



Group Report: Web Software Development

COMP1640

Tutor: Dzung Lai Manh

Name of group: group 4

Members of group:

Name	Roles
Nguyen Trung Duc	Programmer
Vu Manh Hung	Scrum Master, Web Designer
Pham Van Toan	Programmer
Ta Van Duc	Programmer, Web Designer
Mac Xuan Hung	Database Designer, Programmer
Igbo Kingsley Chinedu	Tester

Artefacts:

URL to our video/screencasts: <https://youtu.be/18-sNkgUWgs>

URL to code repository:

<https://github.com/ducnt0706/etutoringreact/tree/master/myetutoring>

URL to our website: <https://etutoring-contact.web.app/>

User name and password:

Roles	User Name	Password
Admin	ducnt0706@gmail.com	comp1640
Tutor	ducntgch17377@fpt.edu.vn	nguyentrungduc
Student	hungvm0510@gmail.com	nhokidlee01

Notice: Because of Google's identity verification feature, if you are required to enter the code you are entering please contact us at contact +84379345690, +84961209037 or +84355637668

Link to Presentation: https://docs.google.com/presentation/d/10uGCNa7F-chyalw_ZonSh86xlTcZHHkZSa7BkK33QGg/edit?fbclid=IwAR0q0fYKkipMJDg0cRJqxFHJCWY5qmyUpF5fDLLglolLtrvbSObsQpLy5k#slide=id.g7750455d92_0_181

URL to Scrum Document(Product Backlog, Sprint Backlog, Meetings):

https://docs.google.com/spreadsheets/d/1-RUEJYo5Lny6isgM2q6tOEQ82aXr2Vx12254QAtZBOg/edit?fbclid=IwAR3Ac7qYK7_udesc65RiesaK3J-v5ul3KfM_-c6Qfa6x-XVUBBoITCYwIGI#gid=494110541

URL to Other Document: <https://trello.com/b/LUvsNa1p/etutoringcomp1640>

Technologies:

- **Front-end:** Html, Css, Bootstrap framework, React.
- **Back-end:** Firebase SDK (Firestore, Firebase storage, Firebase authentication).
- **Database:** Using noSQL database as Firestore on Firebase

Contents

1) Introduction	5
2) Requirement	5
2.1 Requirement	5
3) Site Design	6
3.1 Use Case Diagram	6
3.2 ERD	7
3.3 User Roles	8
4) Design Wire Frame	9
4.1.1 Home Page	9
4.1.2 Login Account:	10
4.1 Admin Wire Frame	10
4.1.1 Admin Assign	11
4.1.2 View Information Of Students & Tutor had Assign	12
4.1.3 Tutor Statics	13
4.1.4 Student Statics	14
4.2 Tutor Wire Frame	14
4.2.1 Tutor Home Page:	15
4.2.2 Daily Post:	16
4.2.3 Create New Meeting:	17
4.2.4 Tutor Contacts:	19
4.2.5 Tutor Messages:	20
4.3 Student Wire Frame	21
4.3.1 Student Dashboard	21
4.3.2 Student Contact	22
4.3.3 Student Message	23
5) Product Review	23
5.1 Functional Requirement	23
5.1.1 Authentication by Google API	24
5.1.2 Firebase Authentication on Websites	25
5.2 Admin Interface	26

5.3 Tutor Interface	29
5.3.1 Meeting.....	29
5.3.2 Post	31
5.3.3 Contact.....	34
5.3.4 Message	34
5.4 Student Interface	35
5.4.1 Meeting.....	35
5.4.2 Post	36
5.4.3 Contact.....	37
5.5 Non-Functional Requirements.....	38
5.5.1 Security	38
5.5.2 Firebase Realtime Database Rules.....	38
5.5.3 Firebase Security Rules for Cloud Storage.....	39
5.5.4 Clean Code	39
5.6 Code Snippets	41
6) Scrum Document	49
7) Apply The Scrum With Agile Method	50
7.2 Product Backlog	50
7.3 Daily Of The Meeting	52
7.4 Sprint Backlog	56
7.5 Burndown Chart.....	60
8) Testing	64
8.1 Test Item	64
8.2 Test List Of Function	65
8.3 Test environment.....	66
8.4 Test Log:	66
8.5 Test case summary.....	77
9) Summary.....	78
10) References	79

1) Introduction

In this course we are required to work in groups and apply agile work practices and record our meetings appropriately to form a website to support students and their tutor in exchange and in meetings, etc. The name of our web site is E-Tutor. We are required to use the scrum model to complete our product. In this report I will outline the functionalities of the product that our team has completed and provide facilities such as wire frame, ERD diagram to explain more clearly about our products. In the next we will also give our insights on the scrum model and information about the team meetings to prove that we have applied the scrum model as our product.

2) Requirement

2.1 Requirement

After reading the requirements and referring to the documentation, we have identified the goals that we need to achieve in order to build a complete product and meet the needs of users. Below is a detailed listing of the functions that we need to build:

- First All student must have a personal tutor.
- Members authorized by staff will have the ability to allocate or reassign individual tutors to students. After completing the student and their tutor will receive notification email.
- One tutor can support many students at the same time.
- All students and their tutors must use the E-Tutor system for messaging and scheduling and record offline or online meetings or upload their documents and comments.
- Email between students and their personal tutors is only used to inform events, meeting information, not other information.
- Student and staff data will be accessed from the university's MIS system. Maintenance feature will not be implemented in this project.
- Each student will have their own personal dashboard and summarize their interaction with their personal tutors.
- The Similarly each individual tutor will also have a panel of their individual tutees that can be sorted and filtered accordingly.
- Authorized staff will have access to the dashboards of other staff, and to individual dashboards for students.
- The interface must be suitable for all devices (eg mobile phones, tablets, desktops).

Basically, an authorized admin will be able to assign the tutor and their student. Once the admin has finished assigning students to tutors, these people will receive an email notification. In addition, all admin, student and their tutor will have their own dashboard. Tutor has the ability to create offline and online meeting schedules between me and students, they can chat and exchange information with each other, etc. So here are all off the requirement that we have to do to build our product. In the next, we will show about how we implement these requirements and prioritize them.

3) Database Design

After identifying the objects as well as their functions, the next thing to do is to model the relationship between them so that we can gain a better overview of the product.

3.1 Use Case Diagram

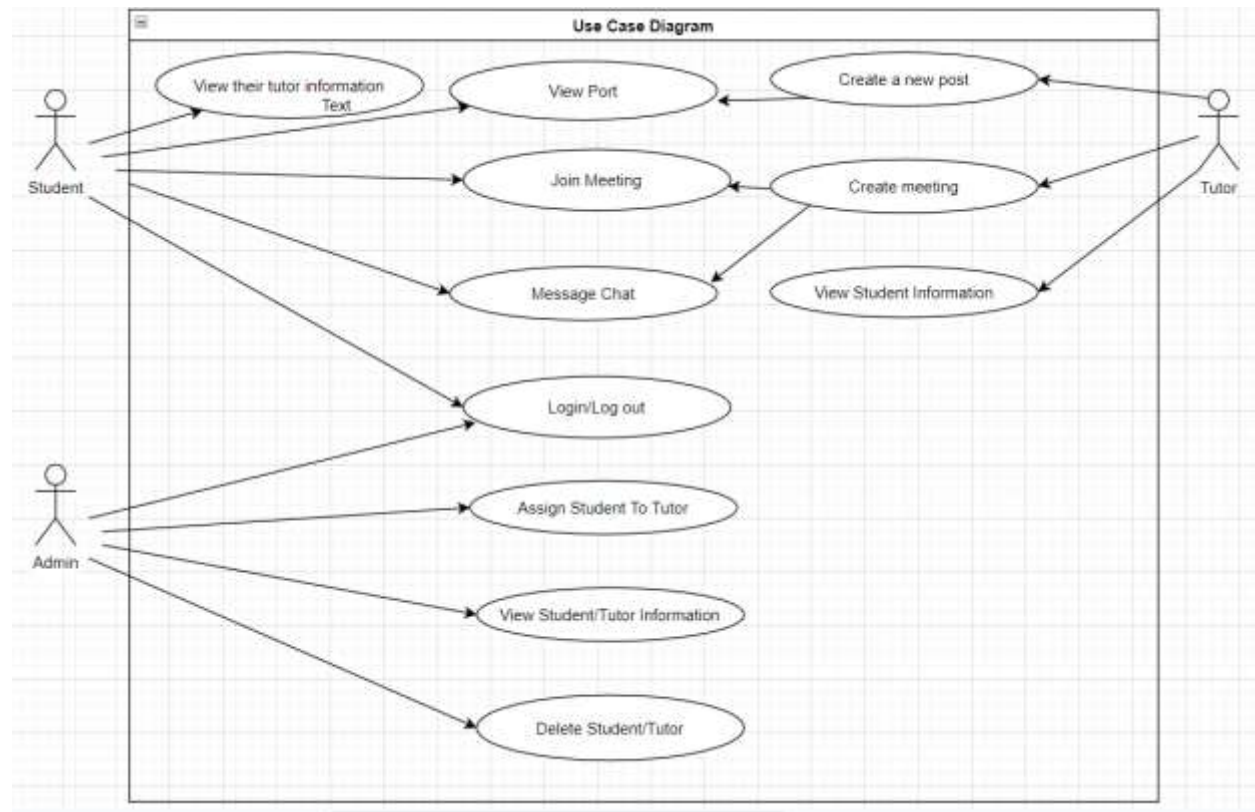


Figure 1: E-tutoring System Use case Diagram

This is a general outline of the features of tutor, student and admin when logging into our system. It is an important basis for us to determine the direction for our products. These features will be summarized in the section below.

3.2 ERD

In this project, we use NoSQL instead of SQL because they bring a lot of benefits. First, we easily store data and queries. Next, we easily work with unstructured form data with NoSQL. Moreover, changing data (adding, editing and deleting) is easy and convenient, not as complicated as SQL. And finally, because of the lack of ACID of transactions and data consistency, NoSQL DB can be extended and run on many machines easily.

Because of using NoSQL Database, the data of the tables are independent of each other. Therefore, when creating a collection, we will add the necessary data without regard to other collections.

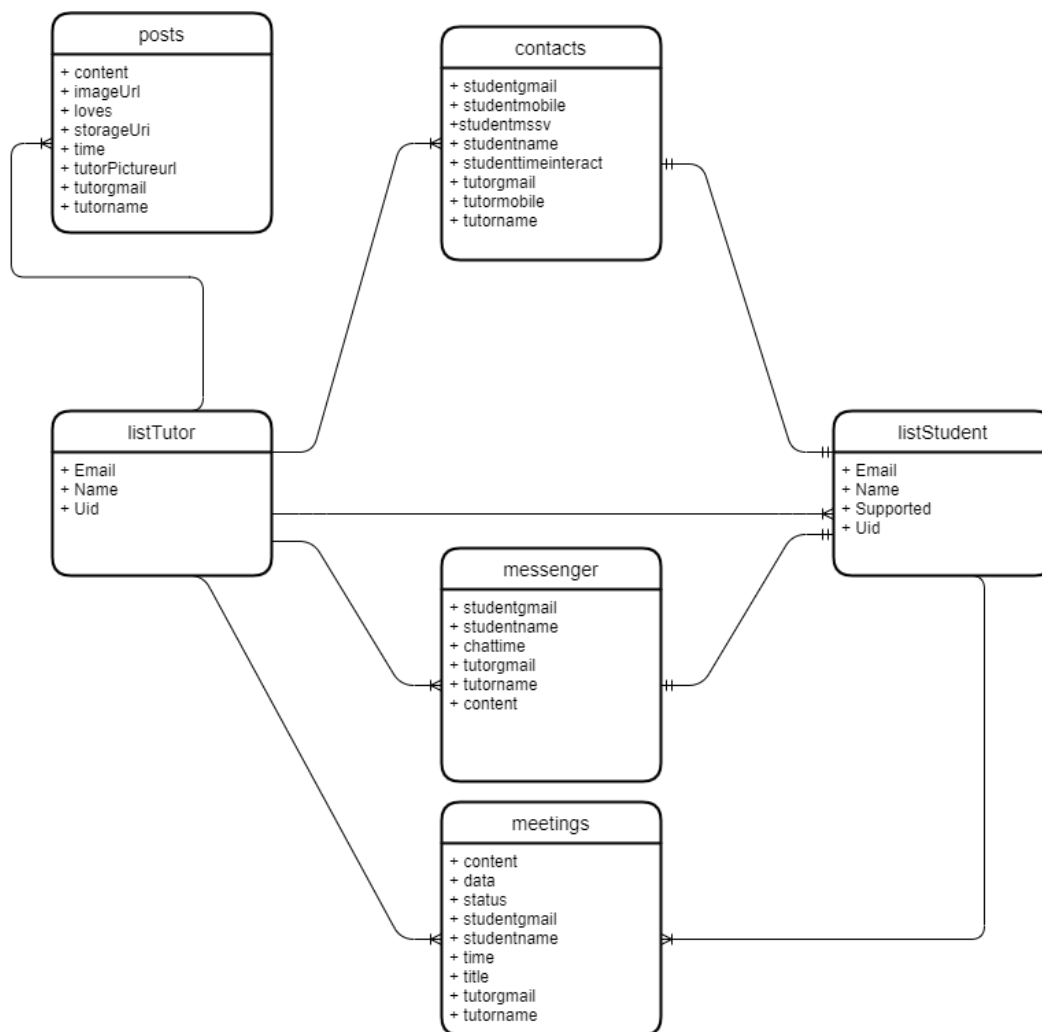


Figure 2: ERD

This is our ERD, we used Firebase to store data. There are 6 entities including **listTutor**, **listStudent**, **posts**, **contacts**, **meetings** and **messenger**. Firstly, listTutor entity stores information of tutor including Email, Name and Uid. Secondly, listStudent entity is similar to listTutor but it has more than one information that is **Supported**. Supported is the tutor who tutoring this student. The relationship between listTutor and listStudent is one-to-many relationship because one tutor can tutor many students but student can be tutored by one tutor.

Thirdly, posts entity is the post of tutor, it stores content (content of post), imageUrl (image of post), loves (total love of post), storageUrl, time (posted time), tutorPictureurl (picture of tutor), tutorgmail (gmail of tutor) and tutorname (name of tutor). The relationship between listTutor and posts is one-to-many relationship because one tutor can have many posts, but one post can be posted by one tutor.

The four is contacts entity, it stores information of tutor (tutorgmail, tutorname and tutormobile), student (studentgmail, studentmobile, studentmssv and studentname) and last interaction time (studenttimeinteract). The relationship between listTutor and contacts is one-to-many relationship and between listStudent and contacts is one-to-one relationship because one tutor has many contacts of students but one student has only one contact of one tutor (supported).

Next this is meetings entity, it stores information of tutor (tutorgmail and tutorname), student (studentgmail and studentname) on this meeting and meeting (content, data, status, time and title). The relationship between listTutor and meetings and listStudent and meetings is one-to-many because tutor can create more than one meeting with one student.

And messenger entity is the final, it stores information of chat between one tutor and one student including studentgmail, studentname, tutorgmail, tutorname, chattime (when content is sent) and content (content of each message). The relationship between listTutor and messenger is one-to-many relationship because tutor has many messenger (students who he supports) and between listStudent and messenger is one-to-one relationship because one student has one supported.

3.3 User Roles

As you can see, we have broken down into 3 different user types with specific features for each role. Here I will summarize to have a more overview.

First, the admin will be able to view tutor and student details. They can assign students to tutors and can track students who have not interacted for days to take action.

For tutors, after being assigned students can view the information of the students they are assigned, can create new posts to post information and create online and offline meeting appointments with their students.

For students after logging in they can view the post of their tutors and participate in meetings and can chat with their tutors. They can also view information of their tutor

4) Site Design

After identified the objects of the product as well as the functions of those objects, we got a better overview of what we needed to do. However, to be able to build a website is not enough. We need a general overview so we can have a more detailed picture of how objects are arranged, how well they are sized, etc. It makes us build the web site easier and not having to edit too much. Here are the wire frame designs for admin, tutor and their student.

The first will be the general interface design for all users. This will include the design home page, the login function into the system.

4.1.1 Home Page

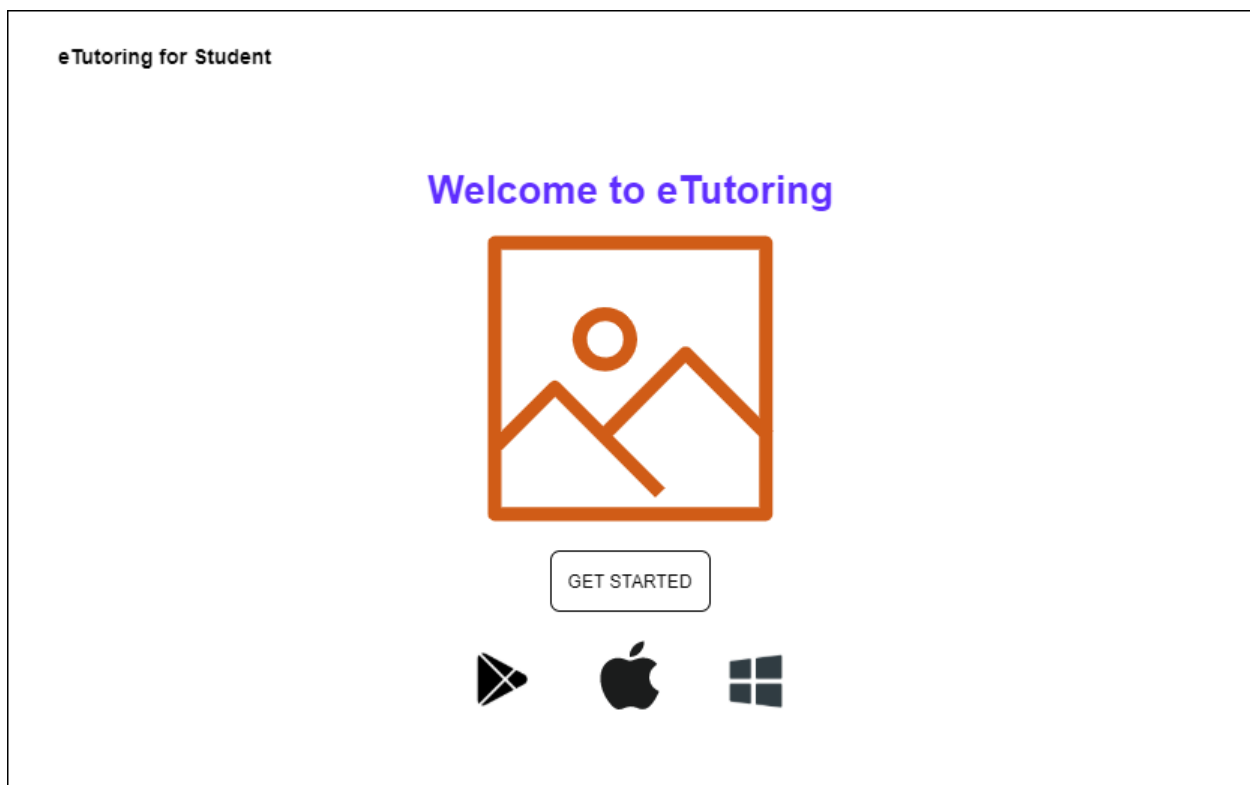


Figure 3: Home Page

When users access the website, the home page interface will appear, the user will click the “Get Started” button to login to his user interface. In it there will be detailed features for each person.

4.1.2 Login Account:

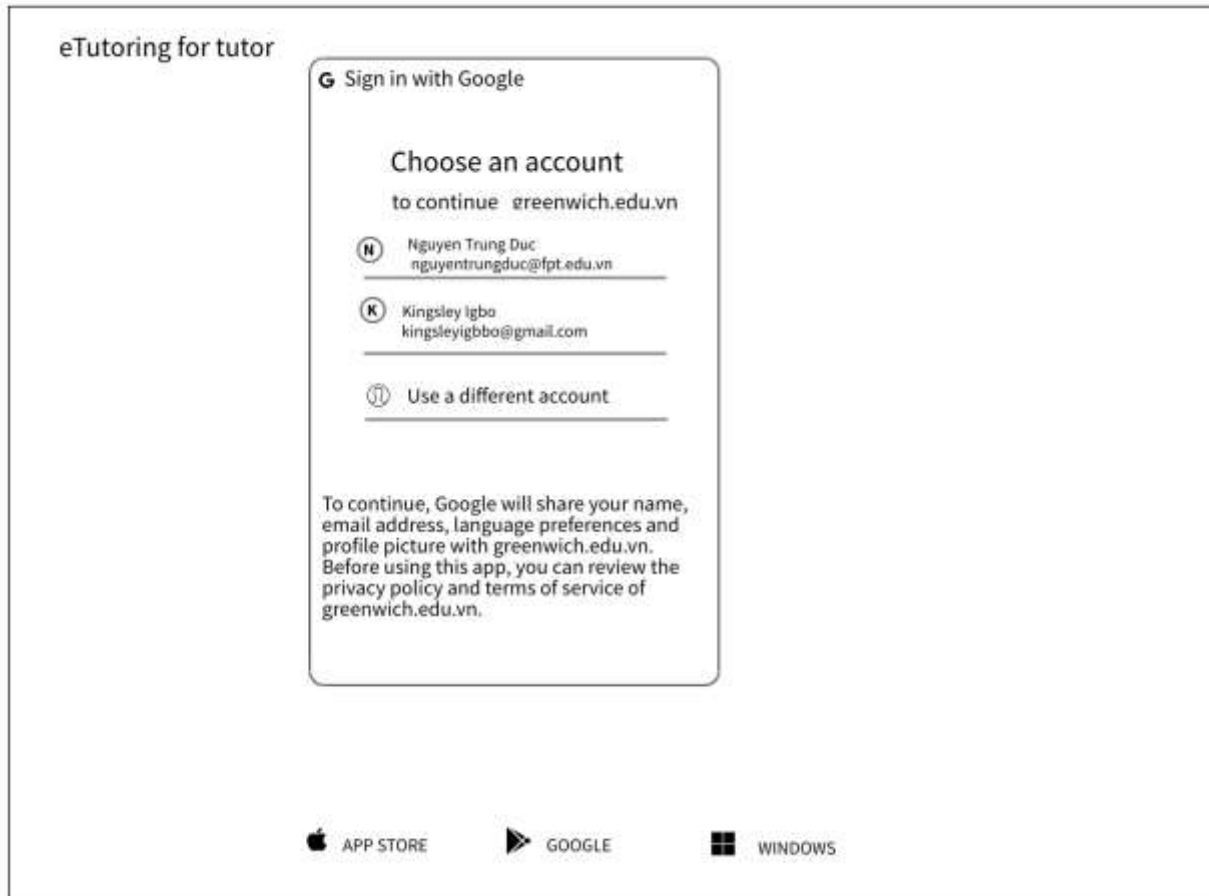


Figure 4: Login Account

After clicking the "Get Started" button, the website will appear Login box for users to log in with the account that they have been provided from before. After successful login, they will be redirected to the interface appropriate to their functions. In the next section, I will show the design interface for each of these functions.

4.1 Admin Wire Frame

In the first we will start with designing wireframe for admin. In this section, I will show you the interface design as well as admin features that can be implemented when logging into the system. When the admin successfully logs in, they will see their dash board interface. In the left corner are the properties "Main" and "Extras" in which when the admin clicks on the "Dash

board" button they will return to the main interface of the site. Inside "Extras" there are two buttons, "Student Statics" and "Tutor Statics". Their use will be detailed in the below.

4.1.1 Admin Assign

The screenshot displays the 'Admin Assign' interface. On the left is a sidebar with a user profile section at the top containing a user icon, 'User's name', and 'Role'. Below this are two sections: 'Main Tasks' with a 'Dashboard' link (house icon) and 'Extra Tasks' with 'Student Statics' and 'Tutor Statics' links (person icons). The main content area has a header with a message icon, a bell icon, and a user icon. Below the header is a box with two tabs: 'Provide For Student The Best Things' and 'Attention And Exchange'. The 'Provide For Student The Best Things' tab is active, showing a 'Select Tutor' dropdown menu with three options: 'Hung0510@gmail.com', 'phamvantuan@gmail.com', and 'Nguyenlungduc@gmail.com'. A red circle and arrow highlight the 'Select Tutor' label. To the right is a 'Select Student' dropdown menu with three options: 'Abc@gmail.com', 'Duy@gmail.com', and 'Thanh@gmail.com'. A red circle and arrow highlight the 'Select Student' label. Below these two dropdowns is an 'ASSIGN' button.

Figure 5: Assign Students To Tutor

After the admin has successfully logged in, they can choose students and assign them to any tutor. We have designed the website so a tutor can have many students.

Once the assignment is complete, the admin can scroll down to see if there are no students assigned to the tutor yet. In the table will also show the tutor is not enough students so admin can choose to assign students to them.

User's name

Dashboard

Main Tasks

Provide For Student The Best Things
Attention And Exchange

Extra Tasks

Student Statics

Tutor Statics

ASSIGN

Tutor		
Vu Manh Hung	Hung0510@gmail.com	01664810161
Pham Van Toan	phamvantoan@gmail.com	01664810161
Trung Duc	trungduc@gmail.com	01664810161

Student have no support		
ABC	Abc@gmail.com	GCH17180
Abc123	Duy123@gmail.com	GCH0705
Abc456	Duy789@gmail.com	GCH0705
Abc567	Duy456@gmail.com	GCH0705
Abc789	Duy241@gmail.com	GCH0705
Duy	Duy442@gmail.com	GCH0705

Figure 6: View List Of Students Without Support

4.1.2 View Information Of Students & Tutor had Assign

User's name

Dashboard

Main Tasks

Provide For Student The Best Things
Attention And Exchange

Extra Tasks

Student Statics

Tutor Statics

List Of Assigned

Student ID	Student Name	Student Email	Tutor Name	Tutor Email	Delete
GCH17180	Vu Manh Hung	Hung0510@gmail.com	Duc Trung	Duc@gmail.com	Delete
GCH0705	Pham Van Toan	phamvantoan@gmail.com	Hung	Hung12@gmail.com	Delete
GCH0705	Pham Van Toan	phamvantoan@gmail.com	Hung	Hung12@gmail.com	Delete
GCH0705	Pham Van Toan	phamvantoan@gmail.com	Hung	Hung12@gmail.com	Delete
GCH0705	Pham Van Toan	phamvantoan@gmail.com	Hung	Hung12@gmail.com	Delete

Figure 7: View Information Of Students & Tutor had Assign

After the student has been assigned to the tutor, the admin can continue scrolling down to see the details of the student's name, email and student code of students who have already assigned the tutor and information about their tutor. They also can click on “Delete” button to delete student out of table.

4.1.3 Tutor Statics












<div>  <div> <div>User's name</div> <div>Role</div> </div> </div> <div> <div>Main Tasks</div> <div>  <div>Dashboard</div> </div> <div>Extra Tasks</div> <div>  <div>Student Statics</div> </div> <div>  <div>Tutor Statics</div> </div> </div>	<div> <div>Static Tutor</div> <div>    </div> </div>		
	<div>  <div> <div>User's name</div> <div>User's ID</div> </div> <div> <div>Message</div> </div> <div> <div>Mobile</div> <div>0987654321</div> </div> <div> <div>Email</div> <div>user1@gmail.com</div> </div> </div>	<div>  <div> <div>User's name</div> <div>User's ID</div> </div> <div> <div>Message</div> </div> <div> <div>Mobile</div> <div>0987654321</div> </div> <div> <div>Email</div> <div>user2@gmail.com</div> </div> </div>	<div>  <div> <div>User's name</div> <div>User's ID</div> </div> <div> <div>Message</div> </div> <div> <div>Mobile</div> <div>0987654321</div> </div> <div> <div>Email</div> <div>user3@gmail.com</div> </div> </div>


Figure 8: View Tutor's Information

When Admin click on "Tutor Statics" the interface button will automatically switch to another page. Here admin can see all the details of each tutor such as email, phone number, name of tutor, etc.

4.1.4 Student Statics




User's name




Role

Main Tasks




Dashboard

Extra Tasks






Student Statics



Tutor Statics

Static Students



Student have supported

Student ID	Student Name	Student Email	Student mobile	Last time Sign in	Day No Interaction
GCH17180	Vu Manh Hung	Hung0510@gmail.com	01664810161	02 May 2020 04:43 pm	1 day
GCH0705	Pham Van Toan	phamvantoan@gmail.com	0988902154	01 May 2020 04:43 pm	0 day
GCH0705	Pham Van Toan	phamvantoan@gmail.com	0988902154	04 May 2020 04:43 pm	3 days
GCH0705	Pham Van Toan	phamvantoan@gmail.com	0988902154	05 May 2020 04:43 pm	4 days
GCH0705	Pham Van Toan	phamvantoan@gmail.com	0988902154	03 May 2020 04:43 pm	2 days

Figure 9: View Student's Information

Similarly, when the admin clicks on the "Student Statistic" button they will be able to view the details of the students who have been assigned tutors as their names. student ID, email, phone. In addition, they can see how these students have been dormant for days and then identify and handle the offending students.

4.2 Tutor Wire Frame

In the next section will be a wireframe design with the interface and functions for the user as a tutor:

4.2.1 Tutor Home Page:

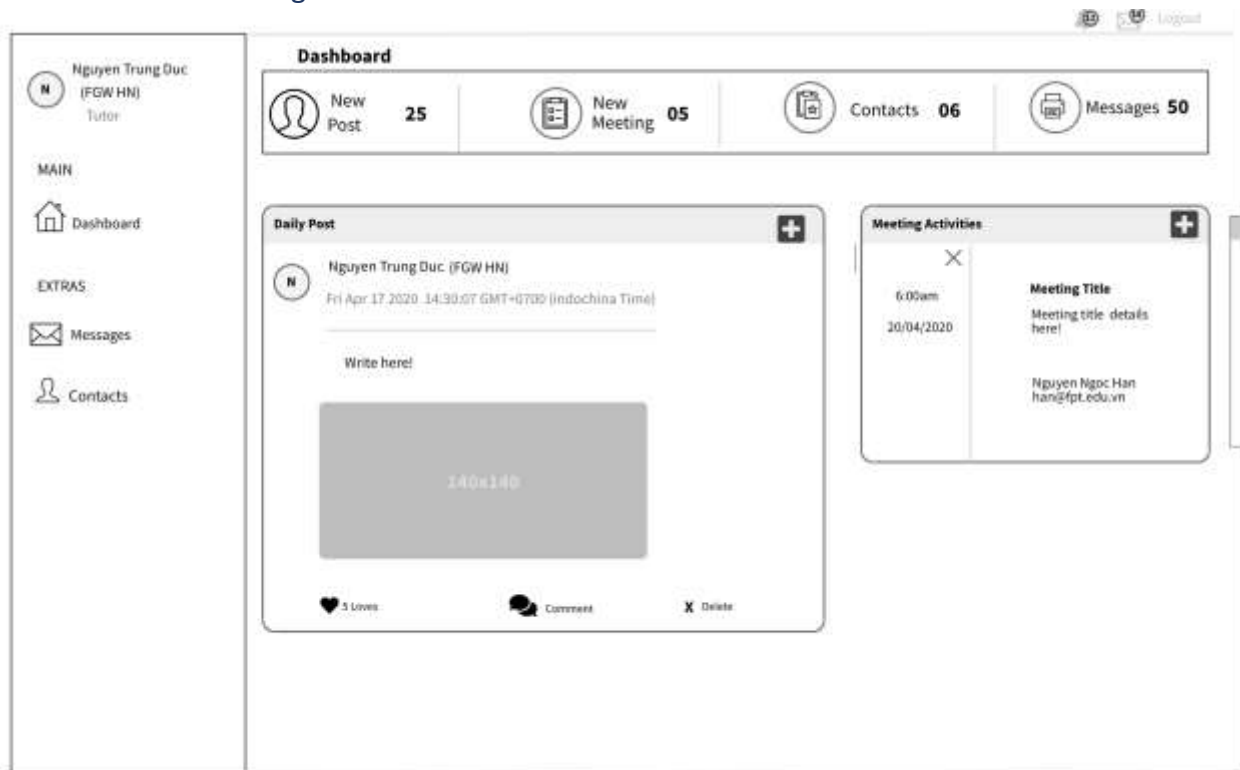


Figure 10: Tutor home page

If the login is successful, the tutor can access the Dashboard. On the left side of the web page will display their information including their name and role as well as their duties. Some examples when tutor clicking on the "Main" button will take you to the home page and "Extras" where containing Messages and contacts. In the upper right corner there are 3 icons representing 3 functions including "Message", "Notification" and "Log Out". Below are the meeting activities containing the title, time and date of the meeting.

Besides the meeting activities, the daily post contains news feeds, images or attachments. Below the attachment are the like, comment and delete features.

4.2.2 Daily Post:



Figure 11: Daily Post

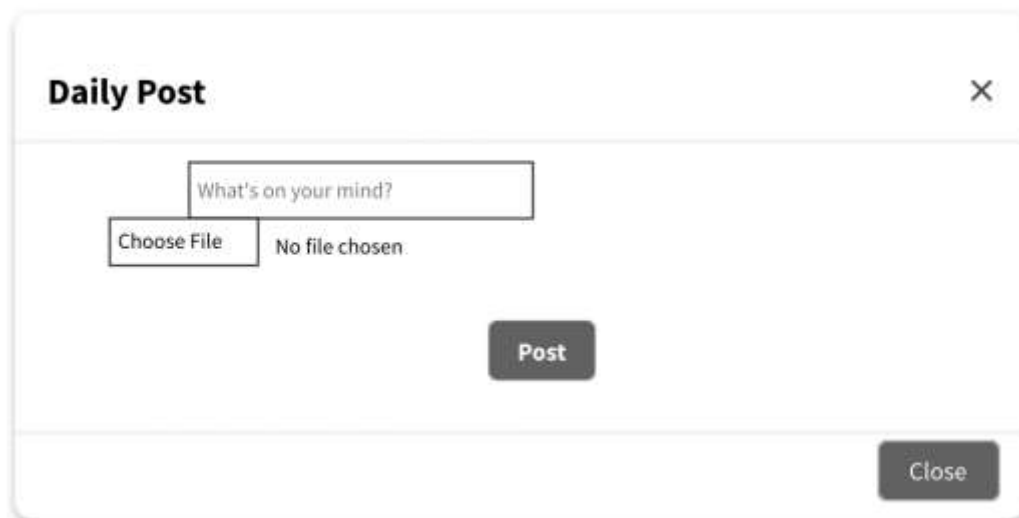


Figure 12: Daily Post Example

This is the interface when the tutor wants to create a new post. The top box is where the tutor writes an information and below is the "Choose File" button where tutor selects a file from the local storage device that could be an image. If no file is selected, it will display "No file chosen" to prompt the user to select a file. And a button below "Post" to post information to the group. At the bottom right is a "Close" button to exit the form when the tutor finishes the work.

4.2.3 Create New Meeting:



Figure 13: Example Of Create New Meeting

The plus (+) icon on the right side of the meeting activities signifies creating a new meeting. When tutors want to make an appointment with their students, they just need to click the plus sign to create a new meeting.

The image shows a 'New Meeting' dialog box. It has a title bar with the text 'New Meeting' and a close button (X). The main area contains the following elements from top to bottom: a text input field labeled 'Title'; a large text area labeled 'Content here!'; a date picker showing 'mm/dd/yyyy'; a time picker showing 'hh:mm'; a radio button selected for 'waiting for student to verify!'; a text input field labeled 'Student email'; a text input field labeled 'Student Name'; a 'Done' button; and a 'Close' button in the bottom right corner.

Figure 14: Example Of Create New Meeting

This is the form to create a new meeting. At the top is the "Title" button where the tutor enters the title of the meeting. Below is where the tutor enters the content for the meeting. Below the content is a date picker in which the tutor can choose month, day and year. Next is the time. The tutor can choose how long the meeting will last. Next is the radio button that the tutor clicks while waiting for the student to verify. The two sections below are the email and student name with a button that will send the meeting information entered into the meeting activity in the control panel. Finally, a close button at the bottom right to close the page.

4.2.4 Tutor Contacts:

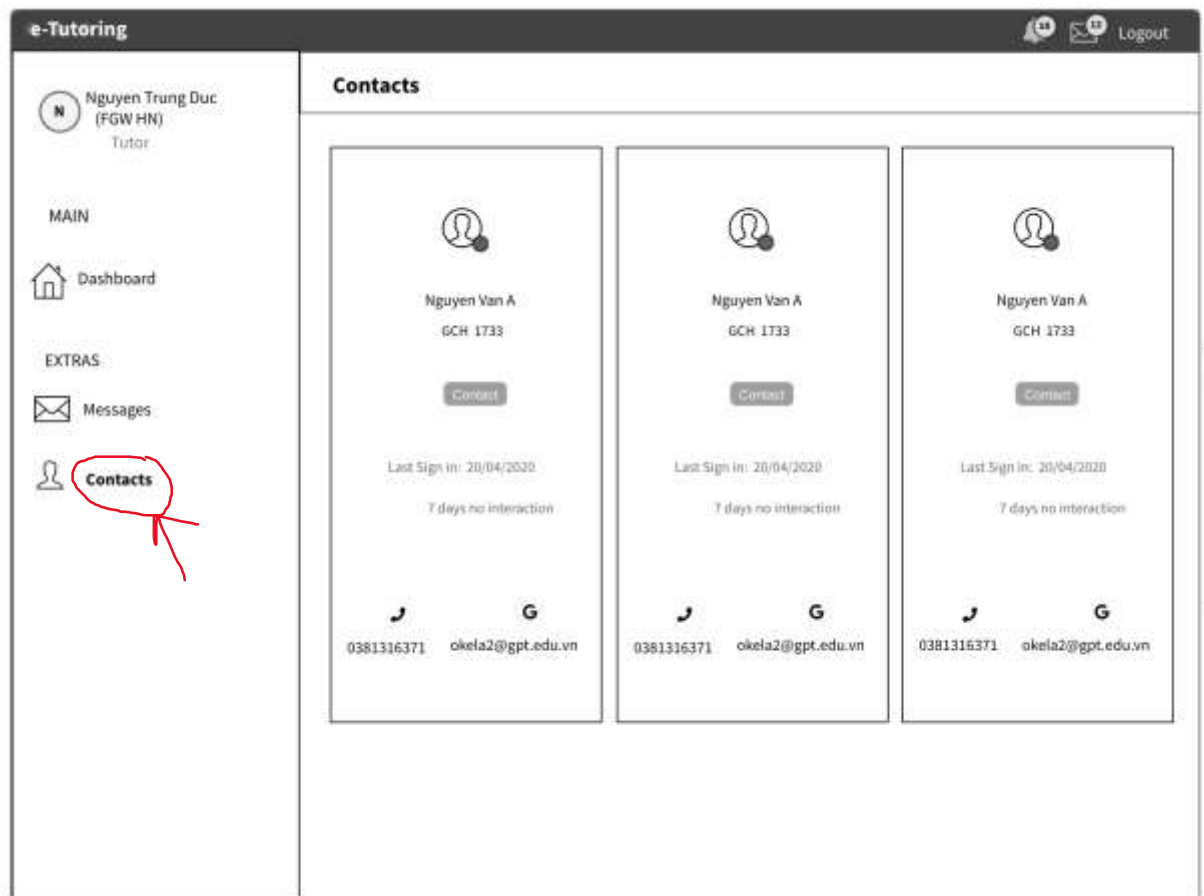


Figure 15: Tutor Contacts

When Tutor clicks the "Contact" button on Extra, it displays the contact between the student and their tutor. In the contact information, there are 4 information including: "User name", "User ID", Phone number of user "and" User email ". In addition, there is a message function to chat with this user.

4.2.5 Tutor Messages:

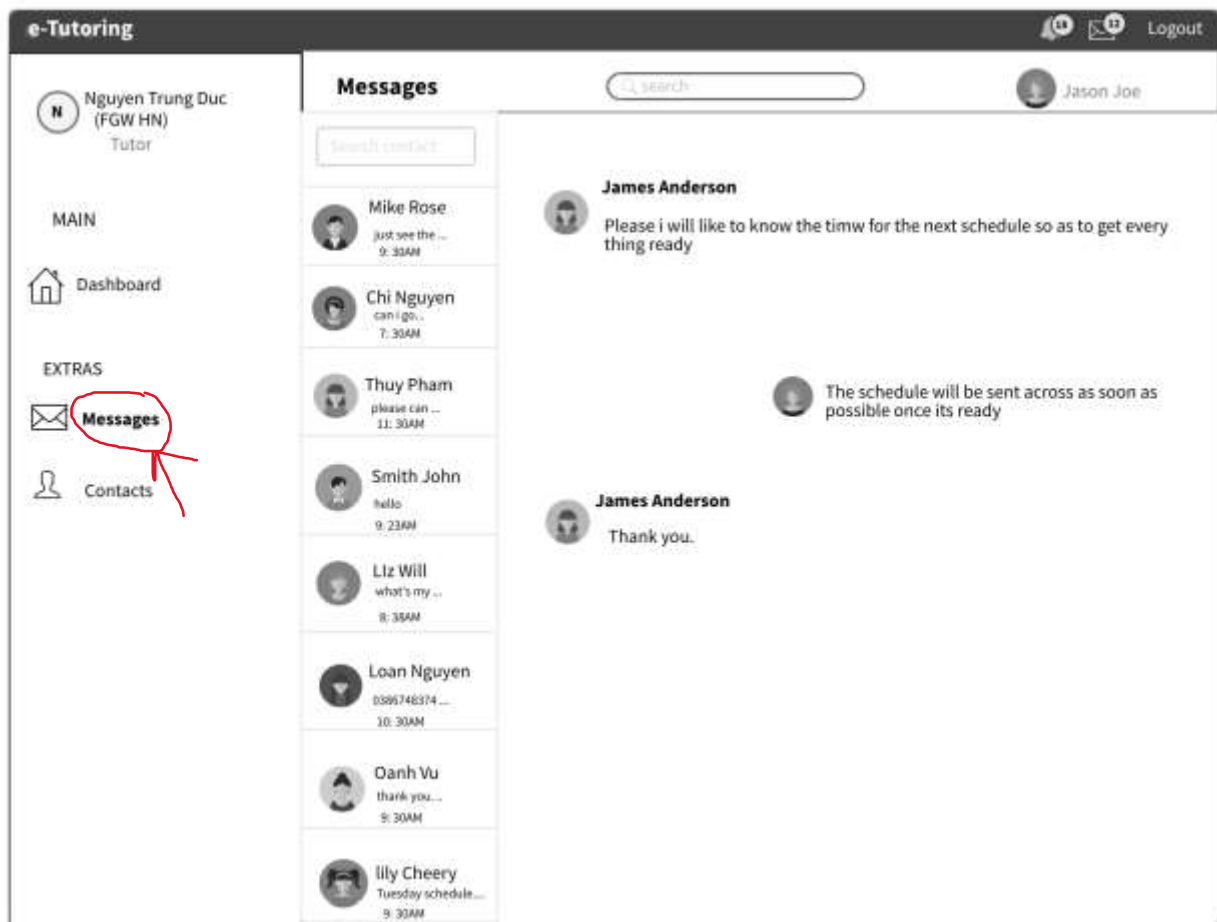


Figure 16: Tutor Messages

When the user clicks the "Message" button on the Extra, it displays the chat page where the tutor can exchange information with their students through the message. Display student contact with a preview of the message and the time of the message. "Content" is where the tutor writes the message and click on the arrow icon to send it to the student.

Above is the entire Wireframe design interface and tutor features. In the next section we will show the Wire frame design about the interface as well as the features of Student.

4.3 Student Wire Frame

In this section, I will show you the design of the interface and features of students when using the system.

4.3.1 Student Dashboard

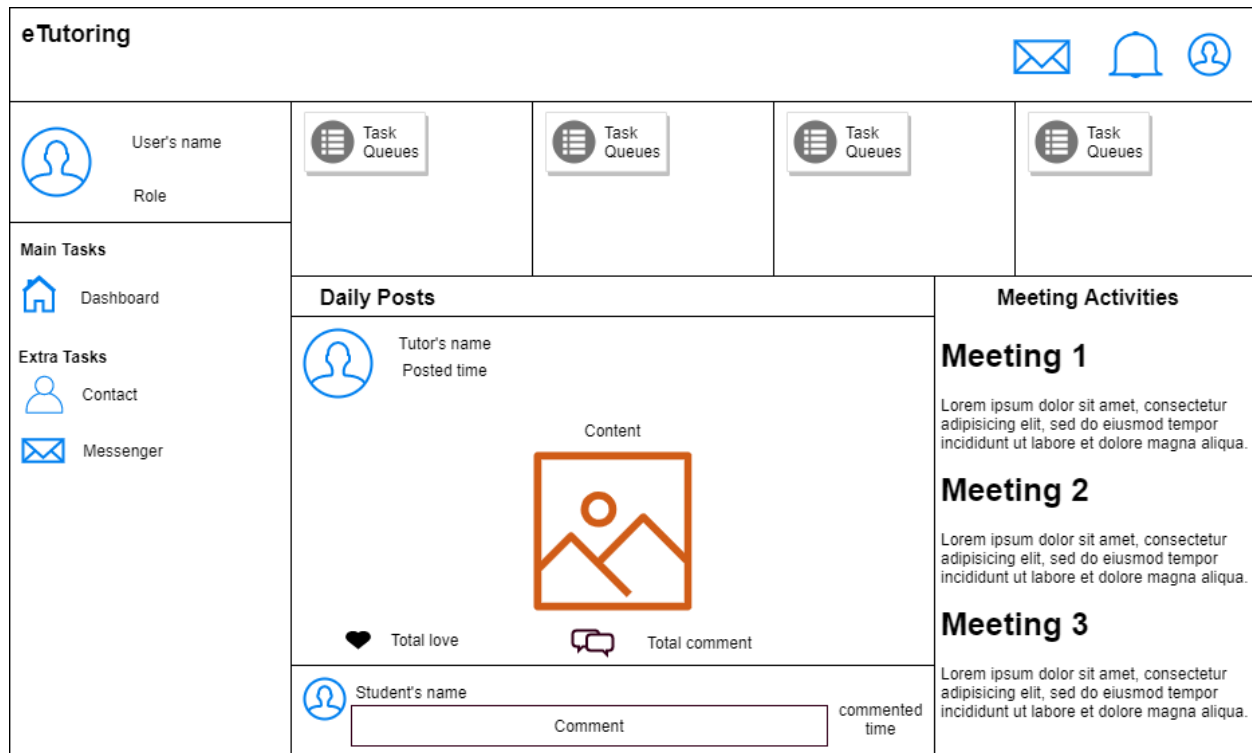


Figure 17: Student Dash board

After login successful, students will access the **Dashboard** on **Main Task**. On the left of website, there are their information including name and role and their tasks including **Main Tasks** and **Extra Tasks**. In the upper right corner there are 3 icons representing 3 functions including **messages**, **notifications** and **personal information**. Right below is the main information of the page including **student tasks**, **daily posts**, and **meeting activities**. In daily posts, it will show the posts of their tutor, they can view, love and comment. In meeting activities, it will show the meeting between student and their tutor.

4.3.2 Student Contact

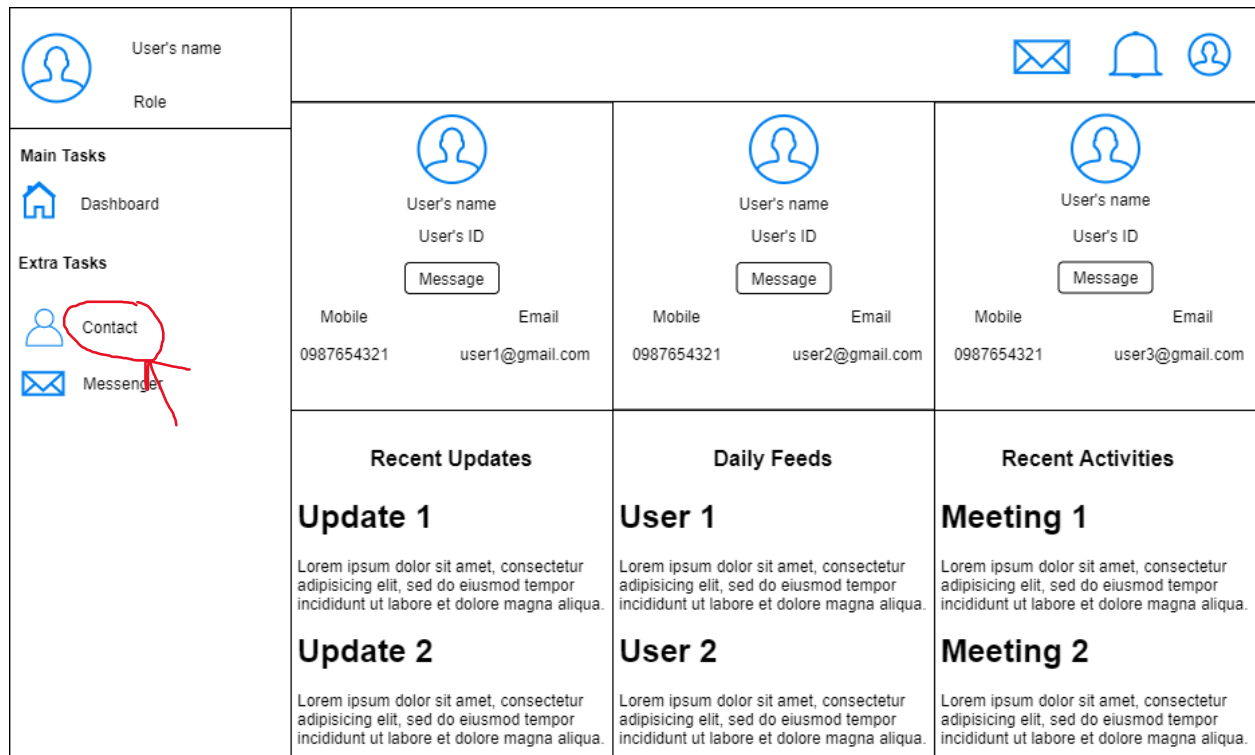


Figure 18: Student Contact

When student click on the "Contact" button on Extra Tasks, it will show the contact of their tutor. In contact information, there are 4 information including "User's name", "User's ID", "User's mobile phone" and "User's email". In addition, there are a function is "Message" to chat with this user.

4.3.3 Student Message

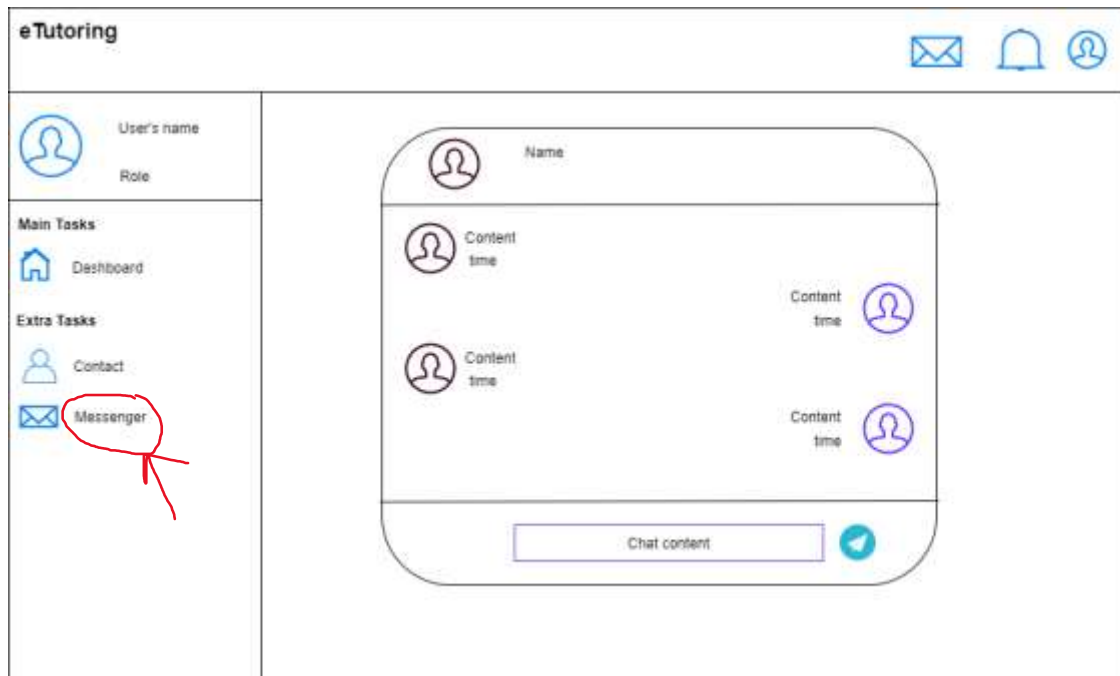


Figure 19: Student Message

Finally is "Messenger" button. When student click on it will show the chat between student and their tutor. In chat interface, there are 4 information: "Name of tutor", "Name of student", "Time when chat sent" and "Chat content". The plane paper icon is the send button to send chat content.

5) Product Review

In the next section, I will show you the detailed and realistic images of the system that we have built and give specific examples of the functions of each user.

5.1 Functional Requirement

A system needs functions to meet the needs of the user. A must-have function for a system is the login function. This function is necessary because it protects the user's personal information as well as protects the data for the system. With the system we are building, we want to make it convenient for users, as they do not need to waste time on account registration. They can use

google account to login to the system. In this section we will analyze why we use authentication by google API explicitly.

5.1.1 Authentication by Google API

The system we are building needs to know the identity of the user. This allows the system to store user data in the database. With a specific identity the system will fetch specific data for each user and display it. With the firebase platform that the system is using, firebase authentication provides ancillary services, easy-to-use SDKs and ready-made UI libraries. This helps authenticate users to the system, and our most popular link choice is Google.



Figure 20: Federated identity provider integration

<https://firebase.google.com/docs/auth>

By using Firebase SDK authentication, login method will be added to the system. Authenticate users by integrating a google link identification provider, and allow users to log in with a Google account.

5.1.2 Firebase Authentication on Websites

Users use Google accounts to log in to the system. To do this, need to integrate Google Login using the Firebase SDK. Firstly, I activated Google login in Firebase control panel.

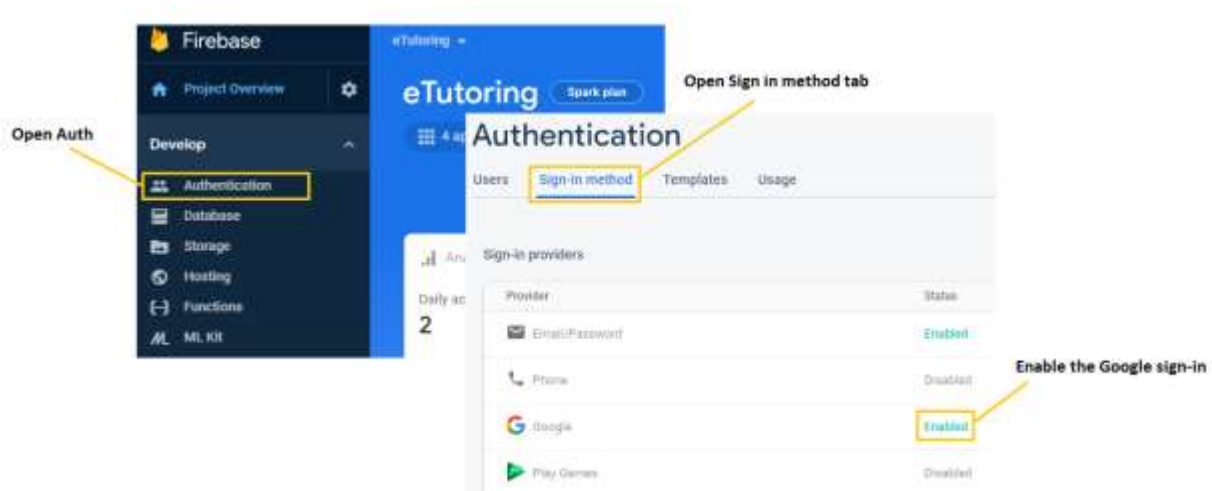


Figure 11: Enable the Google sign-in

The next step is to handle the login flow to handle I have used JavaScript in the following way. (Figure 22) In the first lines of code create a version of the Google provider object, followed by the second line of code validating using the Google provider object a popup window for users to log in with a Google Account there.

```
1 //----- Handle Login -----  
2 // TODO 1: Sign in Firebase with credential from the Google user.  
function signIn() {  
    var provider = new firebase.auth.GoogleAuthProvider();  
    firebase.auth().signInWithPopup(provider);  
}
```

Figure 22: Handle the sign-in

5.2 Admin Interface

After the admin has successfully logged in, they can choose students and assign them to any tutor. We have designed the website so a tutor can have many students.

Once the assignment is complete, the admin can scroll down to see if there are no students assigned to the tutor yet. In the table will also show the tutor is not enough students so admin can choose to assign students to them.

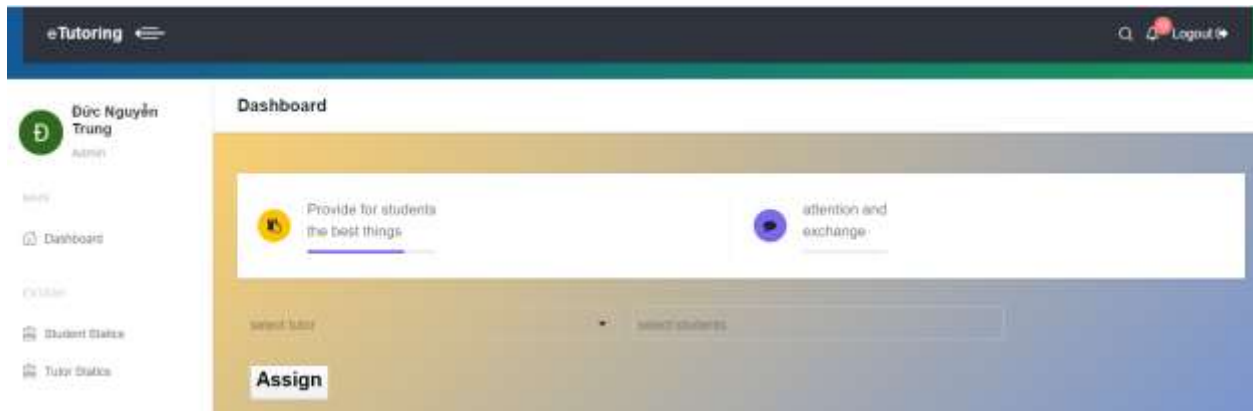


Figure 23: Assign Student To Tutor

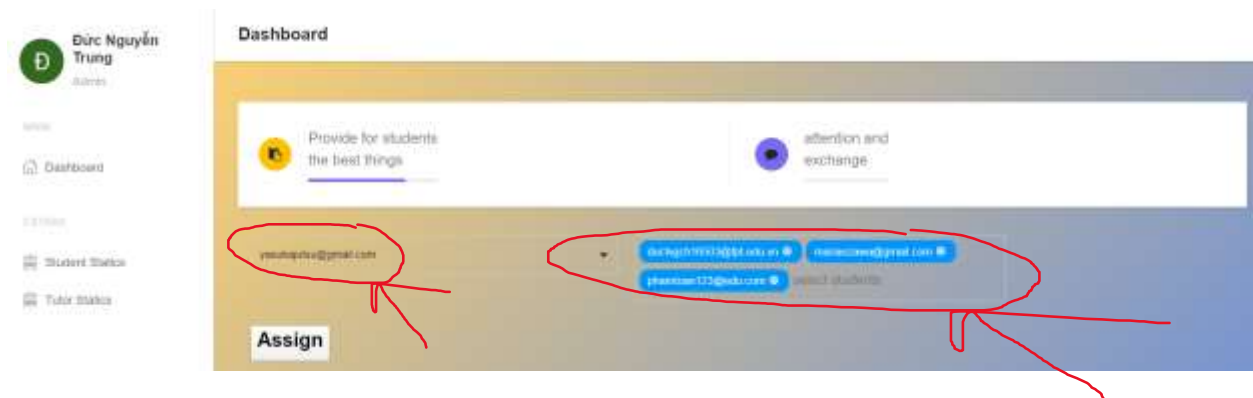


Figure 24: Assign Student To Tutor

As you can see, the admin can choose one tutor and assign multiple students to the tutor at the same time as long as the number does not exceed the allowed limit.

Assign					
Tutors			Students have no support		
Ta Van Duc FGW	yasukajutsu@gmail.c...	0123545626	Ta Van Duc	ductvgch16503@tpt.e...	GCH17637
Nguyen Trung Duc F...	ducntgch17377@tpt...	0371878455	Lee Min Ho	lsmhooo@gmail.com	GCK1234
Mac Xuan Hung FGW	hungtroy7c@gmail.co...	07566262265	ozawa ja	mariaozawa@gmail.c...	GCH123
Mac Xuan Hung	machunghd@gmail.c...	08521515175	Toan Toan	phamtoan123@edu.c...	GCK123

Figure 25: View Student Information Without Support

After finishing the assignment, when the admin scrolls down, you will see a list of students who have not been assigned a tutor.

List Of Assigns					
StudentID	StudentName	StudentEmail	TutorName	TutorGmail	Delete
GCH134	Toan Juda	anhhoankk@gmail.com	Ta Van Duc FGW	yasukajutsu@gmail.com	<button>delete</button>
GCH123	Chi vii	chichi123@gmail.com	Mac Xuan Hung FGW	hungtroy7c@gmail.com	<button>delete</button>
GCH123	Maisida za	dona11223@gmail.com	Mac Xuan Hung FGW	hungtroy7c@gmail.com	<button>delete</button>
GCH17637	Ta Van Duc	ductvgch16503@tpt.ed...	Nguyen Trung Duc PWG	ducntgch17377@tpt.ed...	<button>delete</button>
GCH123	Toan CA	phamtoan2255@gmail...	Mac Xuan Hung FGW	hungtroy7c@gmail.com	<button>delete</button>

Figure 26: List Of Assigned

As shown above when designing wire frames, administrators will be able to see the information of assigned students and their tutors. When the admin clicks on the "Delete" button, The student will be delete of the table.

Ta Van Duc FGW

yesukajutsu@gmail.com

0123545626

Nguyen Trung Duc F...

ductvgch17377@fpt.edu...

0371878455

King Chinedu FGW

superkingigbo@gmail.com

034589614

Mac Xuan Hung FGW

hungtroy7c@gmail.com

07566262265

Mac Xuan Hung

machunghd@gmail.com

08521515175

ozawa ja

mariaozawa@gmail.com

GCH123

Toan Toan

phamtoan123@edu.c...

GCK123

etutoring-contact.firebaseio.com says

Do you want to delete assign/right ???

OK

Cancel

GCH17637

GCH17363

GCK1234

Figure 27: Delete Student

Assign

Tutors

Students have no support

Ta Van Duc FGW

yesukajutsu@gmail.com

0123545626

Nguyen Trung Duc F...

ductvgch17377@fpt...

0371878455

King Chinedu FGW

superkingigbo@gmail...

034589614

Mac Xuan Hung FGW

hungtroy7c@gmail.co...

07566262265

Mac Xuan Hung

machunghd@gmail.c...

08521515175

Ta Van Duc

ductvgch16503@fpt...

GCH17637

King igbo

kingigbo@gmail.com

GCH17363

Lee Min Ho

ismhooo@gmail.com

GCK1234

ozawa ja

mariaozawa@gmail.c...

GCH123

Toan Toan

phamtoan123@edu.c...

GCK123

Figure 28: Delete Successfully

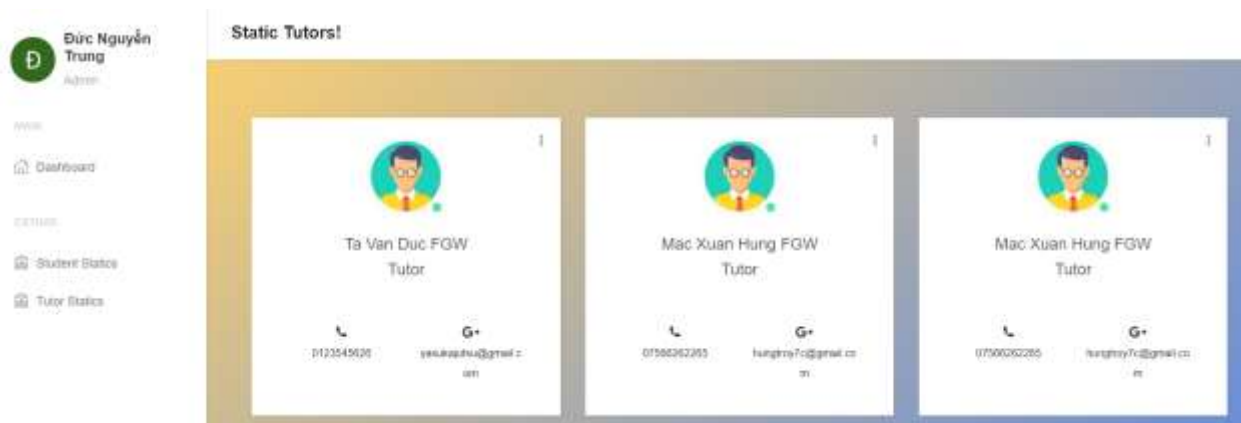


Figure 29: View Tutor Information



Figure 30: View Student Information

5.3 Tutor Interface

As a Tutor, you will have functions that include creating meetings with students, creating posts, communicating with students.

5.3.1 Meeting

The first is the meeting, the image below is the interface of meeting activities, it is located on the right of webpage. The list of created meetings will appear below, once the meeting has been

confirmed by the student, the plane paper icon will change to like icon. Tutor can also delete the meeting by clicking the X mark. In the upper right corner, plus icon is used to create meeting.

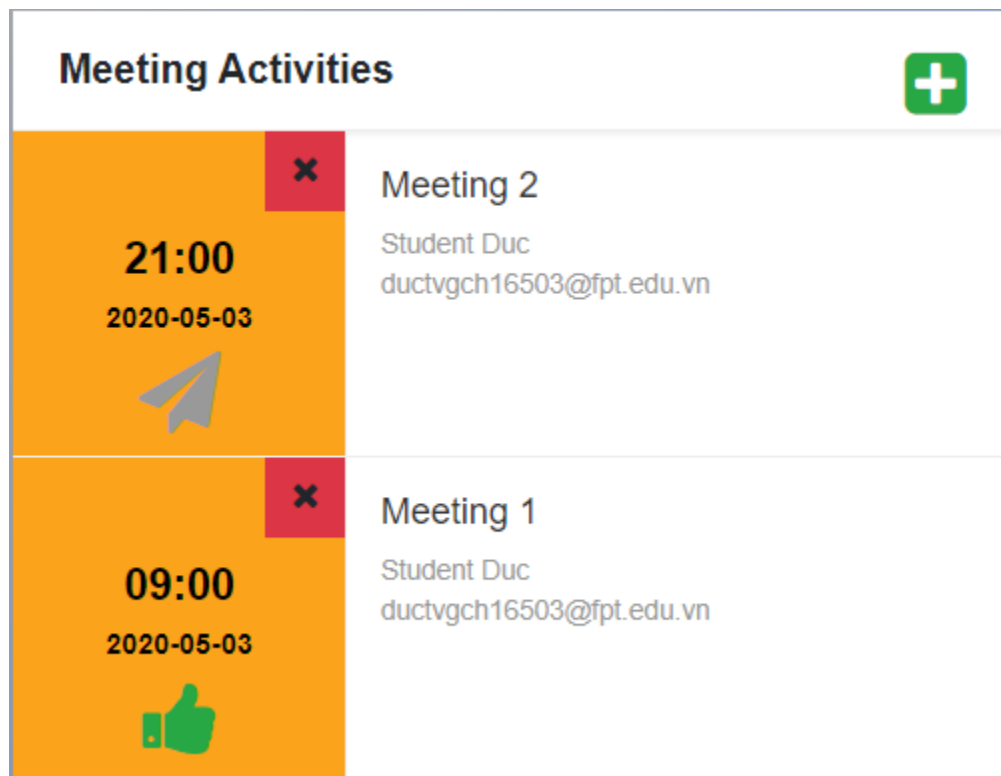


Figure 31: View meeting

After click on it, a creating meeting popup will be show. A meeting consists of 6 information: title, content of meeting, date, time, email and name of student. The meeting will be created after the tutor clicks “**Done!**” Button. If tutor want to stop creating the meeting, the tutor can click the “**Close**” button or the X mark in the upper right corner.

New Meeting×

Title:

Content of meeting...

📅

Date: mm/dd/yyyy

🕒

Time: --:-- --

☐

Waiting for student verify!

Email of student

Name of student

Done!

Close

Figure 32: Create a new meeting

5.3.2 Post

The next is Post, it is located on the middle of webpage. Similar to meeting, to create a new post, tutor must click on plus icon in the upper right corner. Tutor can also delete the post when click on Delete button. Tutor can view other tutor's posts but they can't delete it. Moreover, tutor can love and comment the post.



Văn Đức Tạ

Today at 9:56 AM

Homework for today!



❤ 0 Loves

💬 Comment

✖ Delete

Figure 33: Daily Posts

Tutor can comment and view other comments by clicking in **Comment** button. To return to post, tutor need to click on **Back** button. Moreover, tutor can delete their comment by clicking on red X mark.



Figure 34: Comment in the post

In creating post popup, tutor can write content, upload image and post the post. If tutor want to stop creating the post, tutor can click the **Close** button or the X mark in the upper right corner.

A screenshot of a 'New Post' popup form. The title 'New Post' is at the top left, and a close button (X) is at the top right. Below the title is a text input field containing the placeholder text 'What is your mind?'. Underneath the text field is a file upload section with a 'Choose File' button and the text 'No file chosen'. Below the file upload section is a yellow 'Post' button. At the bottom right of the form is a red 'Close' button.

Figure 35: Create post popup

5.3.3 Contact

Contacts task is located on the menu bar. In contacts task, information of this tutor's students will be showed in there including name, student code, mobile phone, email and last interaction. Moreover, tutor can chat with student by clicking the Contact button in the middle of information table or choose **Messages** task.

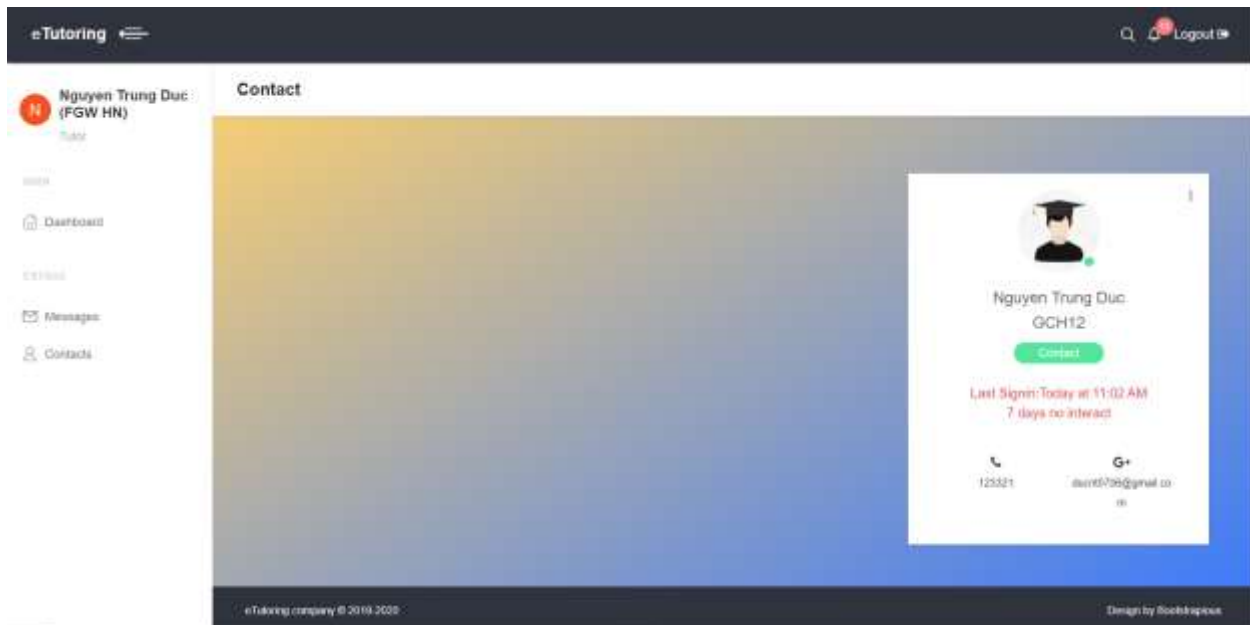


Figure 36: View Student Information

5.3.4 Message

This is the interface of **Messages** task it consists list of student and messages box. Tutor can send message to their students. In the messages box, tutor can delete their sent message by clicking the X mark in the right of this message.

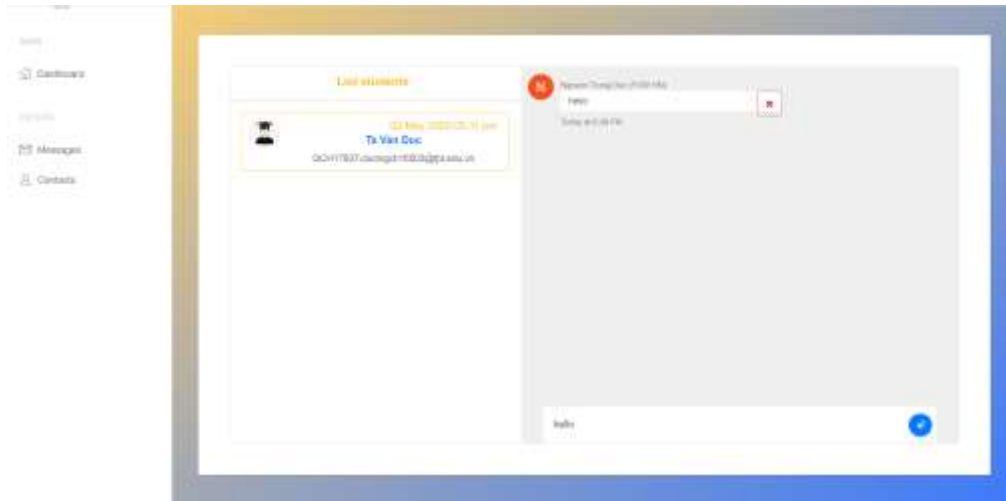


Figure 37: Chat with their student

5.4 Student Interface

As a student, you will also have functions that relate to meeting, post and contact but it has some difference with tutor. The interface of student website is similar tutor's interface.

5.4.1 Meeting

The first is the meeting, students can't add and delete meeting but can confirm the meeting. To confirm meeting, student need click on red check icon, after meeting is confirmed, the plane paper icon will change to like icon.

Meeting Activities	
<div> <div>✓</div> <div>21:00 2020-05-03</div> <div>✈️</div> </div>	<div>Meeting 2</div> <div>Văn Đức Tạ yasukajutsu@gmail.com</div>
<div> <div>✓</div> <div>09:00 2020-05-03</div> <div>👍</div> </div>	<div>Meeting 1</div> <div>Văn Đức Tạ yasukajutsu@gmail.com</div>

Figure 38: Meeting Active

5.4.2 Post

The next is Post, students have less functions than tutor, they can only love and comment their tutor's post. It's mean they don't have creating and deleting meeting.

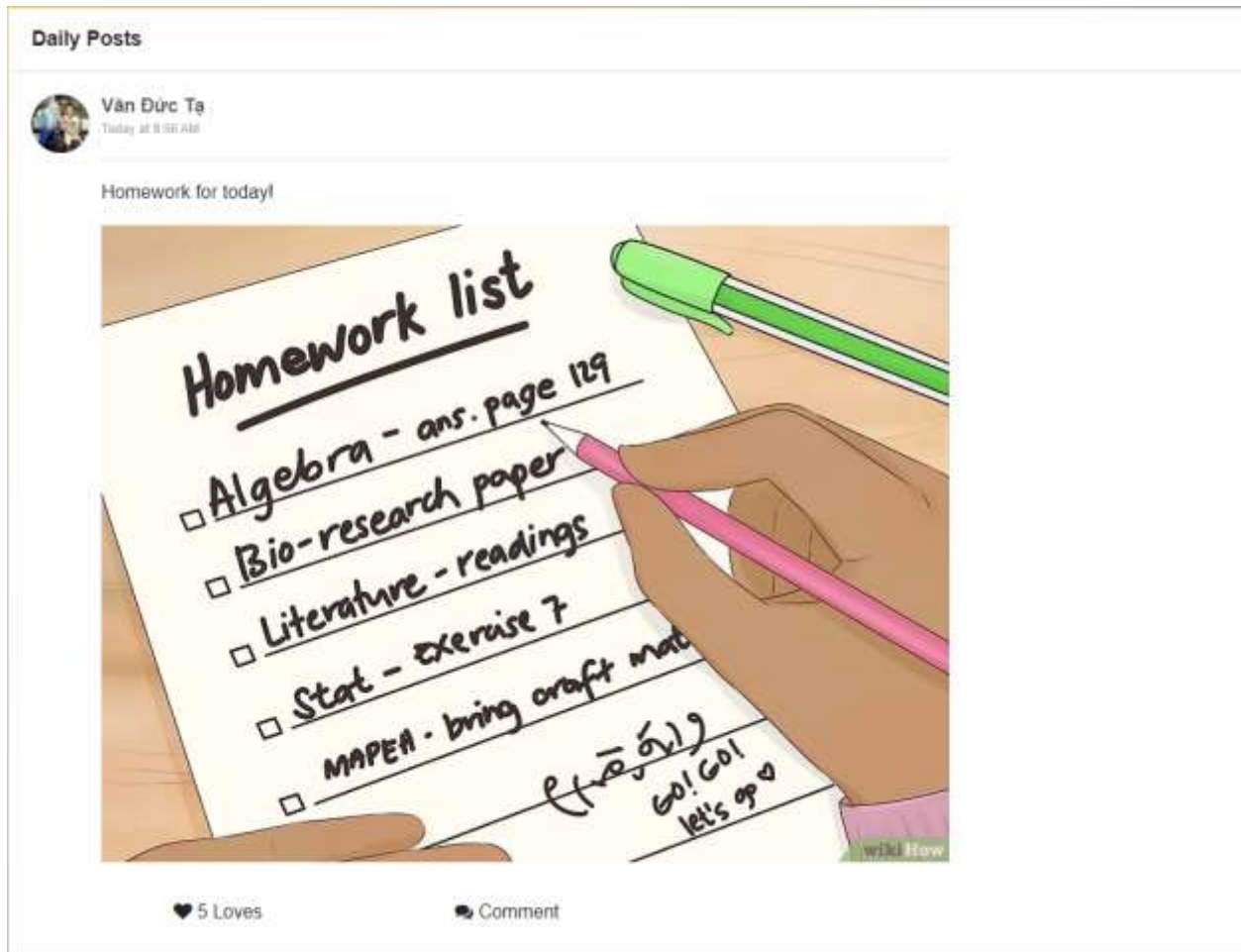


Figure 39: Daily post

To comment and view other comments, student need to click on **Comment** button. To return to post, tutor need to click on **Back** button. Moreover, student can delete their comment by clicking on red X mark.

5.4.3 Contact

In contacts task, information of this student's tutor will be showed in there including name, mobile phone, email and position. Moreover, student can chat with their tutor by clicking the Contact button in the middle of information table.



Figure 40: View Student Information

This is the interface of **Messages** task it consists only the messages box because student has one tutor. Therefore, student can only send message to their tutor. In the messages box, student can delete their sent message by clicking the X mark in the right of this message like tutor.

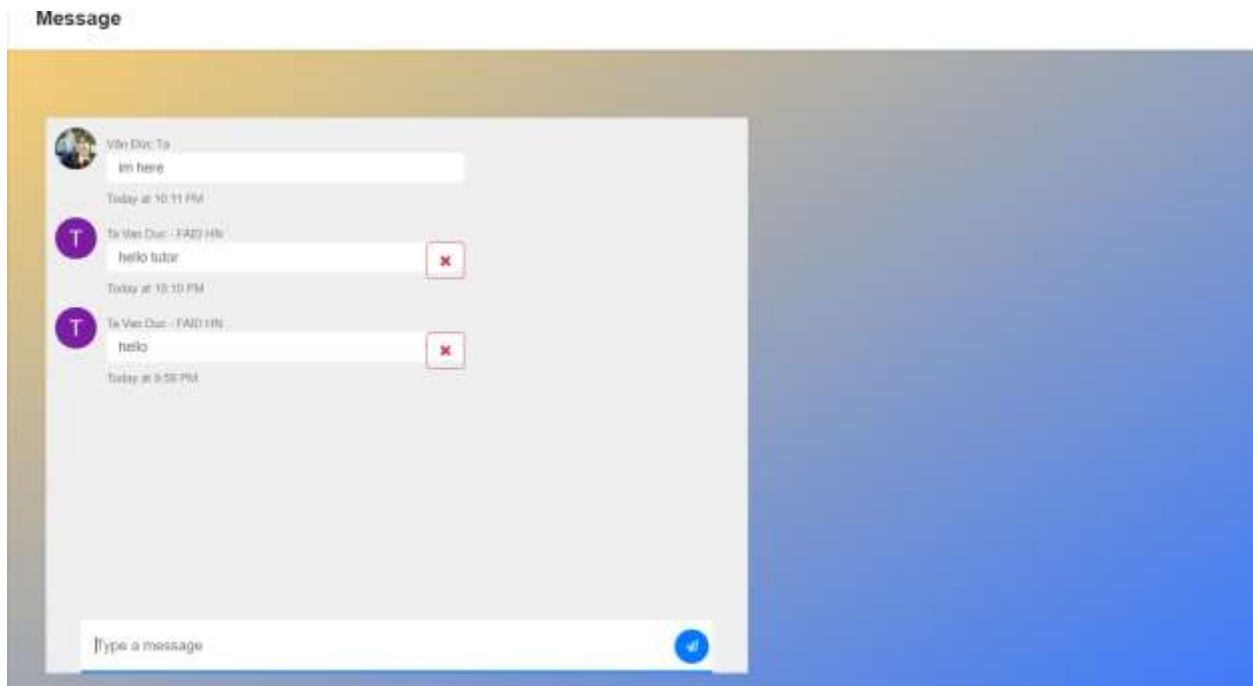


Figure 41: Chat with tutor

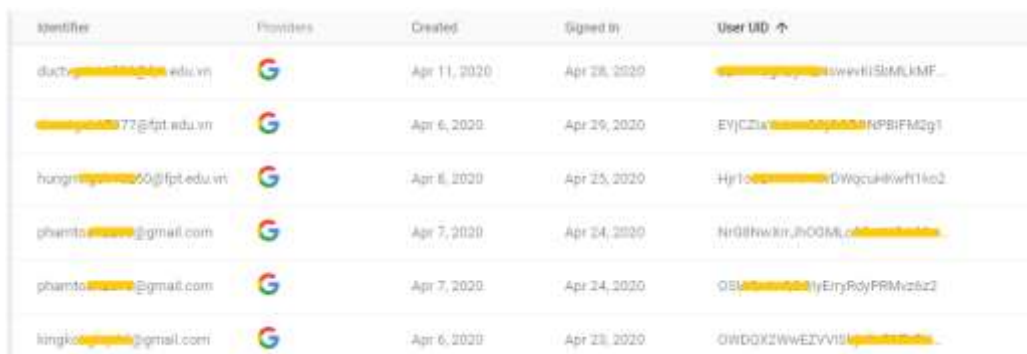
5.5 Non-Functional Requirements

5.5.1 Security

5.5.2 Firebase Realtime Database Rules

Firebase Realtime Database Rules help determine who will read and write access to the system database. This job will be executed automatically continuously, all requests (read, write) will only be done when rules allow. This protects the system database from hackers and from being abused. By default, the system's database rules require Firebase Authentication and only grants full read and write permissions to the authenticated user.

Information includes their unique identifier (UID) as well as associated account data (Figure 42). Will be stored in the auth variable when the user authenticates. By combining Firebase Realtime Database Rules language with authentication information users the system can be able to control access to data based on user identity.









Identifier	Providers	Created	Signed In	User UID
duchy[redacted]@fpt.edu.vn		Apr 11, 2020	Apr 28, 2020	[redacted]wewHISbMkMF...
[redacted]77@fpt.edu.vn		Apr 6, 2020	Apr 29, 2020	EYjCZla[redacted]NPBIFM2g1
hungn[redacted]@fpt.edu.vn		Apr 6, 2020	Apr 25, 2020	Hj1OC[redacted]DWqcuHkwt1ke2
phamto[redacted]@gmail.com		Apr 7, 2020	Apr 24, 2020	NrG8hwkRjhOQM[redacted]
phamto[redacted]@gmail.com		Apr 7, 2020	Apr 24, 2020	QSL[redacted]lyEryRdyPRMvz6z2
kingko[redacted]@gmail.com		Apr 6, 2020	Apr 23, 2020	OWDQXZWWZVVIS[redacted]

Figure 22: User identity

This is an example of how we use the auth variable to ensure that each user can only write to their specific path and cannot write to other users' data.

```

{
  "rules": {
    "users": {
      "$user_id": {
        // grants write access to the owner of this user account
        // whose uid must exactly match the key ($user_id)
        ".write": "$user_id === auth.uid"
      }
    }
  }
}

```

Figure 43: Grants write access

5.5.3 Firebase Security Rules for Cloud Storage

Similar to Database Rules, security is one of the most complex parts of system development. Authentication (who the user is) and authorization (what the user can do) are hard work. To make it easier, I used Firebase Security Rules for Cloud Storage, which made it easier for me to authorize users and validate requests. With this rule it will help me complete the complex processing.

For the example shown in (Figure 44), this system always requires the default Firebase Authentication to perform any read or write operation on all files.

```

rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}

```

Figure 44: Rules for Cloud Storage

5.5.4 Clean Code

The more webpage and functions are created, the more code is written; therefore, our code will be very troublesome if not neatly organized.

Regarding how to organize the code, I break down my code into as many parts as possible. First I will divide it into three main directories consist **admin, student and tutor**. In each directory, I store all the code related to that object and divide their function into separate code directories.

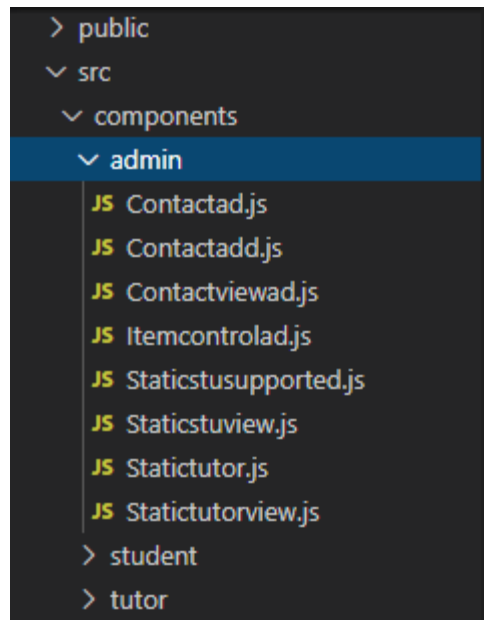


Figure 45: Directory code

Each code directory has only one function and no more than 50 lines of code. Therefore, each code directory is very brief, clear and feels comfortable for readers.

```
src > components > admin > JS Contactviewad.js > Contactviewad
1  import React, { useContext } from 'react';
2  import Contactad from '../Contactad';
3  //TODO: get data form post provider
4  import { ContactContext } from '../../providers/Contactprovider';
5
6
7  const Contactviewad=()=> {
8
9      const contacts=context(ContactContext);
10
11      return (
12          <div className="container-fluid">
13              <div className="row">
14                  {contacts.map(contact => <Contactad {...contact} key={contact.id} />)}
15              </div>
16          </div>
17      );
18  };
19
20  }
21
22  export default Contactviewad;
```

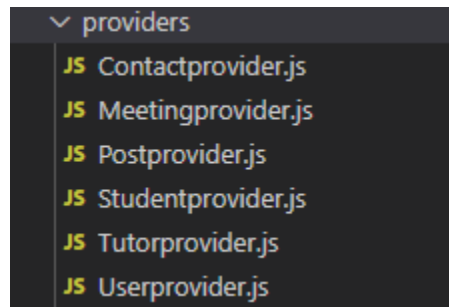
Figure 46: Detailed directory code

5.6 Code Snippets

In this project, we use 3 main libraries and frameworks including reactjs for front-end, firebase for back-end and packages management by nodejs. Firstly, the website interface is built on the component class model of reactjs. This means that the site will be broken down into smaller elements such as the navbar, sidebar, meeting group, and so on. This not only improves the performance of the website, but also makes it easy for programmers to manage the project code.

MVC Structure of website:

We use the main language to build websites which is react according to MVC model with slight variation. However, the concept of MVC remains the same. Accordingly, the web application is divided into three main components Model, View, and Controller. However, as I said above, the names of these components will be changed, but the content will remain the same.



Model folder

```

6  class Tutorprovider extends Component {
7      state = {tutors:[]}
8
9      componentDidMount = () => {
10         firestore.collection('tutors').onSnapshot(snapshot =>{
11             const tutors = snapshot.docs.map(doc => {
12                 return {
13                     tutormobile: doc.data().tutormobile,
14                     tutorgmail: doc.data().tutorgmail,
15                     tutorname: doc.data().tutorname,
16                 }
17             });
18             this.setState({tutors});
19         })
20     }
21
22     render() {
23         const {children}=this.props;
24         return (
25             <TutorContext.Provider value={this.state}>{children}</TutorContext.Provider>
26         );
27     }
28 }
29
30 export default Tutorprovider;

```

Provider Model

Similarly, the next component of the project is View, but in react, we call it screens, screens are user interfaces, or in other words it is the interface between providers and routes (Controller). There is a small change here, the screens will be divided into smaller components called components, this helps the code to be clean and increase the performance of the application.

```

11 class Homepage extends Component {
12     render() {
13         return (
14             <div className="content-inner color-gradient-grey">
15                 <header className="page-header">
16                     <div className="container-fluid">
17                         <h2 className="no-margin-bottom">Dashboard</h2>
18                     </div>
19                 </header>
20                 <div>
21                     <section className="dashboard-counts no-padding-bottom ">
22                         <div className="container-fluid">
23                             <Itemcontrol/>
24                         </div>
25                     </section>
26                     <br><br>
27                     <section className="updates no-padding-top">
28                         <div className="container-fluid">
29                             <div className="row">
30                                 <Postprovider>
31                                     <Postview/>
32                                 </Postprovider>
33                                 <Meetingprovider>
34                                     <Meetingview/>
35                                 </Meetingprovider>
36                             </div>
37                         </div>
38                     </section>
39                 </div>
40                 <Footer/>
41             </div>
42         );
43     }
44 }

```

Screen example (View)

Finally, the routes class corresponds to the controller. The routes handle requests from the user to render the appropriate interface. Besides, it also manipulates data model (screen).

```

12
13 const Tutorroute=()=> {
14     const {user}= useContext(UserContext);
15     return (
16         <BrowserRouter>
17             <div>
18                 <Navbar />
19                 <div className="page-content d-flex align-items-stretch">
20                     <Sidebarad role="Admin" name={user.displayName} photoUrl={user.
21                     <Switch>
22                         <Route exact path="/" component={Homepagead} />
23                         <Route path="/staticstudent" component={Staticstu} />
24                         <Route path="/statictutor" component={Statictutor} />
25                     </Switch>
26                 </div>
27             </div>
28         </BrowserRouter>
29     );
30 }
31
32
33 export default Tutorroute;

```

Route example (Controller)

```

1  import React, { Component } from 'react';
2  import Footer from '../components/Footer';
3  import Contactview from '../components/tutor/Contactview';
4  import Contactprovider from '../providers/Contactprovider';
5
6  class Contact extends Component {
7      render() {
8          return (
9              <div className="content-inner color-gradient-grey">
10                 <header className="page-header">
11                     <div className="container-fluid">
12                         <h2 className="no-margin-bottom">Contact</h2>
13                     </div>
14                 </header>
15                 <div>
16                     <section className="dashboard-counts no-padding-bottom ">
17                         <Contactprovider>
18                             <Contactview/>
19                         </Contactprovider>
20                     </section>
21                 </div>
22                 <Footer/>
23             </div>
24         );
25     }
26 }
27
28 export default Contact;

```

Figure 47: React component class

However, the difficulty is how to control the rendering of components according to the user's wishes. As a workaround, we used the react-router-dom library to manage the navigation whenever the user required to render the interface based on the navigation links.

```

8 //TODO: To import router react
9 import {BrowserRouter, Switch,Route} from 'react-router-dom';
10 //TODO: to import Usercontext
11 import {UserContext} from '../providers/Userprovider';
12
13 const Tutorroute=(()=> {
14     const {user}= useContext(UserContext);
15     return (
16         <BrowserRouter>
17         <div>
18             <Navbar />
19             <div className="page-content d-flex align-items-stretch">
20                 <Sidebarad role="Admin" name={user.displayName} photoUrl={user.
21                 <Switch>
22                     <Route exact path="/" component={Homepagead} />
23                     <Route path="/staticstudent" component={Staticstu} />
24                     <Route path="/statictutor" component={Statictutor} />
25                 </Switch>
26             </div>
27         </div>
28         </BrowserRouter>
29     );
30 }
31
32
33 export default Tutorroute;

```

Figure 48: React-router-dom

Next is the backend part of the website, it is built based on firebase libraries and tools, including services such as authentication (login management), firestore (noSQL database), storage (locate files as images, videos....). In order to connect the website to these tools, we use firebase packages, also known as firebase SDK (Support develop Kit), along with the config file to the project's own address. For bellow example, we used the google authentication API login verification service provided by firebase. We can easily recognize that with only 8-9 lines of code, programmers can create a login function using the Google supplier's pop-up window. Programmers can avoid having to handle user account encryption and passwords in the database ... But ensure website security.

```

1  import firebase from 'firebase/app';
2  import 'firebase/firebase-firestore';
3  import 'firebase/firebase-auth';
4  import 'firebase/firebase-storage';
5
6
7  const firebaseConfig = {
8    apiKey: "AIzaSyBHXS8JoCZL2VLiNclsi2dYi0cwsPcZbv4",
9    authDomain: "etutoring-version1.firebaseio.com",
10   databaseURL: "https://etutoring-version1.firebaseio.com",
11   projectId: "etutoring-version1",
12   storageBucket: "etutoring-version1.appspot.com",
13   messagingSenderId: "664401817574",
14   appId: "1:664401817574:web:299cda2a7fcead9a447027",
15   measurementId: "G-TKDTQJJ9Y8"
16 };
17
18 firebase.initializeApp(firebaseConfig);
19
20 export const firestore=firebase.firestore();
21 export const fireauth=firebase.auth();
22 export const firestorage=firebase.storage();
23
24 export const provider = new firebase.auth.GoogleAuthProvider();
25 export const signInWithGoogle = () => fireauth.signInWithPopup(provider);
26 export const signOut = () => fireauth.signOut();
27
28 export const storage=firebase.storage();
29
30 export default firebase;

```

Figure 49: Firebase SDK config

```

authListener = () => {
  fireauth.onAuthStateChanged((user) => {
    if (user) {
      this.setState({user});
    } else {
      this.setState({user:null});
    }
  });
}

```

Figure 50: Firebase authentication

Similarly, throughout the project we can easily handle problems posed as storing user data in the form of a noSQL statement using firestore, or storing files in storage. Everything in the back-end is well supported by firebase.

```
35     handleCreate = (e) => {
36         e.preventDefault();
37         var file=this.state.file;
38         var post = {
39             content: this.state.content,
40             loves: 0,
41             time: new Date(),
42             imageUrl: null,
43             tutorPictureurl: this.getPhotoUrl(),
44             tutorgmail: this.getGmail(),
45             tutorname: this.getUserName()
46         }
47         firestore.collection('posts').add(post).then(function(postRef){
48             // Upload the image to cloud
49             var filePath = firebase.auth.currentUser.uid + '/' + postRef.id + '/' + file.name;
50             return firebase.storage.ref(filePath).put(file).then((fileSnapshot) => {
51                 // Generate a public URL for the file.
52                 return fileSnapshot.ref.getDownloadURL().then((url) => {
53                     // Update the chat message placeholder with the image's URL.
54                     return postRef.update({
55                         imageUrl: url,
56                         storageUri: fileSnapshot.metadata.fullPath
57                     });
58                 });
59             });
60         }).catch(function (error) {
61             console.error('There was an error uploading a file to Cloud Storage:', error);
62         });
63         this.toggleModal();
```

Figure 51: Firebase storage and Fire store

In the example above, to store information from the user's daily posting including the post's content, and the post's file. We used the Firestore to save fields such as content, love drop times, creation time, creator information, and image links to a collect as "posts." Each post will correspond to a document and they will be numbered. automatic and unique. Next the file will be storage in firebase storage, and the public URL in the Firestore for document file will be updated

6) Scrum Document

In this report I will show you my understanding about Scrum Component. There are 3 main sections: People, Ceremonies and Artifact. However, as required, I will not cover all of their components, but their main characteristics. According to (Software Testing Help, 2020)

➤ **People (Team):**

- ✓ A typical Scrum team usually has 5-10 members with different functions creating a variety such as QA, Programmer, UI Designer, etc.
- ✓ Team members must work full time to exchange information and complete functions.
- ✓ Teams organize themselves and divide specific tasks together. In the process of working will often exchange and check progress.
- ✓ After each sprint is completed, team members can choose to change their work.

➤ **Ceremonies (Meeting):**

➤ **Sprint Planning Meeting:**

- ✓ Sections of the Sprint Planning meeting
- ✓ The first task when meeting is to identify the Sprint goals that the team needs to accomplish in the project.
- ✓ Target audience: Product owner, Scrum Master and Scrum Team
- ✓ Participants: The Scrum Master and the Scrum Team will be directly involved in the project development process
- ✓ Create a backlog Sprint

➤ **Sprint Review Meeting:**

In this section team members will present what they accomplished during the sprint stage. Usually, demos will show demos of new features or highlight difficulties encountered during implementation.

Meeting participants are Clients, Management, Product owner and Other engineers.

➤ **Sprint Retrospective Meeting:**

The team continued to discuss with three main questions about what work has been completed, what is continuing to be done and which work will start next. For example, in practice, when building a website, there are team members in charge of designing UI. then the content of the report will be yesterday, what features have been completed yesterday, what features will continue to be built today, and what are they intended to add in the future. After the individual reports can then synthesize information, evaluate the completion of the project.

➤ **Artifacts:**

According to (Adams, n.d.)

✓ **Product Backlog:**

We can understand that it is a list of all the work to be done in the project. Often a combination of story-based work (online allows users to search and replace) and task-based work (improved exception handling). Requirements for a system, expressed as one Priority list of backlog items. Usually created in the Sprint Plan. During the implementation of sprint backlogs can be changed and prioritized before each.

✓ **Sprint Backlog:**

In my personal opinion, the purpose that sprint backlogs is created to serve the achievement of the sprint goals that the team set when implementing a project.

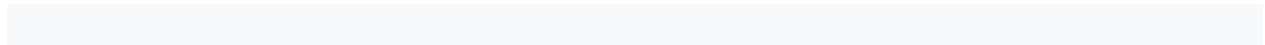
✓ **Burn Down Chart:**

This is a tool to help honestly depict the total Sprint remaining time after each implementation day. In other words, it shows the estimated amount of time the team can complete the work to be done. Of course, in order for the project to achieve the best results, it is best for the team to speed up the time to complete the work because the process of implementing the activities is not always as desired, there are jobs that will not work. could be completed on time due to misjudgment of the difficulty level or encountering other objective problems that could take longer to complete. Therefore, we need to accelerate to complete other tasks to compensate for the extended time period. This helps to ensure that the system can be delivered on time.

7) Apply The Scrum With Agile Method

7.2 Product Backlog

At the request of the project, our team has divided the work to improve the functions of the system. But because we have neither experience nor knowledge, there are some functions that cannot be completed. So in this report I will show only the functions that the group has accomplished and show examples.



A	B	C	D	E	F
ID	AS a/an	I want to ...	So that...	Priority	Sprint
1	As a student	I want to have a personal tutor to guide me	So that I can contact to tutor	Must have	4
2	As a student	I want to messaging to personal tutor	So that I can exchange with tutor	Must have	5
3	As a student	I want to my data is accessed from the university MIS system.	So that I can login/logout	Must have	3
4	As a student	I want to each student will have their own personal dashboard summarising their interaction with their personal tutor.		Must have	1
5	As a student	I want to verify meetings	So that I can be sure to join meeting	Should have	6
6	As a student	I want to commenting on tutor post	So that I can interact with tutor	Should have	7
7	As a student	I want to get notification of events recorded meeting		Could have	9
8	As an authorized member	I want to dashboards to manage student and tutor		Must have	2
9	As an authorized member	I want to llocate or reallocate personal tutors to students.	So that I can manage contact	Must have	4
10	As an authorized member	I want to The student and the personal tutors will get notification emails when this happens		Should have	8

Figure 52: Product Backlog Example

11	As an authorized member	I want to assign to many students at a time to tutor	So that I can save time	Should have	8
12	As an authorized member	I want to filtering students without a personal tutor		Could have	11
13	As an authorized member	I want to filtering students with no interaction		Could have	11
14	As an authorized member	I want to be Notify when a student is absent		Could have	11
15	As a tutor	I want to arrange meetings		Must have	6
16	As a tutor	I want to messaging to students	So that I can exchange with student	Must have	5
17	As a tutor	I want to each personal tutor will have a dashboard of their personal tutees that can be sorted and filtered appropriately		Must have	1
18	As a tutor	I want to my data is accessed from the university MIS system.	So that I can login/logout	Must have	3
19	As a tutor	I want to make of post for uploading document and commenting on them and for blogging	So that I can notify all students	Should have	7
20	As a User	I want to the interface must be suitable for all devices (eg mobile phones, tablets, desktops)		Should have	10
21	As a tutor	I want to get notification when student verify meeting		Could have	9

Figure 53: Product Backlog Example

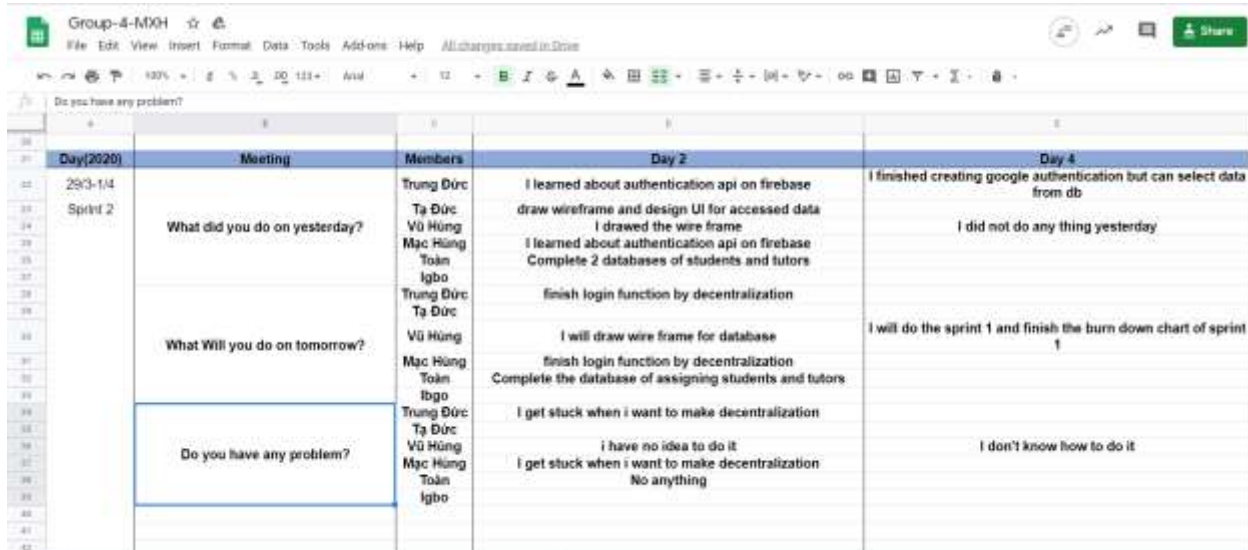
As you can see in the table we have listed all the features that the system needs to ensure the needs of students, tutors and admin management. We also rely on the scrum model to divide the difficulty level for each task we need to perform. It is divided into three levels: Must have, should have and could have. Items rated as Must have and Should have will be prioritized by the priority group. As for the items in the have section, we will try to accomplish as much as we can and within time. After evaluating the priority for each task we will add it to the sprint according to the priority to complete the product.

7.3 Daily Of The Meeting

In the next section, I will show you a list of the meetings that the group has done throughout the production process and the work the members perform as well as the difficulties we face to prove. that my team has followed the scrum model to finish out product.

Group-4-MXH					Share
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive					
100% 123 Arial 12					

Figure 54: Daily meeting in sprint 1



Day(2020)	Meeting	Members	Day 2	Day 4
29/3-1/4	Sprint 2			
		What did you do on yesterday?	Trung Đức: I learned about authentication api on firebase Tạ Đức: draw wireframe and design UI for accessed data Vũ Hùng: I drew the wire frame Mạc Hùng: I learned about authentication api on firebase Toàn: Complete 2 databases of students and tutors Igbo: finish login function by decentralization	I finished creating google authentication but can select data from db I did not do any thing yesterday
		What Will you do on tomorrow?	Trung Đức: I will draw wire frame for database Tạ Đức: finish login function by decentralization Vũ Hùng: Complete the database of assigning students and tutors Mạc Hùng: I get stuck when i want to make decentralization Toàn: i have no idea to do it Igbo: I get stuck when i want to make decentralization No anything	I will do the sprint 1 and finish the burn down chart of sprint 1 I don't know how to do it
	Do you have any problem?			

Figure 55: Daily meeting in sprint 2

A	B	C	D	E
Day(2020)	Meeting	Members	Day 2	Day 4
2/4-5/4		Trung Đức		
Sprint 3	What did you do on yesterday?	Tạ Đức	learn how to design message UI for student and draw wireframe for it	i finished designing message UI for student and drawing wireframe
		Vũ Hùng	Continued do the report group	I finished the burn down chart of sprint 3
		Mạc Hùng		
		Toàn		
		Igbo		
	What will you do on tomorrow?	Trung Đức	design message UI for student and draw wireframe	
		Vũ Hùng	Finish the burn down chart for sprint 3	I will assign the user interface in our report
		Mạc Hùng		
		Toàn		
		Igbo		
	Do you have any problem?	Trung Đức	nothing	
		Tạ Đức	There are many information that i have to add in report	Maybe I don't have problem
		Vũ Hùng		
		Mạc Hùng		
		Toàn		
		Igbo		

Figure 56: Daily meeting in sprint 3

A	B	C	D	E
Day(2020)	Meeting	Members	Day 2	Day 4
6/4-9/4		Trung Đức	i got understand about noSQL database	I finished meeting function
Sprint 4	What did you do on yesterday?	Tạ Đức	learn how to design meeting UI for student and draw wireframe for it	i finished designing meeting UI for student and drawing wireframe
		Vũ Hùng	I finished assign the user interface image in report	I finished the burn down chart of sprint 4
		Mạc Hùng		
		Toàn		
		Igbo		
	What will you do on tomorrow?	Trung Đức	I will find out firestore query	
		Tạ Đức	design meeting UI for student and draw wireframe	I will do the scrum documents
		Vũ Hùng	I will finish the burn down chart of sprint 4	
		Mạc Hùng		
		Toàn		
		Igbo		
	Do you have any problem?	Trung Đức	i get stuck when i select query in firestore	
		Tạ Đức	nothing	No problems
		Vũ Hùng		
		Mạc Hùng		
		Toàn		
		Igbo		

Figure 57: Daily meeting in sprint 4

- Members design front - end create branch wireframe on Github
- Add members to Firebase (User and permissions).
- Adding documents in Trello.
- Talk about types of meeting.
- The members regularly update daily Sprint.

Day(2020)	Meeting	Members	Day 2	Day 4
10/4-13/4		Trung Đức	I learned about firestore to save fire in storage	I finished posting function without no file
Sprint 5		Tạ Đức	I researched how to design ERD	I finished designing ERD
	What did you do on yesterday?	Vũ Hùng	I finished the scrum documents	I finished the burn down chart of sprint 5
		Mạc Hùng		
		Toàn		
		Igbo		
		Trung Đức	Continue learning firebase storage	
		Tạ Đức	design ERD	
	What will you do on tomorrow?	Vũ Hùng	I will draw the burn down chart of sprint 5	
		Mạc Hùng		
		Toàn		
		Igbo		
		Trung Đức	I can't get file from input	
		Tạ Đức		
	Do you have any problem?	Vũ Hùng	I don't know how to evaluate	
		Mạc Hùng		
		Toàn		
		Igbo		

Figure 58: Daily meeting in sprint 5

- Members design front - end update wireframe on Github.
- Members agreed to set id for document in database.
- The members regularly update daily Sprint.

Day(2020)	Meeting	Members	Day 2	Day 4
14/4-17/4		Trung Đức	I researched tutor's functions	I finished my work in this sprint
Sprint 6		Tạ Đức		
	What did you do on yesterday?	Vũ Hùng		
		Mạc Hùng	Complete client-side programming for showing unsupported students and tutors lists	
		Toàn		
		Igbo		
		Trung Đức	I will draw tutor's functions and describe for it	
		Tạ Đức		
	What will you do on tomorrow?	Vũ Hùng		
		Mạc Hùng	Completing client-side programming for showing Contacts lists of students who do not yet have support with tutors	
		Toàn		
		Igbo		
		Trung Đức		
	Do you have any problem?	Tạ Đức		
		Vũ Hùng		
		Mạc Hùng	Having trouble with the lifecycle of ReactJs	
		Toàn		
		Igbo		

Figure 59: Daily meeting in sprint 6

	Date	Meeting	Members	Day 2	Day 4
120	Day(2020)				
121	15/4-21/4				
122	Sprint 7				
123		What did you do on yesterday?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	i researched student's functions I did not do any thing Complete retrieving data of students and tutor from database	i finished drawing student's functions and description for it I finished draw the burn down chart of sprint 6
124		What will you do on tomorrow?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	I will draw student's functions and describe for it I will finish the burn down chart of sprint 6 Complete retrieving the contact data between students and tutors and adding data to the database	I will finished the burn down chart of sprint 7
125		Do you have any problem?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	I have difficulty with complex firebase queries	I have some error when i finish the sprint

Figure 60: Daily meeting in sprint 7

- Members review the role of sprint, continue to complete the functionality that is not yet complete.
- Set goals for the week of completing the necessary functions.

	Date	Meeting	Members	Day 2	Day 4
120	Day(2020)				
121	22/4-25/4				
122	Sprint 8				
123		What did you do on yesterday?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	i learned about route and useContext in react i researched our code I finished the burndown chart of sprint 7	I finished my code I described how our code is clean I finished the burn down chart of sprint 8
124		What will you do on tomorrow?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	I try my best to find solution i describe how our code is clean I will finished the form and check the sprint 8	I will finish the burn down chart of sprint 9
125		Do you have any problem?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	I get stuck to get param from link Some people did not done their sprint	Maybe not

Figure 61: Daily meeting in sprint 8

Done posting can post photos that can like post. Not being able to render the posts according to the tutor's contact from the student's side, students cannot comment on the tutor's post, can't post other resources besides the image. In the near future, there will be a comment section below the post and additional documents!

Day(2020)	Meeting	Members	Day 2	Day 4
25/4-29/4	What did you do on yesterday?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	Finished the Burn down chart for sprint 9	Finished the Burn down chart
	What will you do on tomorrow?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	Do the Evaluation of report	I will check the form and content and fix bugs
	Do you have any problem?	Trung Đức Tạ Đức Vũ Hùng Mạc Hùng Toàn Igbo	I think I don't have any problem	No problem

Figure 62: Daily meeting in sprint 9

The above is a summary of all the meetings of our team during the production process. We divided into 9 sprints and in each 4-day sprint we will meet twice, about 30 minutes each time for team members to talk and talk about what we have done today. first, what to do next and if there are any difficulties for the whole support team to overcome. During the meeting we also set out the next major goals that the group needs to accomplish. Some of the detailed examples have been listed above.

7.4 Sprint Backlog

User Story	Tasks	Member	Status	Estimated Hours of Completion	Day 1	Day 2	Day 3	Day 4	Review
As a tutor, I want to each personal tutor will have a dashboard of their personal tutees that can be sorted and filtered appropriately	Draw wireframe	Chinedu	Done	25x35 - 2 hours	2	0	0	0	0
	Front-end coding for dashboard	Chinedu	Done	20 hours	10	10	5	0	0
	Create database	N.T.D	Done	20 hours	20	15	10	0	0
	Back-end coding	N.T.D	Done	30 hours	30	20	10	0	0
As a student, I want to each student will have their own personal dashboard summarising their interaction with their personal tutor.	Testing	Chinedu	Done	2 hours	2	2	2	0	0
	Draw wireframe	T.V.D	Done	2 hours	2	0	0	0	0
	Front-end coding	T.V.D	Done	9 hours	5	4	0	0	0
	Create database	M.H	Done	5 hours	5	0	0	0	0
As an authorized member, I want to dashboards to manage student and tutor	Back-end coding	M.H	Done	14 hours	6	5	0	0	0
	Testing	T.V.D	Done	1 hours	1	0	0	0	0
	Draw wireframe	Hung Vu	Done	2 hours	2	0	0	0	0
	Front-end coding	Hung Vu	Done	10	10	4	3	0	0
	Create database	Toan	Done	2	5	5	0	0	0
	Back-end coding	Toan	Done	20	15	12	7	0	0
	Testing	Hung Vu	Done	2	2	0	0	0	0

Figure 63: Sprint 1

Group-4-MXH ☆										
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive										
100% Arial 11										
	A	B	C	D	E	F	G	H	I	J
1	User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
2	As a tutor, I want to my data is accessed from the university MIS system.	Draw wireframe	Chinedu	Done	2 hours	1	0	0	0	0
3		Front end coding	Chinedu	Done	5 hours	2	0	0	0	0
4		API Making	N.T.D	Done	5 hours(31/3/2020)	5	0	0	0	0
5		Back-end coding	N.T.D	OnGoing	10 hours	5	5	5	3	3
6		Testing	Chinedu	Not yet	1	1	1	1	1	1
9	As a student, I want to my data is accessed from the university MIS system.	Draw wireframe	T.V.D	Done	3 hours	3	0	0	0	0
10		Front end coding	T.V.D	Done	3 hours	3	0	0	0	0
11		API Making	M.H	Done	4 hours	3	1	0	0	0
12		Back-end coding	M.H	Done	5 hours	2	2	1	0	0
13		Testing								
16	As an authorized member , I want to my data is accessed from the university MIS system.	Draw Wire frame	Hung Vu	Done	2 hours	2	0	0	0	0
17		Front-end Coding	Toan	Done	5	5	3	1	0	0
18		Back-end Coding	Toan	Done	10	10	8	5	0	0
19		Create Database	Toan	Done	3	3	0	0	0	0
20		Test	Chindo	Done	1	1	1	1	0	0

Figure 64: Sprint 2

Group-4-MXH ☆										
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive										
100% Arial 12										
	A	B	C	D	E	F	G	H	I	J
1	User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
2	As a tutor, I want to messaging to students (Base on student code of M.X.H)	Draw wireframe	chinedu	Done	2	1	1	0	0	0
3		Front end coding	chinedu	Done	5	3	5	3	2	0
4		Create collection database	N.T.D	Done	10	7	5	3	0	0
5		Back end coding	N.T.D	Not Done	12/04/2020 - 20 hours	20	20	10	10	0
6		storage file (firebase storage)	N.T.D	Done	10	10	12	5	1	0
11	As a student, I want to messaging to personal tutor	Draw wireframe	T.V.D	Done	2 hours	2	0	0	0	0
12		Front end coding	T.V.D	Done	8 hours	3	5	0	0	0
13		Create collection database	M.H	Done	12/04/2020 - 20 hours	3	4	4	0	0
14		Back end coding	M.H	Done	12/04/2020 - 5 hours	1	3	1	0	0
15		storage file (firebase storage)	M.H	Done	12/04/2020 - 10 hours	3	5	0	0	0

Figure 65: Sprint 3

Group-4-MXH ☆

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

100% 123 Arial 12

User Story

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As a tutor, I want to arrange meetings (Base on code of N.T.D)	Draw wireframe	chinedu	Done	3 hours	3	0	0	0	0
	Font end coding	chinedu	Done	9 hours	7	3	0	0	0
	Create collection database	N.T.D	Done	10 hours	1	0.5	0	0	0
	Back end coding	N.T.D	Done	10 hours	5	8	5	0	0
	Test log	chinedu	Done	2	2	2	2	0	0
As a student, I want to verify meetings	Draw wireframe	TVD	Done	3 hours	3	0	0	0	0
	Font end coding	TVD	Done	15 hours (19/4/2020)	3	4	3	0	0
	Create collection database	M.H	Done	10 hours (19/4/2020)	10	6	3	0	0
	Back end coding	M.H	Done	15 hours (19/4/2020)	5	2	3	0	0

Figure 66: Sprint 4

Group-4-MXH ☆

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

100% 123 Arial 12

User Story

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As a tutor, I want to make of post for uploading document and commenting on them and for blogging	Draw wireframe	Chinedu	Done	1	1	0	0	0	0
	Font end coding	Chinedu	Done	10	10	0	0	0	0
	Create collection database	N.T.D	Done	(23/04/2020) 10	10	5	5	3	0
	Back end coding for posting	N.T.D	Not Done	20 hours	30	20	10	0	0
	Setting save file storage	N.T.D	Done	20 hours	20	25	10	0	0
	Coding comment function	N.T.D	Not Done	20 hours	20	14	8	0	0
	Test	Chinedu		1	1	1	1	0	0

Figure 67: Sprint 5

Group-4-MXH ☆

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

100% 123 Arial 12

User Story

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As an authorized member, I want to locate or reallocate personal tutors to students.	Front end Coding	Toan	Done	5 hours	5	2	0	0	0
	Back-end coding	Toan	Done	5 hours	5	2	0	0	0
	Testing	Chinedu	Done	1 hours	1	1	0	0	0
As an authorized member, I want to assign to many students at a time to tutor	Font end coding	Toan	Done	2 hours	2	0	0	0	0
	Back end coding	Toan	Done	5 hours	5	2	0	0	0
	Testing	chinedu	Done	1 hours	1	1	0	0	0
As an authorized member, I want to filtering students without a personal tutor	Front end code	Toan	Done	5 hours	5	3	1	0	0
	back end coding	Toan	Done	5 hours	8	4	2	0	0
	Testing	chinedu	Done	1 hours	1	1	1	0	0

Figure 68: Sprint 6

User Story	Tasks	member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As an authorized member ,I want to be Notify when a student is absent	Front end code	Toan		5 hours	5	3	0	0	0
	back end code	Toan		5 hours	5	2	0	0	0
	Testing	Chinedu		1 hours	1	1	0	0	0
As an authorized member ,I want to filtering students with no interaction	Front end code	Toan	Done	5 hours	5	2	0	0	0
	back end code	Toan	Done	15 hours	15	10	4	0	0
	Testing	Chinedu	Done	1 hours	1	1	1	0	0

Figure 69: Sprint 7

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As a student,I want to commenting on tutor post	Front end code	M.X.H	Done	1/05/2020 2 hours	2	0	0	0	0
	Back end code	Duc	Done	5 hours	5	1	0	0	0
	Testing	Chinedu	Not yet	1 hours	1	1	0	0	0

Figure 70: Sprint 8

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
As a student, I want to get notification of events recorded meeting	Front end code	M.X.H	Done	2/05/2020 2 hours	2	0	0	0	0
	Back end code	N.T.D	Done	5 hours	5	0	0	0	0
	Testing	Chinedu	Done	1 hours	1	0	0	0	0

Figure 71: Sprint 9

User Story	Tasks	Member	Status	Estimate	Day 1	Day 2	Day 3	Day 4	Review
I want to the interface must be suitable for all devices (eg mobile phones, tablets, desktops)	Front end code	T.V.D	Done	5 hours	5	0	0	0	0
	Testing	Chinedu		1	1	0	0	0	0

Figure 72: Sprint 10

Above are detailed photos of our team's sprint completion process. In these tables we list in detail the tasks that each member must do in sprint. It helps us know how much time we have spent on each job to monitor the completion of the product.

As you can see, most of the work is completed by the members on time. In the next section, there will be burn down charts that help us to evaluate the process as well as the timing.

7.5 Burndown Chart

Day 1, Day 2, Day 3, Day 4 and Review

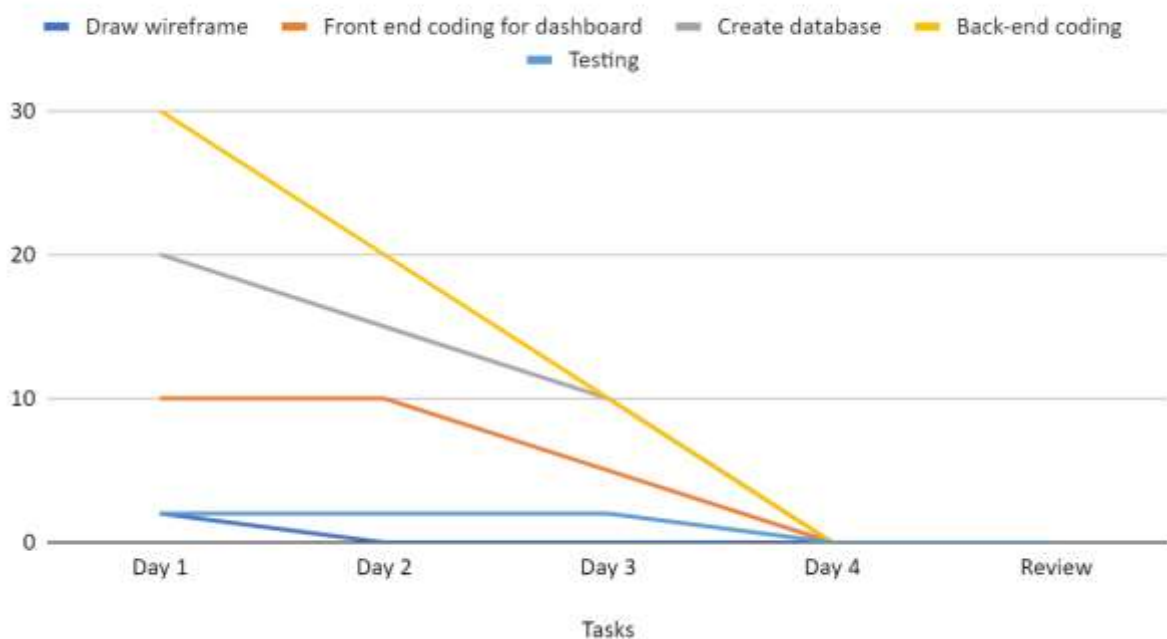


Figure 73: Burn down chart of sprint 1

Day 1, Day 2, Day 3, Day 4 and Review

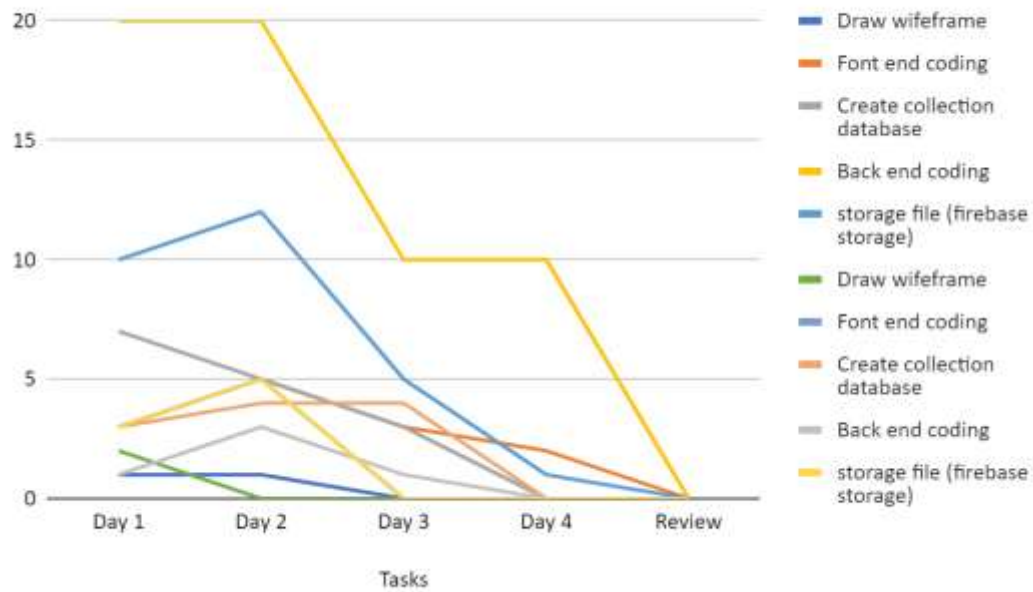


Figure 74: Burn down chart of sprint 2

Day 1, Day 2, Day 3, Day 4 and Review

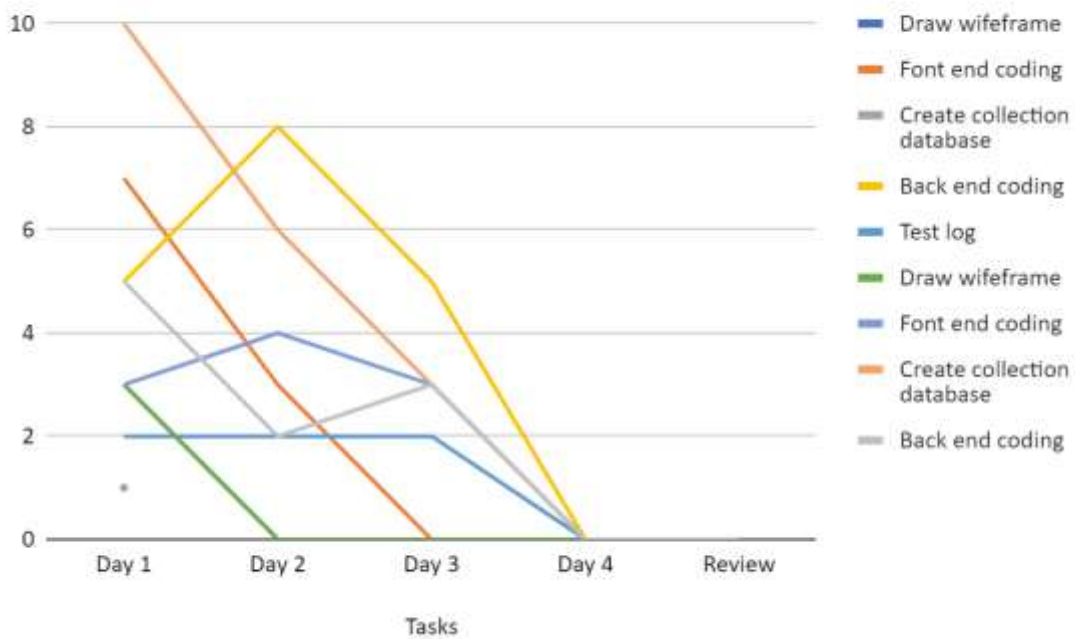


Figure 75: Burn down chart of sprint 3

Day 1, Day 2, Day 3, Day 4 and Review

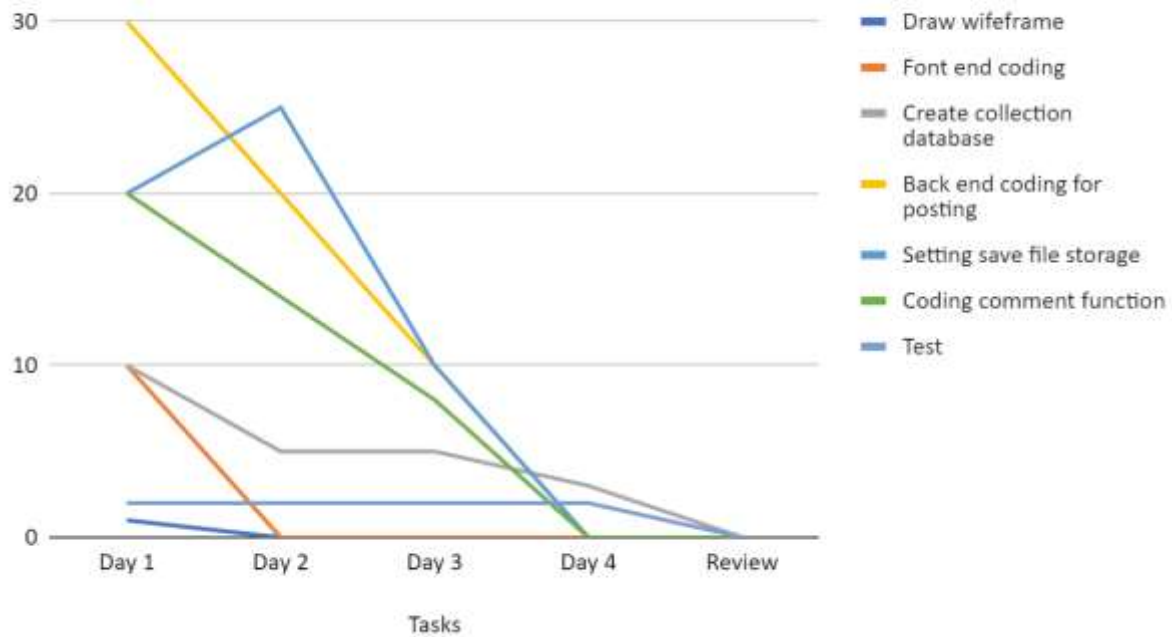


Figure 76: Burn down chart of sprint 4

Day 1, Day 2, Day 3, Day 4 and Review

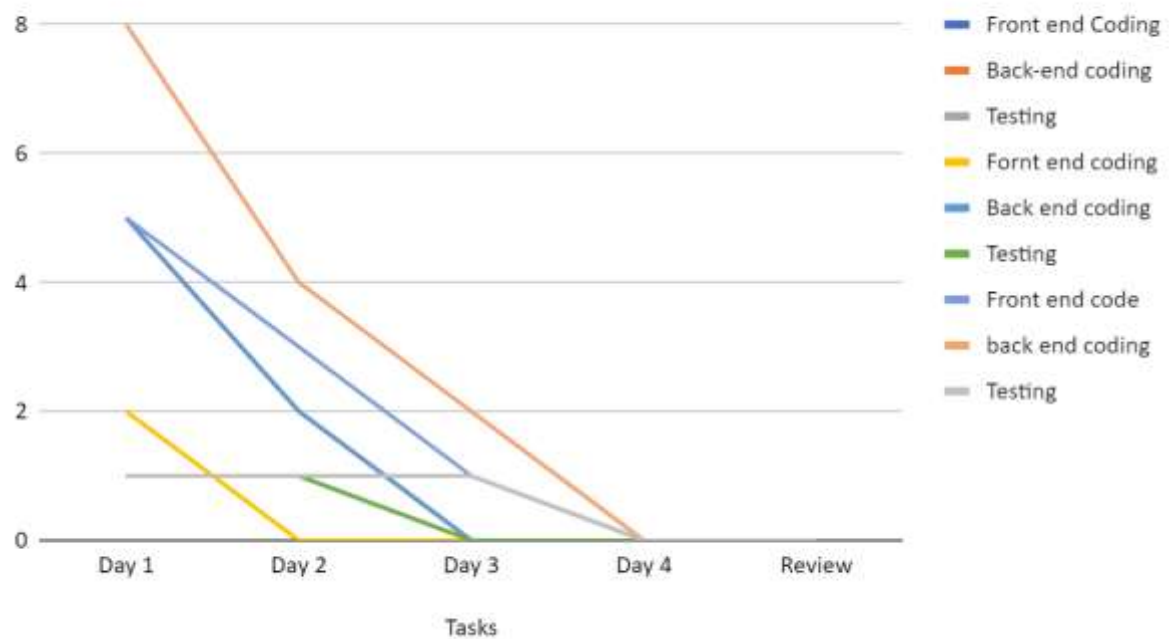


Figure 77: Burn down chart of sprint 5

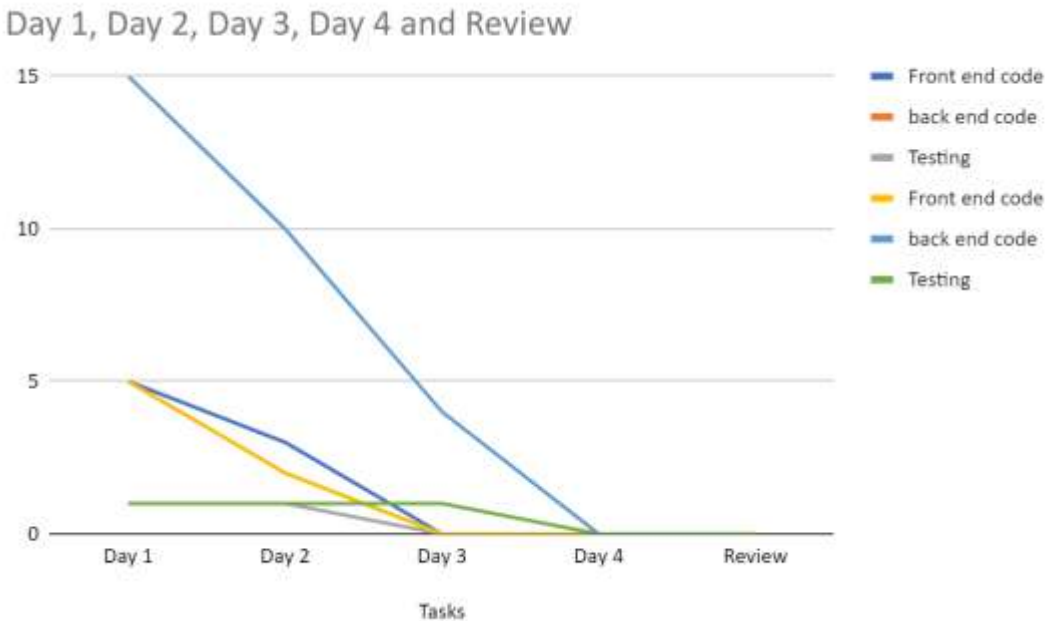


Figure 78: Burn down chart of sprint 6

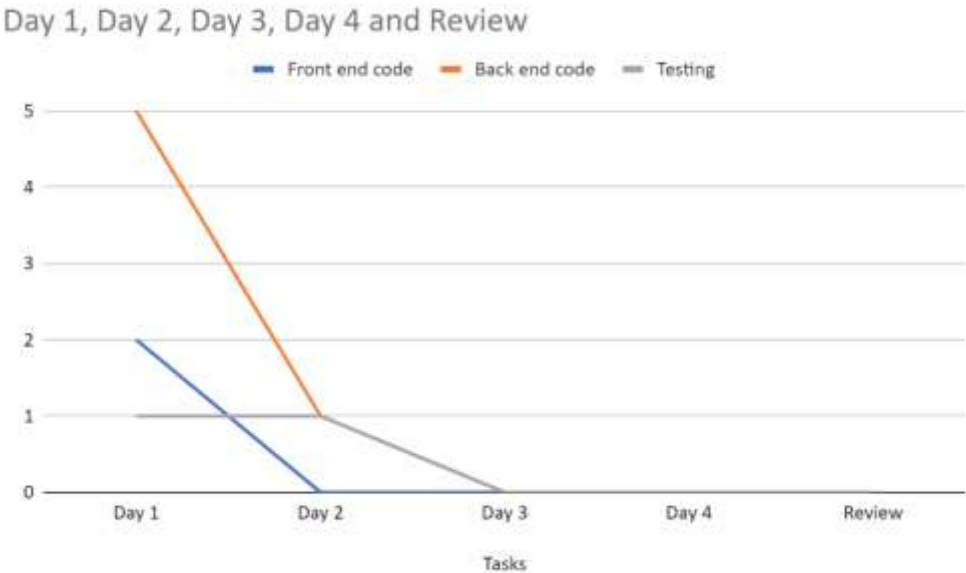


Figure 79: Burn down chart of sprint 7

Day 1, Day 2, Day 3, Day 4 and Review

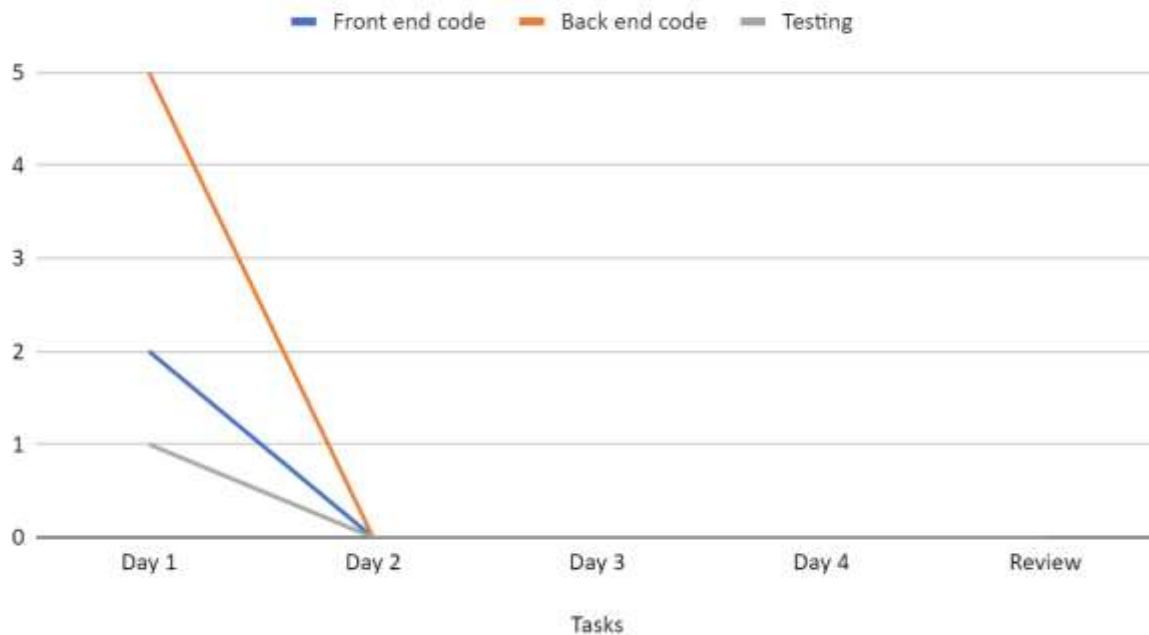


Figure 74: Burn down chart of sprint 8

The importance of burn down charts is undeniable to the process of building our web site. After completing one sprint we take the time to create a burn down chart. It gives us a better overview of what we have done and helps us to assess how much time our team has spent completing that sprint. This helps us assess whether we are on schedule or slowing down. If slow, we must take measures to complete the other sprint earlier to ensure progress with customers. Luckily we don't have any sprint over time. There have been times when the burn down chart has gone up, but we have made it in time to ensure that the product is on schedule.

8) Testing

In this section, I will show hands-on experience and product reviews. In our team, there are 2 members in charge of testing the product so that we can evaluate if there are any bad areas that need to be improved or detect errors during use so we can fix them. before putting the product into the user's hands. Below I will show you some of the tests we have done on our products.

8.1 Test Item

The website will be tested in localhost and in Kingsley's computer.

8.2 Test List of Function

The functions will be tested according to their roles and their various interface:

Login	All the user	Log-in page
View the List of all the users	Admin	User list page
View the tutor profile and Student Profile	Admin, Tutor that is assign to a class, Student that is assign to that class	User profile page
View list of meeting	Admin	Meeting list
Edit user profile	Admin	Admin Dashboard
Assigning of tutor and student	Admin	Admin Dashboard
Tutor and student change	Admin	Admin dashboard
Add, update and delete meeting	Tutor	Tutor page
View the list of Meeting Post	Tutor, Student assigned to the meeting, Admin	Admin Dashboard
Add post and attaching of file	Tutor, Student assigned to the meeting	Tutor dashboard
Edit post in meeting	Tutor or the student that posted	
Creating Schedule	tutor	Tutor Dashboard
Deleting Schedule	Tutor	Tutor Dashboard
Notification when a schedule is created, edited or deleted	Student in that meeting and tutor assigned	Student Dashboard

Pass / Fail criteria

These criteria will be evaluated according to the actual to the actual result and will then be equated to the expected result.

8.3 Test environment

Operating System: Window

Ram: 4 giga byte (Minimum)

Memory: 120 Giga bytes

IDE: Visual Studio code, Text Editor

Database:

Internet: Required

8.4 Test Log:

Login Test Log:

Test Scenario: Verify on entering an valid user id and password, the user can log in

1	Login	Login with a registered google account	Log in	Log in	PASS	02/05/2020
2	Login	Login with an unregistered google account	Cannot log in	Cannot log in	PASS	02/05/2020

View dashboard Test case:

3	View Dashboard	Sign in with student account to view dashboard	Showing student information like, messages, meeting schedules, comment in a meeting, likes	Showing student information like, messages, meeting schedules, comment in a meeting, likes	PASS	02/05/2020
4	View Dashboard	Sign in with Tutor account to view dashboard	Showing Tutor information like, messages, creating meeting schedules, comment in a meeting, likes	Showing Tutor information like, messages, creating meeting schedules, comment in a meeting, likes	PASS	02/05/2020
5	List of student that have not	Sign in with Staff account to view student that	List of students that is not interactive	List of students that is not interactive for	PASS	02/05/2020

	been interactive	have not been interactive	for the past 7/28 days	the past 7/28 days		
	List of student that without tutor	Sign in with Staff account to view student without tutor	List of students without tutor	List of students without tutor	PASS	02/05/2020

Check UI profile Test log:

6	Checking UI profile	Click on the profile page to view user information	Details of user information	User information not clear as it only shows the name	FAIL	02/05/2020
7	Checking UI profile	Click on the profile page to view user information	Details of the user information	User information clearly shows name and user information	PASS	03/05/2020

View all students Test Log:

8	View all students	View student name and email list of all students	All data are listed on a clear format: name and email	All student's data was listed but the name and email was interchanged	FAIL	03/05/2020
9	View all students	View student name and email list of all students	All data are listed on a clear format: name and email	All student's data was listed in the correct format: name and email	PASS	04/05/2020

View all Tutors test Log:

10	View all Tutors	View all tutor from staff dash board	Displaying tutor information	A tutor listed in grid	PASS	04/05/2020
11	View all Tutors	View all tutors from student dashboard	Displaying Tutor assigned to the student	All tutors assigned is listed in a grid	PASS	04/05/2020

Assigning new Student to Tutor test log:

12	Assign new tutors to student	Select a student from the list and allocate the student to a tutor	Assign successful	Assign successful	PASS	04/05/2020
13	Assign new tutors to student	Select a student from the list and relocate the student	Student relocated successful	Student can't be relocated	FAIL	04/05/2020
14	Assign new tutors to student	Select a student from the list and change the tutor of the student	Tutor change successfully	Tutor can't be changed	FAIL	04/05/2020

Search Student Test log:

15	Search Student	Search for student in the search box with the correct email address	Search successful	No search box for student that already have a tutor	FAIL	04/05/2020
16	Search Student	Search for student in the search box that has no tutor	Search with student name	Search found	PASS	04/05/2020

Search Tutor Test log:

17	Search Tutor	Search for Tutor in the search box with the correct email address or name	Search successful	Display the search ed tutor on the search result	PASS	04/05/2020
18	Search Tutor	Search for Tutor in the	Search not found	Search not displays	PASS	

		search box with an incorrect email		nothing on the search result		
--	--	---	--	------------------------------------	--	--

Create meeting Test log:

19	Create meeting	Click on the plus sign at the top of the meeting activities in the tutor dashboard and select the title, content, start time, date and student information	Meeting created successfully	Adds a new meeting	PASS	04/05/2020
20	Creating meeting	Creating a meeting with an empty information	Error message	Adds a new meeting	FAIL	04/05/2020

Search meeting Test Log:

21	Search meeting	Search meeting by a particular key word. E.g. search by title	Find the meeting searched by Title	No search option	FAIL	04/05/2020
22	Search meeting	Search meeting by an invalid search format. E.g. search by student name	Invalid search format, search not found.	No search option	FAIL	04/05/2020

Edit meeting Test Log:

23	Edit meeting by changing the date and the time	Select date and time and select edit to change the date and time	Meeting updated successfully	Meeting created can't be edited	FAIL	04/05/2020

24	Edit meeting by changing the date and the time	Select datetime less than the require datetime	Error message: invalid datetime	Meeting created can't be edited	FAIL	04/05/2020
----	--	--	---------------------------------	---------------------------------	------	------------

View all meeting Test Log:

25	'View all meeting'	View all the meeting that the user had created	Display all the meeting	Displays list of meetings	PASS	04/05/2020

Uploading file on meeting Test Log:

26	Upload file	'Choose an file with a correct file format and upload on the daily post	Uploaded successful	File uploaded	PASS	04/05/2020

27	Upload file	'Choose a file with an incorrect file format and upload on the daily post	Error message: incorrect file format	File uploaded successfully	FAIL	04/05/2020
-----------	-------------	---	--------------------------------------	----------------------------	-------------	------------

Tutor Comment on meeting Test Log:

28	Comment in a Daily post	Comment with a tutor on a daily post	Comment sent and received	Tutor comments successfully on a daily post	PASS	04/05/2020

Student Comment on meeting Test Log:

29	Comment in a meeting	Comment with a student on a meeting	Comment sent and received	Student comments successfully on a daily post	PASS	04/05/2020

Notification received on Meeting created Test Log:

30	Receive a notification when a meeting is created or change	Check the message on the message icon to reveal the content	Meeting created message stating the correct date and time with the correct content and	Meeting is created but the notification icon didn't reveal the content	FAIL	04/05/2020

Chat Test Log:

31	chat	Click on the chat icon from the student and send message to the tutor	Message send and receive from Student to tutor	Message sent to tutor	PASS	04/05/2020
32	Chat	Click on the chat icon from the Tutor and	Message send and receive from Student to tutor	Message sent to student	PASS	04/05/2020

		send message to the student				
33	Chat	Click on the chat icon from the Tutor or student and send a message without any content	The send button will not be operational.	Message sent	FAIL	04/05/2020
34	Chat	Click on the chat icon from the Tutor or student and send a message without any content	The send button will not be operational.	Message Sent	FAIL	04/05/2020

8.5 Test case summary

- ⇒ **Number of case tested: 34**
- ⇒ **Number of test cases passed: 21**
- ⇒ **Number of test cases failed:13**
- ⇒ **Number of test cases solved: 0**

In this test, when we build a web site, we also perform the test every time a sprint is completed. At the beginning, we encountered a lot of errors when performing the test. Then the whole group

will meet and find ways to fix the errors to improve the product. So you can see in our test log diagram that there are quite a few failures. However, although we tried hard, but at the end of the project, we still could not fix all the errors that appeared in the system but most of the main functions required by the course work were completed and tested. work. If we have more time, we will be able to improve the system to minimize errors and create the best user experience.

In the next section, my general assessments of the entire system as well as the group report completion process.

9) Summary

In summary, in this group report we have covered the whole process of our team completing the construction of websites to support tutors and E-tutor students. We conducted a requirement analysis to determine the course work course requirements to best meet the user's requirements. Use case diagrams, Erd has also been deployed to serve the website creation process. Next, we have built wire frames for each user type and their specific functions so that when implementing the interface code, it will be easier to implement. After completing the site, we showed photos of user dash boards and functions to prove we have made those features.

The other part is how we show the security of our system, how we interact with the Google API, how the Fire base security works, and both Functional requirements and Non-functional requirements. We also explain the code of important system functions like login, react-router-dom, etc. And we have also implemented clean code to increase the aesthetics. And we also show our findings about the MVC model to prove that we understand the methods we use.

In the next we also provide evidence that we have applied the Scrum model to our project. We have shown the general analysis and evaluation of the priority of each requirement and put them into each sprint according to priority. At each sprint we create a Daily meeting table to discuss what has been done and see if there are any difficulties that need to be addressed. The Product Backlog and Burn down chart have also been used to statistically and align website completion time in line with the deadline set by customers.

In the last part is a summary of the test results our team performed during the construction of the web site. It is the basis for us to detect system errors and be able to fix them in the shortest time.

However, there are still some limitations such as we still have a lot of errors in the system and cannot complete all the features that the course work out. The process of applying Scrum to website building has not been good. More detailed reviews and personal comments will be presented in the individual report. Thank you for reading our report and wish you have a nice day.

10) References

(n.d.). Retrieved from <https://firebase.google.com/docs/auth>

Adams, S. (n.d.). *Computing & Mathematical Sciences*. Retrieved from https://drive.google.com/file/d/1rDVJ6czlU2tS4evxW9o8A0M721lNu_xX/view

Software Testing Help. (2020, 04 22). Retrieved from <https://www.softwaretestinghelp.com/agile-scrum-methodology-for-development-and-testing/>