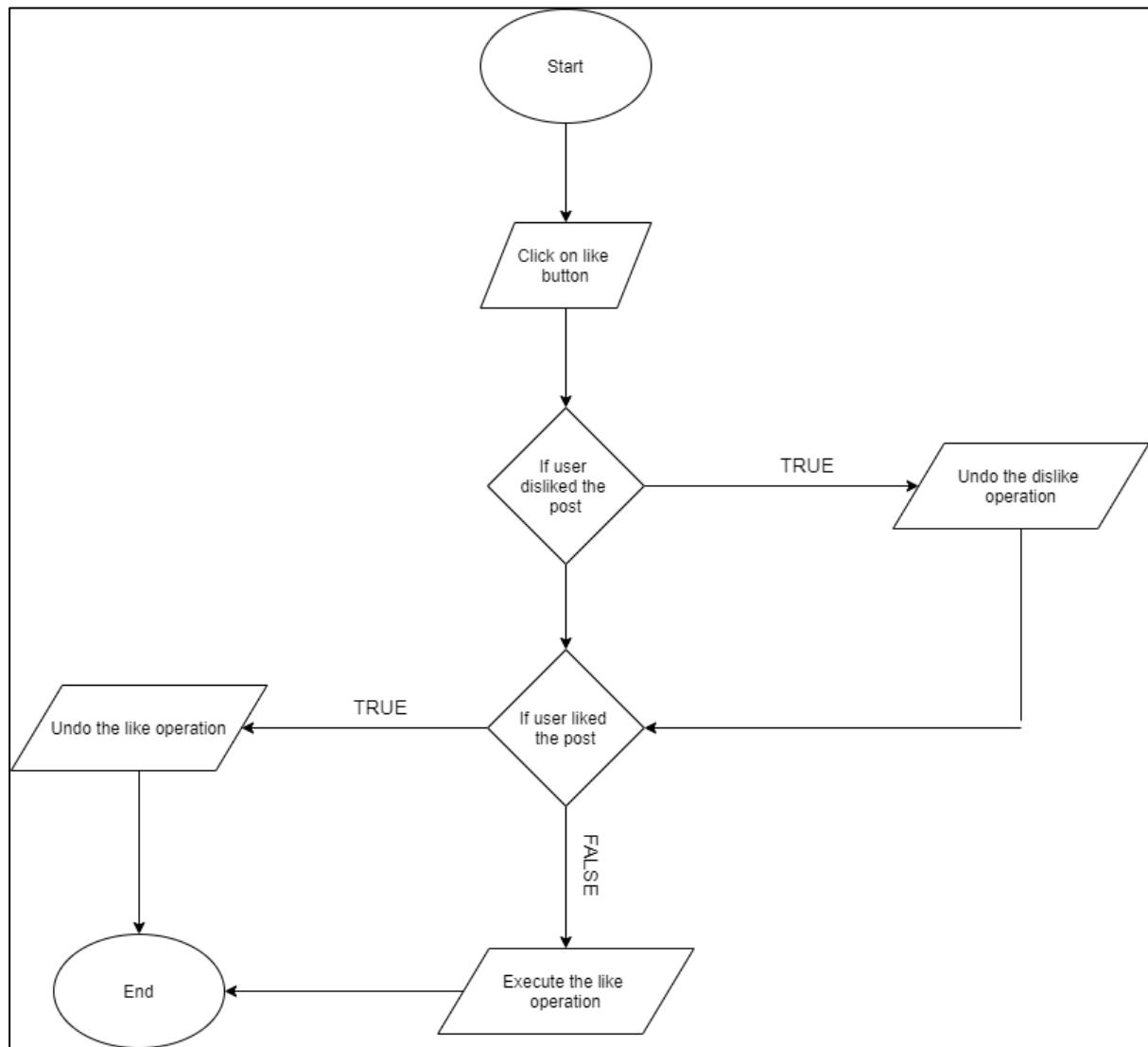


Figure 89: Flowchart for liking post

6.12 Flowchart for Disliking Post

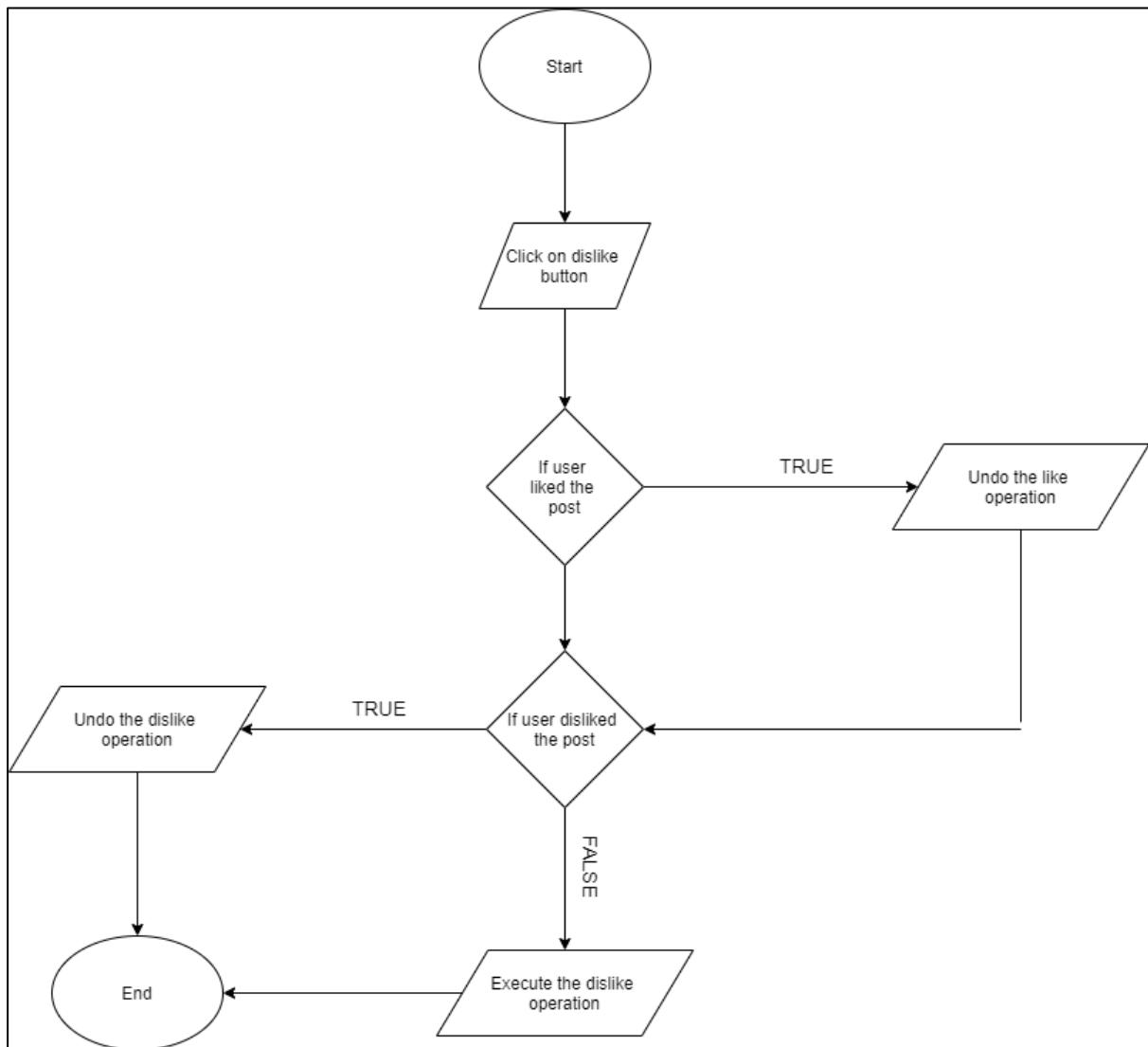


Figure 90: Flowchart for disliking post

6.13 Flowchart for Sending Message

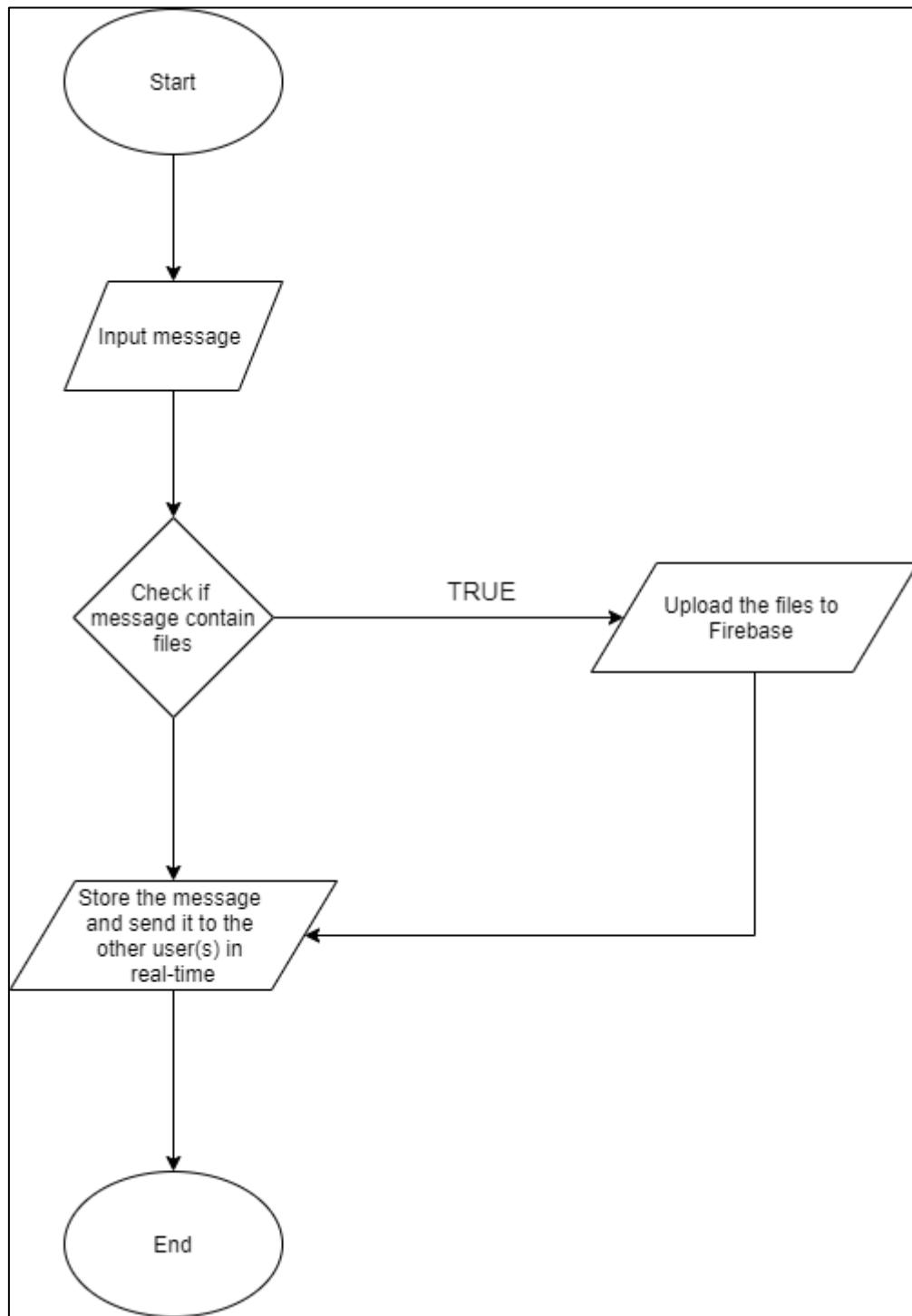


Figure 91: Flowchart for sending messages

6.14 Flowchart for Deleting Message

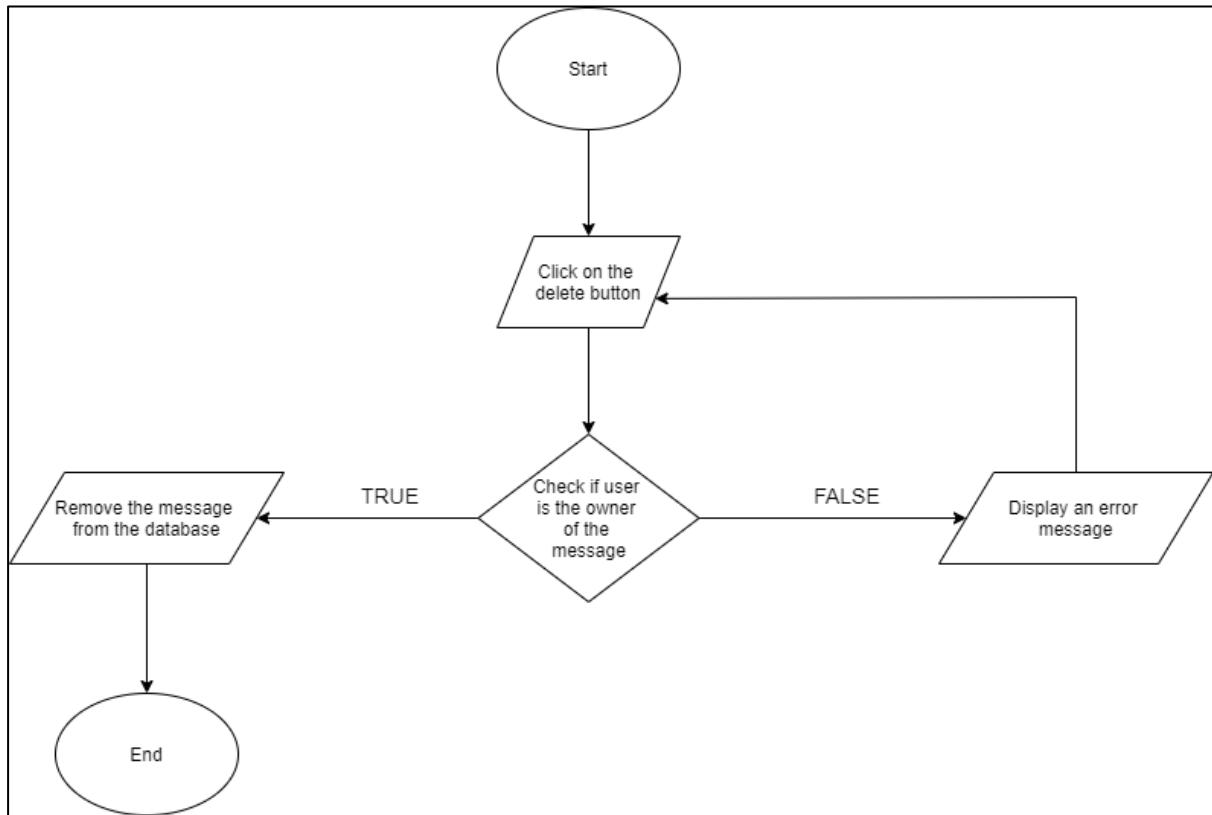


Figure 92: Flowchart for deleting message

6.15 Flowchart for Creating Issue

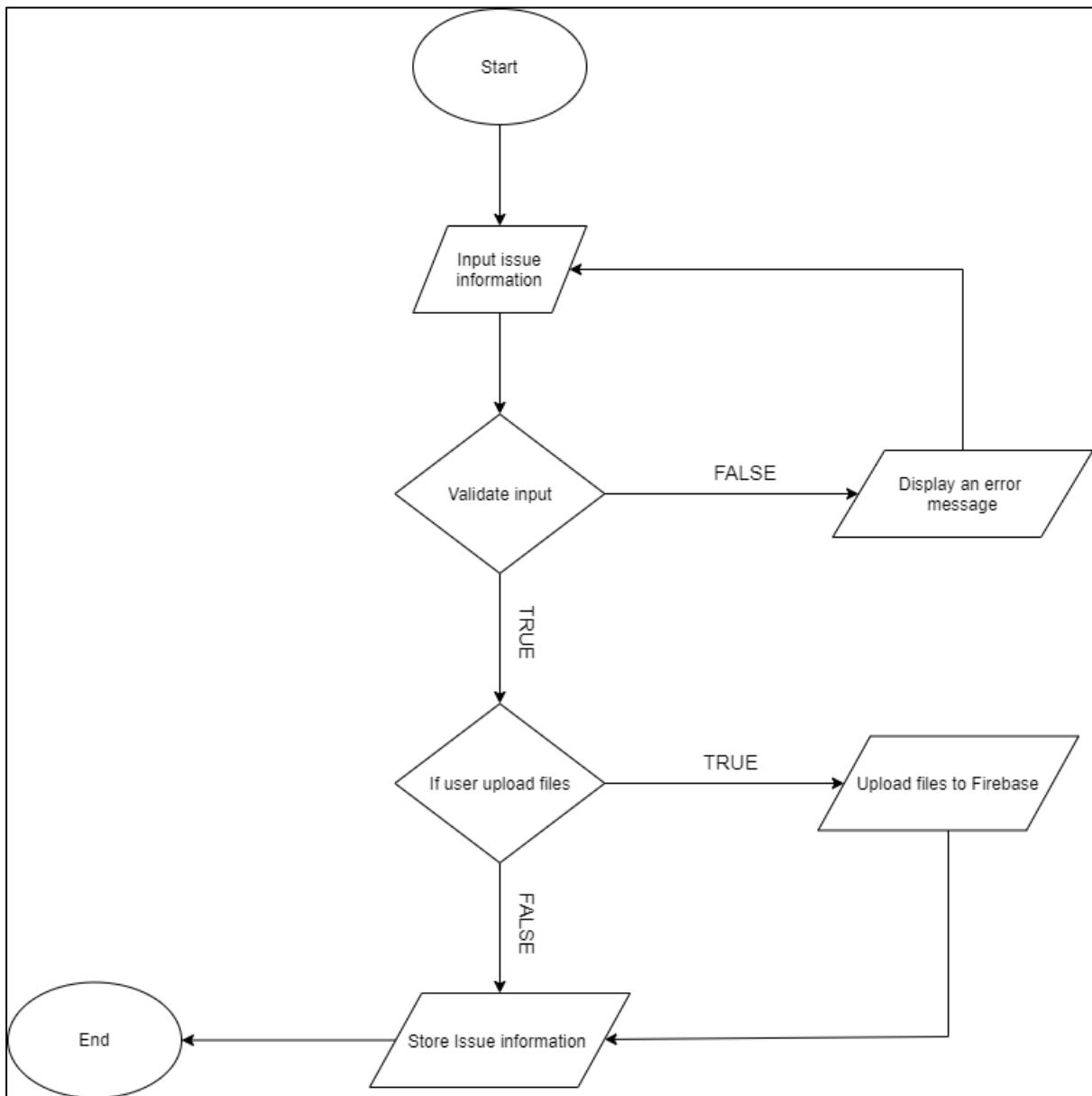


Figure 93: Flowchart for creating issue

6.16 Flowchart for Updating Issue

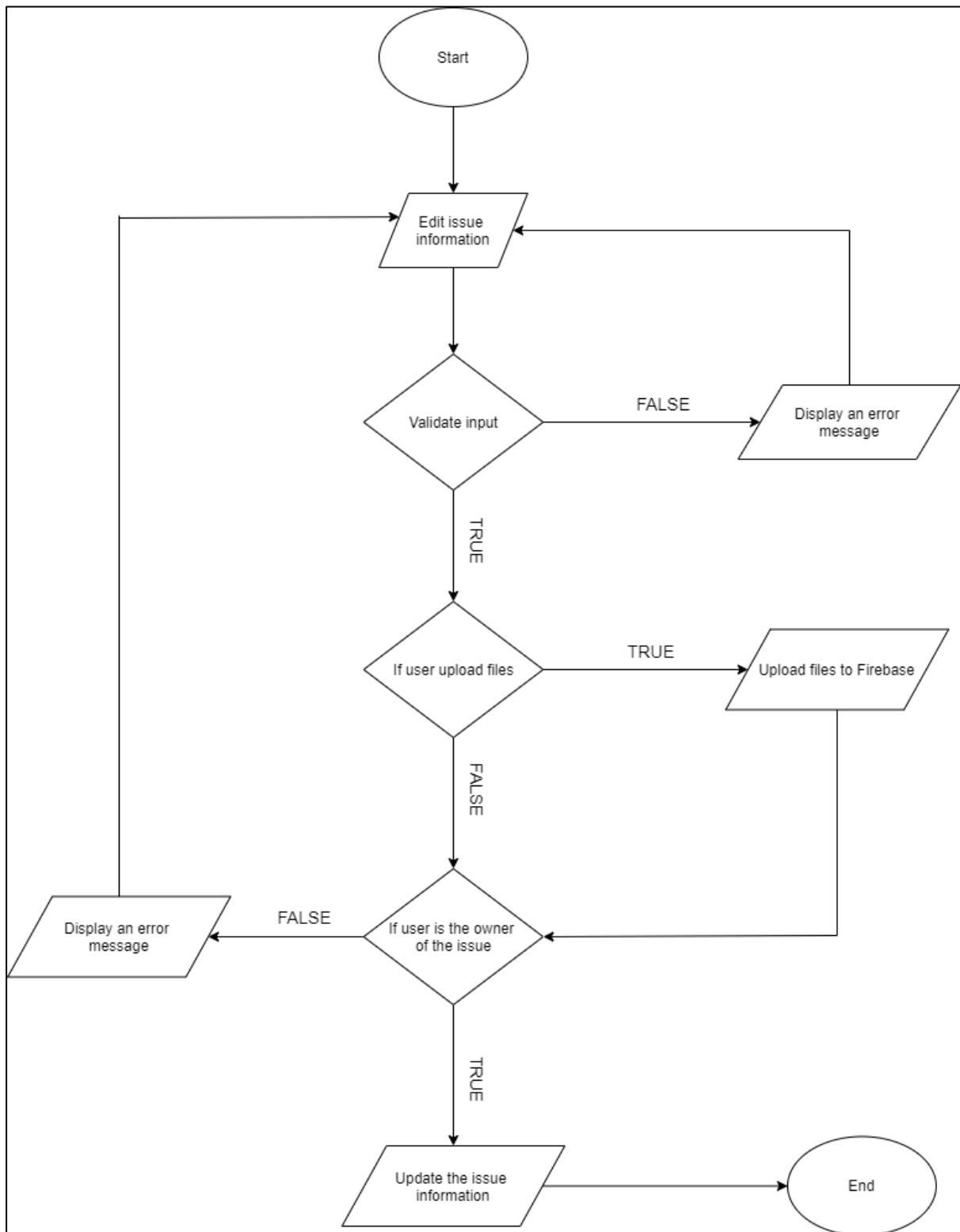


Figure 94: Flowchart for updating issue

6.17 Flowchart for Deleting Issue

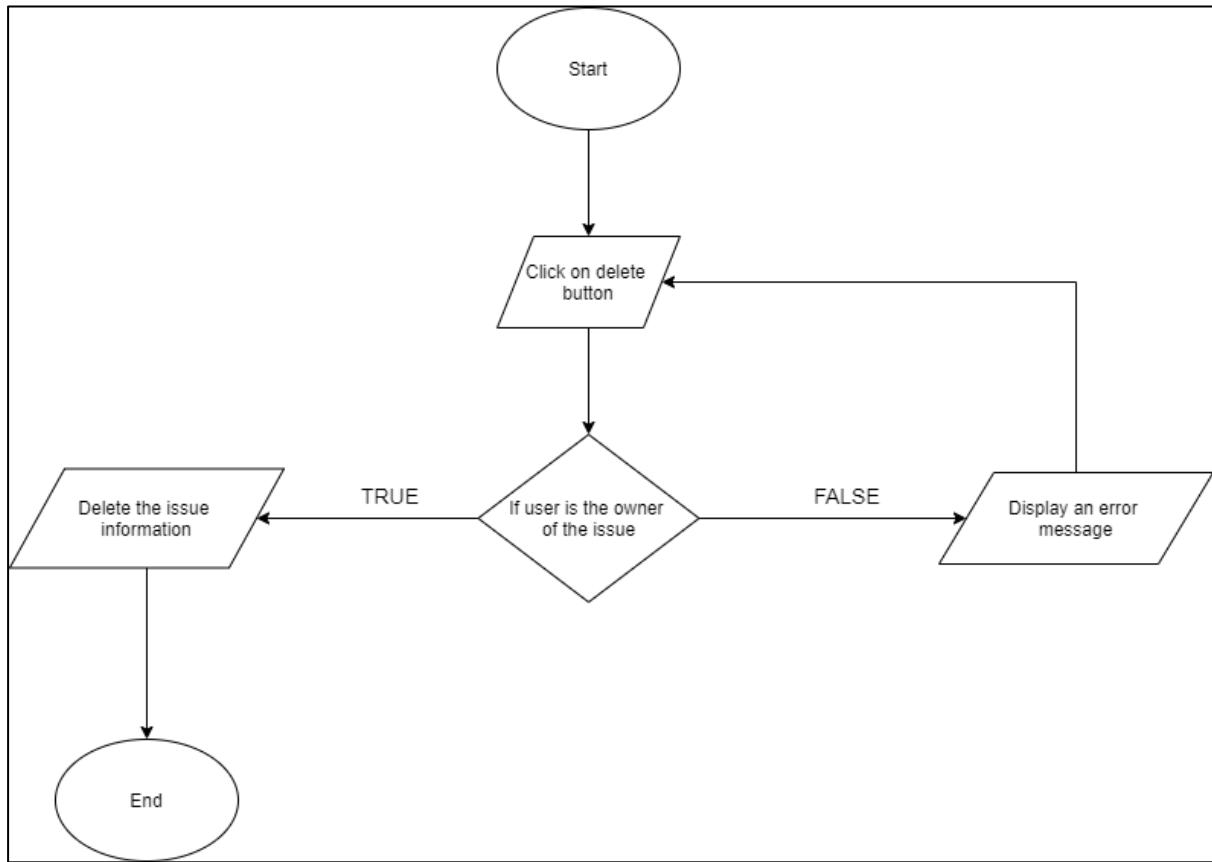


Figure 95: Flowchart for deleting issue

7 TESTING

7.1 Test plan

7.1.1 Test plan identifier

- **Test plan identifier title:** Manual Test plan for E-Tutor
- **Version:** 1.0.0
- **Level:** Manual Testing
- **Product Technologies:** Node.js, Express.js and MongoDB
- **Platform serving:** Web platform
- **Testing environment:** Developing environment (dev stage) and Staging environment (live stage)

7.1.2 Introduction

These test plans are made to identify bug, function fault, weakness of the design, system and functional of our website. The testing objective not only focus on data validations, functional, logical, authentications/authorizations but also focus on the designing of the website, to make a good experience for user.

7.1.3 References

- Product backlog
- Sprint backlogs
- Test cases and test logs

7.1.4 Test artefacts

E-Tutor Web Application (version 1.0.0)

7.1.5 Test features

- Website interface
- Log in/out
- Create/Edit/Delete/View students
- Create/Edit/Delete/View tutors
- Create/View group

- Allocates student to tutor
- Create/View meeting
- Create/View comments
- View students/tutor profile
- Make plan and notify

7.1.6 Test Approach

The test will be performed manually.

7.1.7 Features failed

- Can't postpone a meeting
- Can't cancel a meeting
- View user card

7.1.8 Deliverables

- Test plan (this document)
- Test cases and test log

7.1.9 Estimates

- Estimate time: 12 hours
- Estimate starting date:
 - + Sprint 1: 30/03/2020
 - + Sprint 3: 09/04/2020
- Estimate ending date:
 - + Sprint 1: 31/03/2020
 - + Sprint 2: 12/04/2020
- Estimate human resources: 2

7.1.10 Schedule

In sprint 2, we didn't do any test because there a lot of work was accrued from sprint 1, so we decide to finish it.

- 25/03/2020:
 - + Writing test cases for sprint 1

- 30/03/2020:
 - + Testing login, log out and reset password
 - + Testing send message
 - + Testing view and delete message
- 06/04/2020:
 - + Writing test case for sprint 3
- 09/04/2020
 - + Testing allocate student to tutor
 - + Testing display all allocated student or tutor
 - + Testing display all unallocated student and tutor
- 12/04/2020
 - + Testing de-active user
 - + Testing active user
 - + Testing delete user

7.1.11 Responsibilities

- Oluwasami Obanijesu Israel (Scrum Master and Team Leader)
- Idris Usman Fadil (Tester)
- Ngo Vinh Thai (Tester)

7.1.12 Approvals

Test case and test plan was approved by Product Owner (PO) and Mr. Oluwasami Obanijesu Israel (Scrum master and team leader).

7.2 Test Cases and Test Logs

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. There are things considered when writing a test case

- As far as possible, write test cases in such a way that you test only one thing at a time.
Do not overlap or complicate test cases.
- Ensure that all positive scenarios AND negative scenarios are covered.

- Language:
 - + Write in simple and easy-to-understand language.
 - + Use active voice instead of passive voice: Do this, do that.
 - + Use exact and consistent names (of forms, fields, etc).
- Characteristics of a test case:
 - + Accurate: Exacts the purpose.
 - + Economical: No unnecessary steps or words.
 - + Traceable: Capable of being traced to requirements.
 - + Repeatable: Can be used to perform the test over and over.
 - + Reusable: Can be reused if necessary.

The test case for our group members is as shown below:

	Sprint 1									
1	User story	Case	Type	Input	Action	Expected Result	Actual Result	Status	Date	Note
2	As a tutor, I want to create group chat so that I can chat with my students	Send message	Normal	"Hello"	Enter desired message into text box and click on the send button	Store the message into the database and send it to the end-user(s) in real-time	Store the message into the database and send it to the end-user(s) in real-time	Passed	21st of March, 2020	
3			Normal	"Assignment.pdf"	Click on upload button, select file and send	Upload file to the server and send to end-user(s) in real time	File wasn't uploaded	Failed	30th of March, 2020	
4			Normal	"Screenshot.png"	Click on upload button, select file and send	Upload file to the server and send to end-user(s) in real time	File wasn't uploaded	Failed	30th of March, 2020	
5			Validation	"GTA V.exe"	Click on upload button, select file and send	Reject the file and display an error message	Reject the file and display an error message	Passed	30th of March, 2020	
6		Delete message	Normal	N/A	Right click on the message to be deleted and delete it	Remove the message from the conversation	Remove the message from the conversation	Passed	30th of March, 2020	
7		View message	Normal	N/A	Click on a conversation	Display all messages in the conversation	Display all messages in the conversation	Passed	21st of March, 2020	
8		Login	Normal	Username: "user" Password: "1234"	Enter a correct data into the login form	Log the user into the system and re-direct them to the homepage	Log the user into the system and re-direct them to the homepage	Passed	17th of March, 2020	
9			Validation	Username: "user" Password: "1234"	Enter an incorrect data into the login form	Display an error message	Display an error message	Passed	17th of March, 2020	
10		Logout	Normal	N/A	Click on the logout button	Log the user out of the system and re-direct them to the homepage	Log the user out of the system and re-direct them to the homepage	Passed	17th of March, 2020	
11		Reset password	Normal	Email: "1234@gmail.com"	Click on reset password and enter a valid email	Send a reset password token link to the entered email	Send a reset password token link to the entered email	Passed	18th of March, 2020	
12			Validation	Email: "1234@gmail.com"	Click on reset password and enter an invalid email	Display an error message	Display an error message	Passed	18th of March, 2020	

Figure 96: Test Cases for Sprint 1

Sprint 1

- **Send message:** we had a test for sending messages, what type of messages could be sent such as text, files (pdf, exe... which all failed), Text are allowed. Deleting and viewing messages which passed the test.
- **Login:** In the same sprint we also made test validations for basically login, which are passwords and usernames this is very important because we want in the space bar for username words only and not numbers while the password space bar accepts numbers alone, the validation is to check and display an error if any of these are violated.

- **Logout:** this is just to test the functionality of the button and to do this test, when you are logged into the system located and click the logout button to logout of the system.
- **Reset password:** Lastly on the first sprint, if the reset button is clicked and entered a valid mail, it sends what is required which is the ability to reset the password if the email is valid, the validation is to check if the mail is valid mail and if not valid displays an error.

	A	B	C	D	E	F	G	H	I	J
24	Sprint 3									
25	User story	Case	Type	Input	Action	Expected Result	Actual Result	Status	Date	Note
26	As a Administrator, I want to allocate or reallocate personal tutors to students so that Students can have personal tutors	Allocate Students to Tutors	Normal	N/A	Click on allocate students to tutor button, choose a tutor, and then choose less than or equal to 10 students	Allocate the chosen students to the chosen tutor and send an email to each and every one of them	Allocate the chosen students to the chosen tutor and send an email to each and every one of them	Passed	9th of April, 2020	
27			Validation	N/A	Click on allocate students to tutor button, choose a tutor, and then choose more than 10 students	Display an error message	Display an error message	Passed	9th of April, 2020	
28		Display all allocated students or tutors	Normal	N/A	Click on view users, select the option for viewing allocated students or tutors	Display all allocated students or tutors in a tabular form	Display all allocated students or tutors in a tabular form	Passed	9th of April, 2020	
29			Normal	N/A	Click on view users, select the option for viewing unallocated students or tutors	Display all unallocated students or tutors in a tabular form	Display all unallocated students or tutors in a tabular form	Passed	9th of April, 2020	
30			Normal	N/A	Click on view users, select the option for viewing unallocated students or tutors	Display all unallocated students or tutors in a tabular form	Display all unallocated students or tutors in a tabular form	Passed	9th of April, 2020	
31		Create students or tutors accounts	Normal	N/A	Fullscreen: "Israel Obanjesu" Email: "israel@gmail.com" Role: "Admin" Description: "" Picture: "profile.png"	Navigate to the add user page and fill the form with appropriate information	Store the provided information into the database and display a successful message	Passed	12th of April, 2020	
32			Validation	N/A	Fullscreen: "Israel Obanjesu" Email: "israel@gmail.com" Role: "" Description: "" Picture: "profile.png"	Navigate to the add user page, fill the form and leave out important information such as role	Display an error message	Display an error message	Passed	12th of April, 2020
33			Normal	N/A	Fullscreen: "Israel Obanjesu"					
34	As a Administrator, I want to deactivate/delete a user so that users don't lurk around when they are no longer needed	Deactivate a user	Normal	N/A	Click on "Deactivate" button for the user	Deactivate the user and hide them from other users in the application	Deactivate the user and hide them from other users in the application	Passed	12th of April, 2020	
35		Activate a user	Normal	N/A	Click on "Activate" button for the user	Activate the user and make them visible to other users in the application	Activate the user and make them visible to other users in the application	Passed	12th of April, 2020	
36		Delete a user	Normal	N/A	Click on "Delete" button for the user	Remove the user information from the database and the application	Remove the user information from the database and the application	Passed	12th of April, 2020	
37	As a Tutor, I want to create meetings so that I can arrange and record meetings with my students	Create and request for a meeting	Normal	Title: "Introduction" Description: "Know more about each other" Date: "13/04/2020" Members: "Israel, Cuong"	Click on a date in the calendar and fill in the appropriate information	Send a meeting request to the members' email	-----	Incomplete		
38		Postpone a meeting	Normal	Date: "15/04/2020"	Click on a meeting and select postpone	Postpone the meeting and inform the members of the new date via email	-----	Incomplete		
39		Cancel a meeting	Normal	Reason: "Change of Mind"	Click on a meeting and select cancel	Cancel the meeting and inform the members of the reason via email	-----	Incomplete		
40										
41										

Figure 97: Test Cases for Sprint 3

Sprint 3

- **Allocate student to tutors:** This just shows how a tutor can be allocated to student has no validation. It allows the allocation of 10 or less student to a tutor and the absent

validation is to check if the student are more than 10 but the validation is not applicable in this situation so no error displayed.

- **Display allocated/unallocated student:** Secondly the test also shows how display all allocated or unallocated students or tutors.
- **Created student or tutors account:** In the second half of the third sprint, it shows a normal and validation test type, the normal accepts data into “full name, email, role, description, pictures” the use of the validation is to make sure the correct data in being inputted correctly in the right spaces.
- **Deactivate a user:** This has no input, this test is make sure a user is deactivated alongside hiding them from other users in the application
- **Activate a user:** This is a successful test which when clicked makes a deceased user visible in the application, this can be archived by clicking the activate button
- **Delete a user:** This is also a successful test, this removes user information from the application and the database.

Sprint 4

Sprint 4									
User Story	Case	Type	Input	Action	Expected Result	Actual Result	Status	Date	Note
As a Tutor, I want to create meetings so that I can arrange and record meetings with my students	Create and request for a meeting	Normal	Title: "Introduction" Description: "Know more about each other" Date: "13/04/2020" Members: "Israel, Cuong"	Click on a date in the calendar and fill in the appropriate information	Send a meeting request to the members' email	-----	In-Progress		
	Postpone a meeting	Normal	Date: "15/04/2020"	Click on a meeting and select postpone	Postpone the meeting and inform the members of the new date via email	-----	In-Progress		
	Cancel a meeting	Normal	Reason: "Change of Mind"	Click on a meeting and select cancel	Cancel the meeting and inform the members of the reason via email	-----	In-Progress		
As a Student, I want to create issues / requests so that my tutor can view my issues / requests	Create issue	Normal	Title: "Issue 1" Description: "Issue Desc" Files: "assignment.docx" Participants: "Israel, Cuong"	Navigate to the create issue page and fill the form with the appropriate information	Store the issue information into the database and display a successful message	Store the issue information into the database and display a successful message	Passed	19th of April, 2020	
	Validation		Title: "" Description: "" Files: "assignment.docx" Participants: ""	Navigate to the create issue page, fill out the form and leave out important information	Display an error message	Display an error message	Passed	19th of April, 2020	
	View issues	Normal	N/A	Navigate to the issues page	Display all the issues that the user is involved in	Display all the issues that the user is involved in	Passed	19th of April, 2020	
	Edit issue	Normal	Title: "Issue 1" Description: "Issue Desc" Files: "assignment.docx" Participants: "Israel, Cuong"	Navigate to the issues page and click on the edit icon for the issue to be edited, and then update the form data	Update the issue information and display a successful message	Update the issue information and display a successful message	Passed	20th of April, 2020	
	Validation		Title: "" Description: "" Files: "assignment.docx" Participants: ""	Navigate to the issues page and click on the edit icon for the issue to be edited, and then update the form data and leave out important information	Display an error message	Display an error message	Passed	20th of April, 2020	
	Delete issue	Normal	N/A	Navigate to the issues page and click on the delete icon for the issue to be deleted	Delete the issue and remove it from the website	Delete the issue and remove it from the website	Passed	20th of April, 2020	
As a Tutor, I want to comment on students' issues / requests so that my students can view my comments	Create comment	Normal	Comment: "Example"	Click on an issue, and fill in the comments form	Store the comment in the database and display it on the website	Store the comment in the database and display it on the website	Passed	22nd of April, 2020	
	View comments	Normal	N/A	Click on an issue	Display the issue details and the comments attached to it	Display the issue details and the comments attached to it	Passed	22nd of April, 2020	
	Delete comment	Normal	N/A	Click on an issue, and delete a comment	Remove the comment from the database and remove it from the website	Remove the comment from the database and remove it from the website	Passed	22nd of April, 2020	

Figure 98: Test Cases for Sprint 4

- **Create/View/Edit issues:** this function help student report their issues to their tutor so tutor can know what struggle student is facing right now, from there they can help student to handle their problem
- **Delete issues:** this when the problem has been solved so they can delete it, make room for another issues in future
- **Create/View comments:** this help tutor can answer question, problem or issues that student have
- **Delete comment:** this when tutor is not satisfied with their answer so they can delete it and create a new one for their student

Sprint 5

Sprint 5									
User Story	Case	Type	Input	Action	Expected Result	Actual Result	Status	Date	Note
As a tutor, I want to create group chat so that I can chat with my students	Send message	Normal	"Hello"	Enter desired message into text box and click on the send button	Store the message into the database and send to the end-user(s) in real-time	Store the message into the database and send to the end-user(s) in real-time	Passed	2nd of May, 2020	
		Normal	"Assignment.pdf"	Click on upload button, select file and send	Upload file to the server and send to end-user(s) in real time	Upload file to the server and send to end-user(s) in real time	Passed	2nd of May, 2020	
		Normal	"Screenshot.png"	Click on upload button, select file and send	Upload file to the server and send to end-user(s) in real time	Upload file to the server and send to end-user(s) in real time	Passed	2nd of May, 2020	
		Validation	"GTA V.exe"	Click on upload button, select file and send	Reject the file and display an error message	Reject the file and display an error message	Passed	2nd of May, 2020	
	Delete message	Normal	N/A	Right click on the message to be deleted and delete it	Remove the message from the conversation	Remove the message from the conversation	Passed	2nd of May, 2020	
	View message	Normal	N/A	Click on a conversation	Display all messages in the conversation	Display all messages in the conversation	Passed	2nd of May, 2020	
As a Tutor, I want to create meetings so that I can arrange and record meetings with my students	Create and request for a meeting	Normal	Title: "Introduction" Description: "Know more about each other" Date: "13/04/2020" Members: "Israel, Cuong"	Click on a date in the calendar and fill in the appropriate information	Send a meeting request to the members' email	Send a meeting request to the members' email	Passed	2nd of May, 2020	
	Postpone a meeting	Normal	Date: "15/04/2020"	Click on a meeting and select postpone	Postpone the meeting and inform the members of the new date via email	-----	Not Done		
	Cancel a meeting	Normal	Reason: "Change of Mind"	Click on a meeting and select cancel	Cancel the meeting and inform the members of the reason via email	-----	Not Done		
As a Student, I want to view my personal dashboard so that I can see a summary of my interaction with my tutor	View Student Information	Normal	N/A	Log in as a student and navigate to the dashboard page	Display all relevant and necessary information for the student	Display all relevant and necessary information for the student	Passed	2nd of May, 2020	
As a Tutor, I want to view my personal dashboard so that I can view, sort and filter my personal tubes	View Tutor Information	Normal	N/A	Log in as a tutor and navigate to the dashboard page	Display all relevant and necessary information for the tutor	Display all relevant and necessary information for the tutor	Passed	3rd of May, 2020	
As a Administrator, I want to receive notifications from students so that I can view students' complaints/requests	Receive notifications	Normal	N/A	Allocate students to a tutor	Send an email notification to the specified students and tutor	Send an email notification to the specified students and tutor	Passed	3rd of May, 2020	
		Normal	N/A	Create a meeting	Send an email notification to the organizer and the meeting members	Send an email notification to the organizer and the meeting members	Passed	3rd of May, 2020	
	Responsiveness	Normal	N/A	Open the website	Display the website properly across all screen resolutions	Display the website properly across all screen resolutions	Passed	4th of May, 2020	

Figure 99: Test Cases for Sprint 5

- **Send/View messages:** this help tutor to make a group chat with student, in here tutor can answer question for everyone, and other student if having the same problem can know what to do.
- **Delete message:** this help if, student or tutor sending the wrong message.
- **Create/Postpone meeting:** this help tutor can arrange a meeting with their student in order to have a direct conversation and to know student's progress.

- **Cancel meeting:** this help when tutor has unexpected work that they have to cancel the meeting.

7.3 Test Evidence

7.3.1 Test for sending messages

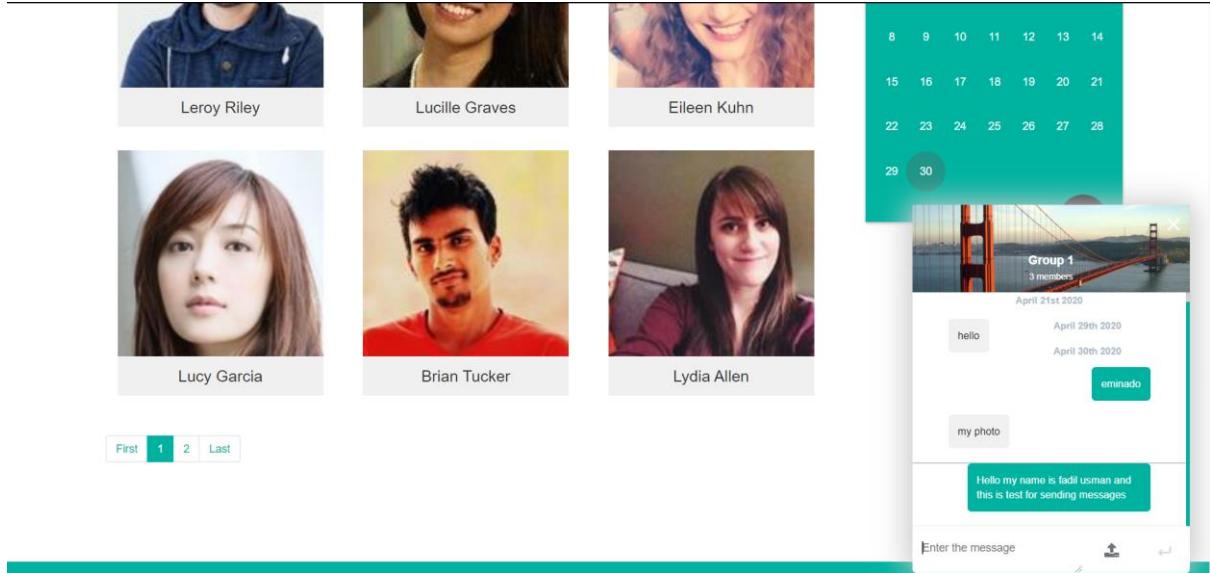


Figure 100 Test for sending messages

In the above picture, the test is for sending messages, typing messages in the chat box and from the above image the test is successful. However, I tried uploading a pdf file but failed to upload.

7.3.2 Test for deleting messages

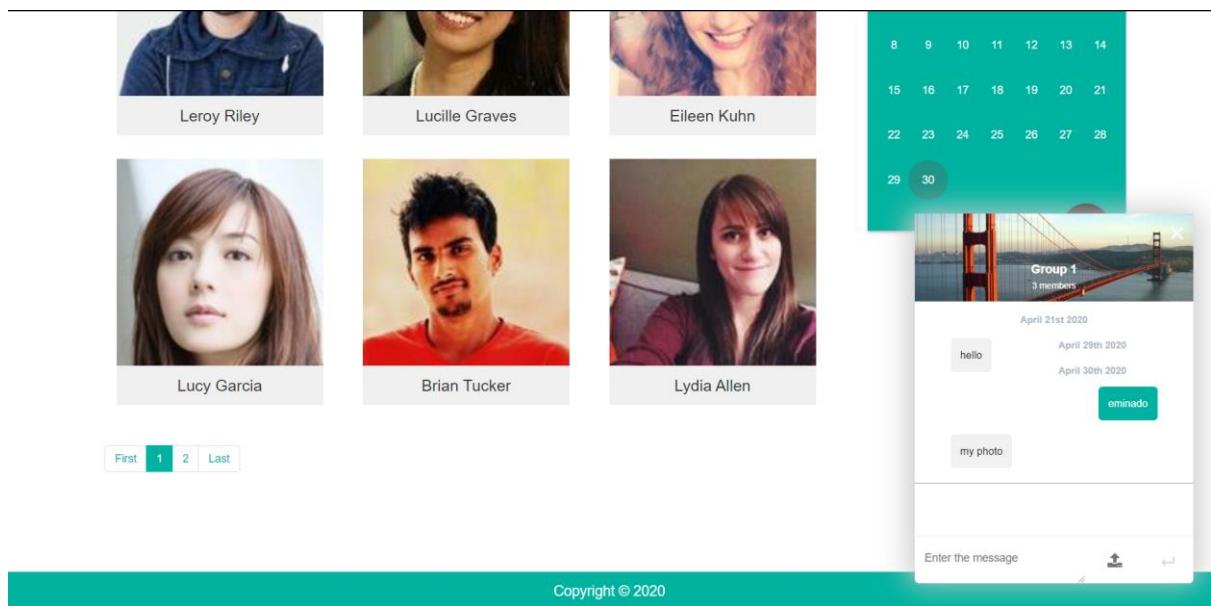


Figure 101 test for delete messages

The above images show a successful test for deleting messages. The present text in the previous images has been deleted.

7.3.3 Test for login with correct input data



Figure 102 Test for login

The above picture is for the login test which was successful, the by the top most right hand side of the screen the logout button is visible.

7.3.4 Test for login with wrong input data (validation)

The screenshot shows the E-Tutor login interface. At the top, there are two error messages in red boxes: "Error: Passwords do not match" and "Error: Could not find a user with that email". Below these, the login form has fields for "E-Mail Address" (1234@gmail.com) and "Password" (four dots). There is a "Remember Me" checkbox, a "Login" button, and a "Forgot Your Password?" link. A "OR" button with a Google icon is also present.

Figure 103 Test for login

The test failed because the email and password used where incorrect and are numbers so it threw an error.

7.3.5 Test for logout



Figure 104 Test for logout

When the logout button is clicked, it changes to login on the top most right hand part of the screen.

7.3.6 Test for reset password.

The screenshot shows the E-Tutor forgot password page. At the top, there is a green success message: "Success: We've received your request. Please check your email to proceed further". Below this is a "Forgot Password" section with an "E-Mail Address" field containing draknoe@gmail.com and a "Submit" button. There is also a "Back To Login" link.

Figure 105 Test for forget password

The above test for resetting password, the message displayed is a successful message because the email is valid.

7.3.7 Test for reset password invalid mail (validation)

The screenshot shows a teal header bar with the text "E-Tutor". Below it is a modal window titled "Forgot Password". Inside the modal, there is a red error message box containing the text "Error: Could not find user with that email!". Below the error message is a form field labeled "E-Mail Address" with the value "1234@gmyt". At the bottom of the modal are two buttons: "Submit" and "Back To Login".

Figure 106 test for rest password

Above is a picture, which shows the test is incorrect because the email entered was invalid.

7.3.8 Test for allocation of students to tutor

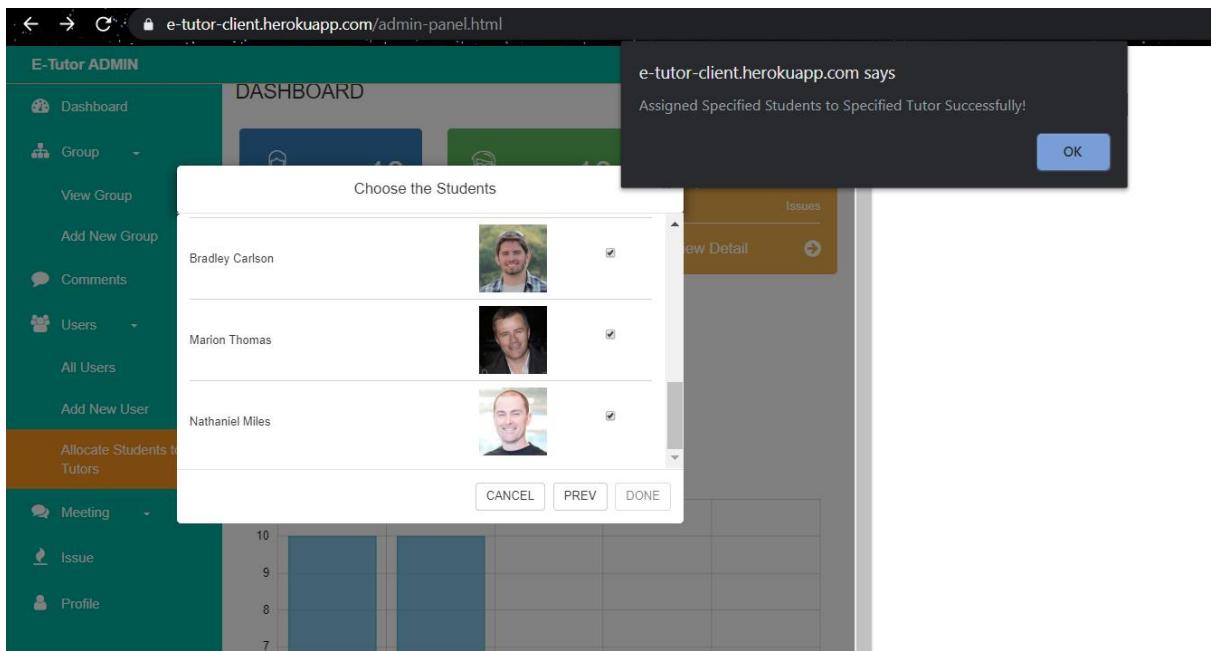


Figure 107 Allocation student to tutor

The above images shows that 3 student were allocated to a selected tutor and no error because a tutor can only take ten or less.

- Test for allocated student to tutors (validation)

7.3.9 Test for display all allocated students

No	Name	Email	Tutor	Role	DEACTIVATE	VIEW USER CARD
1	Pedro Ortiz	pedro.ortiz@example.com	Stacy Smith	student	DEACTIVATE	VIEW USER CARD
2	Tanya Hoffman	tanya.hoffman@example.com	Stacy Smith	student	DEACTIVATE	VIEW USER CARD
3	Timothy Harper	timothy.harper@example.com	Stacy Smith	student	DEACTIVATE	VIEW USER CARD
4	Jackson Lane	jackson.lane@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD
5	Meghan Garcia	meghan.garcia@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD
6	Priscilla Hudson	priscilla.hudson@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD
7	Pamela Oliver	pamela.oliver@example.com	Lucille Graves	student	ACTIVATE	VIEW USER CARD
8	Bradley Carlson	bradley.carlson@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD
9	Marion Thomas	marion.thomas@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD
10	Nathaniel Miles	nathaniel.miles@example.com	Stacy Smith	student	ACTIVATE	VIEW USER CARD

Figure 108 All Allocated students

The above images is shows all allocated, student to view this click on users, click on all users, click on show all allocated students.

7.3.10 Test for display all allocated tutors

No	Name	Email	Tutor	Role	DEACTIVATE	VIEW USER CARD
1	Leroy Riley	leroy.riley@example.com	N/A	tutor	DEACTIVATE	VIEW USER CARD
2	Lucille Graves	lucille.graves@example.com	N/A	tutor	DEACTIVATE	VIEW USER CARD
3	Stacy Smith	stacy.smith@example.com	N/A	tutor	DEACTIVATE	VIEW USER CARD

Figure 109 all allocated tutors

The above images is shows all allocated, tutor to view this click on users, click on all users, click on show all allocated tutor.

7.3.11 Test for displaying unallocated tutor

The screenshot shows the 'Users' section of the E-Tutor ADMIN interface. The left sidebar has a teal background with various icons and links: Dashboard, Group, Comments, Users (selected), Meeting, Issue, and Profile. The main area is titled 'Users' and has a dropdown menu 'Show All Unallocated Tutors'. A table lists 7 unallocated tutors:

No	Name	Email	Tutor	Role	Action	Action
1	Eileen Kuhn	eileen.kuhn@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
2	Lucy Garcia	lucy.garcia@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
3	Brian Tucker	brian.tucker@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
4	Lydia Allen	lydia.allen@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
5	Marion Terry	marion.terry@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
6	Pamela Fields	pamela.fields@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>
7	Savannah Reid	savannah.reid@example.com	N/A	tutor	<button>DEACTIVATE</button>	<button>VIEW USER CARD</button>

Figure 110 all unallocated tutors

The above image shows unallocated tutors.

7.3.12 Test for creating students.

The screenshot shows the 'Add User' form. At the top, a green success message says 'Success: User Created Successfully. View Users?'. The form fields are as follows:

- Full Name: fadi usman idris
- Email: kheengsmoke03@gmail.com
- Role: Student
- Select a Role: (empty)
- Picture: Choose File (capture2.PNG)
- Select a File: (empty)

At the bottom is a blue 'ADD NEW USER' button.

Figure 111: Test for creating students

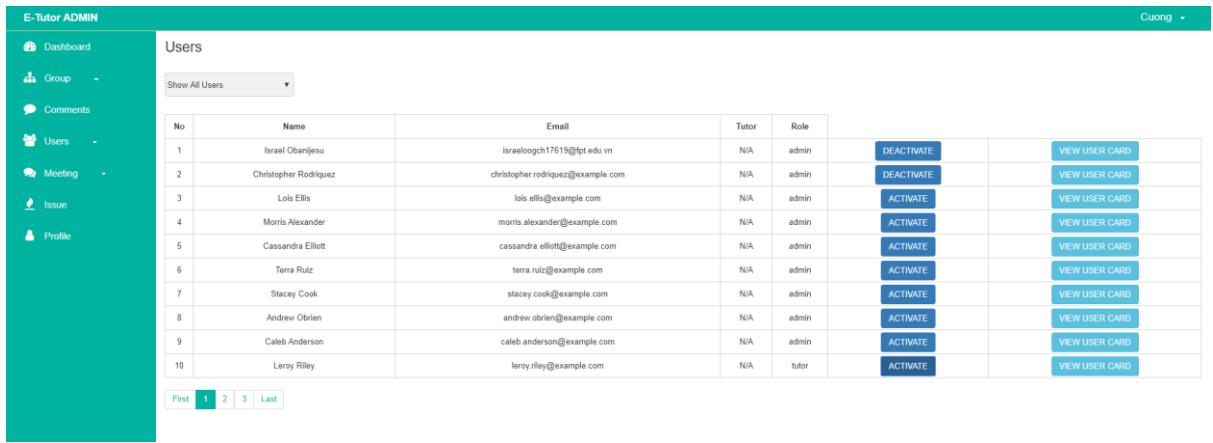
7.3.13 Test for creating students(validation)

The screenshot shows the 'Add User' form with validation errors:

- Full Name: kheengsmoke03@gmail.com
- Email: fadi12 (with validation error: Please match the requested format)
- Role: Admin
- Select a Role: (empty)
- Picture: Choose File (No file chosen)
- Select a File: (empty)

Figure 112: Test for creating students (validation)

7.3.14 Test for deactivating user

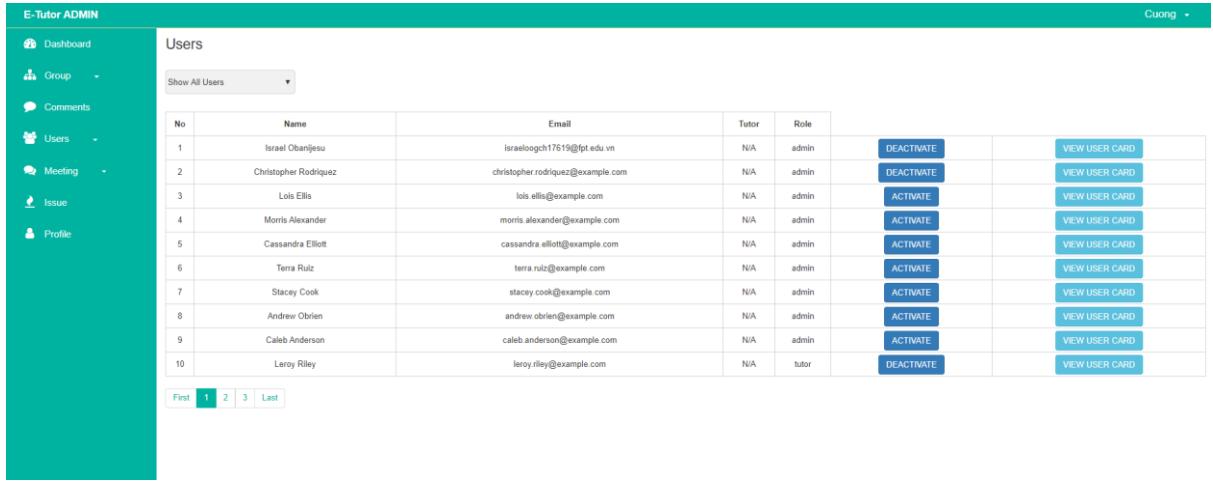


No	Name	Email	Tutor	Role	DEACTIVATE	VIEW USER CARD
1	Israel Obanjesu	israeloogch17619@fpt.edu.vn	N/A	admin	DEACTIVATE	VIEW USER CARD
2	Christopher Rodriguez	christopher.rodriguez@example.com	N/A	admin	DEACTIVATE	VIEW USER CARD
3	Lois Ellis	lois.ellis@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
4	Morris Alexander	morris.alexander@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
5	Cassandra Elliott	cassandra.elliott@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
6	Terra Ruiz	terra.ruiz@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
7	Stacey Cook	stacey.cook@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
8	Andrew Obrien	andrew.obrien@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
9	Caleb Anderson	caleb.anderson@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
10	Leroy Riley	leroy.riley@example.com	N/A	tutor	ACTIVATE	VIEW USER CARD

Figure 113 deactivate user

The above images are shows list of deactivated user's line 10 was just deactivated.

7.3.15 Test for activating user



No	Name	Email	Tutor	Role	DEACTIVATE	VIEW USER CARD
1	Israel Obanjesu	israeloogch17619@fpt.edu.vn	N/A	admin	DEACTIVATE	VIEW USER CARD
2	Christopher Rodriguez	christopher.rodriguez@example.com	N/A	admin	DEACTIVATE	VIEW USER CARD
3	Lois Ellis	lois.ellis@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
4	Morris Alexander	morris.alexander@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
5	Cassandra Elliott	cassandra.elliott@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
6	Terra Ruiz	terra.ruiz@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
7	Stacey Cook	stacey.cook@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
8	Andrew Obrien	andrew.obrien@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
9	Caleb Anderson	caleb.anderson@example.com	N/A	admin	ACTIVATE	VIEW USER CARD
10	Leroy Riley	leroy.riley@example.com	N/A	tutor	ACTIVATE	VIEW USER CARD

Figure 114 Test for activate user

The test for the active user is successful as show in line 10 to do this click on users, click on all user and click on activate.

7.3.16 Test for creating and requesting for a meeting

The screenshot shows the 'Add New Meeting' form. The left sidebar has 'Dashboard' selected. The main form has a green success message bar at the top. The 'Title' field contains 'project check'. The 'Description' field contains 'project checker'. The 'Date' field shows '05/09/2020 12:00'. There is a 'Select Participants' button and a 'ADD NEW MEETING' button.

Figure 115: Test for creating and requesting for a meeting

7.3.17 Test for creating a issue

The screenshot shows the 'Add Issue' form. The left sidebar has 'Issues' selected. The main form has a green success message bar at the top. The 'Title' field contains 'reportcheck'. There is a 'DESCRIPTION' button.

Figure 116: Test for creating an issue

A successful test for issue creation

7.3.18 Test for viewing issue details

The screenshot shows the 'Issue Details' page for the issue 'reportcheck'. It includes the title 'REport', creator 'Cuong', participants 'Christopher Rodriguez', and date 'May 4th 2020, 11:34:22 pm'. Below this is a PDF icon. The 'Comments' section has a 'Leave a Comment' input field and a 'SUBMIT' button.

Figure 117: Test for viewing issue details

A successful test for test of viewing issue.

7.3.19 Test for editing issue

The screenshot shows the 'Edit Issue' form. It includes fields for 'Title' (containing 'REport'), 'Description' (containing 'report checkz'), and a file upload area ('Select File' with 'Choose Files' and 'No file chosen'). There is also a 'Select Participants' button and a prominent green 'UPDATE ISSUE' button at the bottom.

Figure 118: Test for editing issue

A successful test for editing issue, after the update button is clicked, it load back to the view issue page with the updated contents

7.3.20 Test for deleting issue

The screenshot shows the 'All Issues' table. A modal dialog box displays the message 'Issue Deleted Successfully!'. The table lists five issues with columns for 'No', 'Title', 'Description', 'Author', 'Date', and actions ('View', 'Edit', 'Delete').

No	Title	Description	Author	Date	Action	Action	Action
1	REport	report checkz	Cuong	May 4th 2020, 11:34:22 pm			
2	Issue 5	Issue 5 Description	Israel Obanjewu	May 3rd 2020, 3:49:47 am			
3	Issue 3	Description: This is...	Israel Obanjewu	April 29th 2020, 11:14:43 pm			
4	Issue 2	This is an issue wit...	Israel Obanjewu	April 28th 2020, 11:06:30 pm			
5	Issue 1	Issue 1 Description	Israel Obanjewu	April 20th 2020, 7:42:23 pm			

Figure 119: Test for deleting issue

When the delete is pressed, it shows a feedback message from the system saying the text is deleted, it is a successful test for delete issue.

7.3.21 Test for creating comment

The screenshot shows a web interface for creating a comment. At the top, there is a section titled "Issue 5 Description" with two PDF file attachments. Below this is a "Comments" section where a user named Cuong has posted a comment. The comment text is "a very good and well detail PDF file". There are edit and delete buttons next to the comment. Below the comments section is a "Leave a Comment" input field with a placeholder "Comment...". A green "SUBMIT" button is located at the bottom of the input field.

Figure 120: Test for creating comment

A comment has been created and it is successful test

7.3.22 Test for viewing comment

The screenshot shows a web interface for viewing comments. The left sidebar includes links for Dashboard, Group, Comments, Users, Meeting, Issue, and Profile. The main area is titled "Comments" and displays a table of comments. The table has columns for No., Comment, Author, Post / Issue, Date, and three red "Delete" buttons. The data in the table is as follows:

No	Comment	Author	Post / Issue	Date	
1	Comment 3	Cuong	Issue 1	May 2nd 2020, 5:45:42 pm	<input type="button" value="Delete"/>
2	Comment 4	Cuong	Issue 1	May 2nd 2020, 5:45:48 pm	<input type="button" value="Delete"/>
3	Comment 5	Leroy Riley	Issue 2	May 2nd 2020, 6:02:00 pm	<input type="button" value="Delete"/>
4	a very good and well detail PDF file	Cuong	Issue 5	May 4th 2020, 11:59:54 pm	<input type="button" value="Delete"/>

Figure 121: Test for viewing comment

A successful test for viewing comment as shown above

7.3.23 Test for deleting comment

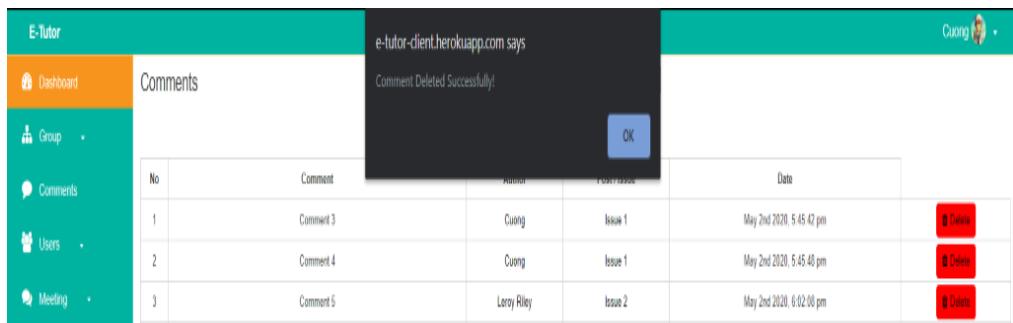


Figure 122: Test for deleting comment

This is a successful test for deleting a comment, the system sends a feedback to the user indicating the comment has been deleted.

8 DEVELOPMENT PROCESS

8.1 User Stories

Getting User stories is the one of the most important step in agile methodology because it provide a very good way to define product with clarity. And with a well-defined, it will help business analyst, scrum master articulate the functionality of product without implementation details and technicalities.

id	as a/an	i want to...	so that...
1	Administrator	Allocate or reallocate personal tutors to students	Students can have personal tutors
2	Student	Send messages to my tutors	I can chat with my tutors
3	Tutor	Create a group chat	I can chat with my students
4	Tutor	Create request for meetings	I can arrange and record meetings with my students
5	Student	Create Issues / Requests	My tutor can view my issues / requests
6	Tutor	Comment on students' issues / requests	My students can view my comments
9	Tutor	Receive email notifications	I will be informed when a student is allocated to me
10	Student	View my personal dashboard	I can see a summary of my interaction with my tutor
11	Tutor	View my personal dashboard	I can view, sort and filter my personal tutees
12	Administrator	Restrict user access to certain pages and features	Users can only see what they are supposed to
13	Administrator	Deactivate/Delete a student/staff	Users don't lurk around when they are no longer needed
14	Administrator	Receive notifications from students	I can view students' complaints/requests
15	User	Login	I can have access to the application
16	Administrator	Create students and tutor accounts	Students and tutors can login with the account details
17	User	Access the application interface from all devices (mobile phones, tablets, desktops)	Use the application interface on all devices (mobile phones, tablets, desktops)

Figure 123: User Stories

8.2 Product Backlog

Product backlog is ordered list of tasks that is known to be needed in the product and it informs how sprints will be planned. It is a simple to-do list which each item in product backlog is prioritized and is estimated.

Without an organized product backlog, the team will be very difficult to define which task is high priority, and may focus on some ideas are not valuable that will be a waste of time. There are some advantages of product backlog such as: simple to prioritize, allows to order and immediately see dependencies, allows team to think about products in the long-term.

Forming product backlogs:

- Getting User stories and User requirement: User stories were get by meeting and discussion with the customer

- For each items in user stories, provide an estimation of how much time, effort is involved
- Rank the product backlog, determining the priority order of each task, an item will not be added to the backlog if that item has no value in prioritizing
- Sprint planning: Move all items onto scrum board. Determine each item can be added to sprints
- The product backlog is a living-document so that mean our team need to update the process of the each task when they do it day by day by changing the status of each task.

Product backlog prioritization:

- **Must have:** Product can't deliver on target without this item, not legal or unsafe without it, or project is not valuable without this.
- **Should have:** Important but not vital, maybe painful to leave out but the project still viable or may need some kind of workaround
- **Could have:** Desirable but not as important as Should have, only do if there is extra time

id	as a/an	i want to...	so that...	priority	hours	added in...
1	Administrator	Allocate or reallocate personal tutors to students	Students can have personal tutors	Must Have	80	Sprint 3
2	Student	Send messages to my tutors	I can chat with my tutors	Could Have	90	Sprint 1
3	Tutor	Create a group chat	I can chat with my students	Could Have	100	Sprint 1
4	Tutor	Create request for meetings	I can arrange and record meetings with my students	Should Have	80	Sprint 3, Sprint 4
5	Student	Create Issues / Requests	My tutor can view my issues / requests	Must Have	80	Sprint 4
6	Tutor	Comment on students' issues / requests	My students can view my comments	Must Have	60	Sprint 4
9	Tutor	Receive email notifications	I will be informed when a student is allocated to me	Could Have	60	Sprint 4
10	Student	View my personal dashboard	I can see a summary of my interaction with my tutor	Should Have	20	
11	Tutor	View my personal dashboard	I can view, sort and filter my personal tutees	Should Have	20	
12	Administrator	Restrict user access to certain pages and features	Users can only see what they are supposed to	Could Have	40	
13	Administrator	Deactivate/Delete a student/staff	Users don't lurk around when they are no longer needed	Could Have	50	Sprint 3, Sprint 4
14	Administrator	Receive notifications from students	I can view students' complaints/requests	Could Have	30	
15	User	Login	I can have access to the application	Must Have	10	Sprint 1
16	Administrator	Create students and tutor accounts	Students and tutors can login with the account details	Must Have	10	Sprint 3
17	User	Access the application interface from all devices (mobile phones, tablets, desktops)	Use the application interface on all devices (mobile phones, tablets, desktops)	Could Have	10	

Figure 124: Product Backlog

8.3 Sprints

8.3.1 Sprint 1

At this sprint, we start to make User stories, product backlog and draw wire frame of the website. Some function for back-end will be implemented on this sprint such as: Login and chatting feature.

In the end of this sprint, the front-end team has done the design homepage, design chat and login page but the home page still need to make some changes, all this design is yet to be connected to the server.

For the back-end team, Cuong is new with the technology so he cannot implement authentication alone, so he was asking for help and Israel has helped him to finish his task. The Chat feature partially done by Israel. But all the back-end side are yet to be connected to the Front-end.

User story	Details	Details	Tasks	Assigned to	Estimated Hours	Status	Note
-----			Setup project prerequisites	Israel	24	Done	
			Design wireframes	Israel, Cường	24	Done	
			Design homepage	Israel, Sơn	120	Partially done	Still need to make some changes on the homepage
	Questions	Answers	Design authentication	Israel, Tuấn	24	Partially done	The design is yet to be connected to the server
	What type(s) of login will users be able to use	Email and Password Only	Implement authentication	Israel, Cường	10	Partially done	It is yet to be connected to the design
	Will users be notified after they have logged in	No, they won't, it's not necessary					
	Will users be able to visit the login page after they have logged in	No, users would be prevented from trying to login more than once					
	Acceptance Test						
	Test that a user can successfully login with his / her correct details						
	Test that a user can logout successfully after they have logged in						
	Test that a user can request for a new password if they forgot their password						
	Questions	Answers	Design chat	Israel, Tuấn	24	Partially done	It is yet to be connected to the server
	What kind of messages can users send to each other	Users can send texts, images and PDF files only	Implement chat	Israel	60	Partially done	It is yet to be connected to the design
	How will users be notified after they've received a message	It will be added to the notification list on the user's homepage					
	Will the messaging system work in real-time	Yes					
	Is a student permitted to delete or edit his / her own message	Yes					
	What are the special permissions of a tutor in a group chat	They have no special permissions					
	Acceptance Test						
	Test that users can send texts, images and PDF files in real-time successfully						
	Test that users receive messages in real-time						
	Test that users can edit and delete their own messages successfully						
			Write Test Cases	Israel, Fadil	24	Partially done	Since the server and client has not connected, it is impossible to determine the actual result of each test case
			Write Unit Tests	Israel, Thai	24	Not done	Currently having some issues with writing unit tests
			Deploy	Israel	24	Not done	Since all features are partially done, we couldn't deploy the application

Figure 125: Sprint 1 Backlog

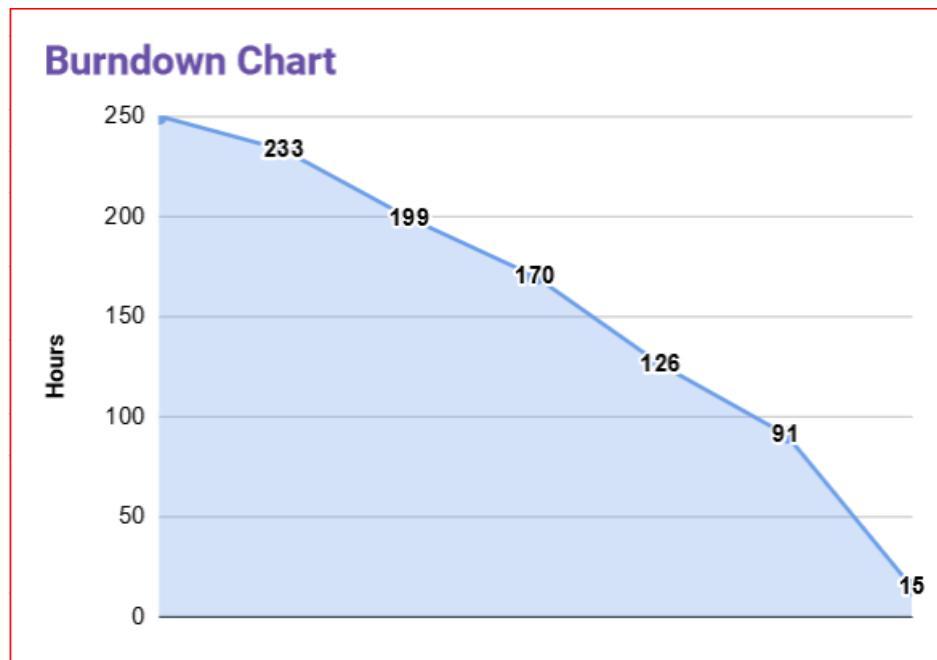


Figure 126: Burndown Chart Sprint 1

8.3.2 Sprint 2

In this sprint, we complete all the task has not been done from previous sprint and try to fix some bugs. We also connect all the sever-side features in sprint 1 to the client-side and also deploy all features of sprint to sever.

The front-end team design the reset, change password and redesign the chat.

All the task have been done well and are on time for this sprint.

User story	Tasks	Assigned to	Estimated Hours of Completion	Actual Hours of Completion	Status
	Complete the homepage design	Son	6	6	Done
	Connect all the server-side features in s	Israel	6	6	Done
	Design the reset and change password	Tuân	6	6	Done
	Redesign the chat	Tuân	6	6	Done
	Deploy all features of sprint 1	Israel	2	2	Done

Figure 127: Sprint 2 Backlog

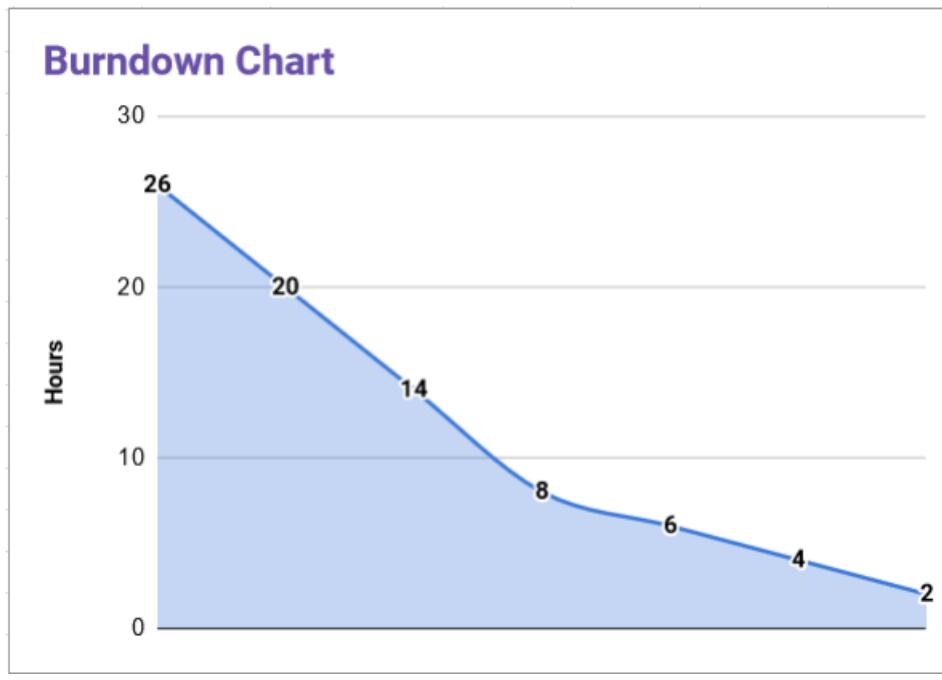


Figure 128: Burndown Chart For Sprint 2

8.3.3 Sprint 3

In this sprint, the front-end team will design the administrator page. This task has partially done but still need to fix some bugs and design more sub-page.

The back-end team need to:

- Implement allocation of students to tutors,
- Implement creation of students and tutors account,
- Implement deactivation and deletion of users account,
- Implement creation of schedules and announcements

At the end of this sprint, all the tasks for back-end team have been partially done, but only the creation of schedules and announcements function has not been done in this sprint due to limited time and unexpected work need to be done such as fixing bugs from the previous sprint.

Connecting all the sever-side logic to the front-end design and Deploy to server also has been done in this sprint.

User story	Details	Details	Tasks	Assigned to	Estimated Hours	Status
			Design administrator page	Son, Tuân	120	Partially Done
	Questions	Answers	Implement allocation of students to tutors	Israel, Cường	80	Done
As a Administrator, I want to allocate or reallocate personal tutors to students so that Students can have personal tutors	How many students can be allocated to one tutor at once How will tutors and students be notified after they have been allocated to each other Can a student be allocated to more than one tutors	Minimum of 1 and maximum of 10 students can be allocated to one tutor at once They will be notified via email No, each student can only have one personal tutor				
	Acceptance Test					
	Test that all unallocated and allocated people are displayed differently Test that students and tutors are notified via email after they have been allocated to each other					
	Questions	Answers	Implement creation of students and tutors account	Israel, Cường	10	Done
As a Administrator, I want to create students and tutor accounts so that students and tutors can login with the account details	How will students and tutors information be stored Is it possible to re-use information of a deactivated account to create a new account	Important information such as: name, username, email and password are required, but profile picture, and description are optional No, emails and usernames can only be used once				
	Acceptance Test					
	Test that students and tutors account are created successfully Test that there is no duplication of account information					
	Questions	Answers	Implement deactivation and deletion of users account	Israel	50	Partially Done
As a Administrator, I want to deactivate users	What is the difference between a deactivated user and a deleted user	A deactivated user is still in the system but only visible to the admin while a deleted user is completely removed from the system				
	Acceptance Test					
	Test that deactivated or deleted user cannot access the application Test that deactivated or deleted user are not visible to other users in the application					
	Questions	Answers	Implement creation of schedules and announcements	Israel	80	Not done
As a Tutor, I want to create meetings so that I can arrange and record meetings with my students	How will meetings be recorded Who can create meetings How will be users be notified when a meeting date is due	They will be logged into the database Only Tutors They will be notified via email				
	Acceptance Test					
	Test that tutors can arrange meetings successfully Test that users are notified via email when a meeting date is due					
			Write Test Cases Connect all the server-side logic to the front-end design and Deploy	Thai, Fadil Israel	24 24	Done Done

Figure 129: Sprint 3 Backlog

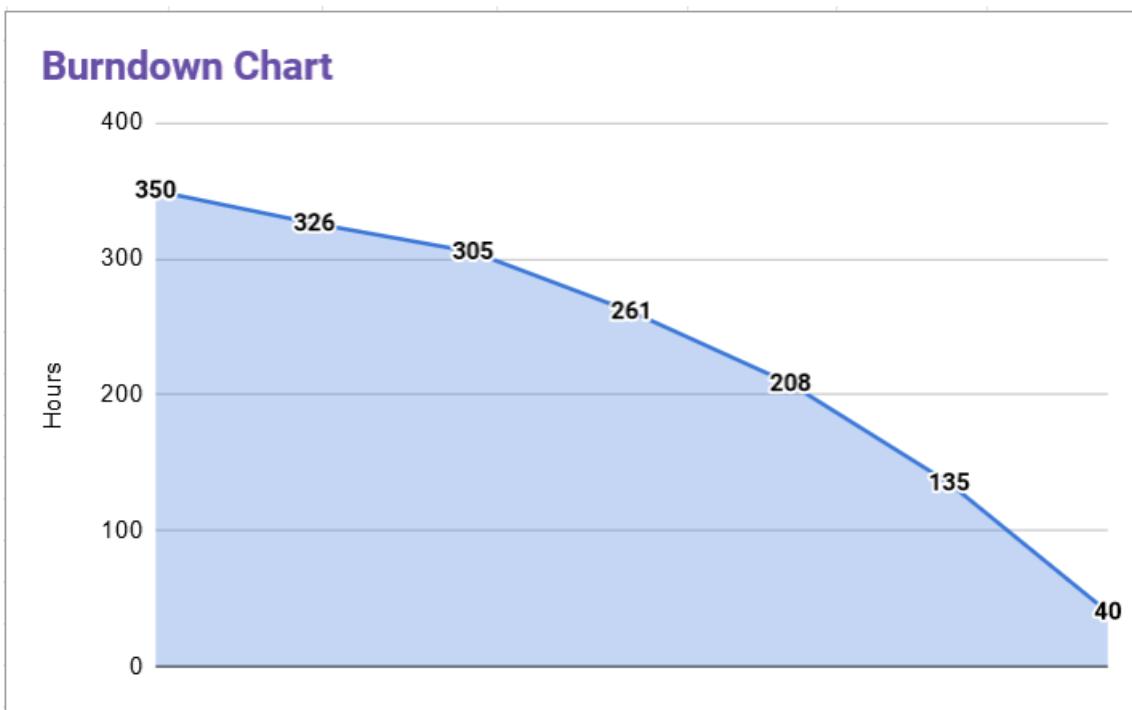


Figure 130: Burndown Chart For Sprint 3

8.3.4 Sprint 4

In the last sprint, our team focused on finishing the remaining features and testing the whole function of the website again after deploying all the features to the Server.

The front side has fixed bugs in the administrator page and designed more sub-pages for admin.

The back-end team need to:

- Implement receivable of notifications via email
- Implement creation of schedules and announcements
- Connect the deletion logic to the front-end design
- Implement comments on issues
- Implement creation and viewing of issues

User story	Details	Details	Tasks	Assigned to	Estimated Hours	Status
-----			Fix the bugs in administrator page and design more pages	Son, Tuấn	80	Done
	Questions	Answers	Implement creation and viewing of issues / requests	Israel, Cường	60	Partially Done
As a Student, I want to create issues / requests so that my tutor can view my issues / requests	How many students can be allocated to one tutor at once How will tutors and students be notified after they have been allocated to each other Can a student be allocated to more than one tutors	Minimum of 1 and maximum of 10 students can be allocated to one tutor at once They will be notified via email No, each student can only have one personal tutor				
	Acceptance Test					
	Test that all unallocated and allocated people are displayed differently Test that students can be allocated to tutors successfully Test that students and tutors are notified via email after they have been allocated to each other					
	Questions	Answers	Implement comments on issues / requests	Israel, Cường	10	Done
As a Tutor, I want to comment on students' issues / requests so that my students can view my comments	How will students and tutors information be stored Is it possible to re-use information of a deleted account to create a new account	Important information such as: name, username, email and password are required, but profile picture, and description are optional No, emails and usernames can only be used once				
	Acceptance Test					
	Test that students and tutors account are created successfully Test that there is no duplication of account information					
	Questions	Answers	Connect the deletion logic to the front-end design	Israel	50	Done
As a Administrator, I want to deactivate/delete a student/staff so that users don't lurk around when they are no longer needed	What is the difference between a deactivated student / staff and a deleted student / staff	A deactivated student / staff is still in the system but only visible to the admin while a deleted student / staff is completely removed from the system				
	Acceptance Test					
	Test that deactivated or deleted student / staff cannot access the application Test that deactivated or deleted student / staff are not visible to other users in the application					
	Questions	Answers	Implement receipt of notifications via email	Israel	30	Done
As a Tutor, I want to receive email notifications so that I will be informed when a student is allocated to me	Acceptance Test					
	Test that emails are sent to users successfully					
			Write Test Cases	Thái, Fadil	24	Done
			Connect all the server-side logic to the	Israel	12	Done

Figure 131: Sprint 4 Backlog

Burnown Chart

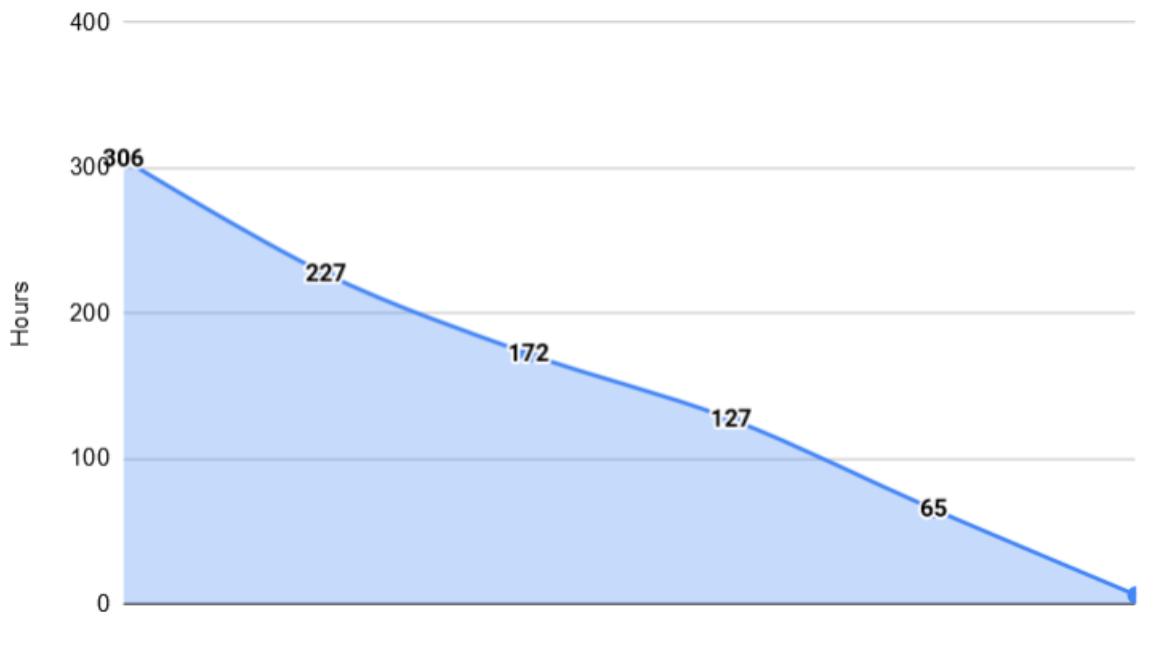


Figure 132: Burndown Chart For Sprint 4

8.3.5 Sprint 5

In the last sprint, our team focus on finishing remaining feature and re-test all test case has failed in previous sprints. At the end, we will deploy all the code to server.

The front-end team still need to fix some bugs in admin page and design blog page.

The tester need to test whole the website again, test that the website is responsive across all device resolutions. All the task has been done well and on time.

As a Administrator, I want to restrict user access to certain pages and features so that users can only see what they are supposed to	Questions	Answers	Implement Page and Data Authorization Logic	Israel	Done
	Acceptance Test				
As a Administrator, I want to receive notifications from students so that I can view students' complaints/requests	Questions	Answers	Implement receival of notifications via email	Israel	Done
	Acceptance Test				
As a User, I want to access the application interface from all devices (mobile phones, tablets, desktops) so that I can use the application interface on all devices (mobile phones, tablets, desktops)	Questions	Answers	Implement responsiveness	Son	Done
	Acceptance Test				
			Write Test Cases	Israel, Thai, Fadil	Done

Figure 133: Sprint 5

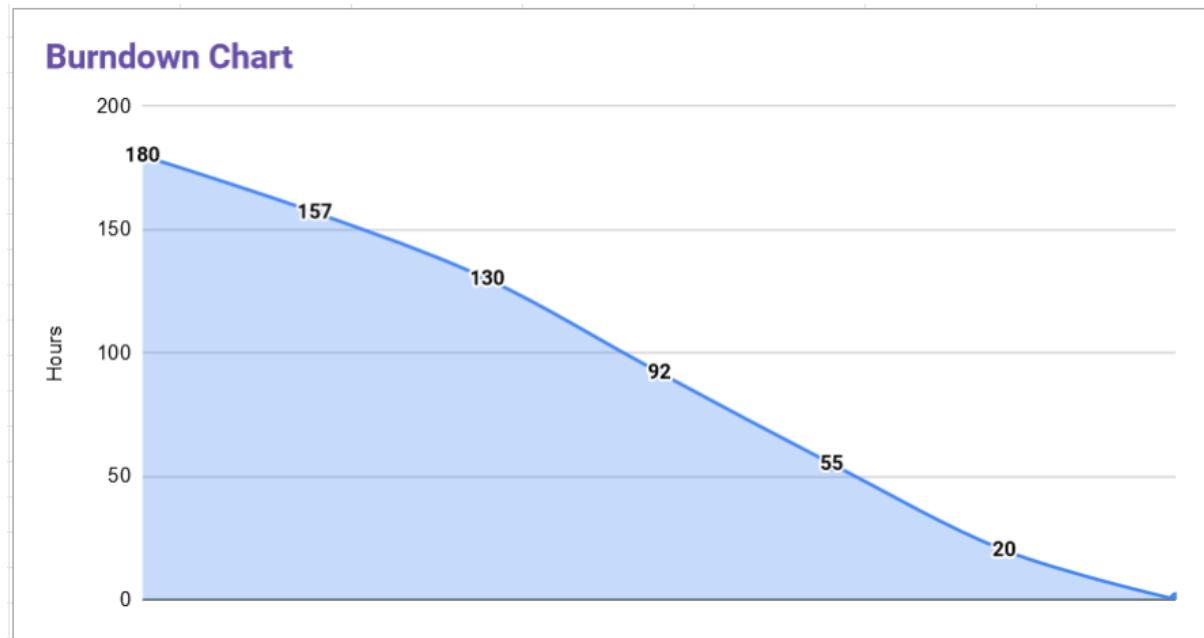


Figure 134: Burndown Chart For Sprint 5

8.4 Release Burndown

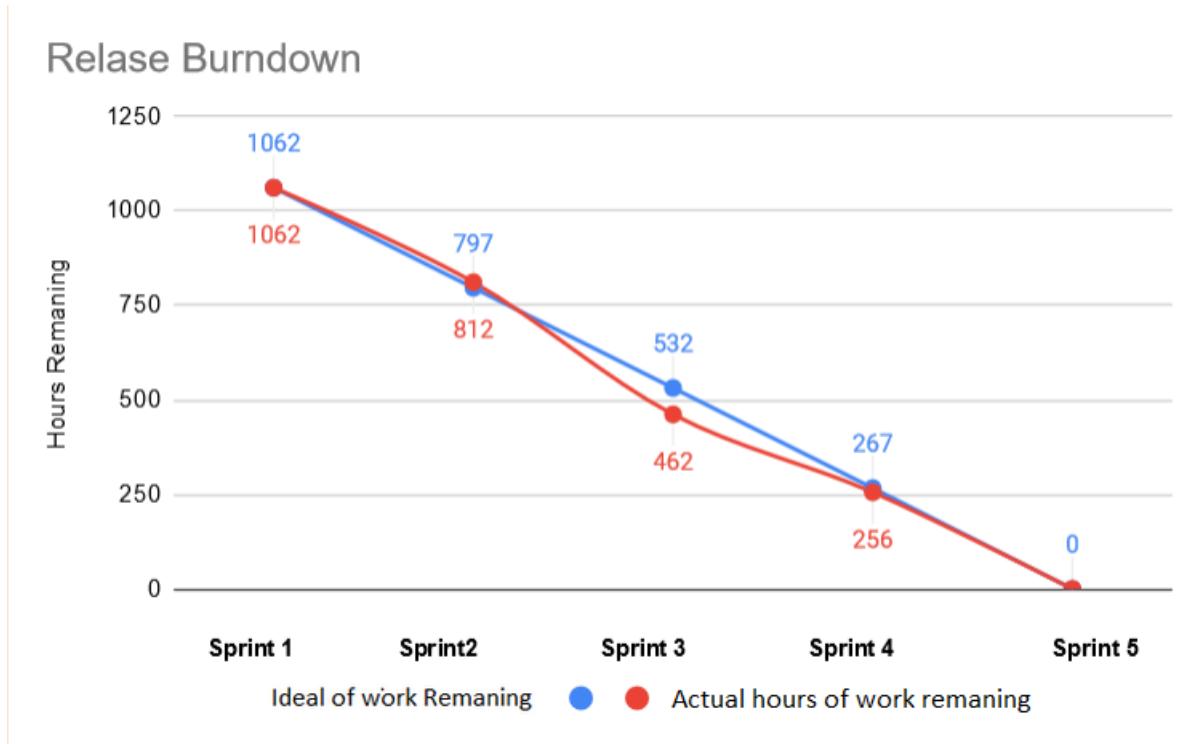


Figure 135: Release Burndown

8.5 Team Challenges

Due to some reasons, our team cannot have offline meeting, we only can discuss and communicate with each other via group chat and sometime the internet not quite good. By using the group call function, we can talk to each other but quite noisy. When a member has a bug cannot fix by himself, other members very hard to debug with him and find the root cause.

The technical leader has chosen Node.js, Express.js, ReactJS, and MongoDB for this project, but those technical are very new for all members. And the Front-end guys don't know anything about ReactJS, so the leader decided not to use ReactJS anymore, then it will lead to difficulty to connect front-end and back-end together. Cuong is a member of backend team but he does not know everything about those languages, he needs to spend 2 weeks to learn everything from beginning and it lead to some task could not finish on time.

Some members are very busy, so some tasks could not finish on time.

8.6 Team Meetings

Our team plans to conduct offline meetings 2 times a week but due to some reasons, we have decided to have online meetings and discussions via group chat. The meeting and discussion helps us to know the difficulties of each member faced when doing their task and the leader will review individual tasks of each member.

In each meeting, we usually spent 10-20 minutes for each member to talk about their task in 3 points:

- What have you done since the previous sprint?
- What are you planning to do today or next sprint?
- Is there any obstacle preventing you from doing the task?

If any member have difficult when doing their task, all the team will discuss and find a way to solve it. If a task is out of date, another member will be involved in that task.

Below is the meeting minutes:

Date	Description	Attendees
------	-------------	-----------

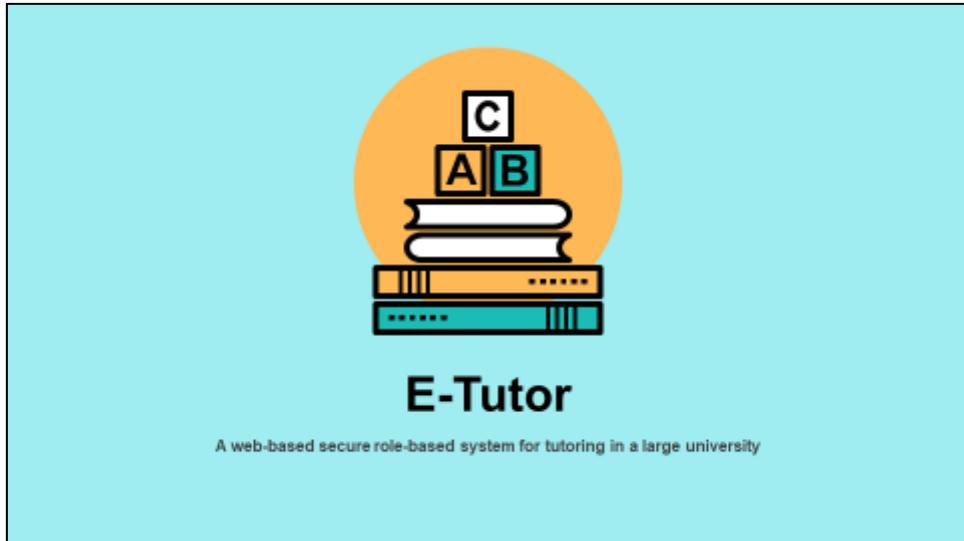
5 th of March, 2020	<p>Discussion on work breakdown structure for the project</p> <ul style="list-style-type: none"> - The leader talk about the structure of the project - The leader chose technology (Node.js, Express.js, MongoDB) and talk about why our team using this technology - Leader assign team role for each member 	Israel, Cường, Thái, Tuấn, Sơn
13 th of March, 2020	<p>Discussion on Front-end, design of the system</p> <ul style="list-style-type: none"> - Making User stories and product backlogs - Discuss about what the Front-end look like - Distribute the task for Front-end team - Distribute the task for Back-end team 	Israel, Fadil, Cường, Sơn
20 th of March, 2020	<p>Leader review weekly activities and update Product backlog</p> <ul style="list-style-type: none"> - Each member in team talk about their task and update the process of each task - Cuong is a member of Back-end team, he need to finish the login function before Sprint 1 end but he cannot finish his task on time because he is quite new with this technology, so he need about 2 weeks to learn about Node.js, Express.js and Mongo DB. The Leader will be involved to do his task because login function is very important function - Front-end team still on track - Our team find out that tester don't have much thing to do at this point of project, so the leader decide that tester will be involved on Front-end team 	All members
27 th of March, 2020	<p>Leader review weekly activities and update Product backlog</p> <ul style="list-style-type: none"> - Tester start writing the test case - The front-end team has small problem about design the chat feature and our team has discussed about find a way to fix that issue. - Cuong has caught up with the back-end team and he is starts to do the CRUD for the user model - Some task has not been done in this Sprint and will continue in next sprint 	All members

3 rd of April, 2020	<p>Leader review weekly all member's task, update Product backlogs</p> <ul style="list-style-type: none"> - Each member in team talk about their task and update the process of their task for the leader. - Front-end team is quite slow and the leader push them to speed up - Tester report for the developer some test case was failed in previous sprint and all developer look into that error and find the root cause. 	All members
10 th of April, 2020	<p>Leader review weekly all member's task, update Product backlogs</p> <ul style="list-style-type: none"> - All the team find out that the project missing blogging feature and decided to add this feature in this sprint. - All the team discuss about what the front-end of blogging feature should look like. - Tester keep reporting the error from previous sprint to the developer to fix. 	All members
17 th of April, 2020	<p>Leader review weekly all member's task, update Product backlogs</p> <ul style="list-style-type: none"> - Each member in team talk about their task and update the process of their task for the leader. - Tester keep reporting the error from previous sprint to the developer to fix. - The leader decide that all the team need to be speed up 	All members
27 th of April, 2020	<p>Leader review weekly all member's task, update Product backlogs</p> <ul style="list-style-type: none"> - All the important task has been done - Leader decided to test all the website again, and all the member need to be involved on testing. 	All members

Table 2: Team Meeting Details

9 PRESENTATION

Below is a presentation summarising few important things about the project:



Our Team

Pham Tran Hong Son Front-End Dev	Truong Quoc Cuong Back-End Dev	Ngo Vinh Thai Tester
Trinh Anh Tuan Front-End Dev	Israel Obanijesu Back-End Dev	Idris Usman Fadil Tester

Main Technologies



Node.js



MongoDB



Express.js

Key Features



Reports

There are a lot of reports like Statistics number of message in last 7 days or average number of message for each tutor



Blogging/Chatting

All student can tutor can messaging, uploading document, comment for blogging



Allocate/Reallocate tutor

Staff can allocate personal tutor to students



Personal Dashboard

Each personal tutor/student will have their dashboards



Email

Email between students and their personal tutors



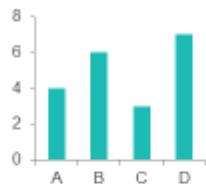
Responsive Design



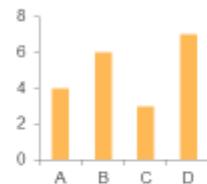
Bootstrap CSS

Using Bootstrap to make responsive website
Support responsive for Desktop, Mobile, Tablet.

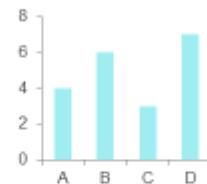
STATISTICS



Number of message in last 7 days



Average number of messages for each personal tutor

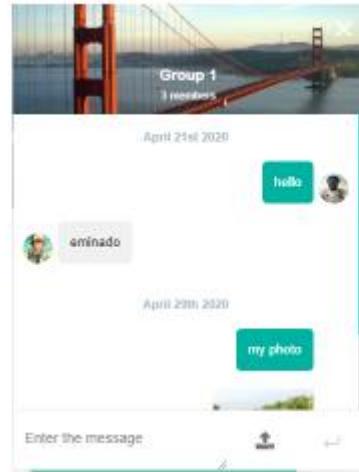


Students with no interaction



Messaging Chatting

Group Message
Private Message



Email Sending

Easy to modify Email specification



Sending Email automatically



For notification Events

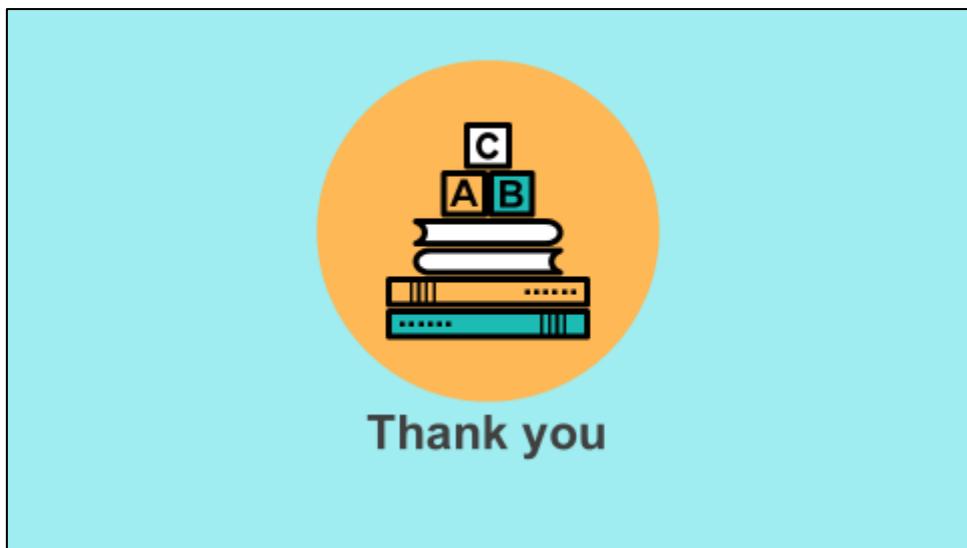


Figure 136: E-Tutor Presentation

10 CONCLUSION

The application “E-Tutor” has been built successfully, covering all the requirements that were specified in the coursework. As a team, we’ve successfully used Agile Scrum methodology for managing the development process of the application.

Several challenges were faced during the course of developing the software application. However, a key highlight of one challenged faced was, the difficulty of developers to correctly estimate the amount of time needed to complete a feature due to complexity. We were able to manage this challenge by systematically monitoring the progress of each developer working on a particular feature daily. As a team, there are times when we also had to support other developer technically, if he’s running out of time.

Overall, the project’s journey has actually given us more knowledge and exposure to some best practices. We can say team work and total commitment is paramount to the success of an application.