F2G SE PROGRAMME **Enterprise Web Software Development**

**Team Report (Team 6)**

*By Le Trung Hieu*

*Nguyen Do Hong Quan*

*Luu Tien Anh*

*Ha Ngoc Linh*

*Instructor: Lai Manh Dzung*

**Hanoi, 2020**

Link github front-end: <https://github.com/quannguyennn/enterpriseweb-frontend>

Link github back-end: <https://github.com/quannguyennn/enterpriseweb-backend>

Link document: <https://drive.google.com/drive/folders/1LTNJkp6qggi6kBTDOlGvb3sO63egS4el?fbclid=IwAR1UtyHsLg6A6cEdMa0JuxVUEfDrcNkQFZ0H0YrZrNg-MqhN-KPoy1nCXv4>

Link video: <https://drive.google.com/drive/folders/1LTNJkp6qggi6kBTDOlGvb3sO63egS4el?fbclid=IwAR1UtyHsLg6A6cEdMa0JuxVUEfDrcNkQFZ0H0YrZrNg-MqhN-KPoy1nCXv4>

Link wireframe: https://www.figma.com/file/42MRFlJtehrFuu25ypxl7W/WEB-ENTERPRISE

Contents

[1. Introduction 4](#_Toc39460713)

[2. Database 5](#_Toc39460714)

[2.1. Entity relationship diagram 6](#_Toc39460715)

[2.2. Database explanation 6](#_Toc39460716)

[3. Site design 10](#_Toc39460717)

[3.1. Review 10](#_Toc39460718)

[3.2. Flowchart 12](#_Toc39460719)

[3.3. Wirefame 15](#_Toc39460720)

[4. Fucntionality 20](#_Toc39460721)

[5. Testing 30](#_Toc39460722)

[5.1. Test plan 30](#_Toc39460723)

[5.2. Test sprint 1 31](#_Toc39460724)

[5.3. Test sprint 2 33](#_Toc39460725)

[5.4. Test sprint 3 37](#_Toc39460726)

[6. Agile methodology 39](#_Toc39460727)

[6.1. Sprint 1: 40](#_Toc39460728)

[6.2. Sprint 2: 41](#_Toc39460729)

[6.3. Sprint 3: 42](#_Toc39460730)

[6.4. Summary meetings minutes: 43](#_Toc39460731)

[Bibliography 44](#_Toc39460732)

# Introduction

My team consists of 4 people: Nguyen Do Hong Quan (developer), Le Trung Hieu (Agile), Luu Tien Anh (Tester), Ha Ngoc Linh (Designer). We need to adopt agile scrum working practices, we are required to construct an online secure job based framework for eTutoring in a huge college. Full subtleties of the framework will be given in addresses. The framework must meet the accompanying standards:

* All students must have a personal tutor.
* Any authorised member of staff can allocate or reallocate personal tutors to students. The student and the personal tutors will get notification emails when this happens.
* Bulk allocation of students to their personal tutor needs to be implemented.
* All students and their tutors are to use the eTutor system for messaging, arranging and recording meetings (both real and virtual), uploading documents and commenting on them, and for blogging.
* Email between students and their personal tutors is to be used only for notification of events recorded in the backend database. No other content is to be sent via email.
* Student and staff data is accessed from the university MIS system. The maintenance of this is outside the scope of this project.
* Each student will have their own personal dashboard summarising their interaction with their personal tutor.
* Each personal tutor will have a dashboard of their personal tutees that can be sorted and filtered appropriately
* Authorised 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)

In addition, the admin can view a few reports from the system such as:

* Statistics:
  + Number of messages in last 7 days
  + Average number of messages for each personal tutor
* Exception reports:
  + Students without a personal tutor.
  + Students with no interaction for 7 days and 28 days.

# Database

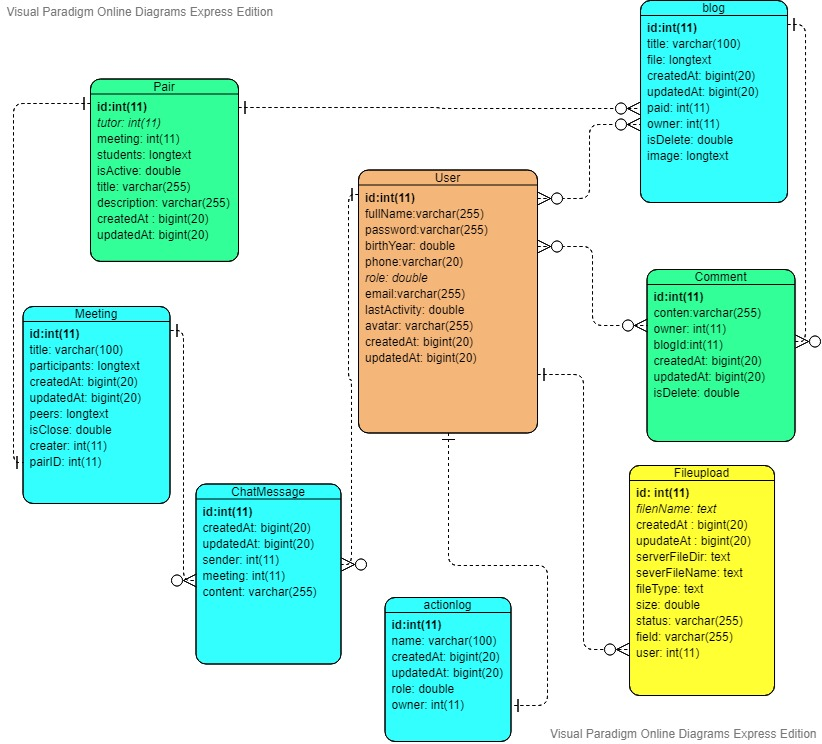
For this project, the developer used MySQL which is a relationship database in order to save the database memory.

The developer applied normalization to keep the database size low by removing duplicate data in a row table, also remove duplicate fields in all tables and digitalized data to reduce the size. For example, in order to video call tutor and student, one of them must create a meeting so the current active meetingId is store in the field meeting of model Pair, another example is the system role is digitalized: 0 is authorized staff, 1 is staff, 2 is tutor and 3 is student.

//viết kỹ hơn

However this project database’s record could increase very fast because this system allow users to create blogs, leave comments and chat message, therefor in order to boost up the query speed and the app performance in general the developer decided to implement the database with some of No SQL properties for example: used JSON to represent data structure, remove foreign keys.

## Entity relationship diagram



## Database explanation

User: This collection record user information such as personal information, credentials and system role.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | User id |
| fullName | Varchar(255) | User fullname |
| password | Varchar(255) | User hashed password |
| birthYear | double | User year of birth |
| Phone | Varchar(20) | User phone number |
| Role | Double | User role in the system:  1: Authorized staff  2: staff  3: tutor  4: student |
| Email | Varchar(255) | User email |
| lastActivity | Double | Timestamp of user last activity |
| Avatar | Varchar(255) | User avatar url |
| createdAt | Bigint(20) | Timestamp when user created |
| updatedAt | Bigint(20) | Timestamp when user update info |

Pair: This collection manage the allocation between tutor and student

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | Pair id |
| Tutor | Int(11) | Id of user which has role of tutor |
| Meeting | Int(11) | Id of current active meeting |
| Students | Longtext | User id which has role of student |
| isActive | Double | Whether this pair is active or not  0: not active  1: active |
| Title | Varchar(255) | Title of this pair |
| Description | Varchar(255) | Description for this pair |
| createdAt | Bigint(20) | Timestamp when pair created |
| updatedAt | Bigint(20) | Timestamp when pair update info |

Blog: This collection stores a blog data. A blog must belongs to a pair.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | Blog id |
| Title | Varchar(255) | Blog content |
| File | Longtext | JSON array of file url |
| Pair | Int(11) | Pair id that includes this blog |
| Owner | Int(11) | User Id who created this blog |
| isDelete | Double | Whether this blog is deleted  0: not deleted; 1: deleted |
| Image | Longtext | JSON array of images url of this blog |
| createdAt | Bigint(20) | Timestamp when blog created |
| updatedAt | Bigint(20) | Timestamp when blog update info |

Comment: This collection record comments data. A comment must belong to a blog

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | Comment id |
| Content | Varchar(255) | Comment content |
| Owner | Int(11) | User Id of whom create this comment |
| blogId | Int(11) | Blog id which this comment belong to |
| isDelete | Double | Whether this comment is deleted  0: not deleted; 1: deleted |
| createdAt | Bigint(20) | Timestamp when comment created |
| updatedAt | Bigint(20) | Timestamp when comment update info |

Meeting: This collection stores meeting data. A meeting must belong to a pair.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | Meeting id |
| Title | Varchar(255) | Meeting title |
| Pariticipants | Longtext | JSON array of user id who is currently in this meeting |
| Peers | Longtext | JSON array of socketid |
| isClose | Double | Whether this meeting is closed  0: not closed; 1: closed |
| Creater | Int(11) | User id of whom create this meeting |
| pairId | Int(11) | Pair id which this meeting belong to |
| createdAt | Bigint(20) | Timestamp when meeting created |
| updatedAt | Bigint(20) | Timestamp when meeting update info |

ChatMessage: This collection stores chat messages data. A chat messages must belong to a meeting.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | Chat message id |
| Sender | Int(11) | User id which send this message |
| Meeting | Int(11) | Meeting id which this message belong to |
| createdAt | Bigint(20) | Timestamp when meeting created |
| updatedAt | Bigint(20) | Timestamp when meeting update info |
| Content | Varchar(255) | Content of the message |

ActionLog: This collection stores user actions data. A user action must belong to a user.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | ActionLog id |
| Name | Varchar(255) | Name of the action |
| Role | Double | User role |
| Owner | Int(11) | User id which this action belong to |
| createdAt | Bigint(20) | Timestamp when action created |
| updatedAt | Bigint(20) | Timestamp when action update info |

FileUpload: This collection stores user uploaded file and images. A user uploaded file must belong to a user.

|  |  |  |
| --- | --- | --- |
| Field name | Type | Description |
| Id: primary key | Int(11) | File id |
| filename | text | Name of the uploaded file |
| serverFileDir | text | Direction folder which save this uploaded file |
| saverFileName |  | Name of the uploaded file on the server |
| filetype | Text | Type of uploaded file (image, word, excel..) |
| Size | Double | Size of the file in kB |
| Status | Double | Status of the file |
| User | Int(11) | User id of whom upload this file |
| createdAt | Bigint(20) | Timestamp when file created |
| updatedAt | Bigint(20) | Timestamp when file update info |
|  |  |  |

# Site design

## Review

The application is built with a role-based security system. Users will be authenticated according to these roles and access to the corresponding screens which authorized by these roles.

According to the project's design, the application will be divided into 4 main screen groups:

|  |  |
| --- | --- |
| Admin | As an admin, user can view personal information of every user. They can create account, delete or edit all profile information. Admin also has permission to check action log of every user and number of messages for each personal tutor |
| Staff | As a staff, user also can view, create, edit or delete account information.  They can allocate or reallocate personal tutors to students |
| Student | As a student, user can view profile of himself |
| Tutor |  |

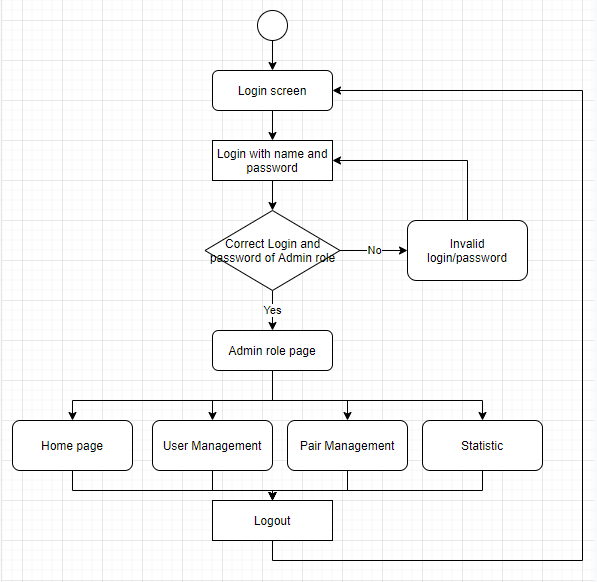
Recently, Single page application is an emerging name in the trend of web development. Although this concept has been around for more than a decade. Many websites have applied this technique. Using Single-page Application (SPA) will be very effective and save bandwidth, as well as for a better user experience.

Based on the advantages of Single page application, e-Tutoring is also designed as a single page application.

All pages use the same top navigation bar and left side bar. With just one button on the top navigation bar, users can show or hide the left side bar which is menu. On top navigation it is also possible to show-hide user information. This design will help users more convenient in using the application, reducing implementation time for develop team.

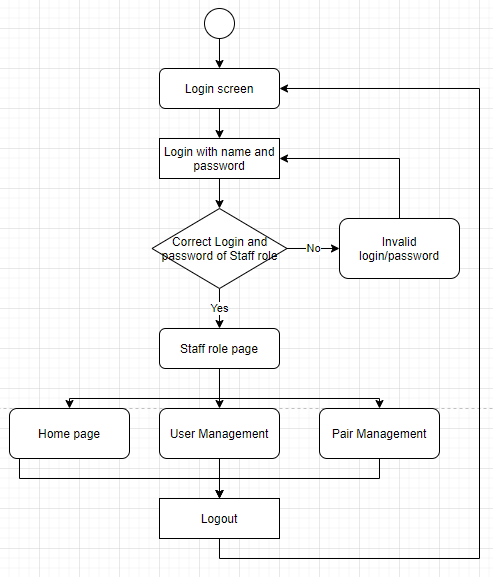


## Flowchart

* Admin

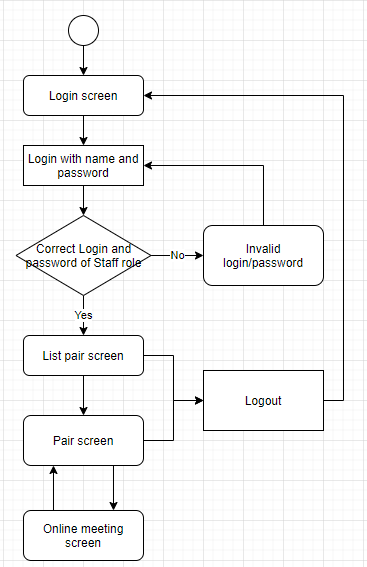
After logging in with the admin account, the admin will gain access to the screens where the admin account is granted. The screens include the default home page, user management, pair management, and statistic. On the Statistic screen, there is also a statistics table for students who have not had any activity in the application for 7-28 days.

* Staff



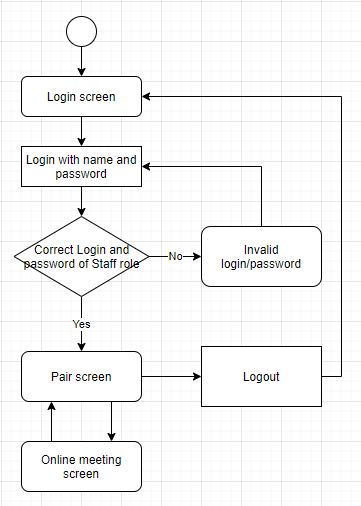
After logging in with the staff account, the staff will have access to screens where the staff account is authorized. Screens accessed as admin accounts but without statistic, including default home page, user management, pair management.

* Tutor



The Tutor account after login will only display the list pair screen. Here is a list of the pairs that have been allocated to this account. Clicking on a pairing will be directed to the paring screen. Users who want to meet online with students will have to create a meeting, then click join to direct to the online meeting screen.

* Student



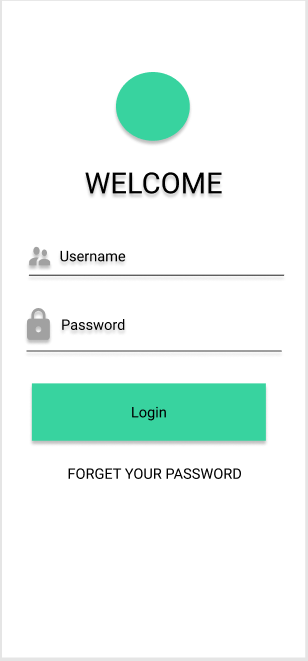
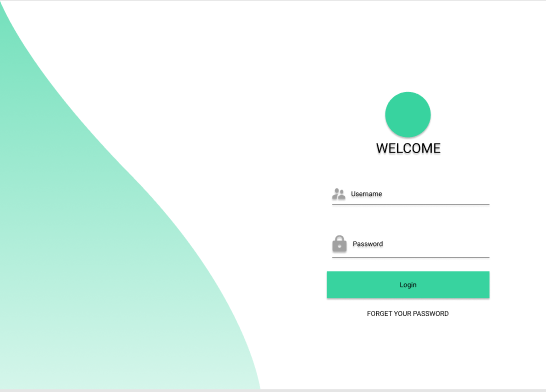
Student accounts will only be accessed on the paring screen. If not paired, this screen will not display anything.

## Wireframe

Because the plague didn't allow us to have much time for working directly, the team decided to use the Figma tool to design the wireframe. Figma is a user interface design tool (UI) that is run on the browser. Figma is a vector-based software that helps designers use it in real time. Because of the advantages that working in real time, members can contribute design ideas at the same time. Figma also does not charge users and has many powerful support tools.

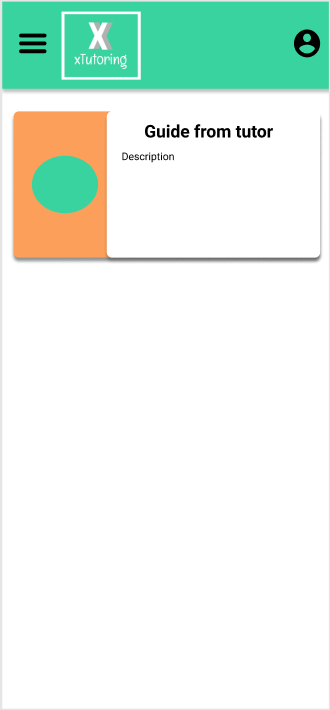
Below is the wireframe of all screens in the application. This design includes both for computers and mobile devices

* Login screen



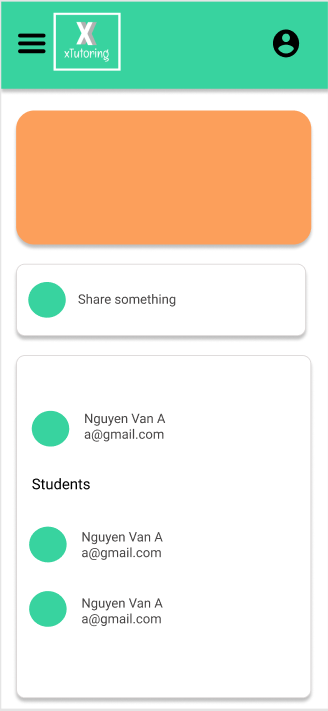
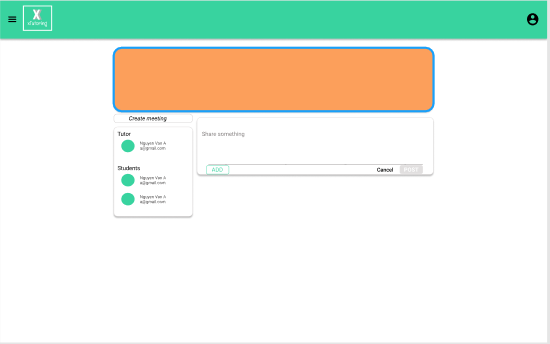
The login screen will ask the user to enter the username and password. If a user has forgotten their password, they can click on the "FORGET YOUR PASSWORD" button, a popup will appear for the user to enter their email. The email with the password reset link will be sent to that email.

* Paring list screen

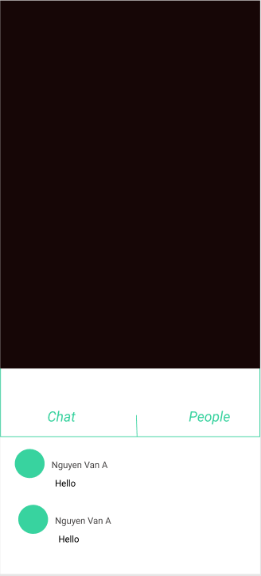
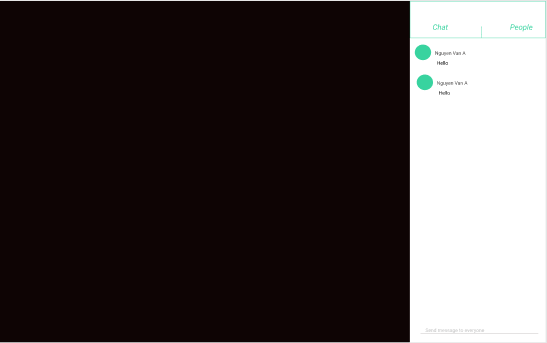


This is the paring list screen. This screen is only available for accounts with role tutors. There will be a list of the pairs allocated by admin (staff) for that tutor account.

* Paring screen

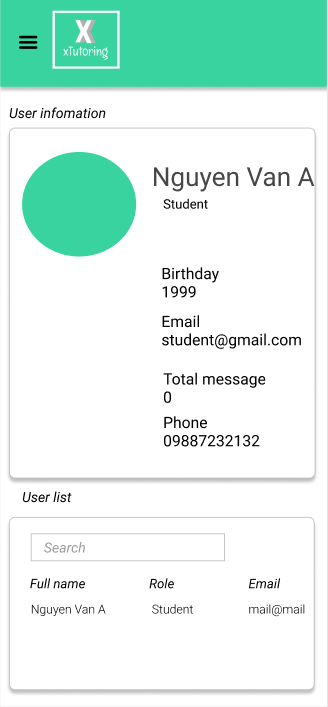


This is a paring screen. This screen will be the work place between tutor and student which are allocated by admin (staff). In there, they can blog, create work appointments, create online meetings.

* Meeting online screen:

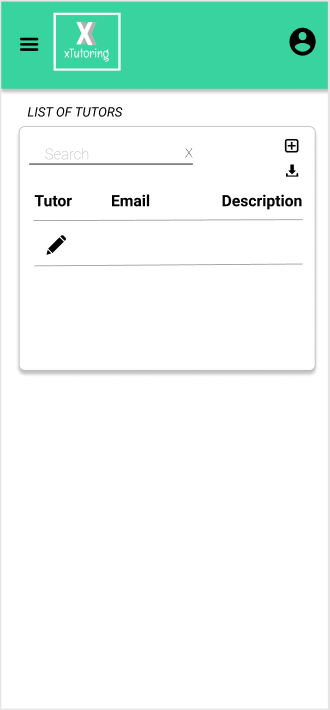
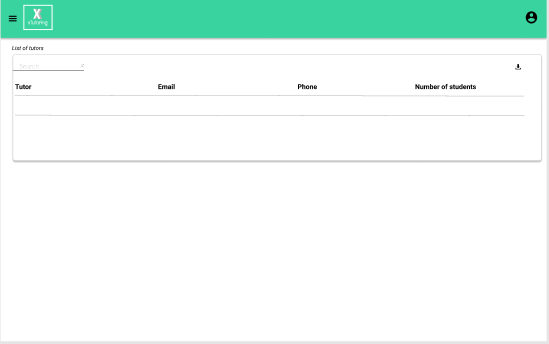
This is the online working screen. They can text or video call.

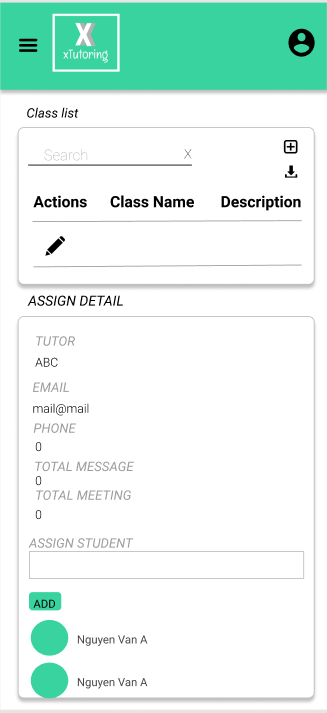
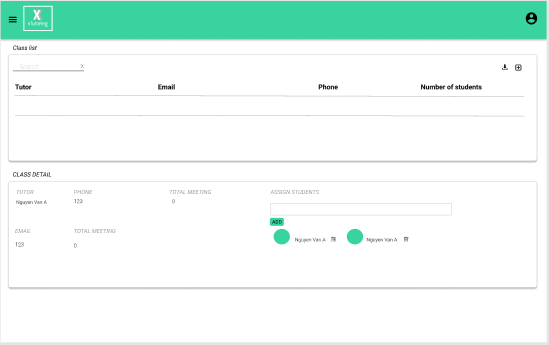
* User management



This is the user management screen. There is all user information. For accounts with admin roles will be able to view, edit, delete and create users. The staff account will only have the function to view profiles of users.

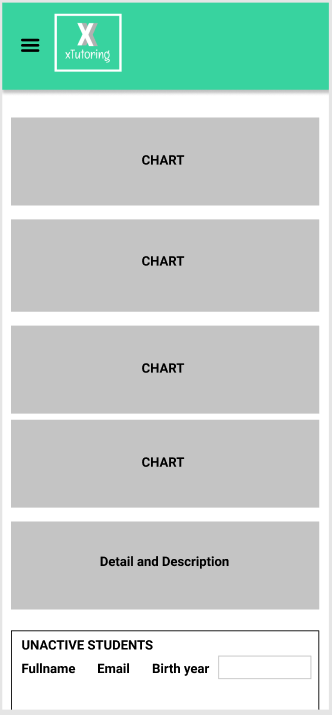
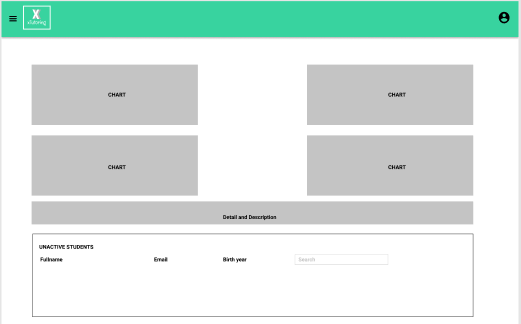
* Pairing management





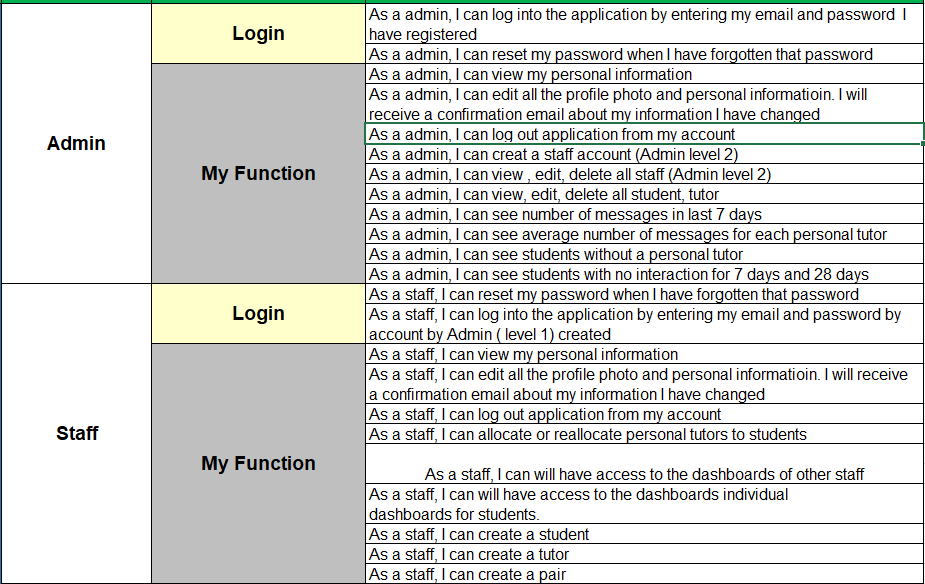
This is the paring management screen. Here, the admin and staff accounts can allocate or unallocate tutors and students. By clicking the name of a tutor, the user will be able to add and delete the allocation.

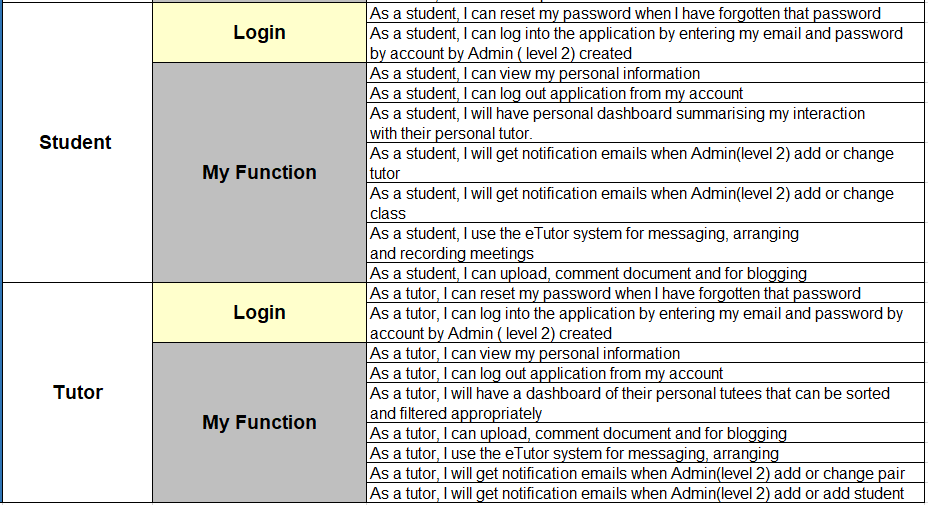
* Statistic screen



This is the Statistic management screen. This screen is only for admin accounts. This screen has activity statistic that take place on the application. There is also information about the unactive-students.

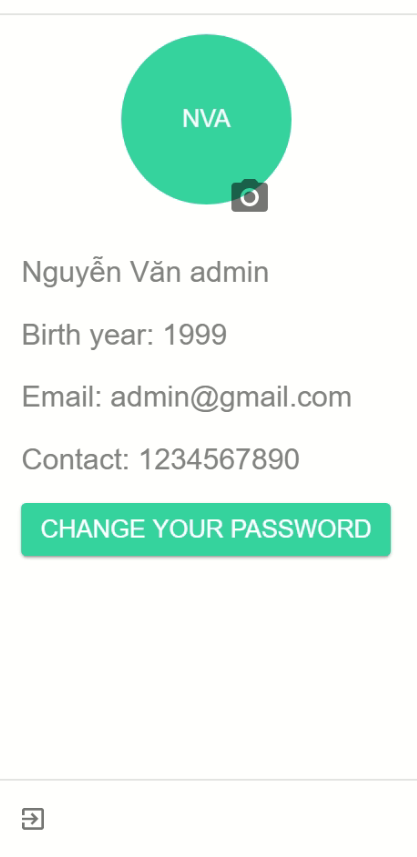
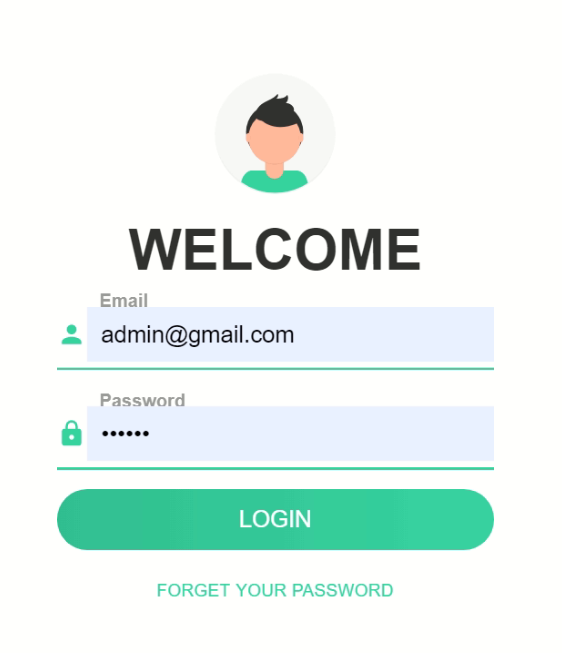
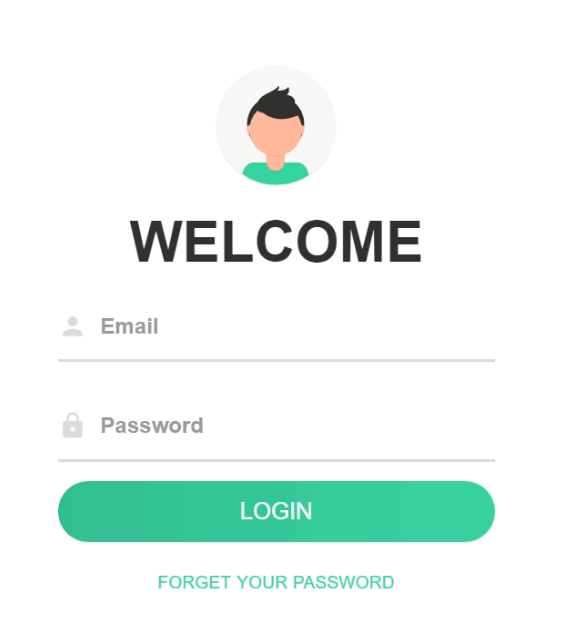
# Functionality

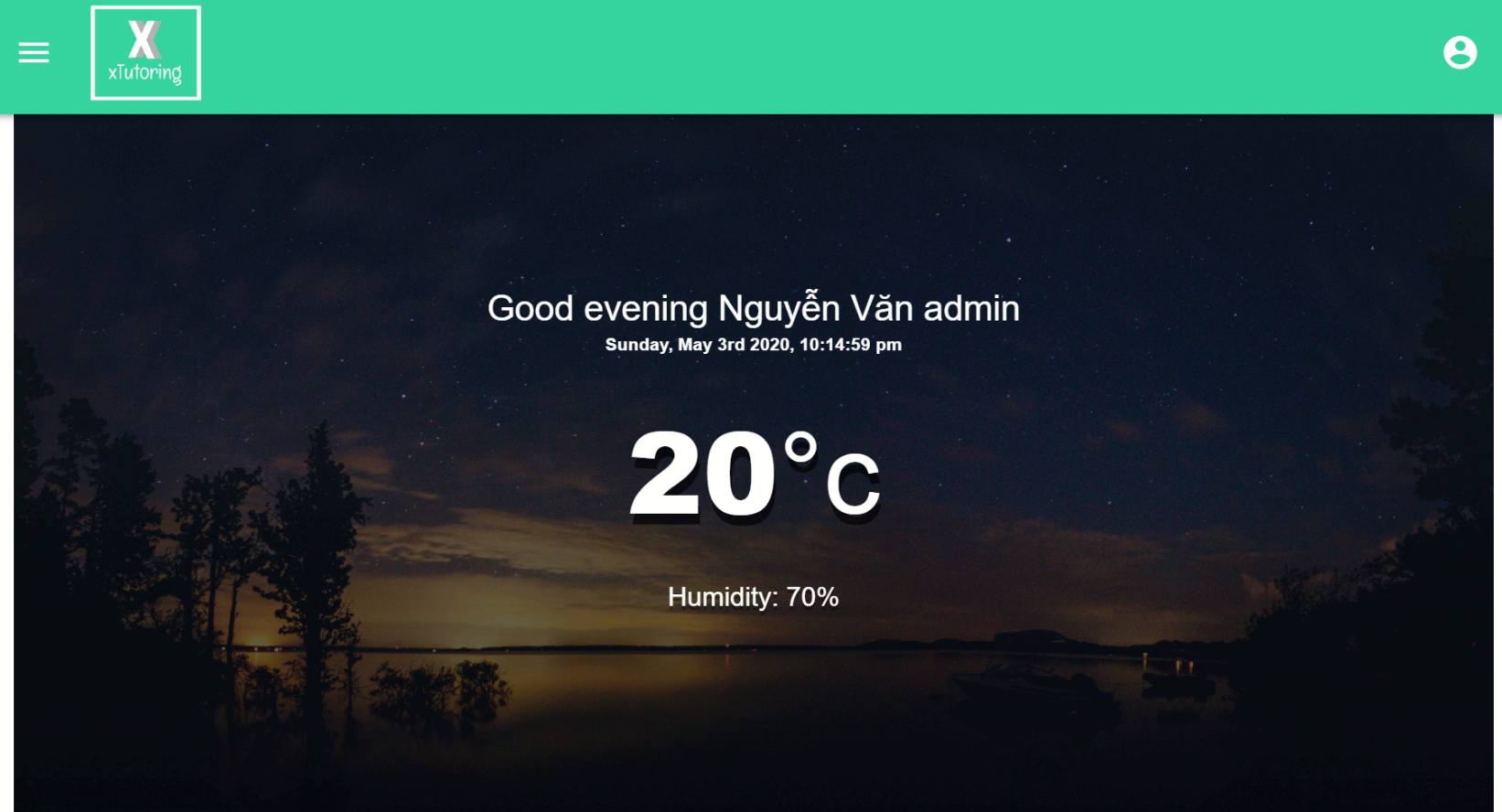




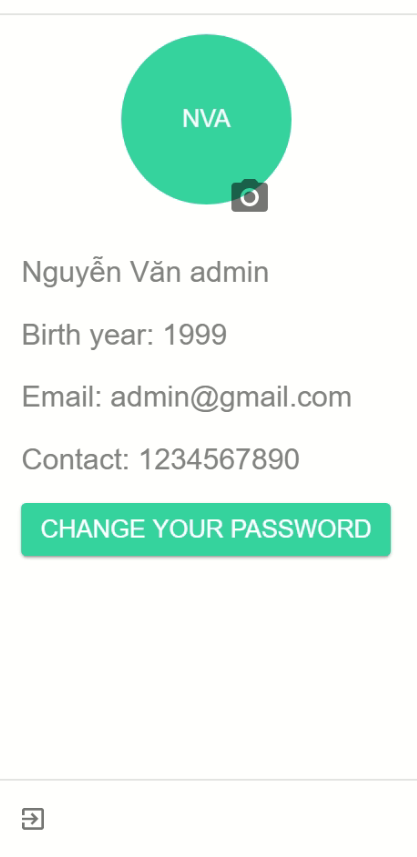
After reading and learning about the system, we have identified users of this system including: Admin, Staff, Tutor and Student. Each person has their own functions to express their rights in the system. In addition to the main functions of the required course work, we also add some functions to improve the system.

* Admin:
  + **Login and Logout (done)**

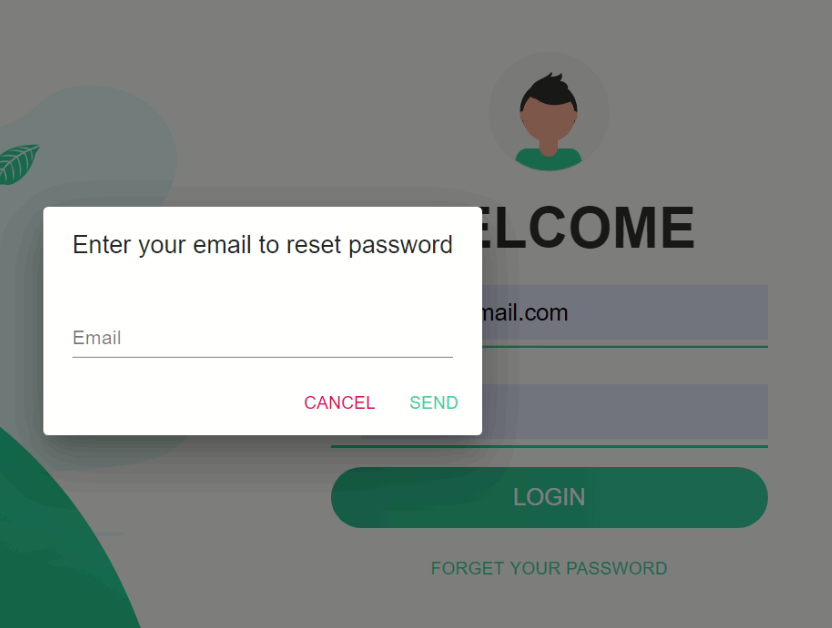
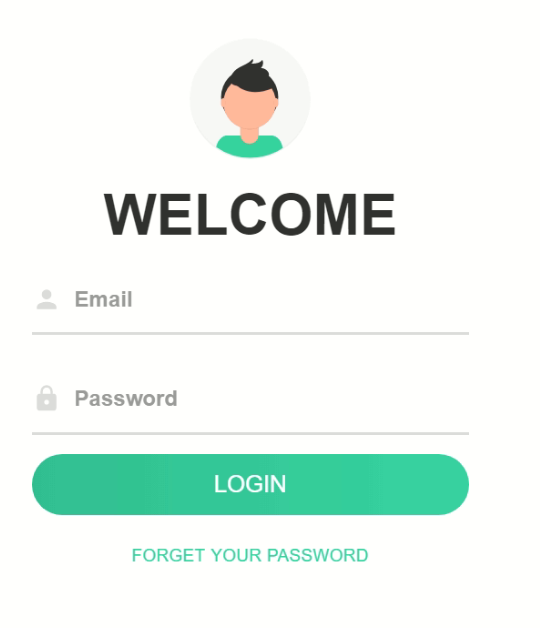




The Log-in and Log-out function is working, you can type your email address and password to log-in, it will access your account as image attached.

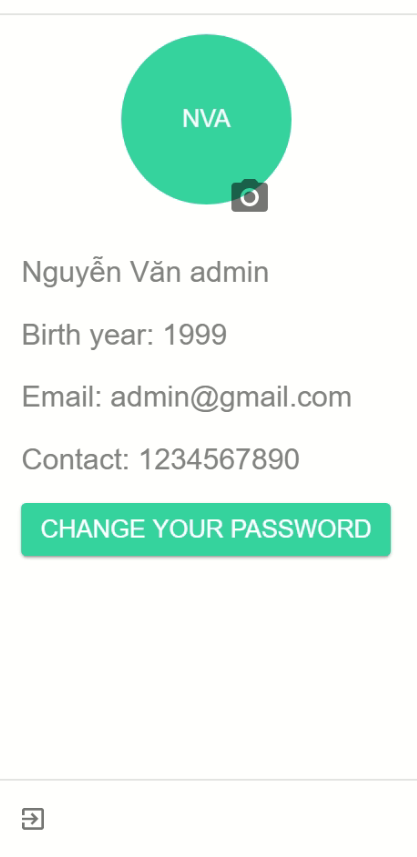
The account is logged out successfully when click and went back to log-in function

* + **Reset password (done)**



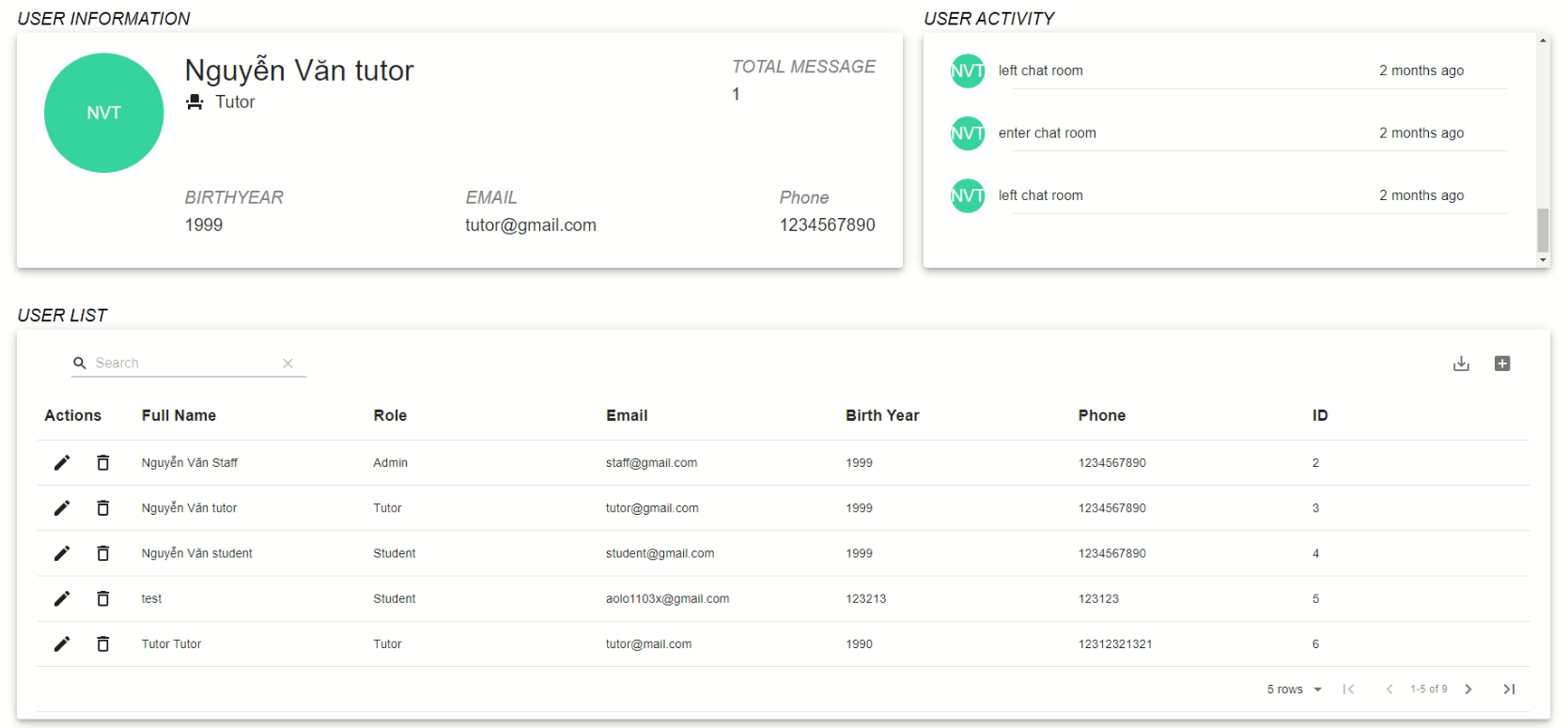
Reset password is working normally. To do this, choose “Forget your password” then type your email address, for example: [admin@gmail.com](mailto:admin@gmail.com) then press “Send”; an automatic email will be sent to your email to renew password.

* + **View and edit profile (50% done)**



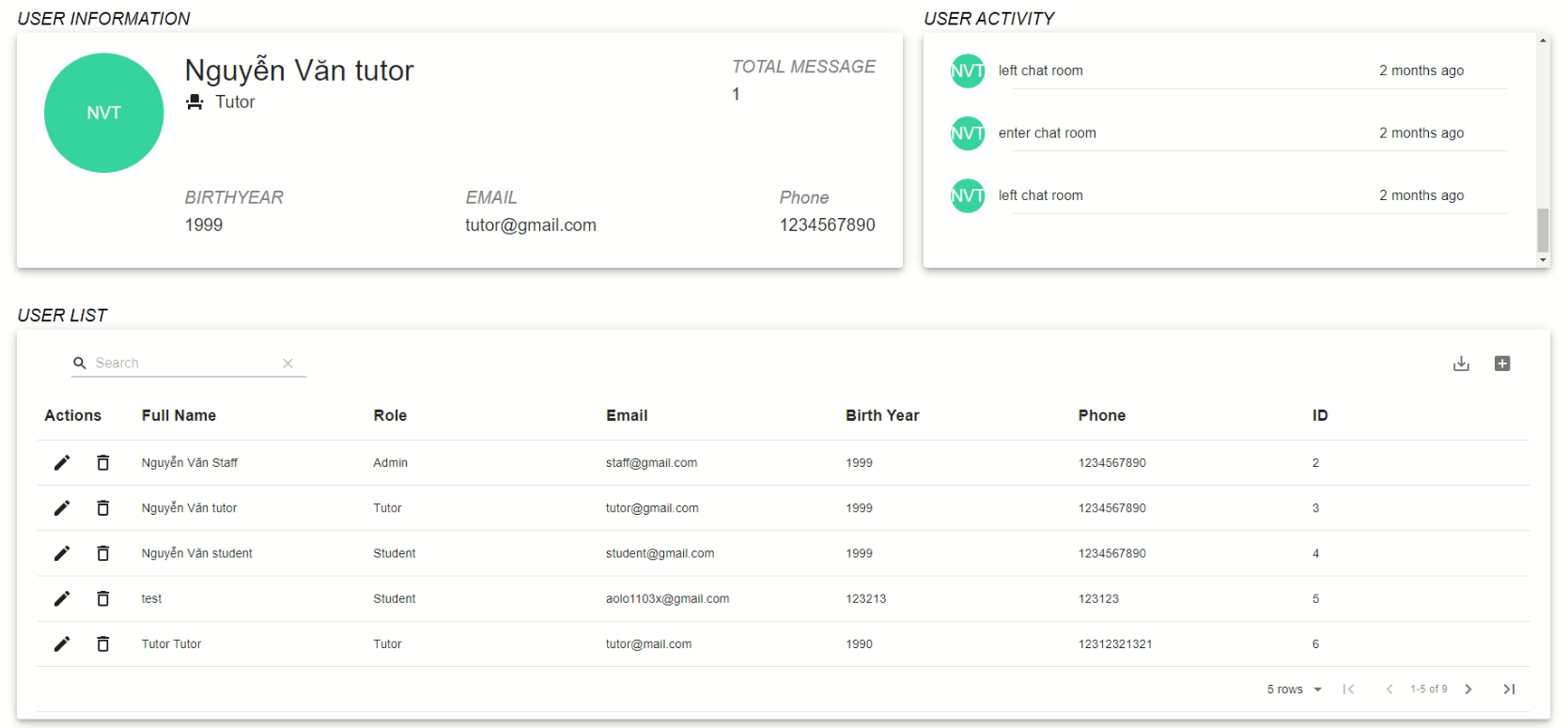
User only can view this profile by clicking on the right-top corner of the page

But the admin can edit his profile by using edit profile user function.



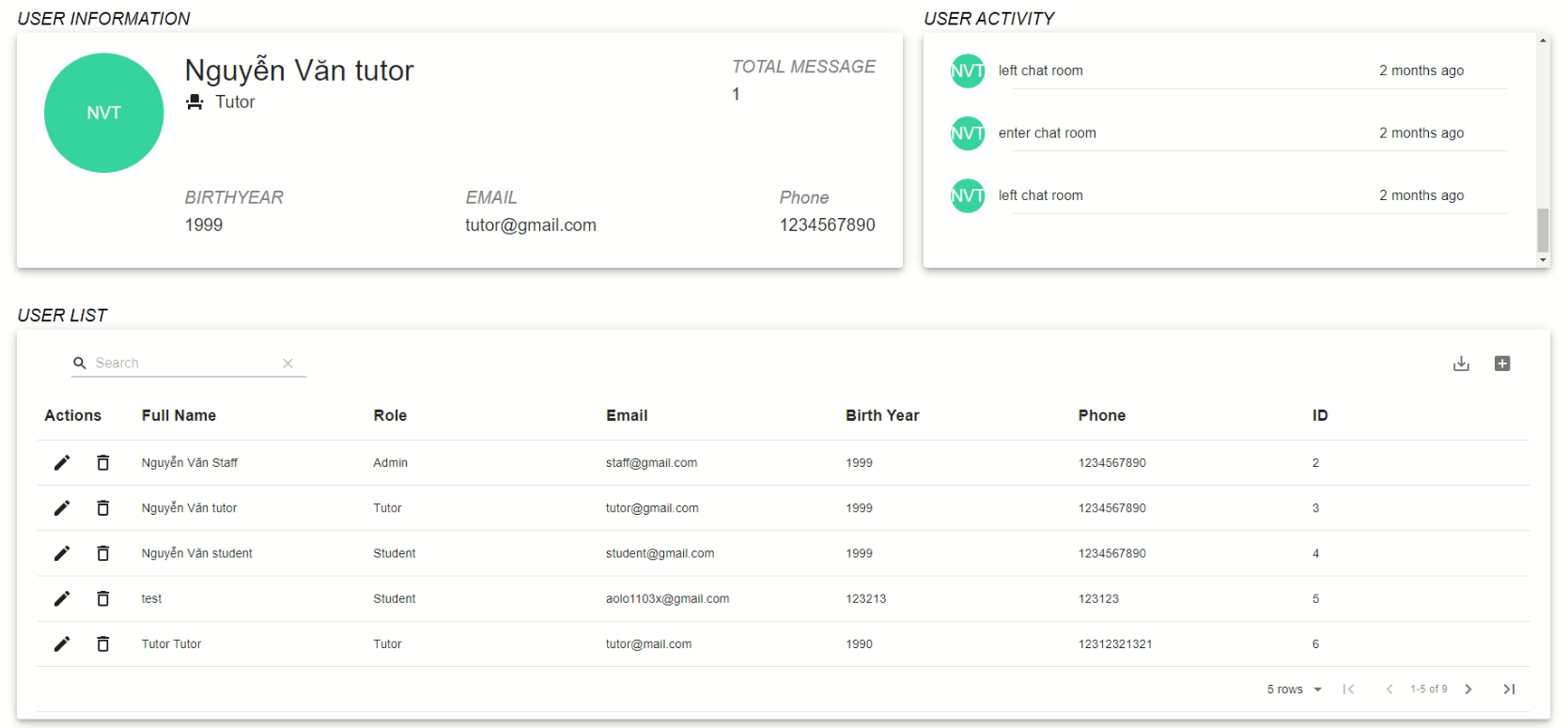
With this function, tap  then choose “User Management”, the screen turns into your profile information for you to view and edit.

* + **View ,create , edit , delete staff (done)**



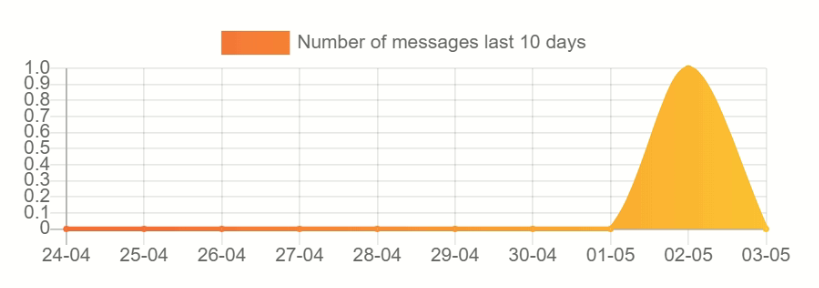
All these functions are working with admin account, you can find it admin profile, choose each symbol to “Edit staff” or “Add staff” or “Delete staff” 

* + **View ,create , edit , delete tutor and student (done)**



All these functions are running in profile admin account, choose each symbol to “Edit student” or “Add student” or “Delete student” .

* + **Report about average number message tutor (not yet available)**
  + **Report about average number of messages in last 7 days (done)**



The report about average number of messages in last 7 days is now running in last 10 days. From homepage, tap  then choose “Statistic”.

* + **Report about students without a personal tutor and no interaction for 7 days and 28 days (not yet available):**

This report has not worked yet. At the moment, it’s working as a report for all the inactive students.

* Staff:
  + **Login and log out (done)**

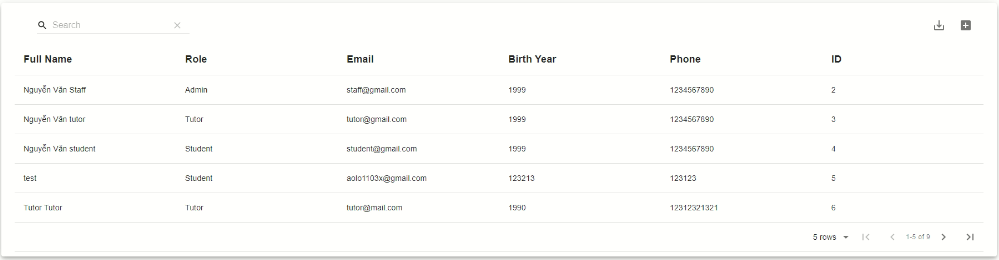
Similar to the admin, after entering the staff account, the user will have access to the screens which are authorized for staff.

* + **Reset password (done)**

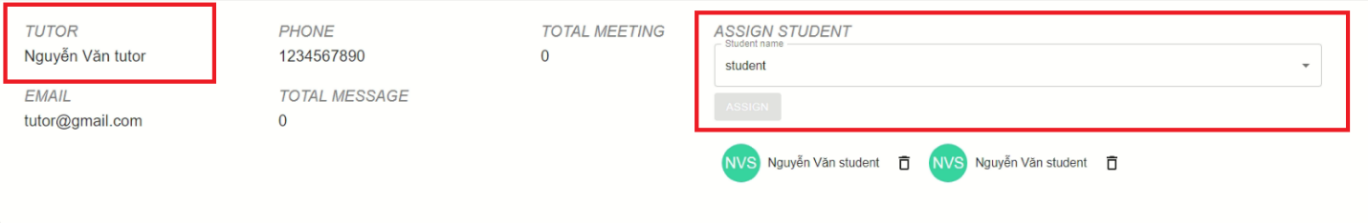
Password reset is identical to admin account.

* + **View and edit profile (done)**

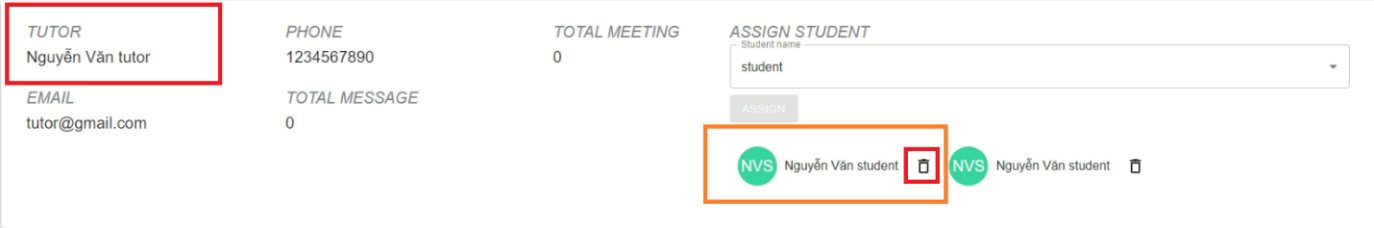
Staff accounts cannot edit their information, but can only be viewed by pressing the button .

* + **Create tutor , student (done)**

Similar to the admin, the staff account also has the function to create tutor and student accounts, but they cannot edit or delete.

* + **Allocate or reallocate personal tutors to students (done)**

Staff can multiple or single assign student to a tutor by typing the student name after the ‘@’ character in the ASSIGN STUDENT box.

They can unallocate by clicking the trash bin icon.

* + **Access to the dashboards of other staff (not yet)**
  + **Access to the dashboards individual dashboards for students (not yet)**
* Tutor:
  + **Login and lout out (done)**

Similar to the admin, after entering the tutor account, the user will have access to the screens which are authorized for tutor.

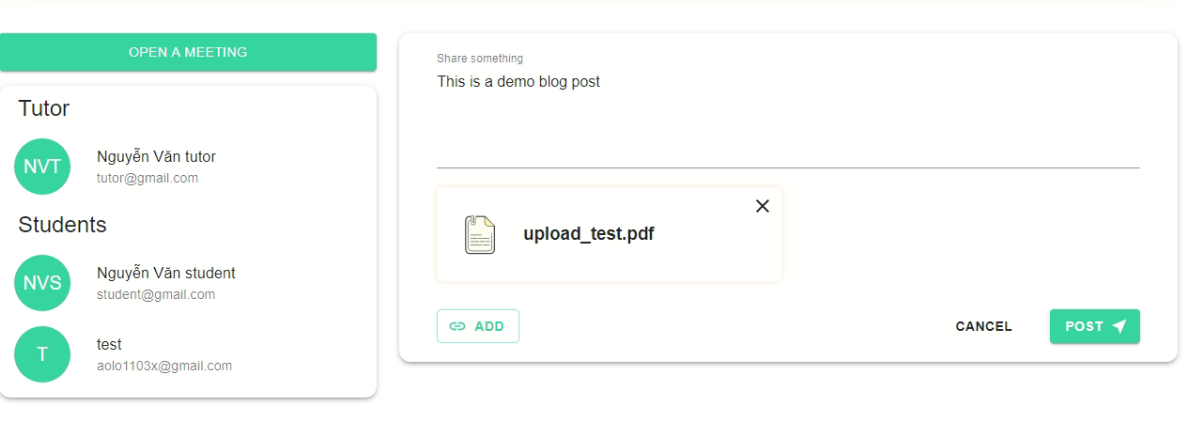
* + **Reset password (done)**

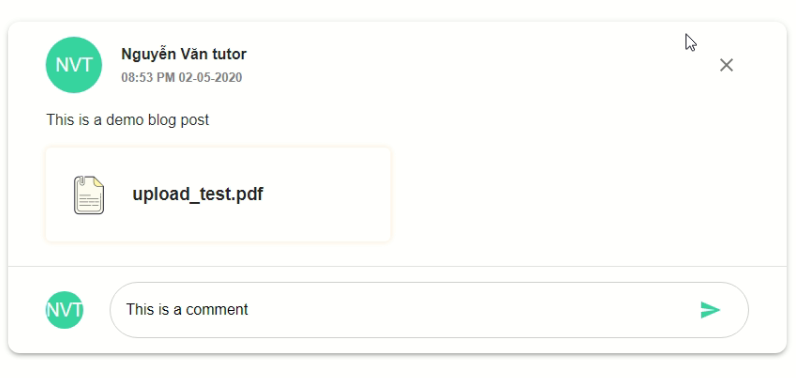
Password reset is identical to admin account.

* + **View and edit profile (50% done)**

Tutor accounts cannot edit their information, but can only be viewed by pressing the icon .

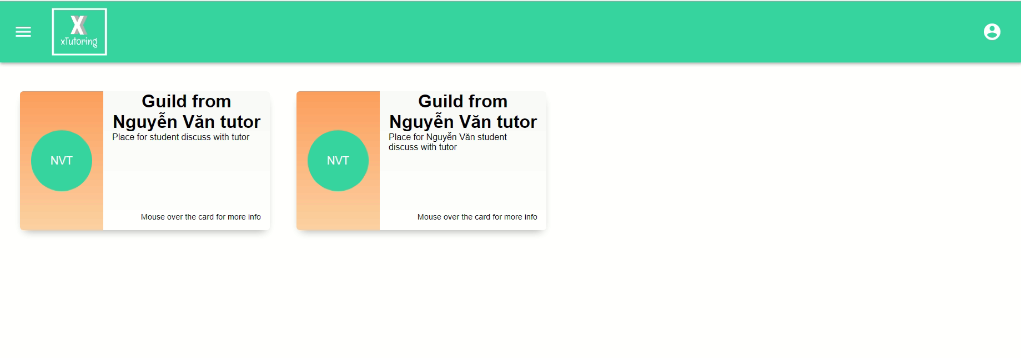
* + **Upload, comment document and for blogging (done)**





In the Pairing Screen, tutor can attach file in the blog, they also can comment on any blog which is allocated to them.

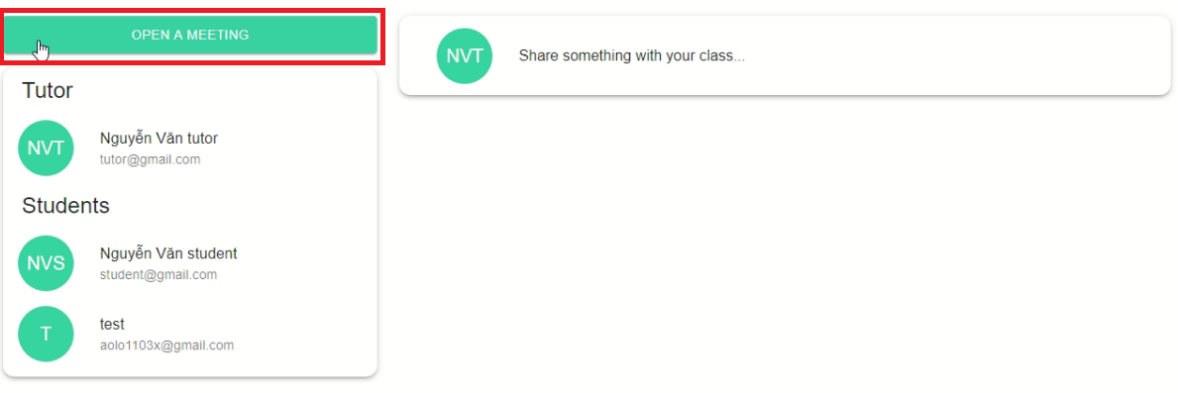
* + **Have a dashboard of their personal tutees that can be sorted and filtered appropriately (50% finished)**



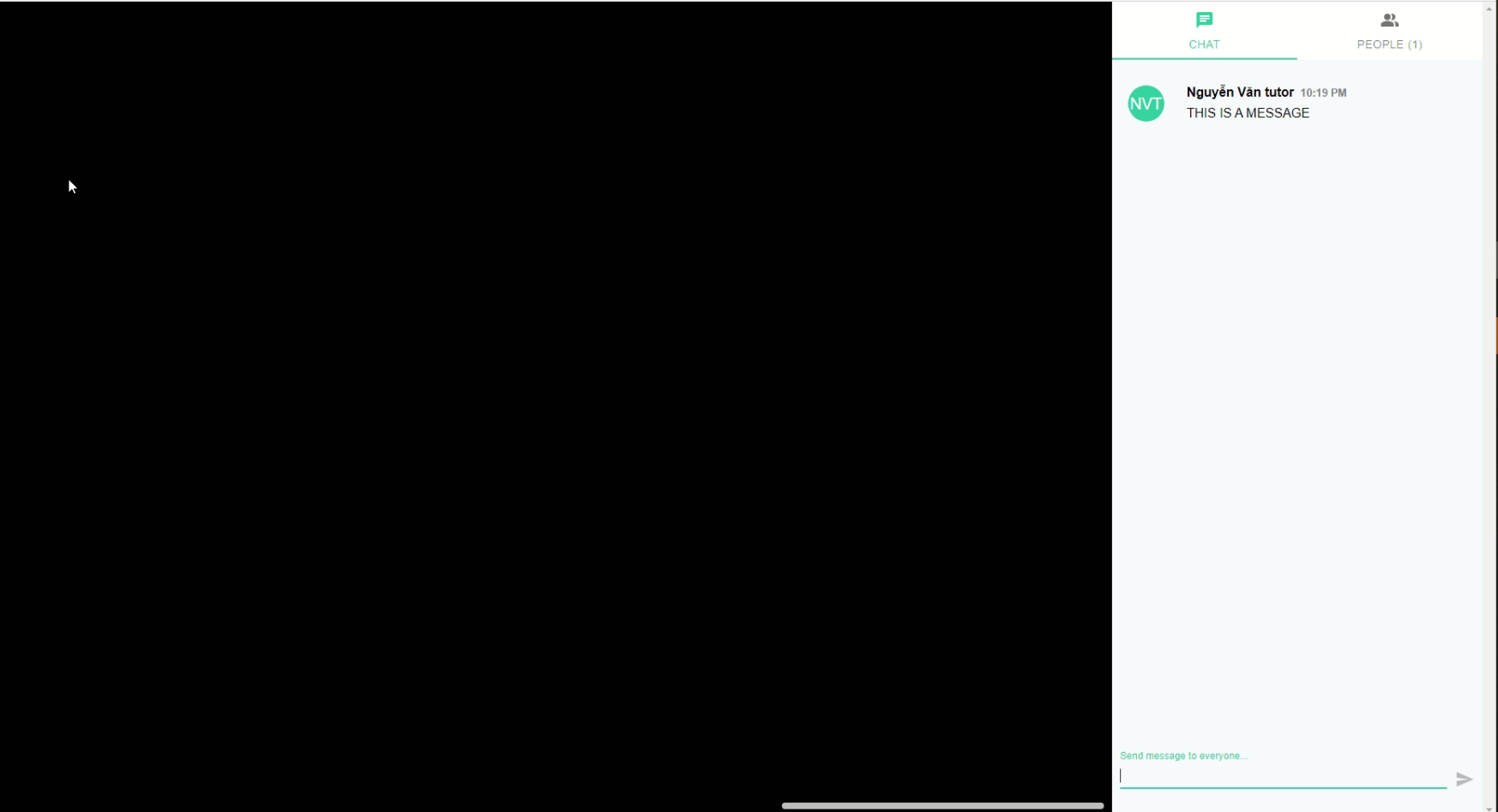
There is a pair list screen for tutor account but the function sort and filter are not ready.

* + **Use the eTutoring system for messaging, arranging and recording meetings (done)**

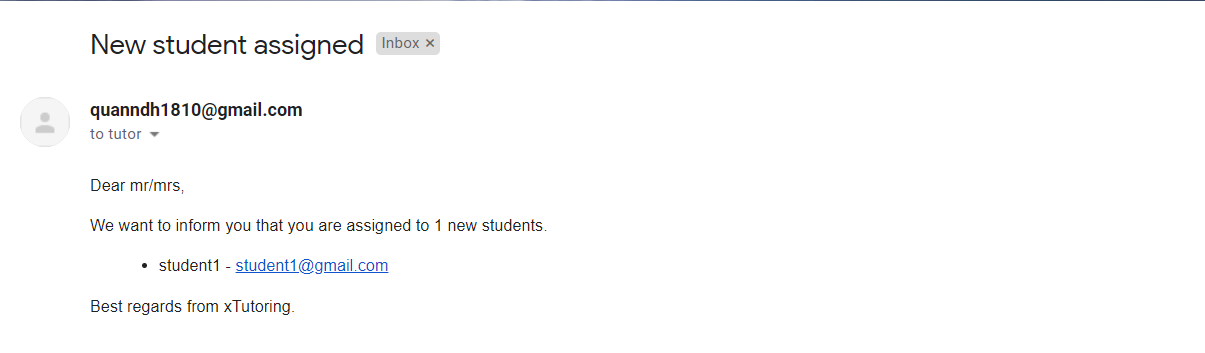
To create an online meeting (to message and video chat), tutor need to open a meeting first by clicking “OPEN A MEETING”



After clicking "OPEN A MEETING", a link to the online meeting screen will be created. All accounts in this paring can see and access the meeting. To access, press "Join"



* + **Get notification emails when Admin(level 2) add or change pair (done)**



This is the notification mail when a tutor is allocated

* + **Get notification emails when Admin(level 2) add or change student (done)**
* Student:
  + **Login and lout out (done)**

Similar to the admin, after entering the student account, the user will have access to the screens which are authorized for student (only paring screen).

* + **Reset password (done)**

Password reset is identical to admin account.

* + **View and edit profile (done)**

Student accounts cannot edit their information, but can only be viewed by pressing the icon .

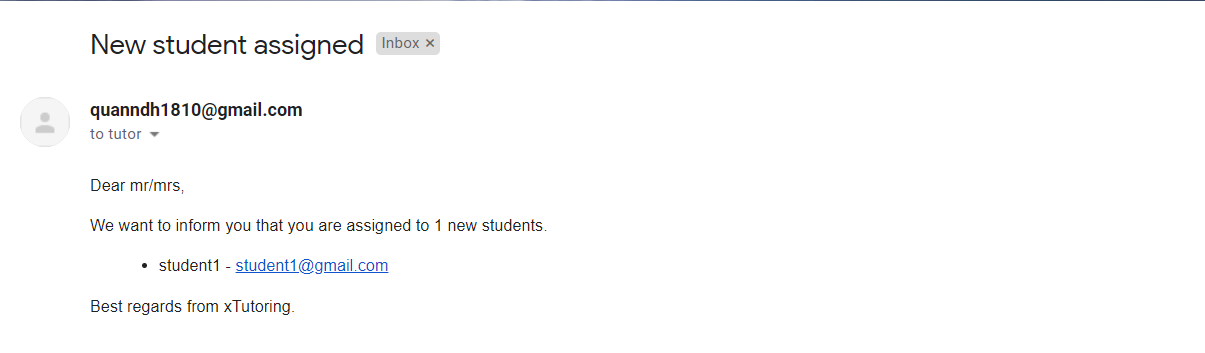
* + **Can upload, comment document and for blogging (done)**

Similar to tutors, students can post blogs, comment, and attach files.

* + **Use the eTutor system for messaging, arranging and recording meetings (done)**

Students can also join video calls, messages similar to tutor.

* + **Get notification emails when Admin(level 2) add or change tutor ( not done)**
  + **Get notification emails when Admin(level 2) add or change pair (done)**



This is the notification mail when a student is allocated.

# Testing

## Test plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test Area | Sub Tests | Target Completion | Status |
| 1 | [Front-end]/[Back-end] Login, logout | Login, logout success with correct accout: Admin, staff, student, tutor | 3/7/2020 | Complete |
| 2 | [Front-end] Student dashboard, Class detail |  | 3/10/2020 | Complete |
| 3 | [Front-end] Tutor dashboard | Screen tutor dashboard | 3/13/2020 | Complete |
| 4 | [Back-end] Student and Tutor Login and logout |  | 3/16/2020 | Complete |
| 5 | [Back-end] Admin, staff, student , tutor edit profile | Edit profile: Admin, staff, student, tutor | 3/19/2020 | Complete |
| 6 | [Back-end] Admin view, edit , delete staff, student, tutor | Admin managent staff, student, tutor | 3/22/2020 | Complete |
| 7 | [Back-end] All role can resert password | Reset password | 3/22/2020 | Complete |
| 8 | [Back-end] Student and Tutor can receive email | Get email | 3/22/2020 | Complete |
| 9 | [Front-end] Chat video, message and Statistics screen for Admin | Show report data to statistics screen of Admin | 3/22/2020 | Complete |
| 10 | [Back-end] Student and Tutor upload file, comment | Upload file type: .doc, .xlsx | 3/22/2020 | Complete |
| 11 | [Back-end] Statistics of Admin | Input data to statistics screen of Admin | 3/25/2020 | Complete |
| 12 | [Back-end] Student and Tutor Chat video | Video call | 3/25/2020 | Complete |
| 13 | [Back-end] Student and Tutor Chat message | Send message | 3/25/2020 | Complete |

## Test sprint 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Owner | Action | Input | Expected Output | Actual Output | Test Result | Device |
| 1 | Admin | Log into the application by entering my email and password I have registered | Email Id: root@gmail.com | eTutor home | eTutor home | Pass | Mobile, PC |
| Password: 123456 |
| 2 | Admin | Log out application from my account | Click logout button | Logout accout and go to login page | Logout accout and go to login page | Pass | Mobile, PC |
| 3 | Admin | Create a staff account (Admin level 2) | Click "user management" | New user created | New user created | Pass | Mobile, PC |
| Click "add" button |
| Full name: Nguyen Van A |
| Role: Admin |
| Email: nguyenvana@gmail.com |
| Bird year: 1998 |
| Phone: 012345678 |
| Click button "save" |
| 4 | Admin | View staff (Admin level 2) | Click "user management" | Display all information of Admin | Display all information of Admin | Pass | Mobile, PC |
| Click "role" |
| 5 | Admin | Edit staff (Admin level 2) | Click "edit" button of "Nguyen Van A" | User updated successfully | User updated successfully | Pass | Mobile, PC |
| Full name: Nguyen Van Edit |
| Click "save" button |
| 6 | Admin | Delete staff (Admin level 2) | Choose "Nguyen Van A" | User deleted successfully | User deleted successfully | Pass | Mobile, PC |
| Click "delete" button |
| Click "save" button |
| 7 | Staff | Can log into the application by entering my email and password by account by Admin ( level 1) created | Email Id: admin@gmail.com | eTutor home | eTutor home | Pass | Mobile, PC |
| Password: 123456 |
| 8 | Staff | Log out application from my account | Click logout button | Logout account and go to login page | Logout account and go to login page | Pass | Mobile, PC |
| 9 | Staff | Create a student | Click "user management" | New user created | New user created | Pass | Mobile, PC |
| Click "add" button |
| Full name: Nguyen Van Student |
| Role: Student |
| Email: nguyenvanstudent@gmail.com |
| Bird year: 1998 |
| Phone: 012345678 |
| Click button "save" |
| 10 | Staff | Create a tutor | Click "user management" | New user created | New user created | Pass | Mobile, PC |
| Click "add" button |
| Full name: Nguyen Van Tutor |
| Role: Tutor |
| Email: nguyenvantutor@gmail.com |
| Bird year: 1998 |
| Phone: 012345678 |
| Click button "save" |
| 11 | Staff | Create pair | Click "class management" | New class created | New class created | Pass | Mobile, PC |
| Click "add" button |
| Class name: test |
| Description: "this is a test class" |
| Number of student: "1" |
| Status: Active |
| Click button "save" |
| 12 | Tutor | Can log out application from my account | Click logout button | Logout account and go to login page | Logout account and go to login page | Pass | Mobile, PC |

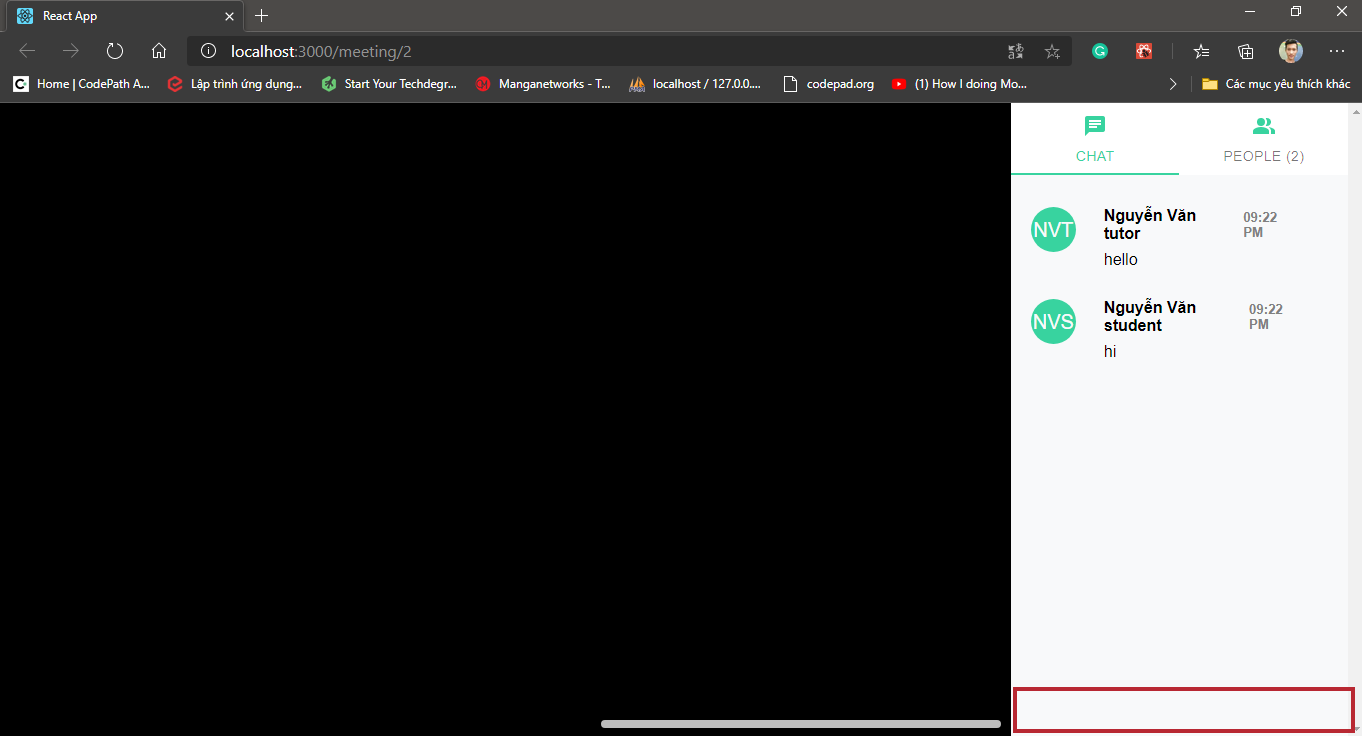
## Test sprint 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Owner | Action | Input | Expected Output | Actual Output | Test Result | Device |
| 1 | Admin | Can reset my password when I have forgotten that password | New password: Test@851853 | Login with new password successful | Login with new password successful | Pass | Mobile, PC |
| Confirm your password: Test@851853 |
| 2 | Admin | View my personal information | Click "Profile" icon | Show my personal information | Show my personal information | Pass | Mobile, PC |
| 3 | Admin | Edit all the profile photo and personal information. I will receive a confirmation email about my information I have changed | Edit avatar and password | Updated successfully | Updated successfully | Pass | Mobile, PC |
| 4 | Admin | View all student, tutor | Click "User Management" | Show all student and tutor | Show all student and tutor | Pass | Mobile, PC |
| 5 | Staff | Can view my personal information | Click "Profile" icon | Show my personal information | Show my personal information | Pass | Mobile, PC |
| 6 | Staff | Can allocate or reallocate personal tutors to students | Click "Class Management" | Class updated successfully | Class updated successfully | Pass | Mobile, PC |
| Choose class |
| Click "Add Tutor" button |
| Input: "Nguyễn Văn tutor" |
| Input: "Nguyễn Văn student" to box add student |
| Click "ADD" button |
| 7 | Staff | View all student, tutor | Click "User Management" | Show all student and tutor | Show all student and tutor | Pass | Mobile, PC |
| 8 | Admin | Edit student | Click "edit" button of "Nguyen Van student" | User updated successfully | User updated successfully | Pass | Mobile, PC |
| Full name: Nguyen Van Edit |
| Click "save" button |
| 9 | Admin | Delete student | Choose "Nguyen Van Edit" | User deleted successfully | User deleted successfully | Pass | Mobile, PC |
| Click "delete" button |
| Click "save" button |
| 10 | Admin | Edit tutor | Click "edit" button of "Nguyen Van tutor" | User updated successfully | User updated successfully | Pass | Mobile, PC |
| Full name: Nguyen Van Edit |
| Click "save" button |
| 11 | Admin | Delete tutor | Choose "Nguyen Van Edit" | User deleted successfully | User deleted successfully | Pass | Mobile, PC |
| Click "delete" button |
| Click "save" button |
| 12 | Student | Can log into the application by entering my email and password by account by Admin ( level 2) created | Email id: student@gmail.com | eTutor home | eTutor home | Pass | Mobile, PC |
| Password: 123456 |
| 13 | Student | Can reset my password when I have forgotten that password | New password: Test@851853 | Login with new password successful | Login with new password successful | Pass | Mobile, PC |
| Confirm your password: Test@851853 |
| 14 | Student | Can view my personal information | Click icon "User" | Show personal information | Show personal information | Pass | Mobile, PC |
| 15 | Student | Log out application from my account | Click icon "Logout" | Logout account and go to login page | Logout account and go to login page | Pass | Mobile, PC |
| 16 | Student | Will have a personal dashboard summarizing my interaction with their personal tutor. | Click "My classes" | Personal dashboard | Personal dashboard | Pass | Mobile, PC |
| 17 | Student | Will get notification emails when Admin(level 2) add or change class | Added by admin to class | Send "New assigned class" to email | Send "New assigned class" to email | Pass | Mobile, PC |
| 18 | Tutor | Can reset my password when I have forgotten that password | Click "Forget your password" at Login dashboard | Login with new password successful | Login with new password successful | Pass | Mobile, PC |
| Input email: "tutor@gmail.com" |
| Check email and click the link to reset password |
| New password: Test@851853 |
| Confirm your password: Test@851853 |
| 19 | Tutor | Can log into the application by entering my email and password by account by Admin ( level 2) created | Email id: tutor@gmail.com | Login successfully and go to tutor screen | Login successfully and go to tutor screen | Pass | Mobile, PC |
| Password: 123456 |
| 20 | Tutor | Can view my personal information | Click icon "personal information" | Show personal information | Show personal information | Pass | Mobile, PC |
| 21 | Tutor | Have a dashboard of their tutor that can be sorted and filtered appropriately | Click "My classes" | Show dashboard of tutor | Show dashboard of tutor | Pass | Mobile, PC |
| 22 | Tutor | Will get notification emails when Admin(level 2) add or change class | Change class by admin | Get notification email | Get notification email | Pass | Mobile, PC |
| 23 | Tutor | Will get notification emails when Admin(level 2) add or change student | Change student by admin | Get notification email | Get notification email | Pass | Mobile, PC |

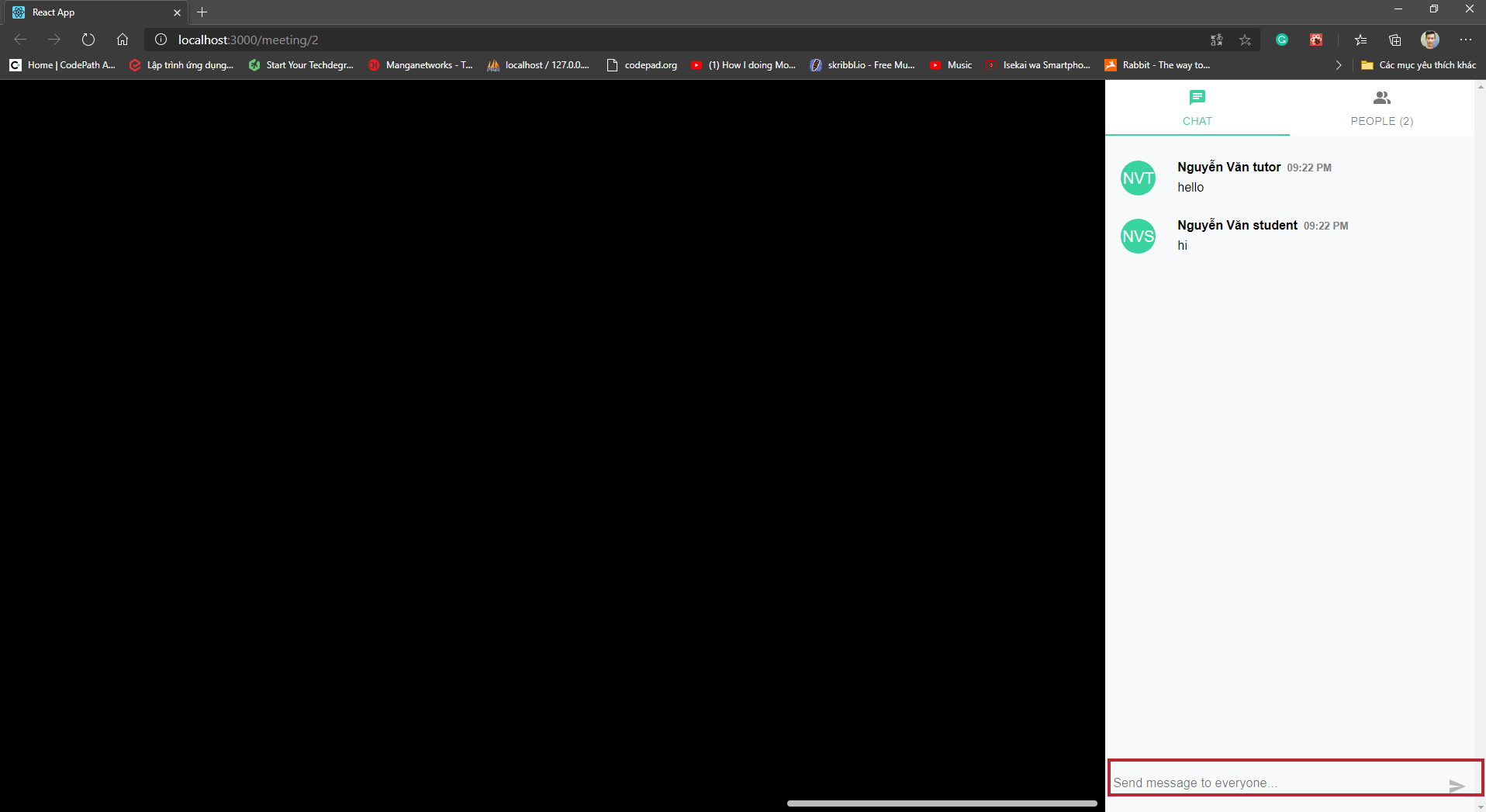
## Test sprint 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Owner | Action | Input | Expected Output | Actual Output | Test Result | Device |
| 1 | Admin | see number of messages in last 10 days | Click "Statistic" | Show report "Number of messages last 10 days" | Show report "Number of messages last 10 days" | Pass | Mobile, PC |
| 2 | Staff | can reset my password when I have forgotten that password | Click "Forget your password" at Login dashboard | Login with new password successful | Login with new password successful | Pass | Mobile, PC |
| Input email: "admin@gmail.com" |
| Check email and click the link to reset password |
| New password: Test@851853 |
| Confirm your password: Test@851853 |
| 3 | Student, Tutor | use the eTutor system for messaging and video call meetings | Send a message in the text box and can see photos from the webcam from both sides | Send message and see video both sides | Send message and see video both sides | Pass | Mobile, PC |
| 4 | Student, Tutor | can upload files | Upload file word type ".doc" | Upload file successfully | Upload file successfully | Pass | Mobile, PC |
| 5 | Student, Tutor | comment document and for blogging | Comment document under files uploaded | Display comment | Display comment | Pass | Mobile, PC |

**Error missing message box of small display**



Error missing message box of small display (1366 x 768)

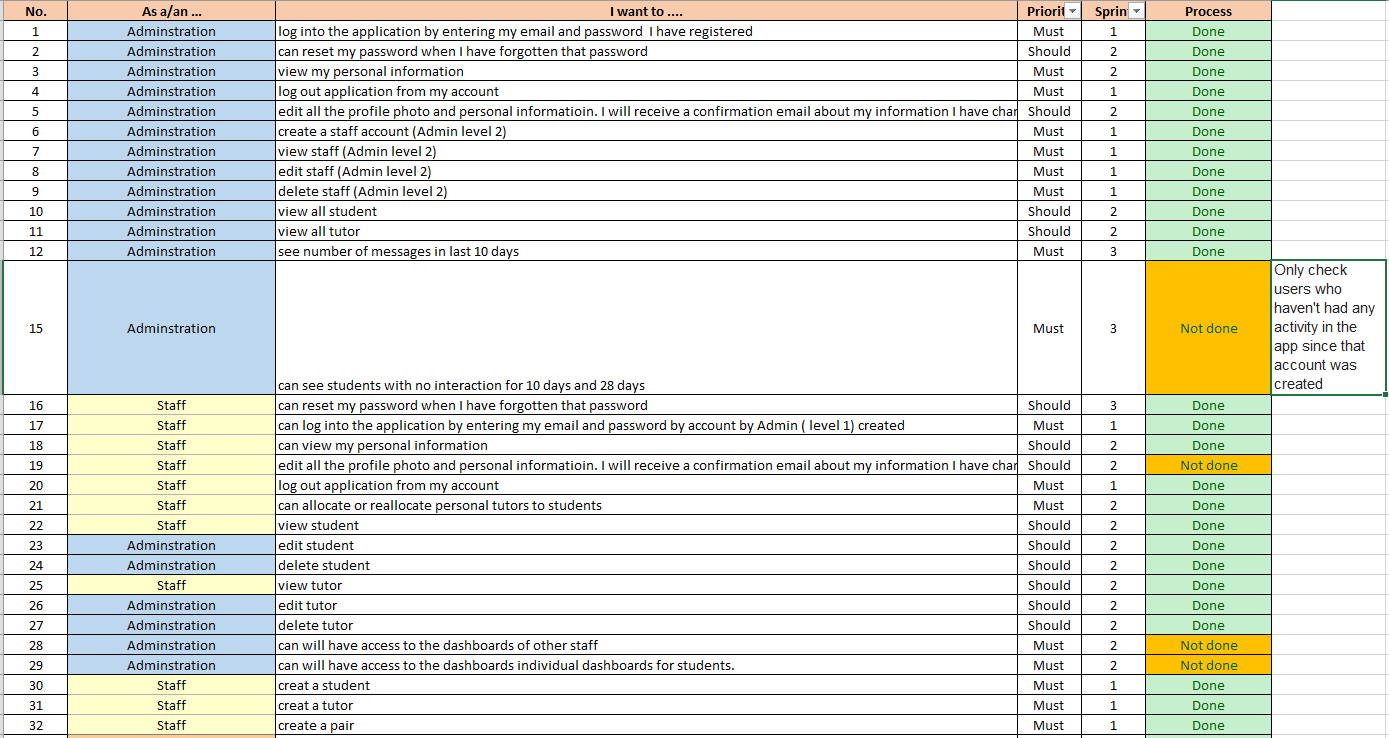


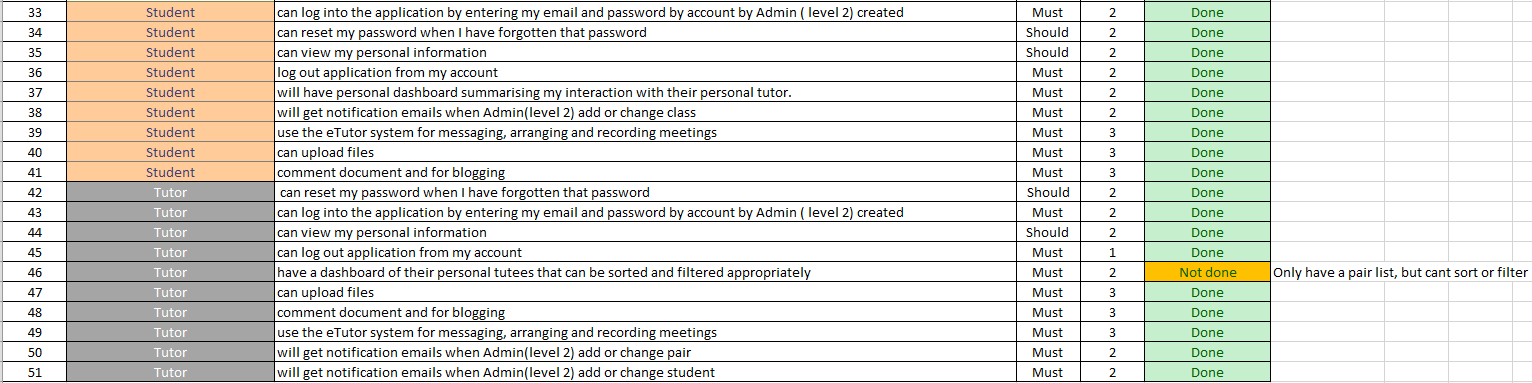
Display normal with display 1920 x 1080

As we can see in the two pictures above, the "chat box" is lost on small screens but not on mobile screens and large screens.

# Agile methodology

After we have the user story, we start to put those functions into the product backlog and then divide them by priority and sprint.

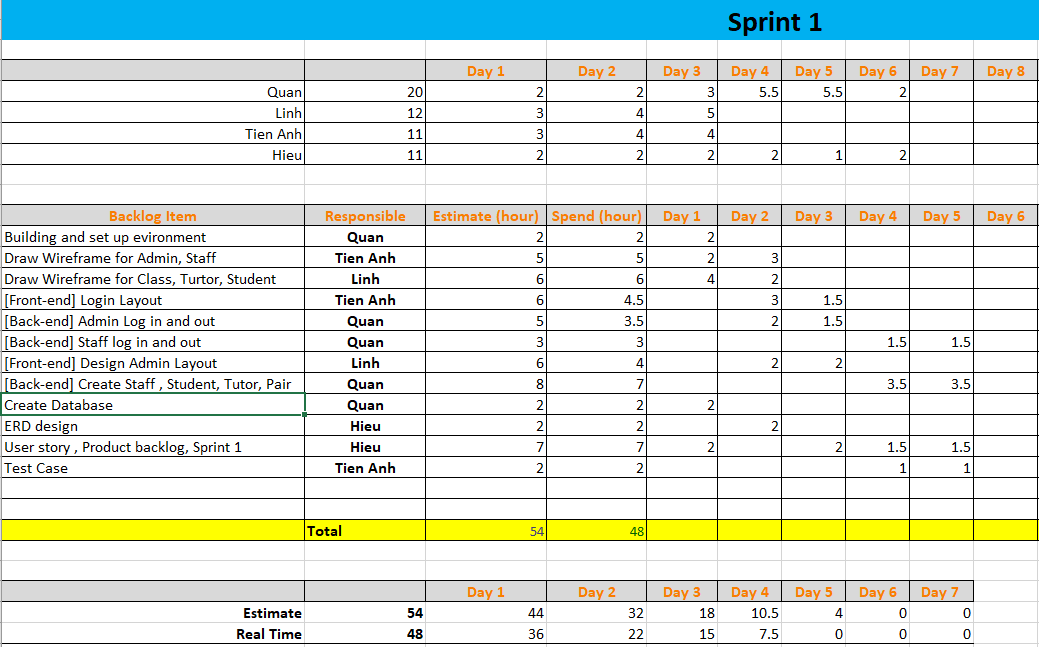


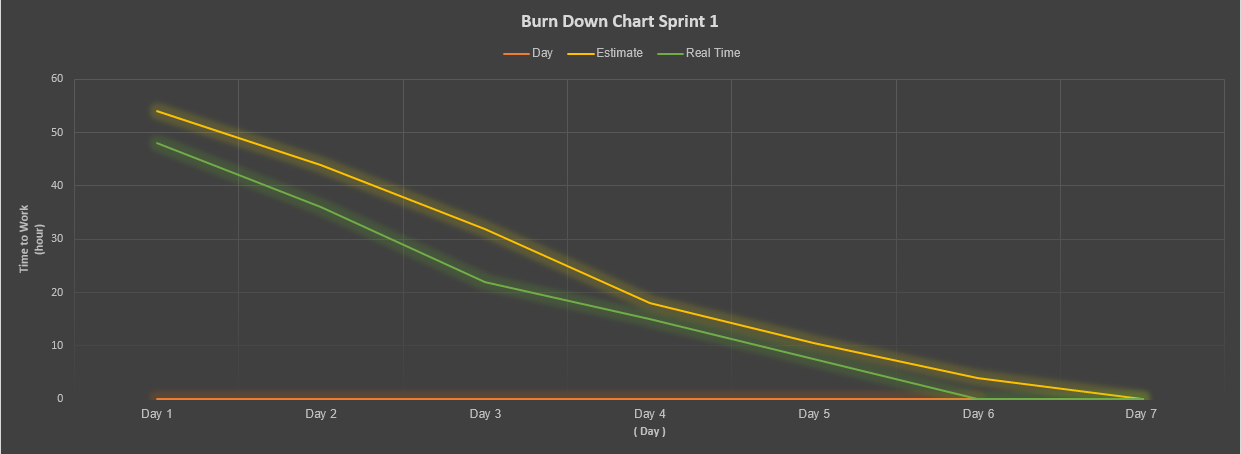


In the product backlog we used MoSCoW to divide the priorities. The MoSCoW strategy is a prioritization procedure utilized in the executives, business examination, venture the board, and programming improvement to arrive at a typical comprehension with partners on the significance they place on the conveyance of every necessity. (Madsen, 2019). Especially here we have divided into Must (have) the required requirements that the course work requires and should (have) these functions are also important but it does not specify the success or failure of the programming.

We then divided it into 3 sprints with a request for a refund between 7 and 14 days:

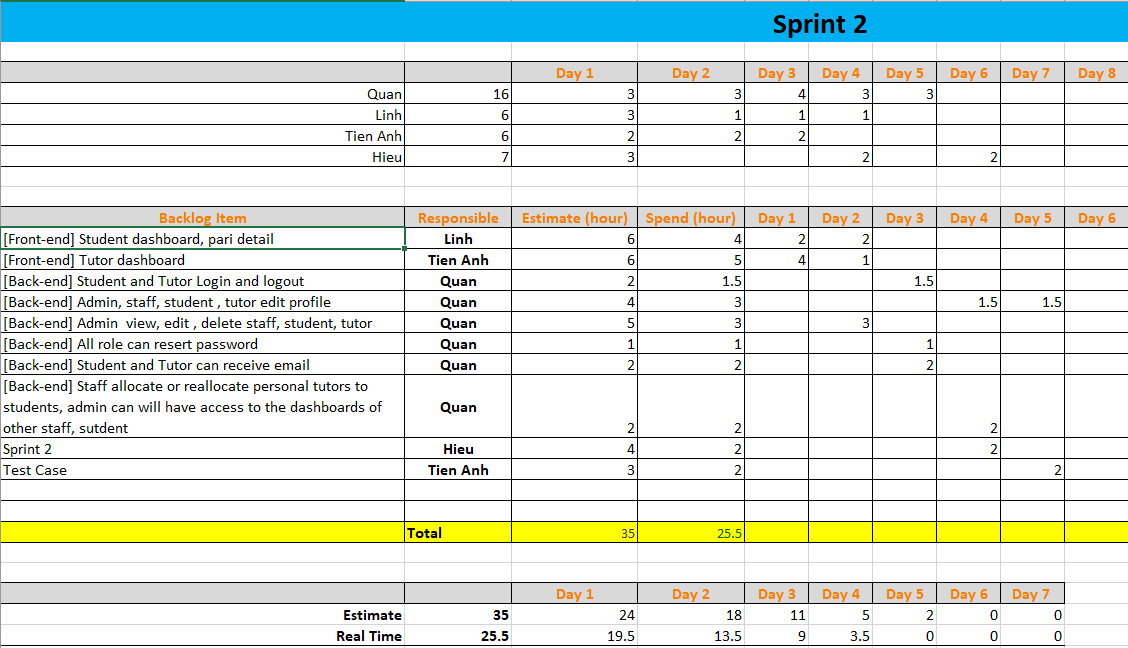
## Sprint 1:

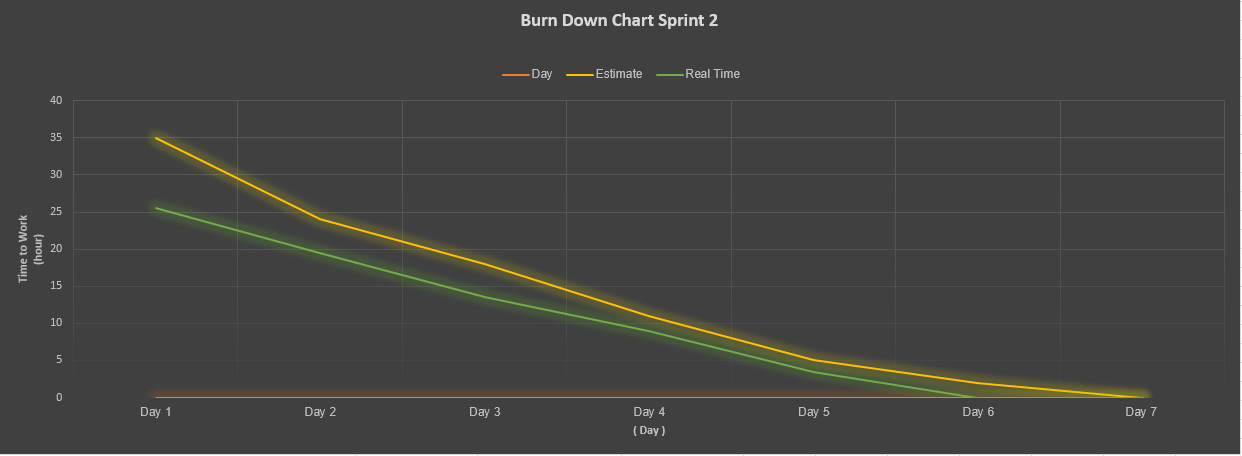




In sprint 1 we focus on learning as well as interface design and create DB for the project we are working on. All team members will give an estimated time to complete the tasks assigned in this sprint 1. We got started in pretty soon and got it done quickly. According to the Burn down chart we completed the project within 6 days earlier than expected one day. So it can be said that we have completed the task in this sprint 1.

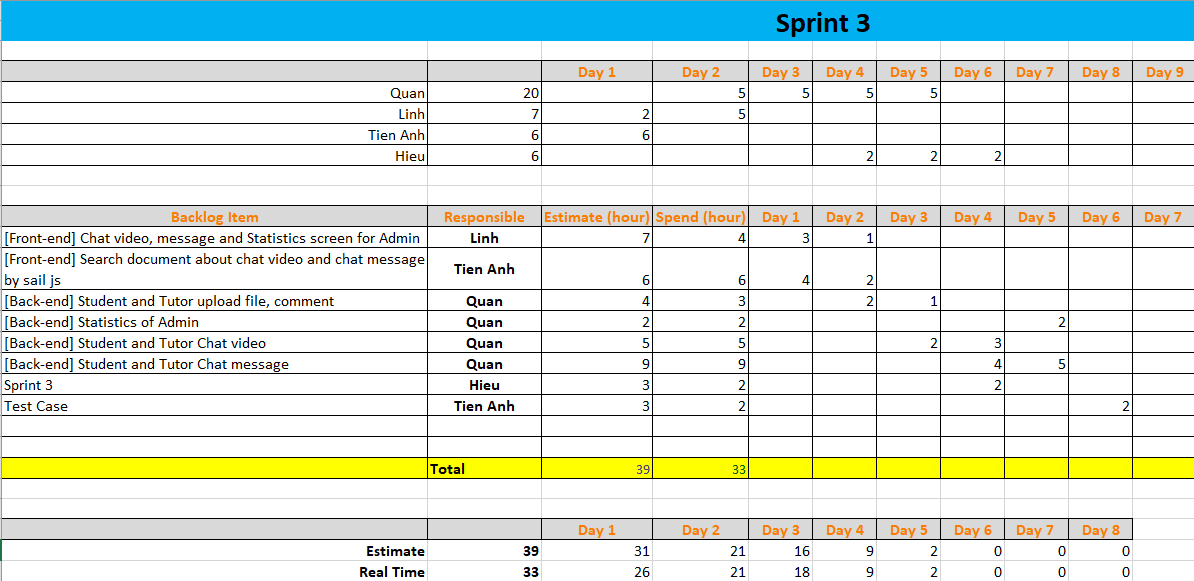
## Sprint 2:

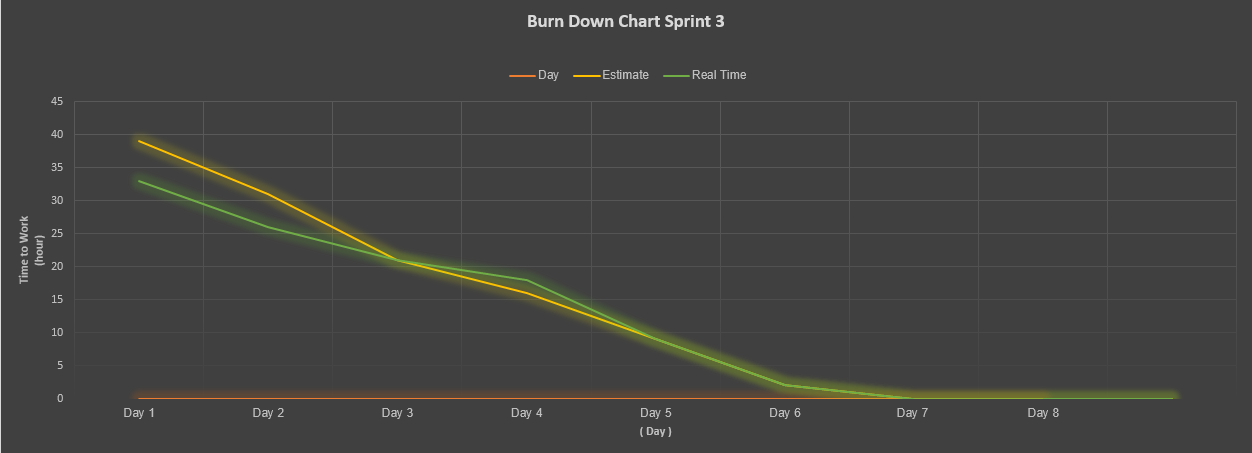




After we have finished sprint 1 and have had enough documentation about sprint 1, we have started to implement sprint 2 shortly, in this sprint 2 we focus mostly on the backend code about create, assign tutor and student in a pair and staff can allocate or reallocate personal tutors to students, admin can will have access to the dashboards of other staff, student. And just like sprint 1 we completed it in 6 days faster than expected 1 day

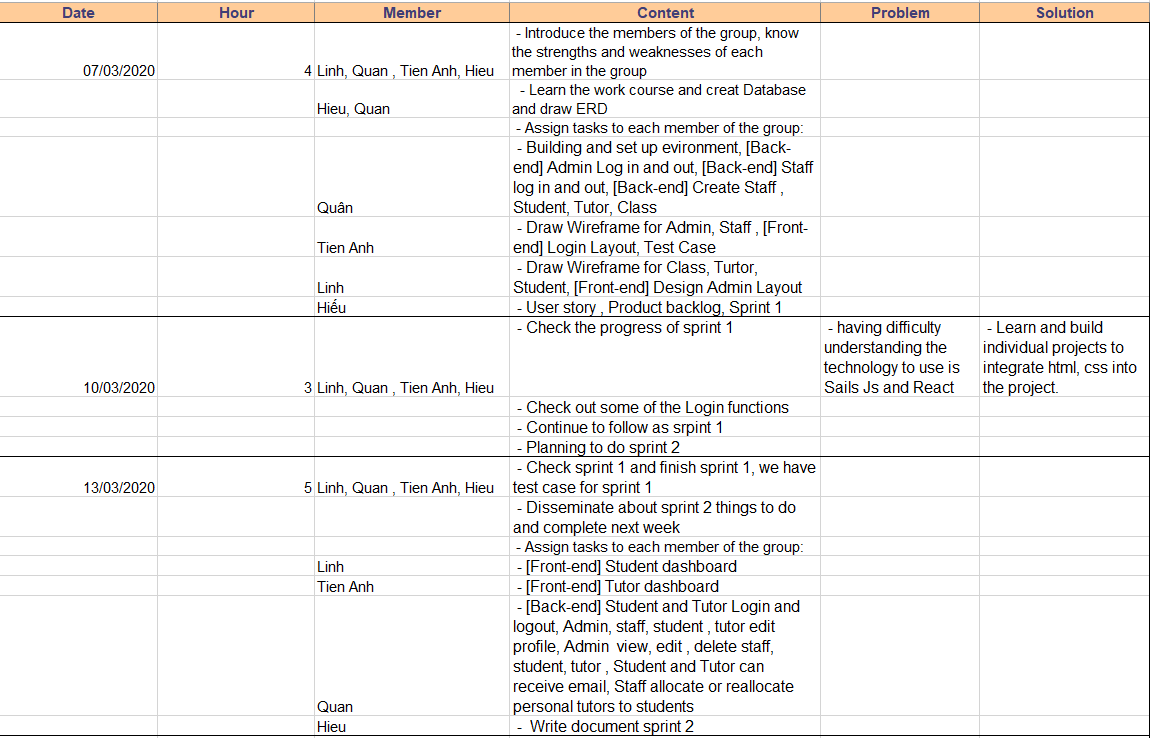
## Sprint 3:

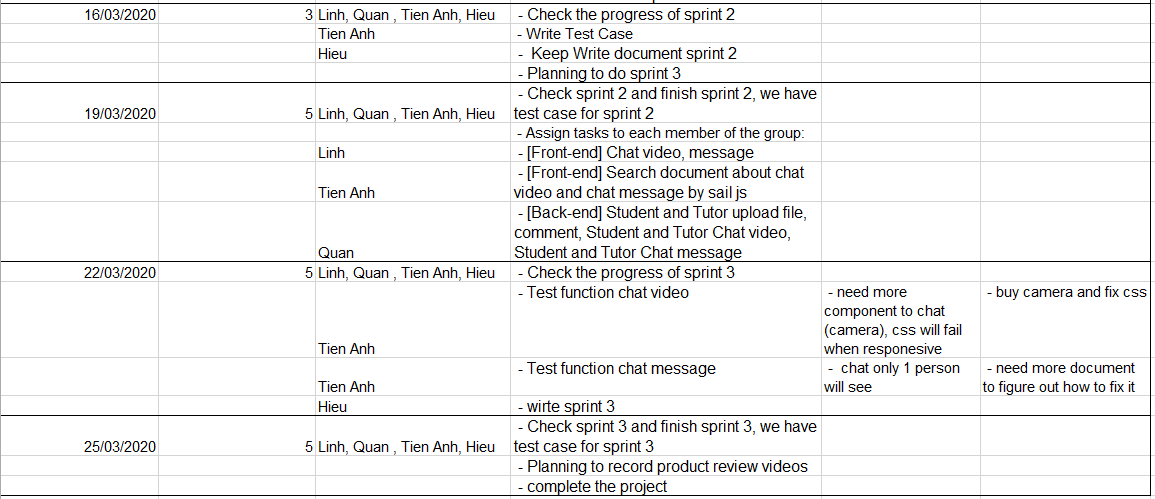




In this sprint 3 we perform the final project functions including admin reports, updates, comment files and video and message chat. It can be seen in the burn down chart that by the fourth day the project progress was not as good as the estimate because we had difficulties and bugs during the video chat and message coding stage of tutor and student. But then we got together to find a solution and then completed it on schedule as originally estimated.

## Summary meetings minutes:





This is a document that records all our meetings during the project. Every three days we sit down and check to see how well each of us has done to see if everyone has any problems and solve it, but we can see that we have successfully completed it. Each meeting we take 3-5 hours to meet depending on the level of work that day. Only in sprint 3 did we have a bit of trouble, but everything was solved and we have perfected the program.

# Bibliography

Madsen, S., 2019. *projectmanager.com.* [Online]   
Available at: https://www.projectmanager.com/training/prioritize-moscow-technique  
[Accessed 20 04 2020].