****

**B.Sc. (Honors) Computing – Final Year (Top-up)**

Enterprise Web Software Development

GROUP Report

Term 2 - MAC

Name of the Group: Group 4

* Nguyen Quang Vinh
* Le Thanh Trung
* ....

Greenwich Course Leader: Mr. Matthew Prichard

Class: TCS2006

Subject’s ID: 1640

Assignment due: 16th April 2020

Assignment submitted: 29th

**ASSIGNMENT BRIEF**

|  |  |  |  |
| --- | --- | --- | --- |
| **Degree** | **Honours Diploma in Computing** | | |
| **Unit number** | Term 2 – Level 6 – CW – COMP1640 | | |
| **Assignment title** | Enterprise Web Software Development | | |
| **Academic Year** | 2019 – 2020 | | |
| **Unit Tutor** |  | | |
| **Issue date** |  | **Submission date** |  |
| **IV name and date** |  | | |

|  |
| --- |
| **Submission Format:** |
| *Format:* The submission is in the form of 1 document  You must use font *Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm.* The reference follows Harvard referencing system.  *Submission:*   * An electronic copy of your work for this coursework should be fully uploaded by midnight (local time) on the Deadline Date. * The last version you upload will be the one that is marked. For this coursework you must submit a single Acrobat PDF document. In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). * For this coursework you must also handling this artefact: Links to repository and screencasts * There are limits on the file size. The current limits are displayed on the coursework submission page on the Intranet * Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions. * Comments on your work will be available from the Coursework page on the Intranet. The grade will be made available in the portal. * You must NOT submit a paper copy of this coursework. * All coursework must be submitted as above   The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See http://www2.gre.ac.uk/current-students/regs for details.  *Note:* **Plagiarism** *is presenting somebody else’s work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student’s coursework; stealing or buying coursework from someone else and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University.*  *All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using. Your work will be submitted for electronic plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.* |
| **Deliverables:** |
| **An Individual PDF Report**  The report must give the URL of the Group Repository, the Screencast and the website and any usernames or passwords needed to access it. The individual component of the marking will be based on your report, so ensure this has evidence that your system meets the specified requirements. The text in your individual report must be entirely your own words. |
| **Assignment Brief and Guidance:** |
| **Scenario**:  This is a group coursework with a maximum of six in the group.  You need to adopt agile scrum working practices, and document your meetings appropriately. Ideally you need a database designer, a programmer, a web designer and a tester, but you should take on all these roles at various stages as part of the project, and more than one person can be in any role at any time. No one is to take the role of project manager.  You will get an individual grade based on your contribution to the team, and for your individual contribution to the product.  **Specification:**  You are required to build a web-based secure role-based system for eTutoring in a large university. Full details of the system will be given in lectures.  The system must meet the following criteria:  • All students must have a personal tutor.  • Any authorized 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 (eg 10 at a time) 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 summarizing their interaction with their personal tutor.  • Each personal tutor will have a dashboard of their personal tutees that can be sorted and filtered appropriately  • 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)  **Assumptions:**  You must clearly state any assumptions you make.  **Reports:**  A number of reports need to be made available. For example:  • Statistics  o Number of messages in last 7 days  o Average number of messages for each personal tutor  • Exception reports  o Students without a personal tutor.  o Students with no interaction for 7 days and 28 days.  **Tasks:**  1. Work as a team using agile scrum methods to develop and test a secure web-based system to meet the above specification.  2. Create a screencast recording (including screen and sound) demonstrating the key functionalities of the system. This needs to be hosted somewhere (e.g. YouTube) that is accessible by the Greenwich moderator  3. Present the finished product to a non-technical audience to try to persuade them to purchase your system.  4. Document the system to an appropriate standard using a weighted scoring model with commentary, including an evaluation of the design process you followed and your reflection on the finished product, and on the contributions of your team members. |

|  |  |  |
| --- | --- | --- |
| **Assessment Breakdown** | | |
| Group Component | **Total:** 60**%** | This will be assessed based on a group report and a group repository created by the group on a secure shared area accessible to the Greenwich markers. Password and URL must be provided in individual reports. Must be suitably structured with a menu. Suggested location: GitHUB, Google Docs, SharePoint 365, own website, DropBox or other repository. |
| Database | 10% | Expect: Security, appropriate data types and validation, clear ERD, referential integrity implemented, enables roles to be implemented |
| Site design | 10% | Expect: Responsive design, clear information architecture for both mobile and desktop, aesthetically pleasing, good usability, meets accessibility criteria |
| Functionality | 10% | Expect: Role based security, submission of reports, email notification, summary and exception reports, UML diagrams, code snippets |
| Testing | 10% | Expect: Test plan, test log, sufficient data to fully test, evidence of testing finding errors, test items linked to user stories in the product backlog |
| Agile methods followed | 10% | Expect: Burn down chart, minutes of meetings, user stories, sprints, product backlogs |
| Screencast and Presentation | 10% | Expect: Professional standard of presentation promoting the product, with contributions by all the team members, Screencast demonstrating all the main features of the product. Screencast can be done by one person. |

|  |  |
| --- | --- |
| **Indicative Grading Criteria** | |
| >=70% | * Well designed system to fully meet the requirements. * Professional standard of report, with appropriate documentation. * High level of individual commitment. * High level of evaluative commentary |
| 60-69% | * Well designed system to meet most of the requirements * Professional standard of report * High level of individual commitment * Limited evaluative commentary |
| 50-59% | * Well designed system to meet most of the requirements * Acceptable standard of report * Good level of individual commitment * Limited evaluative commentary |
| 40-49% | * Acceptable system to meet most of the requirements * Acceptable standard of report * Acceptable level of individual commitment * Limited evaluative commentary |
| <40% | * Poorly designed system * Few requirements met * Poor standard of report * Limited individual commitment * No evaluative commentary |

Contents

[**SCHEDULE** 7](#_Toc33889123)

[**AGILE METHODS FOLLOWED** 7](#_Toc33889124)

[*Burn down chart* 7](#_Toc33889125)

[*Minutes of Meetings* 7](#_Toc33889126)

[*User stories* 7](#_Toc33889127)

[*Sprints* 7](#_Toc33889128)

[*Product backlogs* 7](#_Toc33889129)

[**DATABASE** 8](#_Toc33889130)

[*SECURITY* 8](#_Toc33889131)

[Pros: 8](#_Toc33889132)

[Cons: 8](#_Toc33889133)

[*Validation & Datatypes* 8](#_Toc33889134)

[*ERD*: 8](#_Toc33889135)

[*Database Explaination:* 8](#_Toc33889136)

[**SITE DESIGN** 8](#_Toc33889137)

[*What is Responsive Design* 8](#_Toc33889138)

[*How do we apply it to the project* 8](#_Toc33889139)

[*Interface Design* 8](#_Toc33889140)

[*How is this work* 8](#_Toc33889141)

[**FUNCTIONALITY** 9](#_Toc33889142)

[*The application of Role base system* 9](#_Toc33889143)

[*The Report from the system* 9](#_Toc33889144)

[*The Email from the system* 9](#_Toc33889145)

[*UML diagram* 9](#_Toc33889146)

[**TESTING** 10](#_Toc33889147)

[*Test Plan* 10](#_Toc33889148)

[*Test log* 10](#_Toc33889149)

[*ERROR* 10](#_Toc33889150)

[**SCREENCAST & PRESENTATION** 10](#_Toc33889151)

[**LOCATION REPOSITORY** 10](#_Toc33889152)

[**REFERENCES** 10](#_Toc33889153)

[Figure 1: Scrum Method 8](#_Toc35636582)

[Figure 2: Role base function 13](#_Toc35636583)

[Figure 3: Message function 14](#_Toc35636584)

[Figure 4: Allocating function 16](#_Toc35636585)

[Figure 5: Meeting function 17](#_Toc35636586)

[Figure 6: Upload funtion 18](#_Toc35636587)

[Figure 7: Blog function 19](#_Toc35636588)

[Figure 8: Home Page 20](#_Toc35636589)

[Figure 9: Student Dashboard 20](#_Toc35636590)

[Figure 10: Student Joining Meeting Page 21](#_Toc35636591)

[Figure 11: Tutor List Page 21](#_Toc35636592)

[Figure 12: Tutor Holding Meeitng Page 22](#_Toc35636593)

[Figure 13: Tutor Dashboard 23](#_Toc35636594)

[Figure 14: Document Upload Page 23](#_Toc35636595)

[Figure 15: User Role Management Page 24](#_Toc35636596)

[Figure 16: Add New Role Page 24](#_Toc35636597)

[Figure 17: Admin Dashboard 25](#_Toc35636598)

[Figure 18: Admin Add User Page 25](#_Toc35636599)

[Figure 19: Allocating Class Page 26](#_Toc35636600)

[Figure 20: Allocating Student Page 26](#_Toc35636601)

[Figure 21: Allocating Select Tutor Page 27](#_Toc35636602)

[Figure 22: Blog Comment Page 27](#_Toc35636603)

[Figure 23: Blog Main Page 28](#_Toc35636604)

[Figure 24: Upload Comment Page 28](#_Toc35636605)

[Figure 25: Staff Create new account page 29](#_Toc35636606)

[Figure 26: Admin Change user Profile page 29](#_Toc35636607)

[Figure 27: Users Change Profile page 30](#_Toc35636608)

[Figure 28: Upload Main page 30](#_Toc35636609)

[Figure 29: General Meeting Manager 31](#_Toc35636610)

[Figure 30: Message Board 31](#_Toc35636611)

[Figure 31: Reset Password Page 32](#_Toc35636612)

[Figure 32: Admin Dashboard 32](#_Toc35636613)

[Figure 33: Student List Page 33](#_Toc35636614)

# **SCHEDULE**

* The plan is to use the Agile methods – Scrum – to develop the application. First, because the current team is small (just 6 people) and the Second is because the application is not too big to carry out to begin with.
* The database will need to be done first because it is the exoskeleton of the project, the database must be built base on the customer requirement and developers experience. And after that, the application design will be carried out. This second step should be paid more attention because that is what the stakeholders see on the very first sign, and they may give back a lot of feedbacks. After that the developer team will use Java to build the functions and stick everything together.
* Work distributed for developer members:
* Mr. Vinh: Main Database design/develop, Web Design, and take a part of Programing/Testing
* Mr. Trung: Mainly Testing, take a part of Web Design, take a part of Programing
* Mr. Thong: Mainly Testing/Documenting, take a part of Programing
* Mr. Phong: Mainly Backend Develop, Programing
* Mr. Hieu: Mainly Backend Develop, Programing
* Ms. Hue: Mainly Frontend Develop, Web Design, Programing
* There will be the meeting at every Friday of the week, for the next following 8 weeks. The meeting will be hold at the Coffee House at 22 TruongDinh Street – District 3 – HoChiMinh City. That will be all about what already done during the week, and distributed tasks that need to do for the next week.

# **AGILE METHODS FOLLOWED**

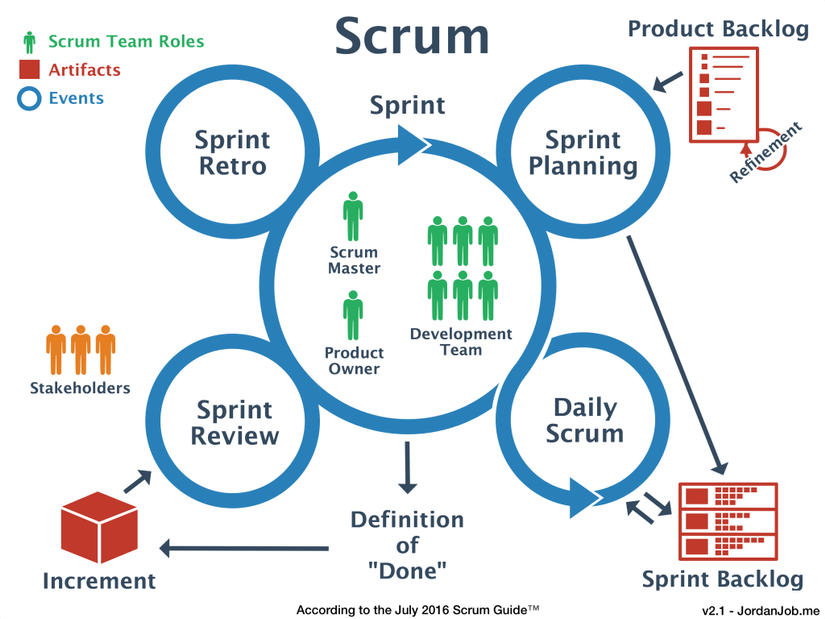


Figure 1: Scrum Method

## What is Scrum?

Scrum offers a structure for the efficient production of diverse goods. ' It should be noted that this is a general structure for the management of complex goods, mainly applications. Nevertheless, Scrum may be viewed as a forum for coordinating activities ranging from agile project management in general, product creation, publicity strategy, teaching and growth. Modular vehicles or other specific work.

The Scrum description is actually published in the Scrum Guide manual and revised periodically by the developers of Scrum Guides.

It should be remembered that Scrum is a system and not a particular process. As implemented, Scrum offers the fundamental framework, paired with certain approaches or techniques to maximize the result.

## Three Roles on scrum

Throughout Scrum, the product engineering team is split into three positions with distinct duties to insure that different projects are streamlined. These three positions are: Brand Manager (Company Seller), Scrum Master and Design Group (Production Group or Project Team).

Product Owner: The person responsible for the success of a project, who defines the requirements and ultimately evaluates the output of software developers.

Scrum Master: A person who has a clear knowledge of Scrum and guarantees that the team can function successfully with Scrum.

Development Team: A cross-functional self-managed team that handles the program configuration specifications grouped in the Software Backlog.

## The Core Values of the Scrum.

Scrum is an agile approach that implements the ideals of the Manifesto for the Creation of Agile Applications Manifesto. In fact, Scrum works on the basis of three fundamental principles, commonly known as Scrum's Three Legs, namely integrity, examination and adaptation.

Transparency: In Scrum, openness is advertised as the most important core principle. For order to work in Scrum, details relating to the production phase will be clear and open. These details can include: marketing vision, consumer needs, job schedule, challenges and obstacles, etc. Through which all have the required knowledge in every position. Make strategic choices to boost job performance. Scrum resources and workshops insure the communication is available to the stakeholders.

Inspection: The continuous review of events in Scrum guarantees the identification of issues as well as approaches so that relevant and valuable knowledge can be rendered accessible to the project participants. A detailed and ongoing analysis is the initiating point for integration and continual development of Scrum.

Adaptation: Scrum is as robust as any other agile product development system. Thanks to this, it gives really strong adaptability. Across the basis of consistent knowledge from review and job processes, Scrum is able to adapt to improvements in a constructive way, adding progress to the project.

## *Burn down chart*

### What is Burndown Chart

The Sprint Burndown map displays revised regular averages of the volume of research that needs to be done. The Sprint Burndown map summarizes the approximate details from the Sprint Backlog.

The Sprint Burndown model allows the manager to realize the actual task situation. When the process moves more gradually than anticipated, it is projected that this Sprint will not be able to finish the task, and the team will concentrate on delivering the most critical content to accomplish the target. Sprint and work for the POs. Especially in this sluggish scenario, the squad finds that they encounter certain challenges in order to meet the target of the Sprint. From this stage, the team can come together to examine the cause of the impediments and the necessary adaptation steps.

However, this map is just an approximation, and the team does not depend too heavily on the map, but can continue to check and adjust on the basis of others realistic research.

The current burndown chart of the whole project

Figure 2: Burndown chart

## *Minutes of Meetings*

Meeting minutes are meeting minutes. The key function is to preserve the meeting material, which is known to be a memo or meeting record.

Meeting minutes play a significant role in some businesses as a deliverables tool to consumers, rather than other daily updates.

The primary aim of meeting minutes is to log anything that occurs in the conference, a report for members, as well as exchange details with non-present individuals and all interested parties. Meetings, despite helping to discourage the loss of such research happens only because of the forgetfulness of any person.

You can log conversation points, suggestions, current situation and actual activities in meeting minutes, as well as task plans, and it can also be used as a job guide to better improve the consistency of the work and support. Carry out more successfully the next meeting.

In the office, multiple sessions frequently take place at varying intervals, such as regular, weekly, monthly, ... It's a responsibility to log meeting minutes for meetings of high volume and relevant material. Needs more meticulousness, more consideration and more elaboration than other people assume.

We will discuss the key points in this article to create successful meeting minutes and resources to help you do it, hopefully it will be useful for those who have trouble writing minutes or those who want to minimize the time taken to produce a meeting minutes.

* Toàn bộ report của minutes of meeting (table) + đưa ra 1 file riêng để gửi riêng cho trường

## *User stories*

* Định nghĩa sơ user stories trong scrum
* Toàn bộ user stories trong dự án (dưới dạng table)

## *Sprints*

Sprint is Scrum's heart, providing the flow, product creation process for users to upload. A sprint can be seen as a project in which the sprint would restrict the duration of the project to 1 month or less.

Image Sprint will have a no more than 30-day Time-box.

The Speed Target remains unchanged at Speed. The sprint will be canceled if the Sprint goal no longer exists

In Sprint Preparation the Scrum team must build the Sprint Target.

The marketing department is responsible for product development and at the end of the Sprint, produce Completed Percentage.

The next sprint will begin as soon as the current sprint ends.

Sprint 0, Layout Sprint or Hardening Sprint are not available.

* Toàn bộ các sprints đã chạy trong dự án (dưới dạng table), sprint backlog + đưa ra 1 file riêng để gửi riêng cho trường
* Dưới mỗi table phải giải thích tại sao có sprints này (ví dụ, Login là function quan trọng then chốt, nên phải có sprints làm Login riêng)

## *Product backlogs*

The User Backlog is where you store the list of requested software functionality. The collection is structured on the basis of each item's value. The higher priority products are at the top of the list and will be chosen for early production by the Marketing Department, the lower priority products will be at the bottom of the list, and produced later.

The Brand Manager is responsible for the monitoring and control of the Backlog. It involves defining material (products to be developed), prioritizing and organizing products, optimizing items, clarifying and answering all issues relating to the product.

The Product Backlog can include elements in these categories:

* Product features
* Failure
* Technical Work
* Analysis research

Things from Product Backlog can be represented in several different ways. A typical way of doing this is to use User Story (a User Story is a concise summary of the product's desired function from the user's viewpoint, it has the form: Is ... I want ... to ...

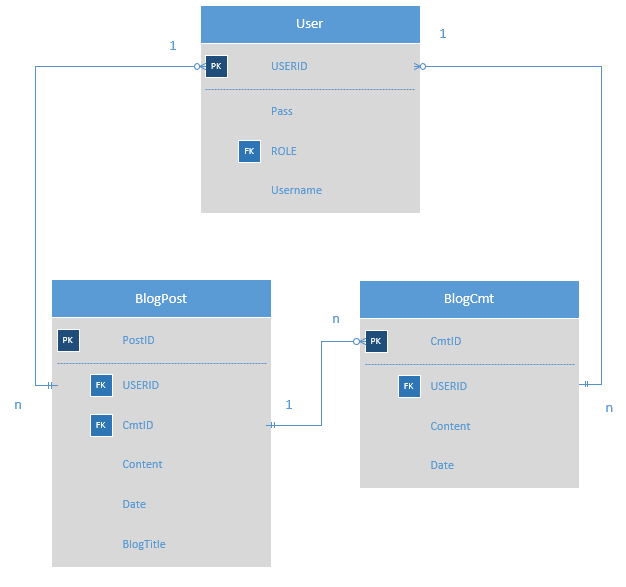
Example: As a consumer I want to see a list of items to select from.) But this is not the only way to pick the correct type for each category, such as using User Situation, User Scenario, etc. It should be succinct, succinct and simple, no matter which approach the team uses.

* Product backlogs của dự án + đưa ra 1 file riêng để gửi riêng cho trường
* Giải thích tại sao có product backlog này (ví dụ: khách hàng yêu cầu Student phải có dashboard)

# **DATABASE**

## *SECURITY*

### Pros of relationship database:

* 

Previously, users also only operated their data by different records. This strategy just fits small-scale data collection environments. But the case needs to handle vast volumes of data, it means that we use the database. Data processing by database services can have beneficial benefits and help to easily store, access, use and modify data and to reach a high degree of job efficiency.

The benefit of data processing is that it prevents replication which helps in a continuity of data which the manager cannot manually monitor. The data formats maintained maintain consistency and data validity. Around the same time, it offers other opportunities for communicating with other team leaders, allowing us to freely share and update data in order to tackle the problem. Solve rapidly developing issues. In short, we can sum it up as the few lines below:

easy to use and understand, because information is stored in tables, organized in rows and columns, much as same as a spreadsheet.

* join table query.
* avoid data duplication.
* avoid inconsistent records.
* better security.
* Tables can be made accessible only to those who need specific information.
* cater for future requirements.
* familiarity.

### Cons of relationship database:

* When you try to put any filed into columns, you experience difficulties.
* Restrict the duration of data areas, which ensures knowledge will be lost if you insert more data into an area than it can handle.
* The output of the reassembly into their more complicated real-world interpretations of basic data structures.
* If you enter complicated data, SQL is tight.
* To build ad hoc requests, you need to learn the layout of the database.
* Lacking framework specified by RDBMS does not permit design transactions to be assisted. Lack of support of specific premise forms such as drawing ...

## *ERD*:

Entity relationship Diagram (without datatype version)

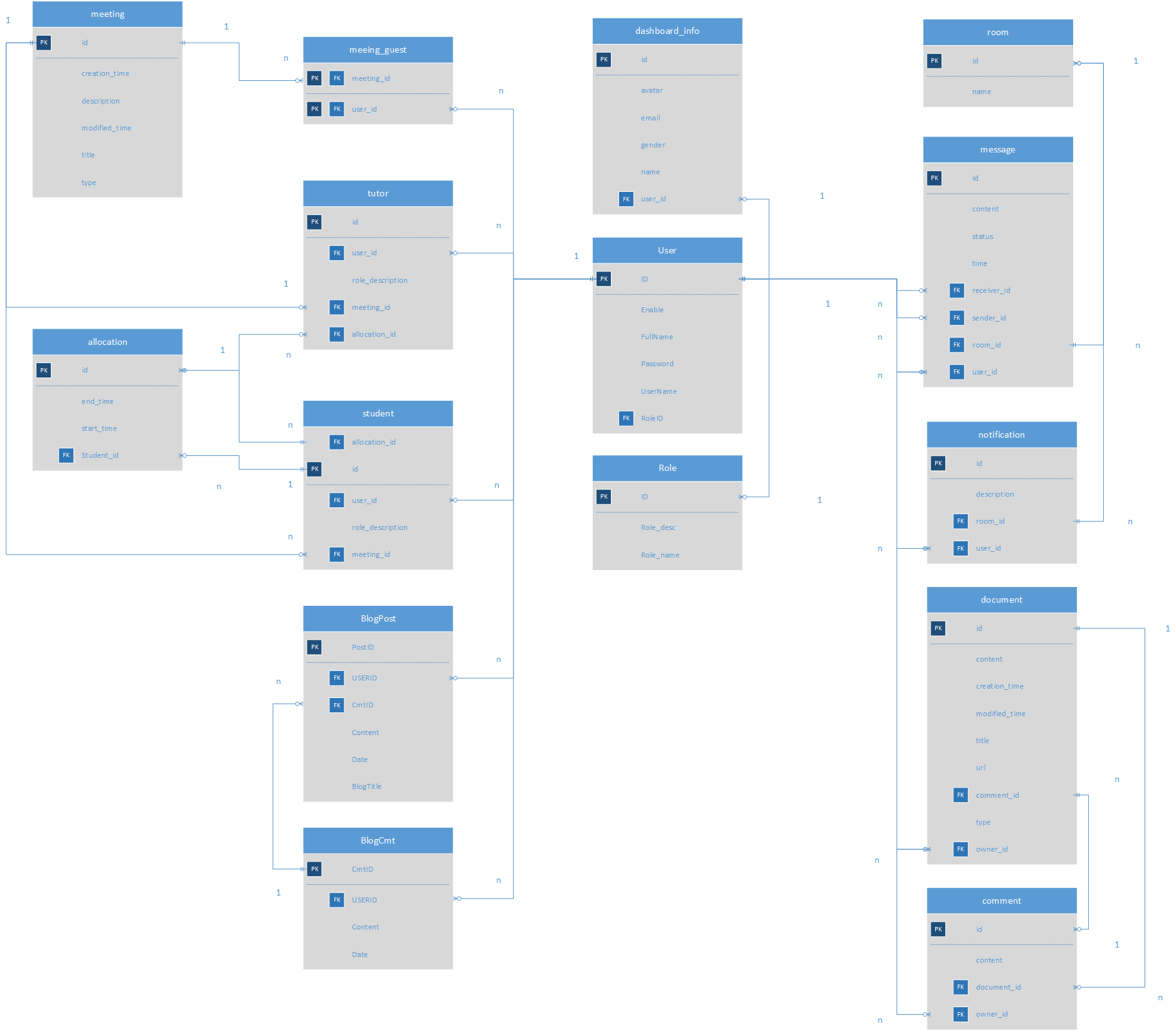


Figure 3: EDR without datatype

There is the ERD of the database that will be used on this project, but this one is the one without the datatype. It shows really clearly the relationship between each table. Although the picture is a quite small, please check the explain below for more information when we dig into the database one-by-one function.

## Entity relationship Diagram (with datatype version)

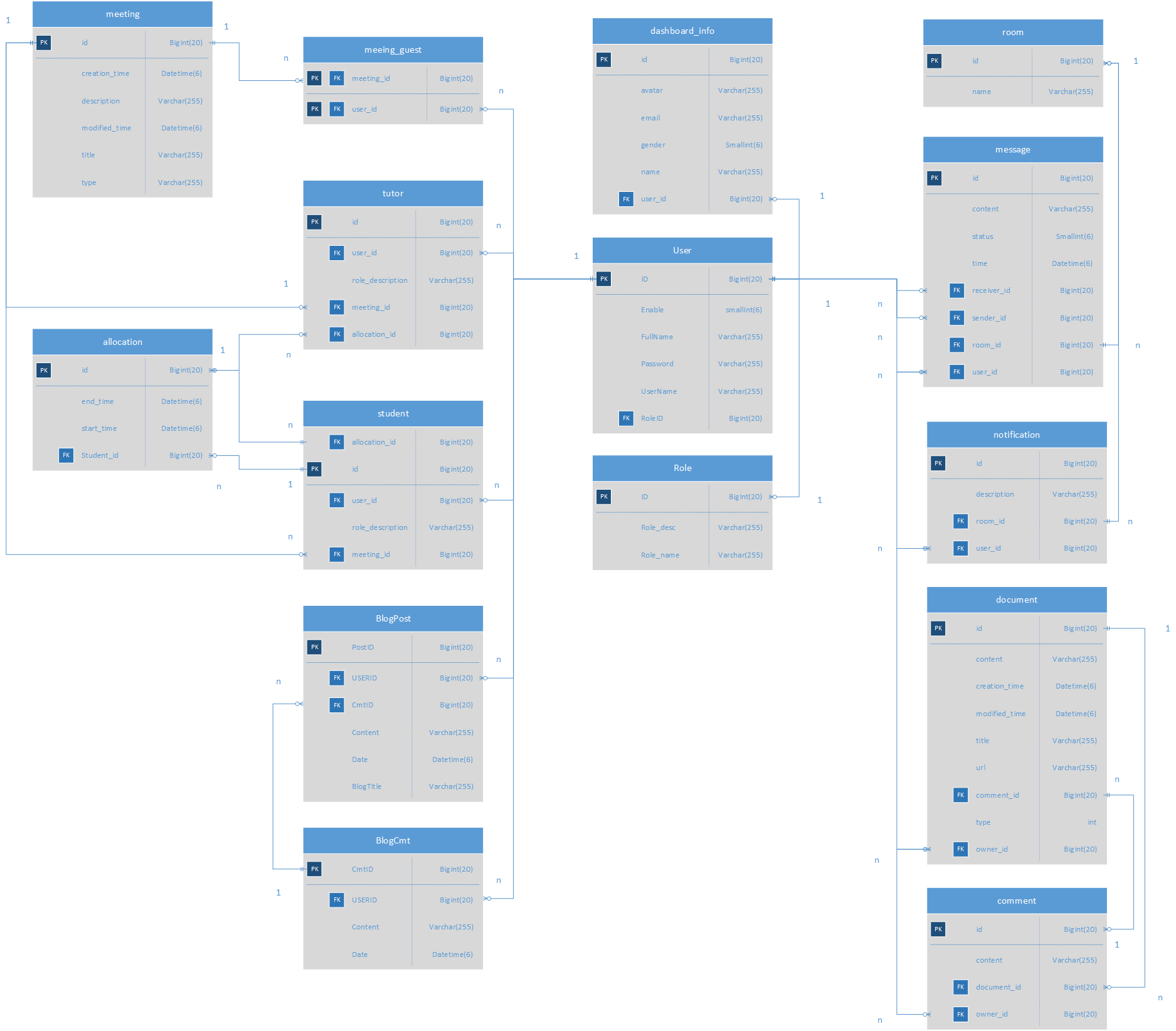


Figure 4: ERD with datatype

Another ERD for the database, but this is the version with datatype and its limit. The picture is quite small when the database itself is quite large, so please check the explain section below for more information.

## *Database Explaination:*

### Role base function

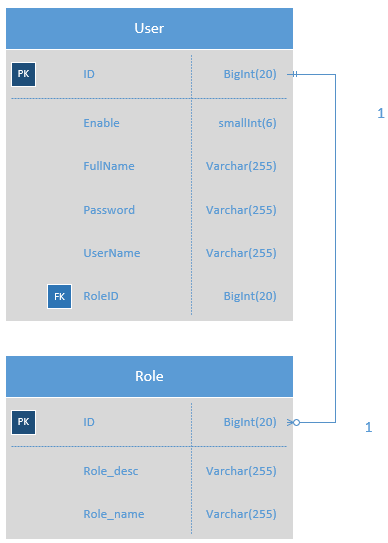
****

Figure 5: Role base function

When user using login function, the table User and Role will be in used. Besides having different UserID, each user also has RoleID to identify which level of authority they have, such as being a Staff, or a Student, or a Tutor ... will be determined by the RoleID. Please keep in mind that the RoleID field may or may not have the same name as the role of user in real life, for example: the RoleID can be 1, but the real-life role of user is Staff.

Since the delete function is necessary for the user, the field “Enable” has been put in, each and every time user log in, the system will check if this field is valid to see the status of the account. Why do we just use the delete function on database, you ask? Simply because this is a relationship database, so delete 1 data may cause the huge problem for the other data, because they have “relationship” with each other, so the best way is to keep it as it is, and then just “hide” them.

### Message function

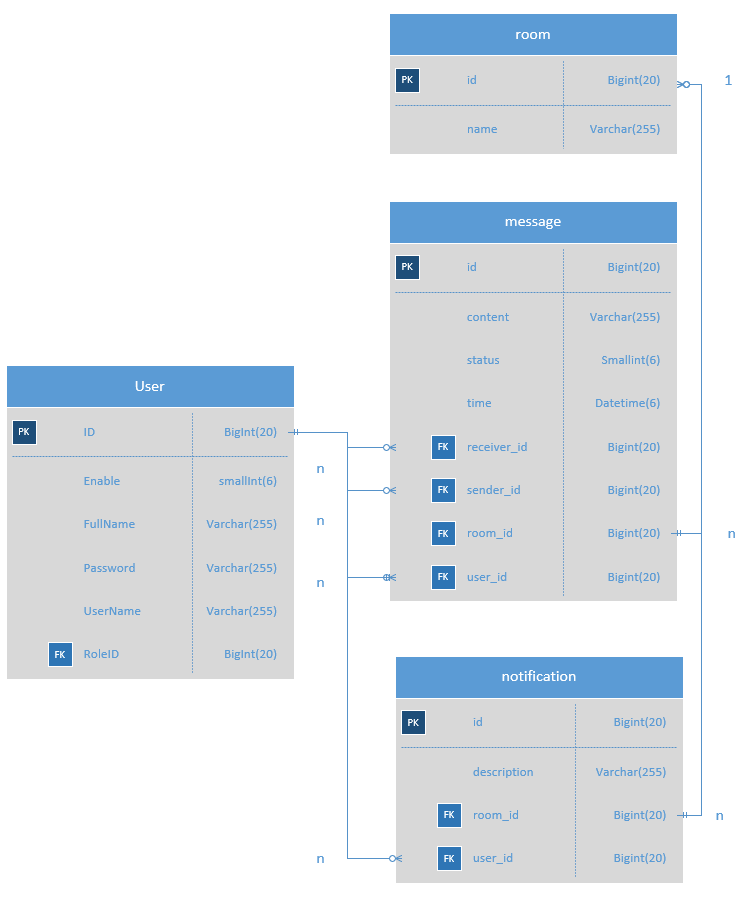


Figure 6: Message function

When the user enters Message Page, the User and Message table will be used. Each User already have an identical UserID, and the SenderID and ReceiverID from Message table will use this characteristic to identify who will be the sender and who is the receiver to help build the Notification and Message functions of the application.

The room table will cover the whole things including both message and notification table, and make it easier to build the front-end code. Every time a message has been sent to the room, we can identify right away who is the owner, and then the system will send the notification to all other members in the room except the sender of the message.

Also, we can identify which notification we get come from what room, and what message will be sent to what room, all has been covered.

### Allocating function

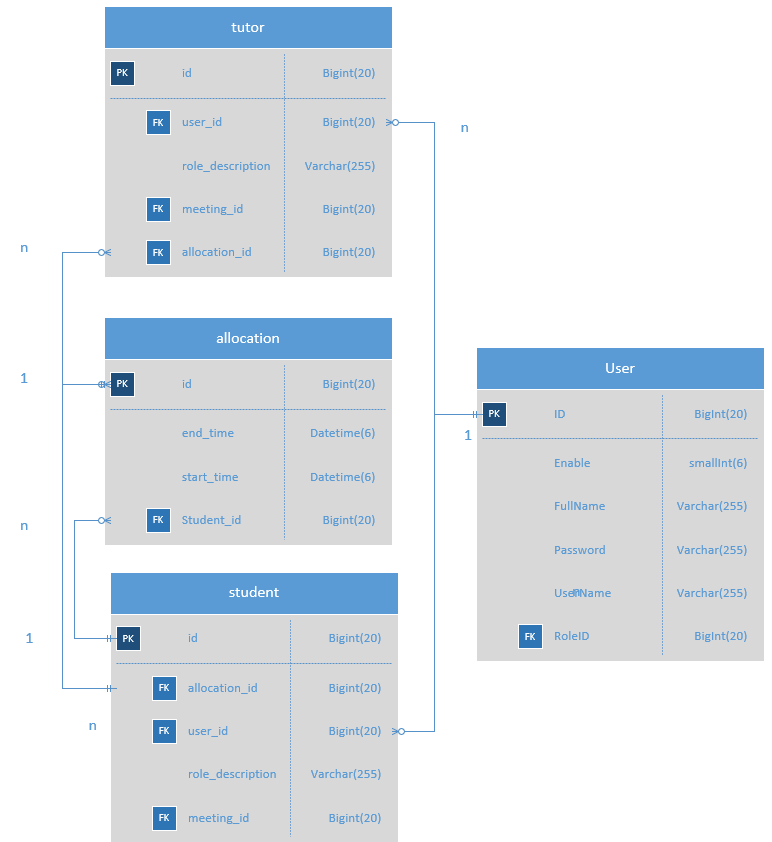


Figure 7: Allocating function

It’s a bit complex but with 4 tables: User, Tutor, Student, and Allocation, that will become easier to build the allocating function. With the Tutor and Student table, the Allocation table will get the right ID for it locating function, whose will be assigned with whom.

Because the relationship in this case somehow is many-many between User and Allocating table, so that we will need to break it down to 1-many and many-1 tables to help present the data better.

### Meeting function

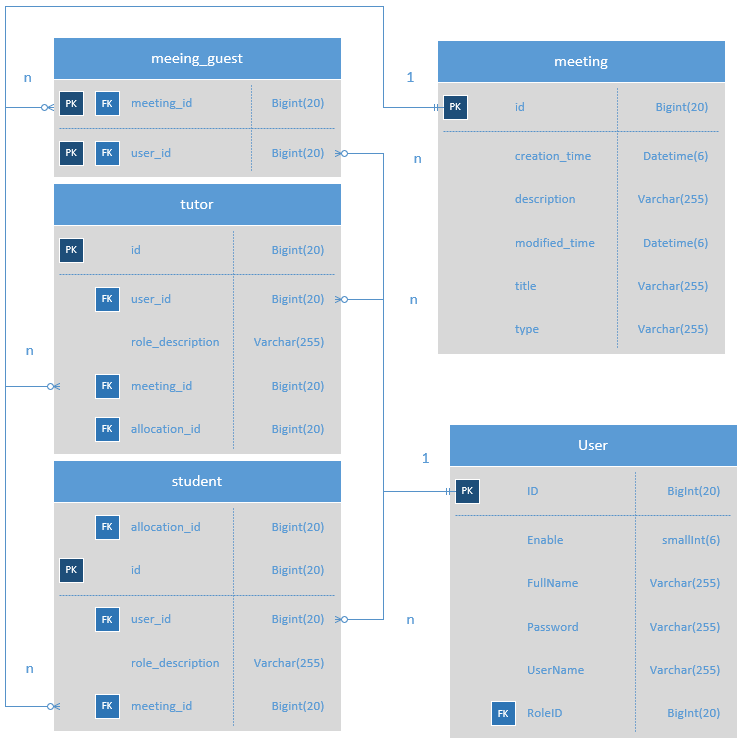


Figure 8: Meeting function

### Document and Comment Function

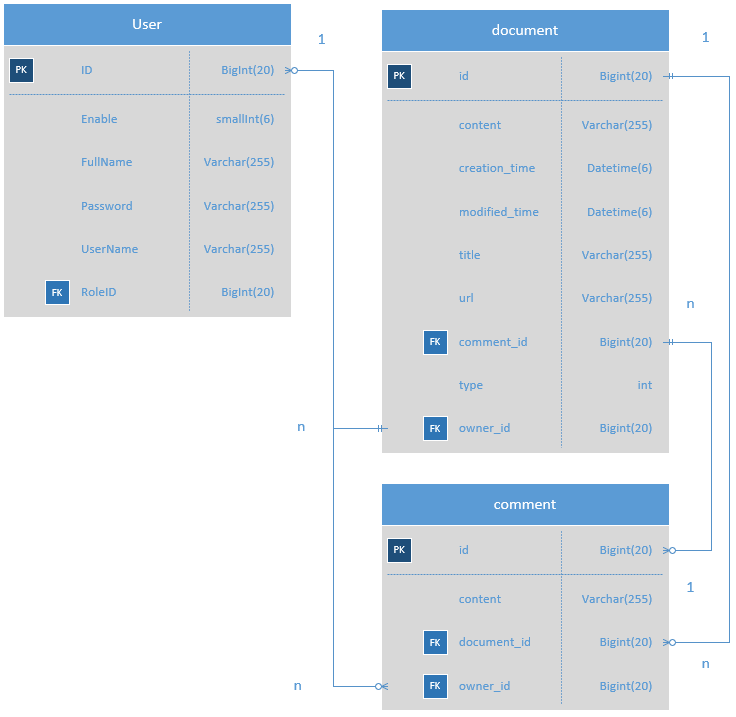


Figure 9: Upload funtion

One of the important functions is upload file. Sometime user will need to upload the file to the internet, and then the other users will come, check the file, and see if there is any problems, and then leave a comment beside this document and expected the change from it’s owner. The OwnerID will be determined through UserID from User table, and also

### Blog function



Figure 10: Blog function

# **SITE DESIGN**

## *What is Responsive Design*

## Responsive Web Development (RWD) is a recent phenomenon in which web design and production systems satisfy both platforms and user experience through screen size and dimension requirements. We will use versatile combination of techniques like versatile map, responsive graphic, and CSS media query for this reason.

## As users switch from a desktop to an iPad or iPhone, the platform changes automatically to match the screen size and scenario for processing. To put it another way, websites require technology that adapts to the system of the customer. This reduces the need for several separate site projects, which minimizes web design time and expense.

## *How do we apply it to the current project:*

### View Port:

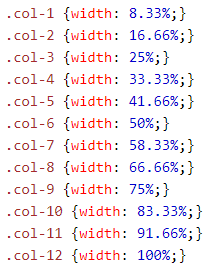
* View port simply is what users see on the website, but we can see that the screen size for different devices is different, so do the View Port. By using meta tag, we can overcome this situation easily



* It simply gives the browser the instruction of how to control the view of the page and set width which will be the set point to change the View Port.
* Few years back, when versatile layout became almost a "luxury" and "premium" for websites, the only aspect that was versatile in column and page layout was that versatility was used in architecture. Email. Email. Photos can quickly split the arrangement and as they are forced forward even structural components may induce arrangement breakage. Versatile architecture isn't really versatile-programmers can produce hundreds of pixels, but they can't change the mainframe configuration to suit a netbook.
* We may now change even further. Photos can be changed dynamically, and we've got workarounds such that the interface rarely fails. The modular display approach is ideal for apps that turn handheld devices such as iPad or iPhone from a horizontal screen to a vertical screen.

### Grid View:

* The page has been divided to many columns, and then base on that, developers can place the elements of the website easier



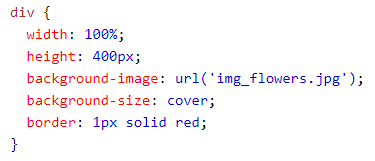
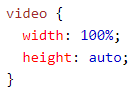
* As we can use css file to determine how big the column is, it is very flexible and easier for us to develop a website using Grid.

### Bootstrap:

* Also, by using bootstrap framework, developers can make the responsive web design faster and easier



Resize elements automatically:

* One of the key elements of the responsive website is the element of the website will be resized base on the device’s screen and so that still make user experience the best feeling
* 
* For the block of code, it is mean that the background image will always be stretch to 100% of the browser size no matter how we change the browsers size.
* In CSS3 we'll use a media query function to construct dynamic layout and responsive materials. The min-width and max-width parameters should do just as we're advising to match the screen and window scale of the user. The min-width property defines a default window and screen width centered on a selection of common models (or similar models) that match the computer to the minimum distance. For every computer that has a screen size below this cap is ignored the stylesheets format. The attribute of max-width does the reverse. Any computer that meets the standard with a toolbar or full screen width does not implement the corresponding stylesheet file.
* The same thing can be applied for the video
* 
* The video on this example will take up all the width of the browser while only take the right amount of height to display the video to the finest quality
* When the picture size is too high, the picture may seem to be of poor quality but we also want it to remain on the site and do not want it to be deleted because it is significant. The image is then split into two groups: one group should be allowed to strip out the unimportant bits while the other group must preserve the picture which will be zoomed in and out.
* Many methods are used to produce versatile photographs that can reach various screen sizes.

## *Interface Design*

And below is the mockup of the website, please keep in mind that it is the mockup, not the final design, so in the end, the real product may look a little bit different

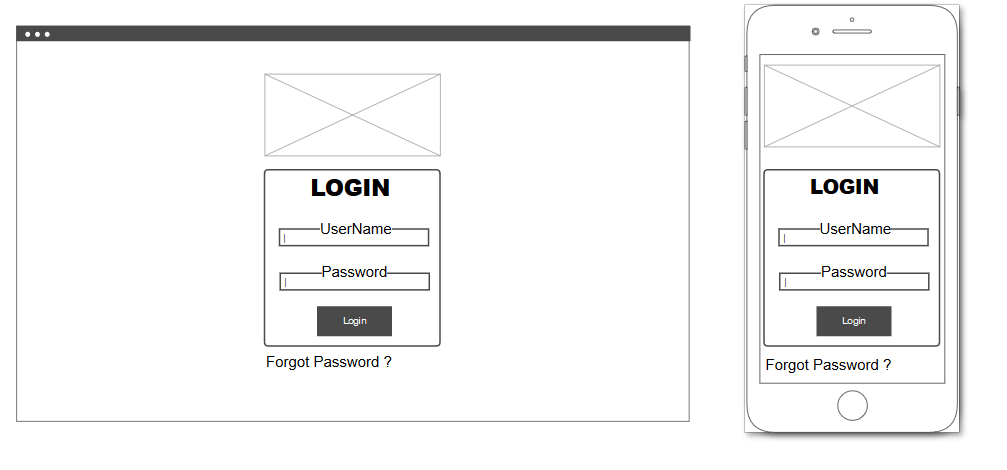


Figure 11: Home Page

Actually there is no homepage, it is the login page instead, when user first access to the website, they will need to use username and password to access to their designated destination, for example, if the Admin login, he will be directed to admin dashboard page, if the Student login, he will be directed to student page

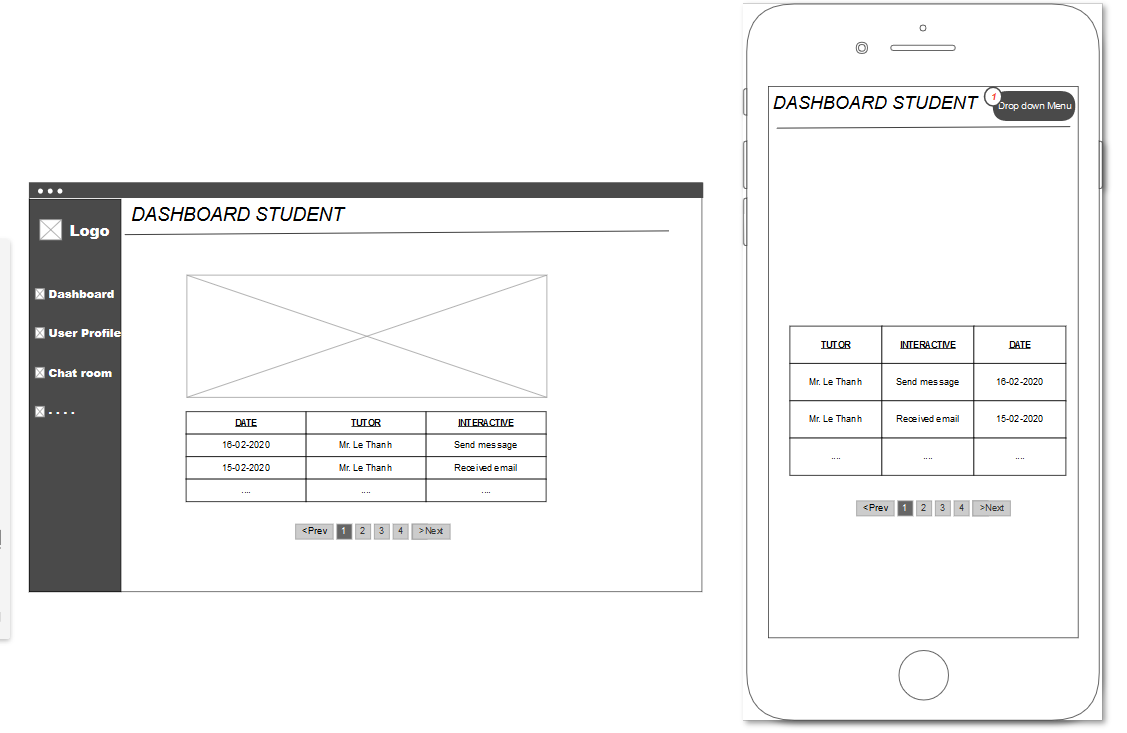


Figure 12: Student Dashboard

This is student dashboard, from this screen, student can access to all of the student’s available functions including checking profile, modified profile (just a few fields are available) or joining the meeting. He can also get notification whenever there is the message or comment from blog/file upload

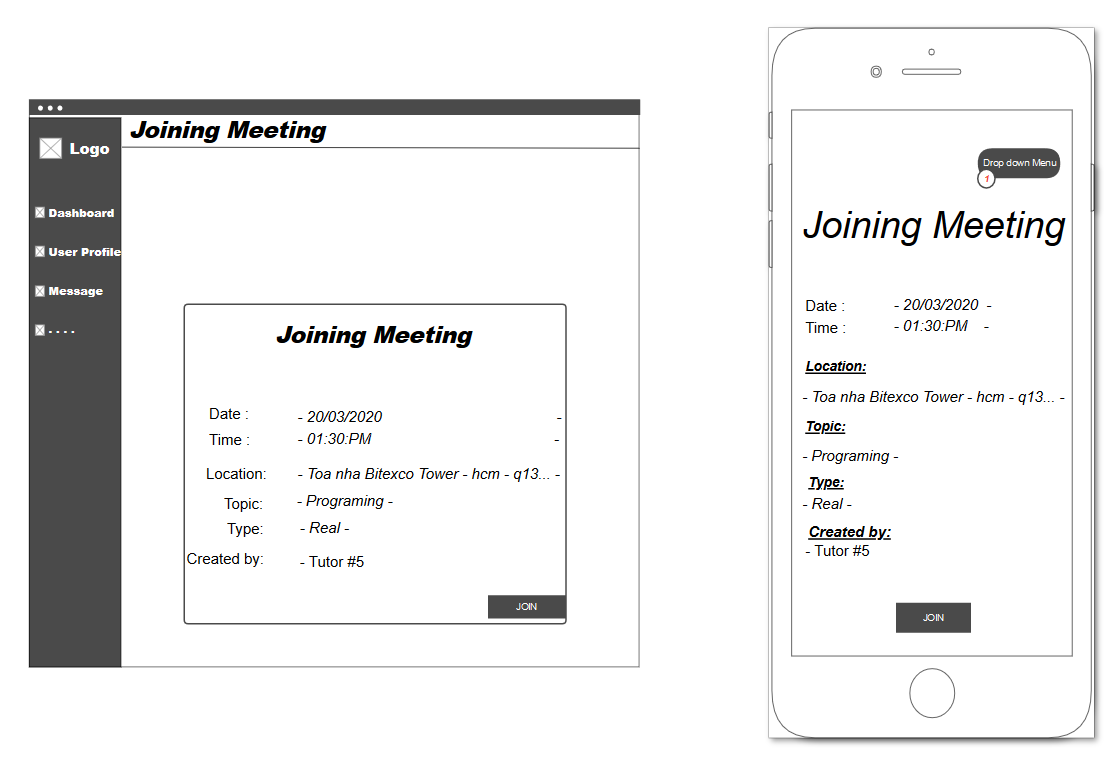


Figure 13: Student Joining Meeting Page

Special function just for student, student can not create meeting but can only joining the available meeting instead. From this screen, user can either choice to access to other page through the menu on the left side, or press “JOIN” button to join a meeting.

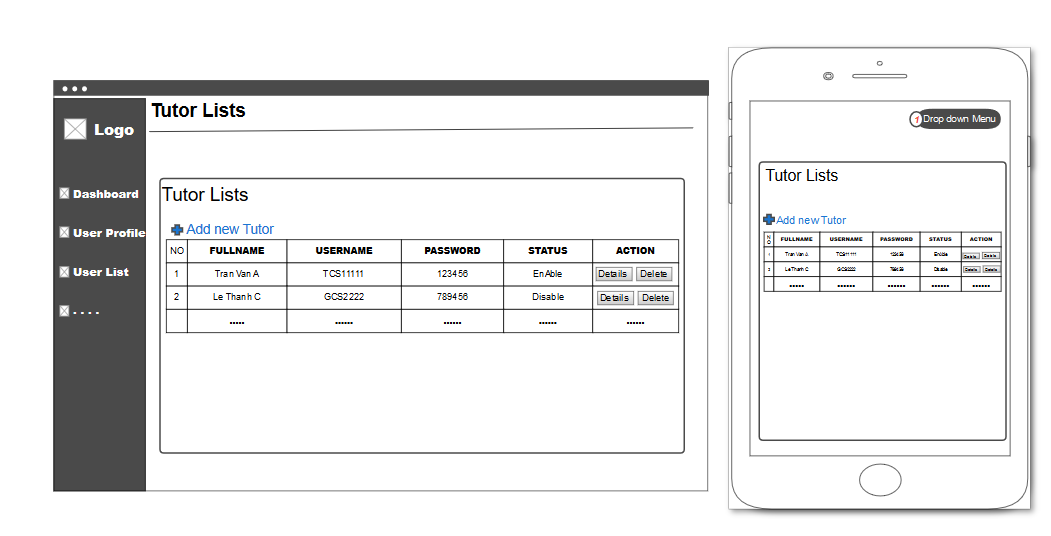


Figure 14: Tutor List Page

This page is tutor list page, from this page, user can access too other available pages on this account. Please note that this page is only available for Staff and/or Admin account. User can see the detail profile of the Tutor by clicking into “DETAILS” button, or delete the account from the system by clicking into “DELETE” button.

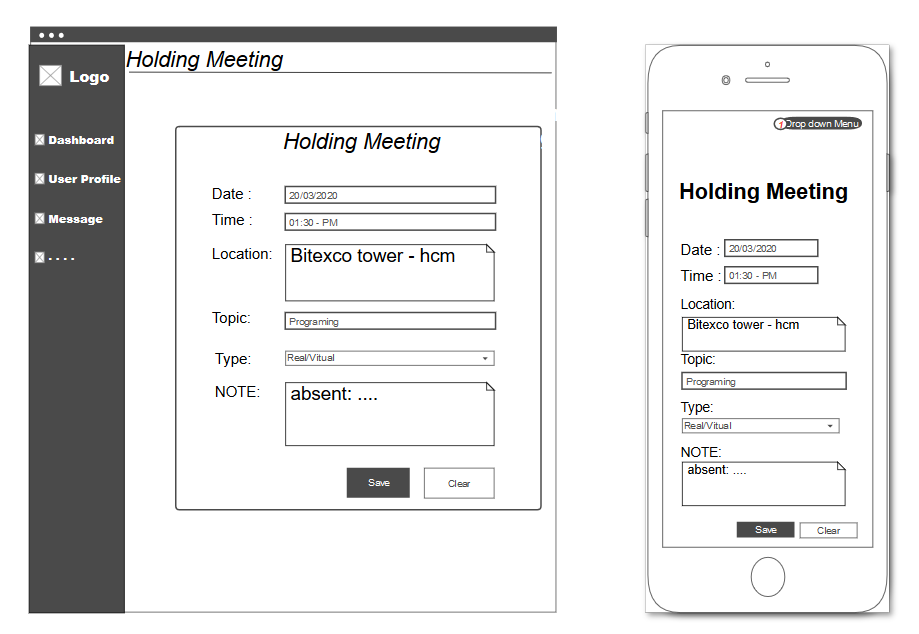


Figure 15: Tutor Holding a Meeting Page

This page is only for Tutor account, Tutor account can create its own meeting and then let the Students join in. User can set the date, time, location, Topic, type (real or virtual) and adding the NOTE also. Press “SAVE” button to save this meeting and press “CLEAR” button to clear all the field to its default.

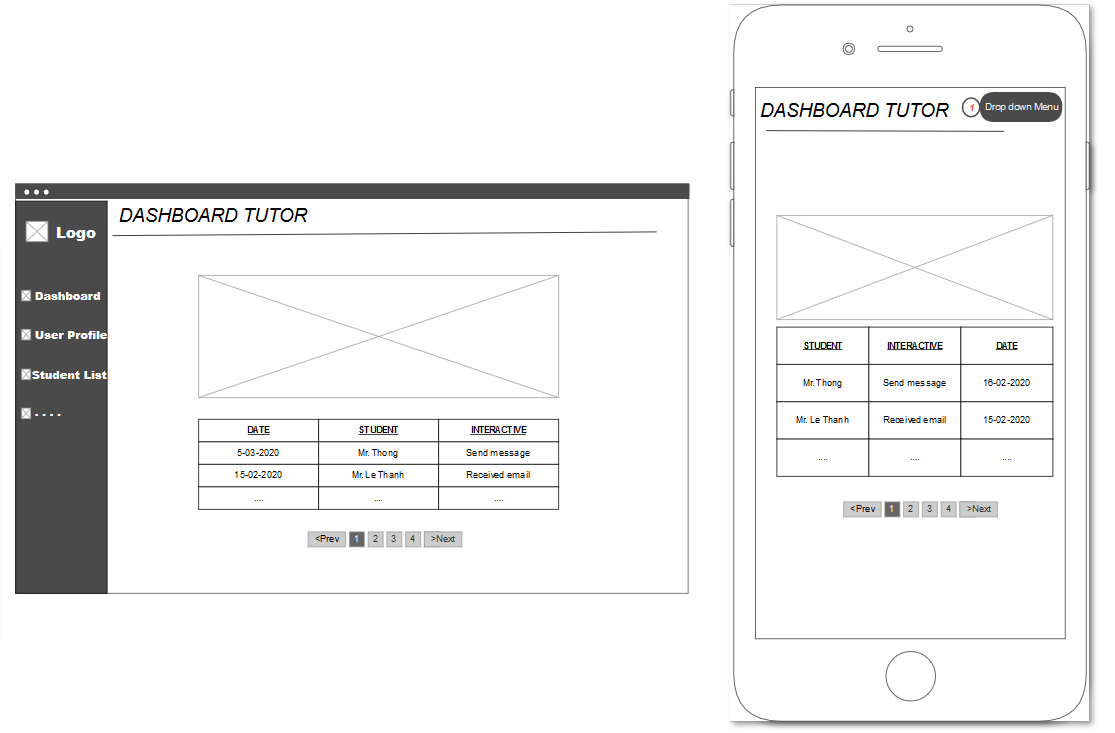


Figure 16: Tutor Dashboard

This Dashboard is only for Tutor. From this dashboard, Tutor can manage what is the message come from who on what day. User can also access to all other available functions of this account through left side menu.

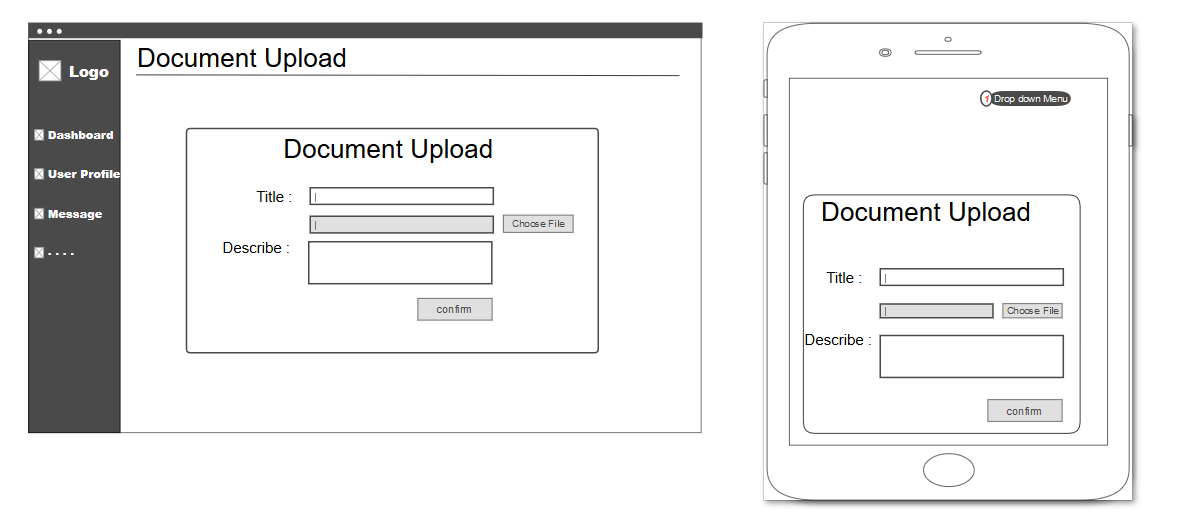


Figure 17: Document Upload Page

This page will be available for both Student and Tutor account. When upload the file, user will need to set the title and the description, and then choose the file on their computer and press “CONFIRM” button to upload it to the server. User can also access to all other available functions of this account through left side menu.

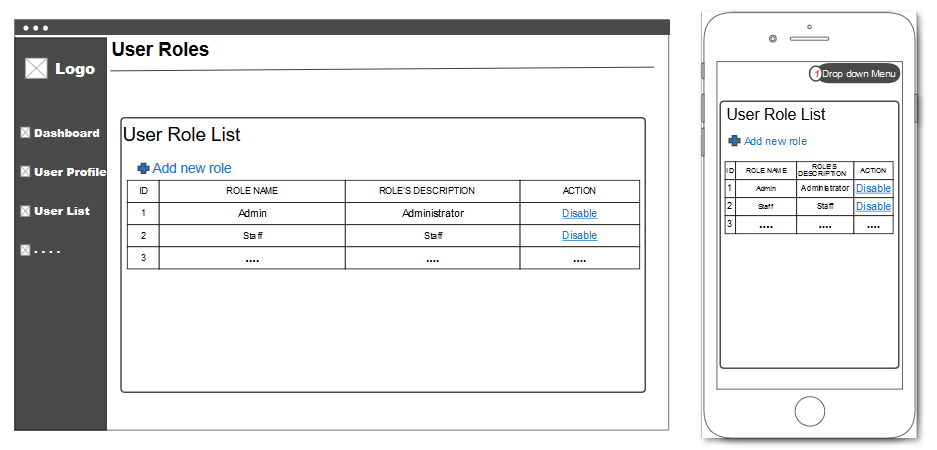


Figure 18: User Role Management Page

This page is only available for Admin account since he can also manager the staff role, too. The only action in this case is “ENABLE” or “DISABLE” the available role of the system and then through them, distributed the access authorize to the others. User can also access to all other available functions of this account through left side menu.

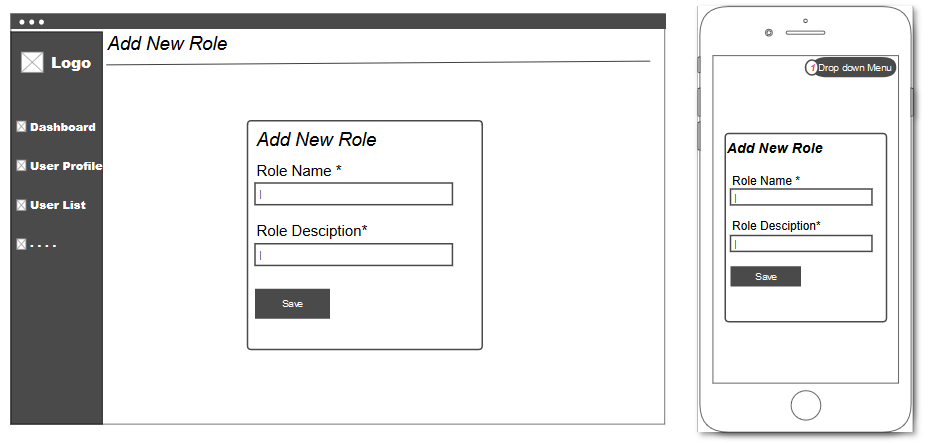


Figure 19: Add New Role Page

Same as above, this page is for Admin account only, to create a new ROLE with the new description to appoint it to other accounts later on. User can also access to all other available functions of this account through left side menu.

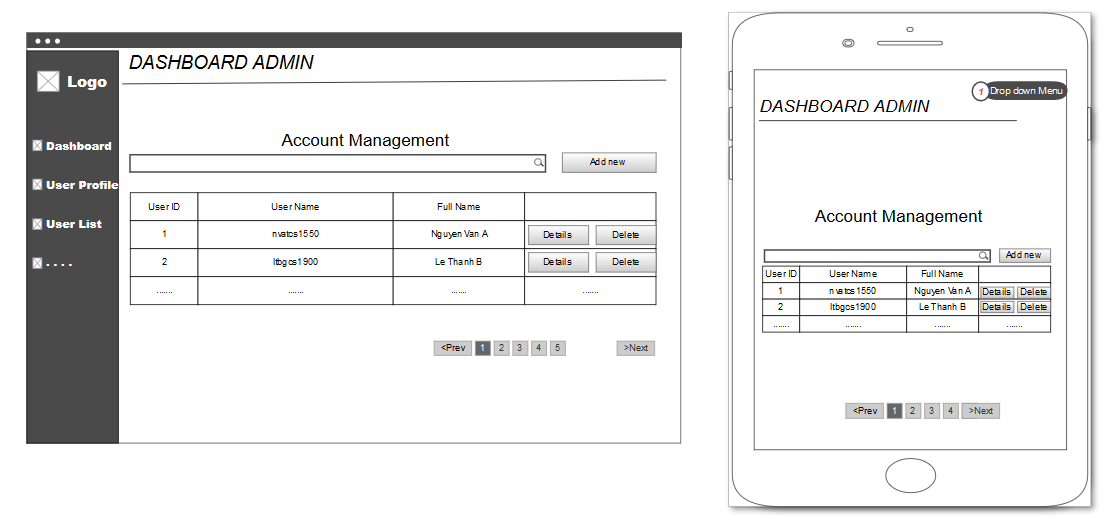


Figure 20: Admin User Management Dashboard

This User management dashboard is for Admin account only. From this dashboard, user can either see the details of the accounts on the list by clicking into “DETAILS” button, or “DELETE” it by delete button, or using “ADD NEW” to create a new user. The Search bar is also available on top to help the manage become easier. User can also access to all other available functions of this account through left side menu.

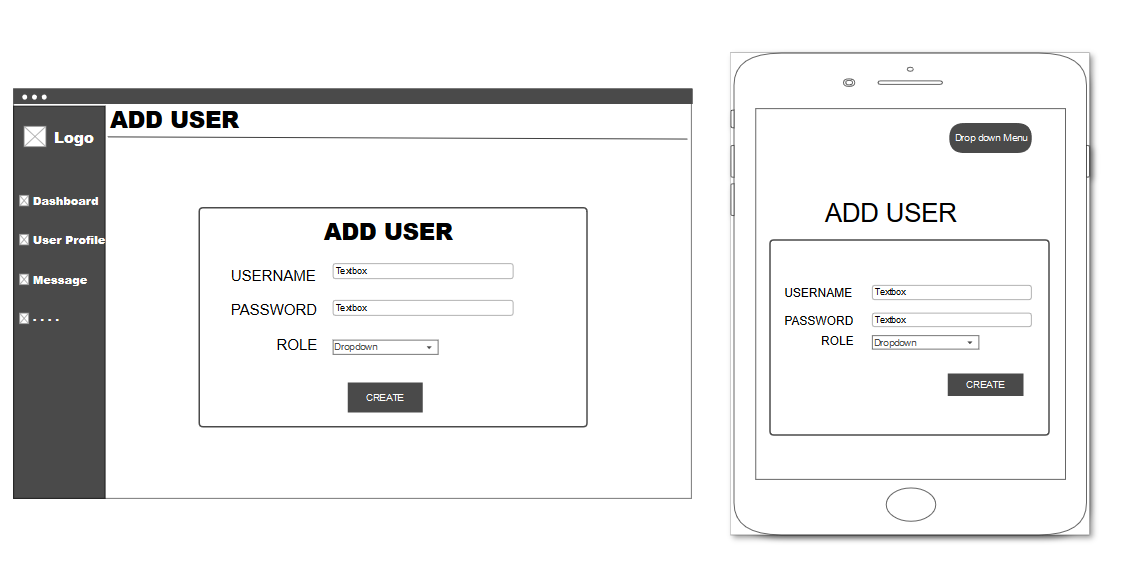


Figure 21: Admin Add User Page

This page is only available for Admin account since user can select the role for the user. Press “CREATE” button to create the new account with that selected username, password, and role. User can also access to all other available functions of this account through left side menu.

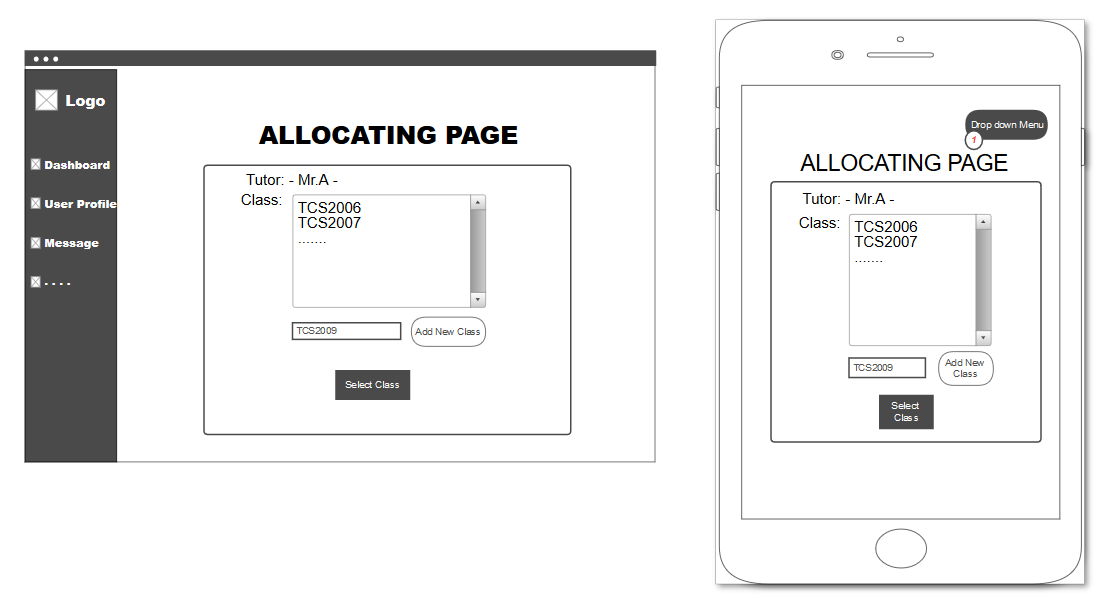


Figure 22: Allocating Class Page

This page is only available for Staff account, this is just one of the group of 3 Allocating Page to assign Students to a Tutor. User can select the class of that Tutor on the list, or simply add new Class by filling the textbox and click into “ADD NEW CLASS” button. After choice the right one, user can press “SELECT CLASS” to go to the next process page. User can also access to all other available functions of this account through left side menu.

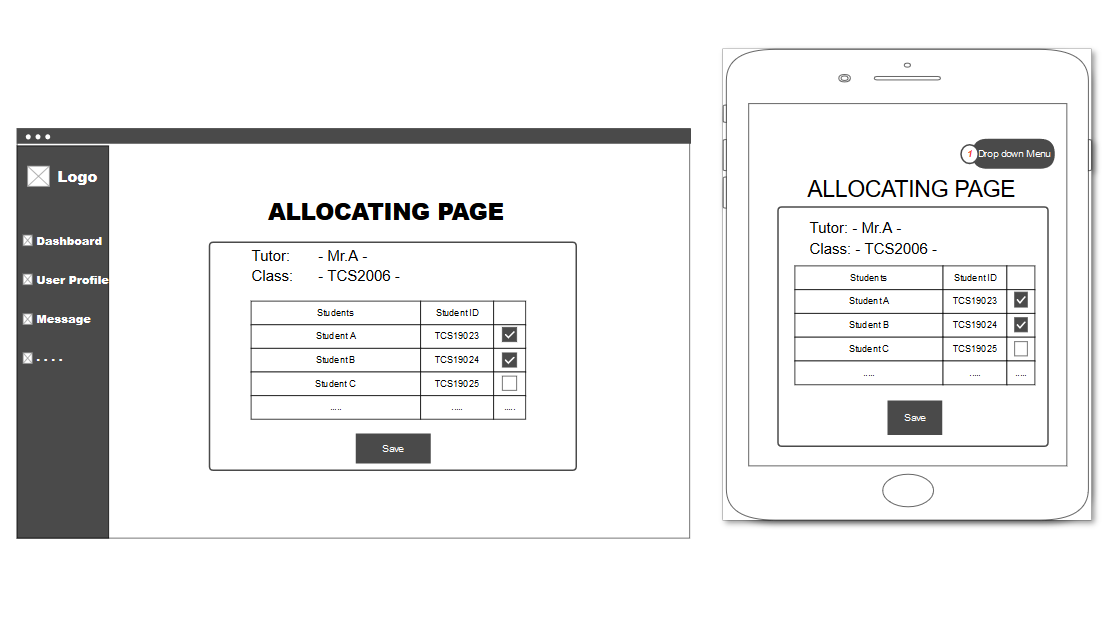


Figure 23: Allocating Student Page

This page is only available for Staff account, this is just one of the group of 3 Allocating Page to assign Students to a Tutor. From this page, user can multi-select the available student for the class and tutor, after press “SAVE” button, the students will be assigned into the right place. User can also access to all other available functions of this account through left side menu.



Figure 24: Allocating Select Tutor Page

This page is only available for Staff account, this is just one of the group of 3 Allocating Page to assign Students to a Tutor. User can select the Tutor name from the list and then press “SELECT TUTOR” to process further. User can also access to all other available functions of this account through left side menu.



Figure 25: Blog Comment Page

This page is used to manage the comments of the old blog, any users with authorize account can access and write the comment here. User can fill the textbox and press enter on their keyboard to post a new comment anytime. User can also access to all other available functions of this account through left side menu.

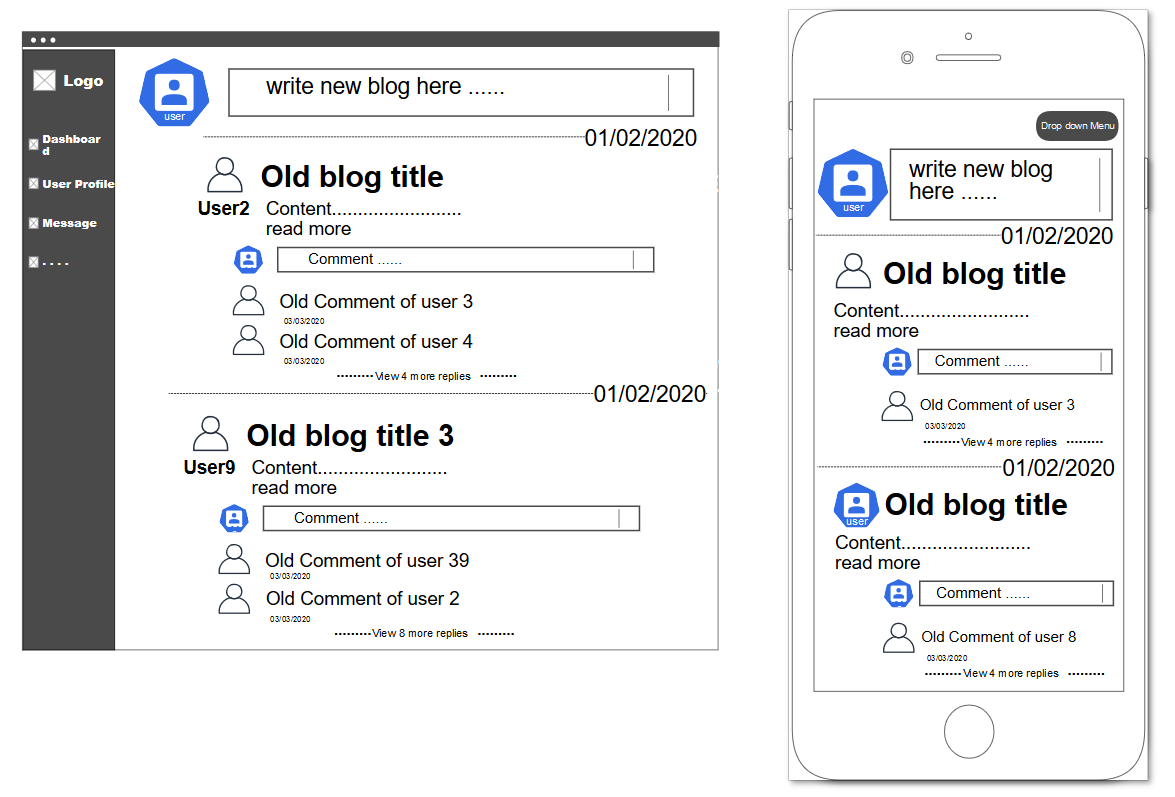


Figure 26: Blog Main Page

This Blog main page can be available for everyone, but only the one with the authorize account can use post a comment or write new blog function. User can also access to all other available functions of this account through left side menu.

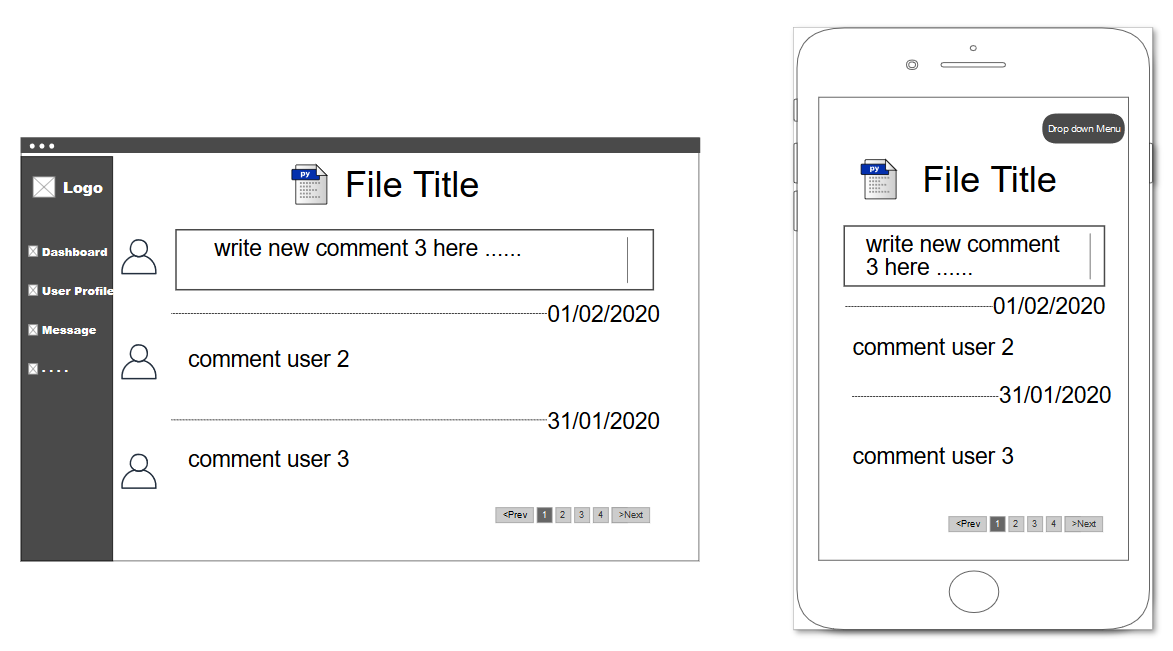


Figure 27: Upload Comment Page

This page is available for authorize accounts, they can post a comment for an uploaded file by filling the textbox and press enter on their keyboard. User can also access to all other available functions of this account through left side menu.

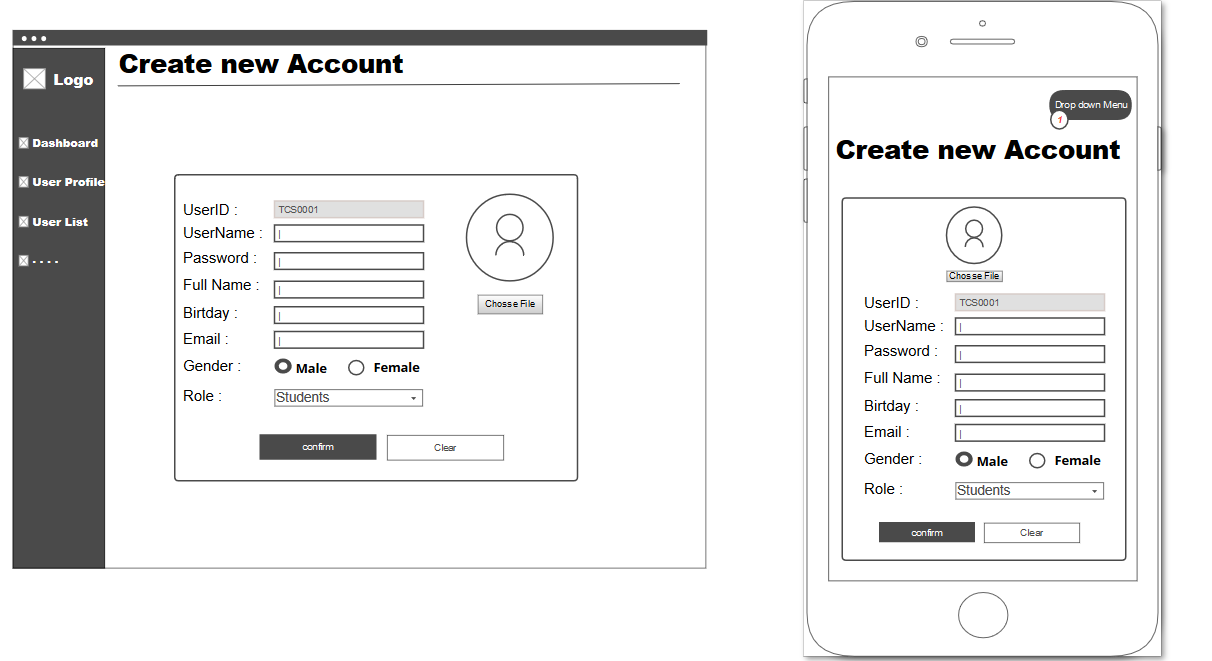


Figure 28: Staff Create new account page

This page is only available for Staff account, its Role choosing option will be limited to only either Student or Tutor. User can fill the textbox with suitable information and press “CONFIRM” button to get the process done. User can also access to all other available functions of this account through left side menu.

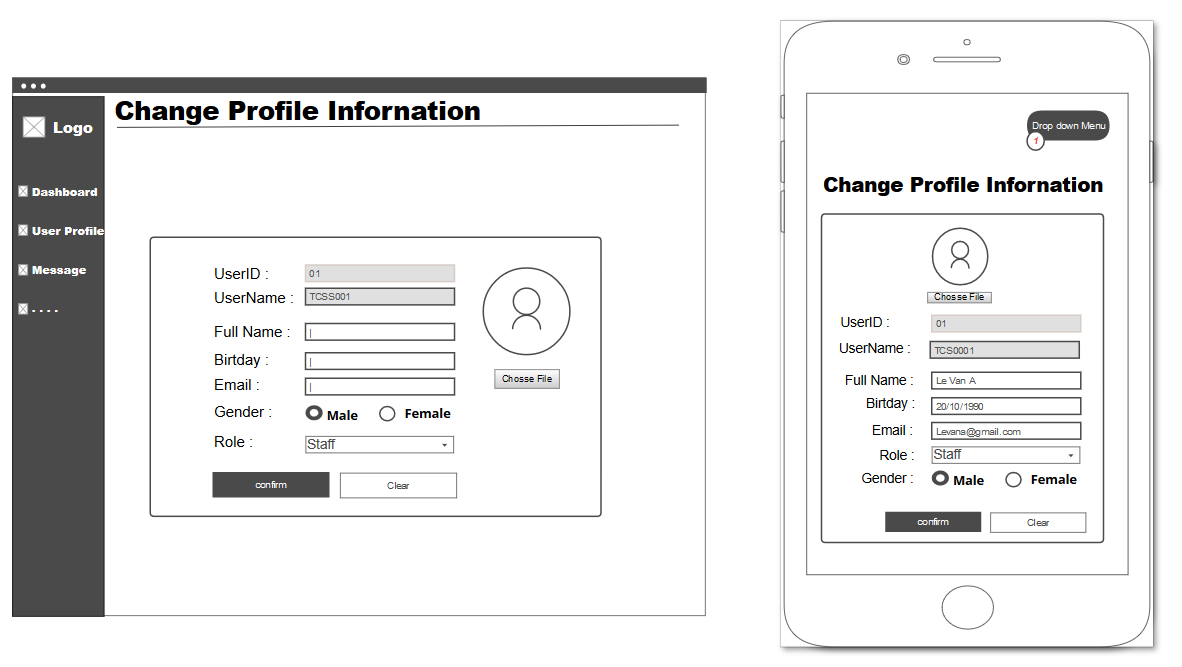


Figure 29: Admin Change user Profile page

This page is only for Admin account, since the Role choosing option will also have Staff role on it, the use of it is similar to the above page. User can also access to all other available functions of this account through left side menu.

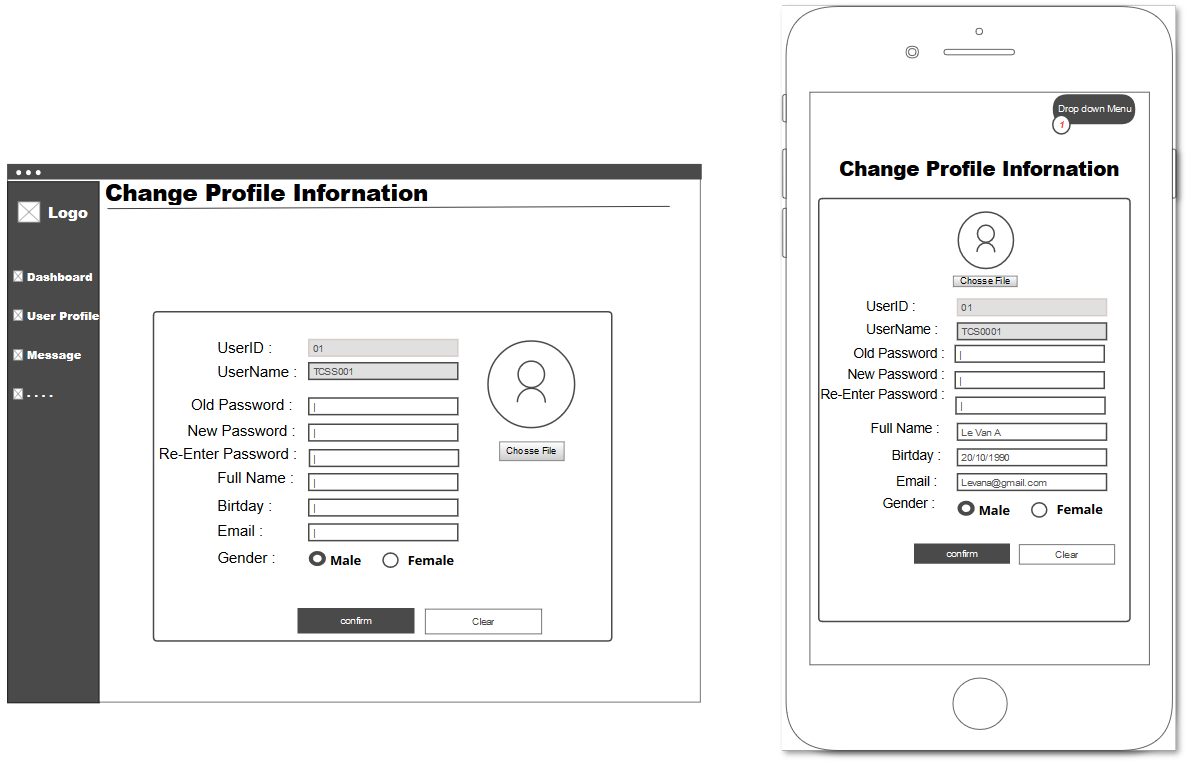


Figure 30: Users Change Profile page

This page is available for Student/Tutor accounts only, since it can not change the role of the account but only the available text field, image, or radio box. Press “CONFIRM” button to confirm the change or “CLEAR” to put all the field into default state. User can also access to all other available functions of this account through left side menu.

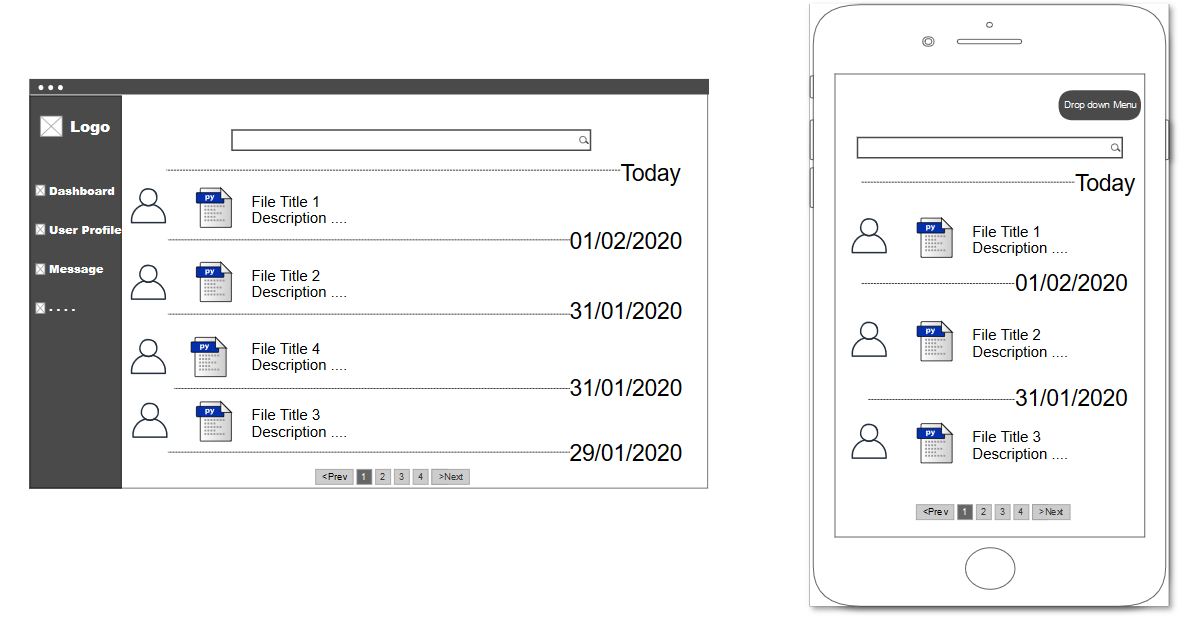


Figure 31: Upload Main page

This page is available for authorize accounts only, to manage the uploaded file on the server, users can see which file is available and then can click on them to go to “Upload comment page”. User can also access to all other available functions of this account through left side menu.

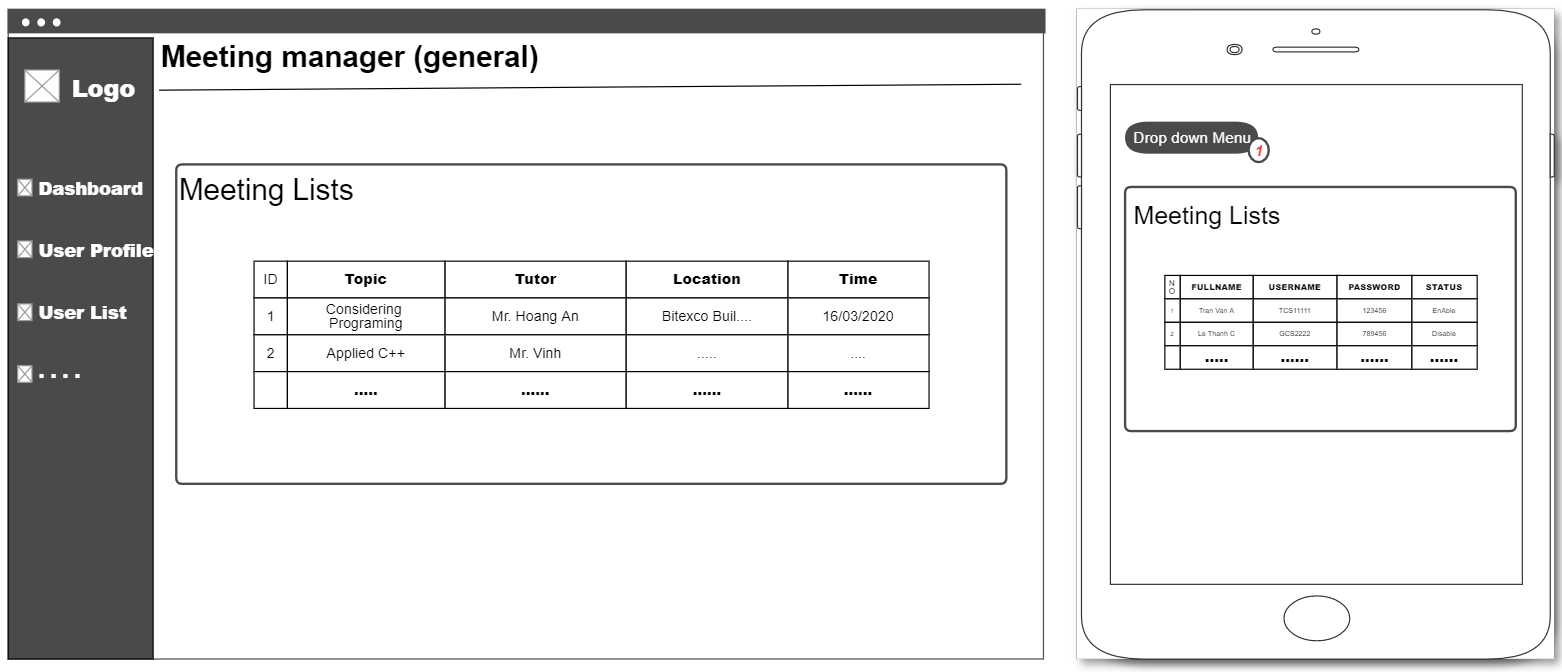


Figure 32: General Meeting Manager

This page is available for every authorize account on the system, used for manage the current available meeting on the system. User can also access to all other available functions of this account through left side menu.



Figure 33: Message Board

This is the Message Board for every authorize users on the system, they can manage which message come from which “Room” and send the reply back without error by filling the text box and using “ENTER” button. User can use Search bar on top to search for either Room name or a Message of that Room. User can also access to all other available functions of this account through left side menu.

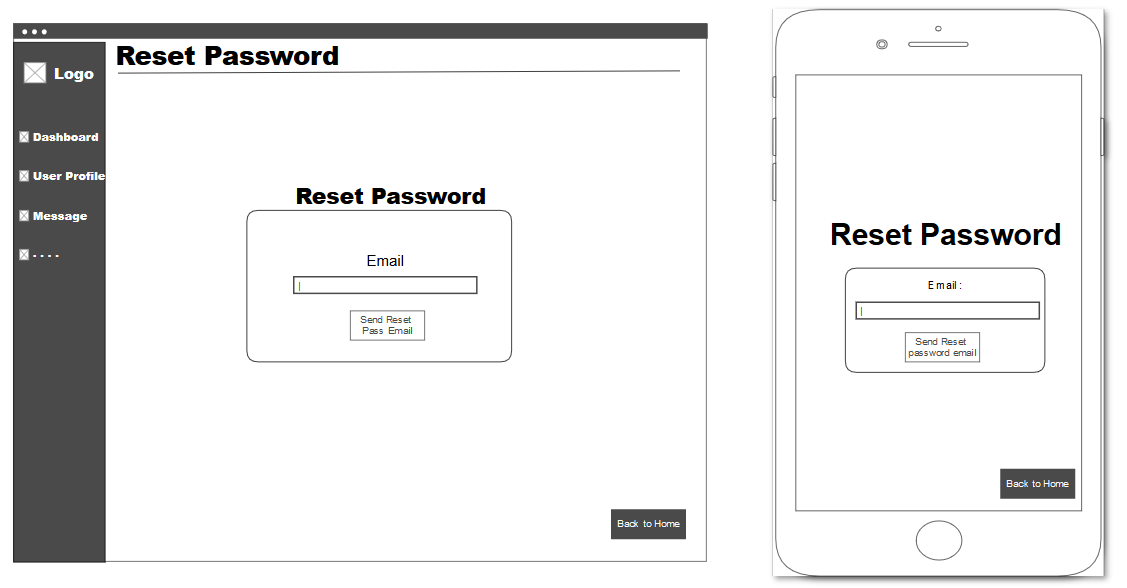


Figure 34: Reset Password Page

This page is available for everyone, they just need to fill in the Email and press the “Send Reset Pass Email” button to get the reset email to reset their password for the account. User can also access to all other available functions of this account through left side menu.



Figure 35: Admin Dashboard

This Dashboard is only available for Admin account. From this page, admin can get a clear view report for the total message of the system per day, average message per person, or see which student have no tutor currently, or non-active student... User can also access to all other available functions of this account through left side menu.

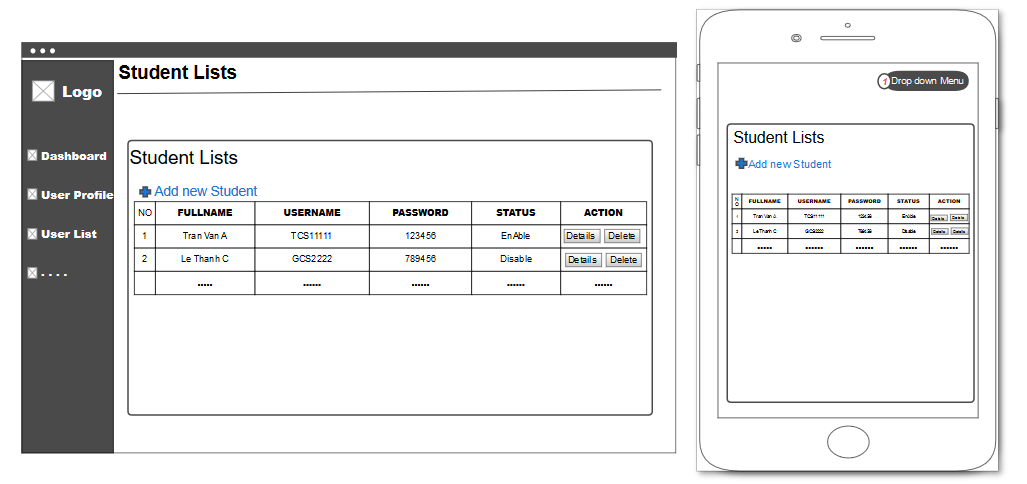


Figure 36: Student List Page

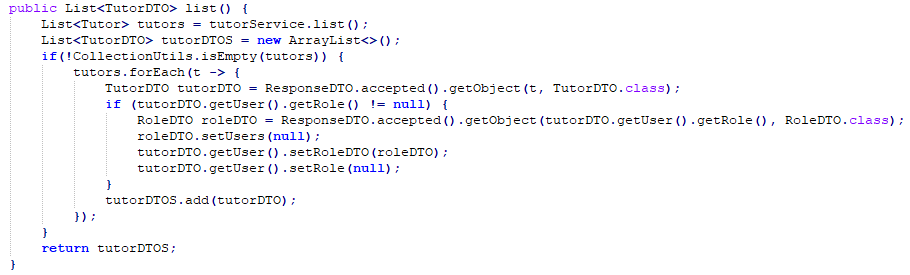
This page is only available for Admin account, since he can also watch the Student dashboard by clicking into “DETAILS” button or using “DELETE” button to delete that student from the system. User can also access to all other available functions of this account through left side menu.

## *How is this work*

* Hình full screen chạy trên máy tính
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao
* Hình bị kéo nhỏ trên máy tính
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao
* Hình trên điện thoại
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao

# **FUNCTIONALITY**

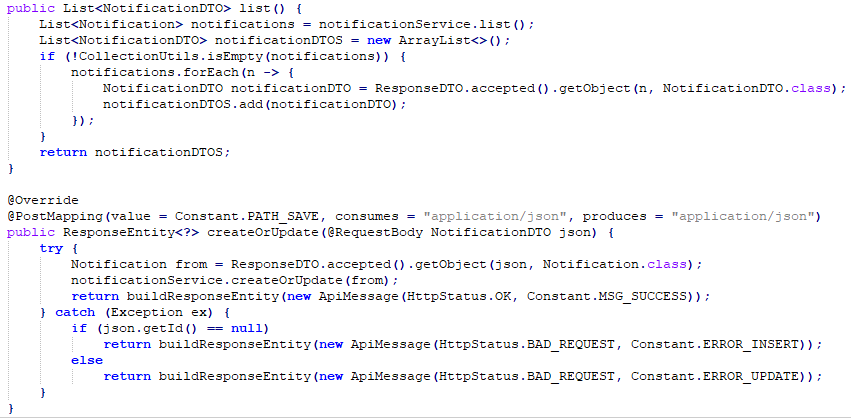
## *The application of Role base system*

* 
* The configuration and maintenance of the safety model is simplified by a function. It is a category of related rights that the individual may obtain. It is difficult to assign or remove rights to users as multiple users are included in a database. So, once you identify roles:
* rights may even be granted or cancelled, and rights granted or cancelled automatically.
* You can build tasks or you can predefine machine functions.

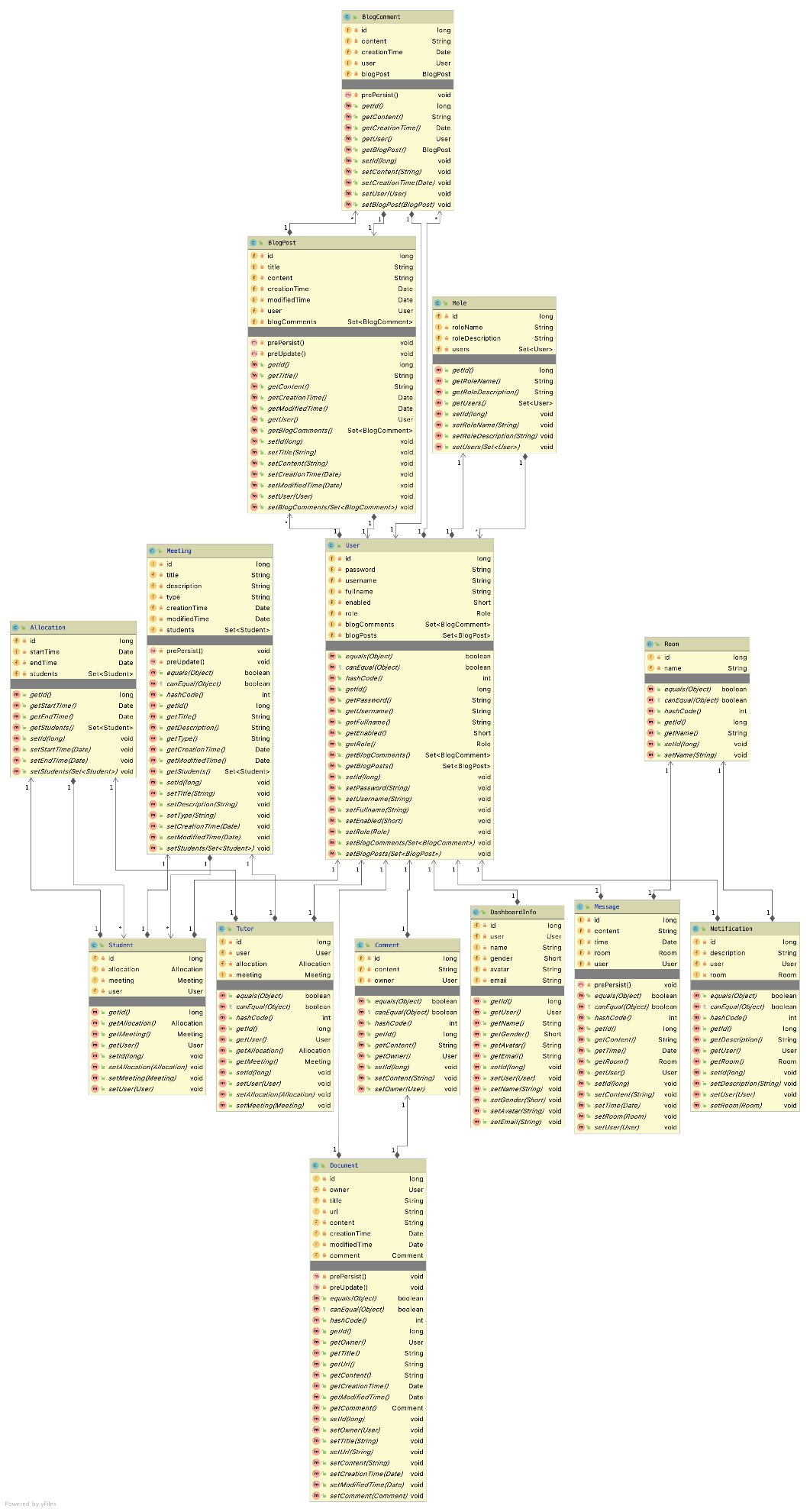
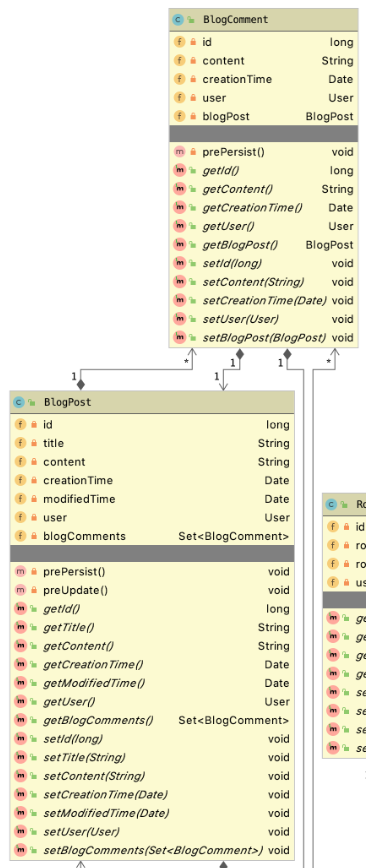
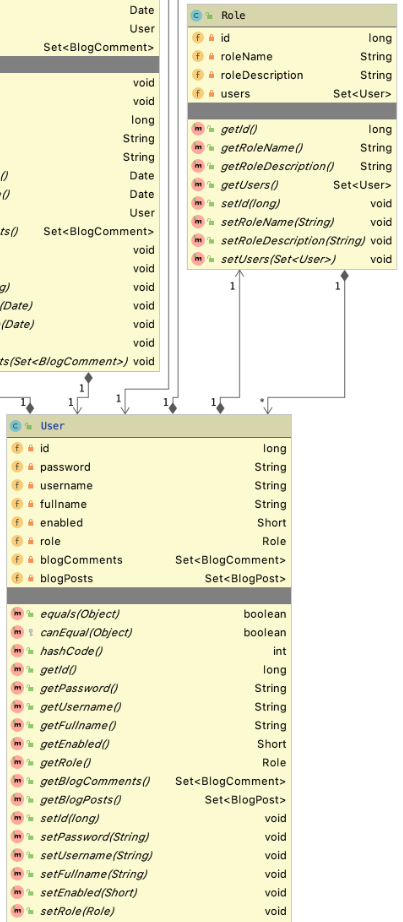
## The Report from the system

* Hình một cái report hoàn chỉnh về Student được xuất ra từ hệ thống
* Giải thích vài trường quan trọng trong cái report
* Hình một cái report hoàn chỉnh về Student được xuất ra từ hệ thống (Exception report)
* Giải thích vài trường quan trọng trong cái report, highlight tại sao report này lại khác report trên

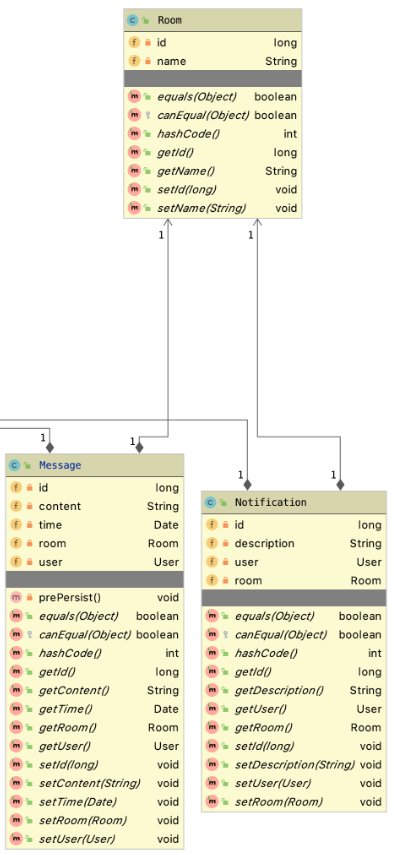
## *The Message from the system*

* Giải thích lại yêu cầu của đề bài về Message
* Hình một đoạn code minh họa về gửi notification khi có message
* 
* Giải thích đoạn code đó
* Hình về một cái notification khi có message trên trang web
* Giải thích, đánh giá cái notification đó có ý nghĩa thế nào với tutor/student

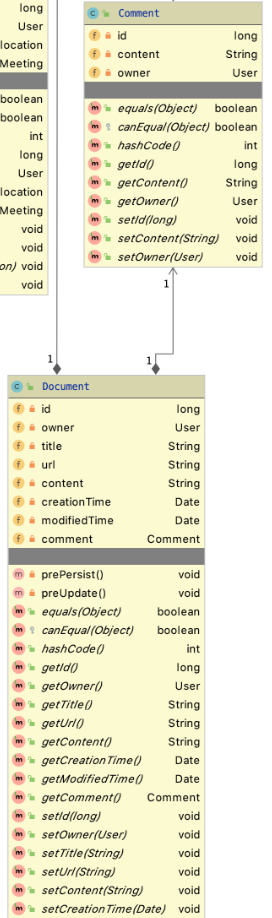
## *UML diagram*

* The Diagram class is a type of static structure diagram that explains the system structure by displaying the classes of the system, its property, its actions and the relationships between artifacts. The unit diagram is the primary object-oriented layout building block.
* 
* Because the image is too big, so I should take some main functions of the application out and explain it in detail for better understanding and viewing
* 
* The first thing is Blog function, as the stakeholders ask, the User must have an ability to post/view/comment a blog, so the table BlogPost and BlogComment will be linked together through BlogComments ID. When user post a Blog, of course there would be a record about who posts, which is the post time, what is the title and how is the content. After that the blog will be appeared on the main blog page for everyone to view, and then anyone with authorized accessibilities can leave the comment on this blog, the comment will also be recorded with some key fields such as the content, time, who post this comment, post on what blog, and the ID. Those two tables be linked together will make it easier for developers to access the data and write the code.
* 

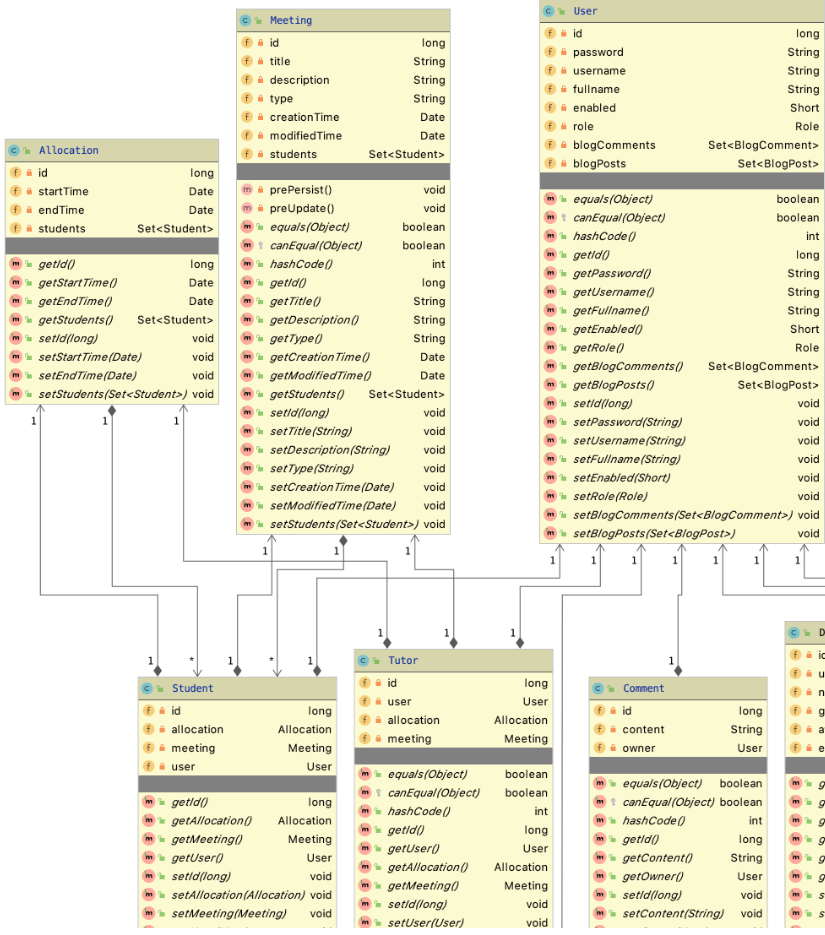
The Role of the user is also an important function, although this function can not be seen from the surface, but it contributed a lot to all of other functions on the program. The stake holder already asked to divide the authorities to many different levels to help in the management, this role will help us do that. By using Role on distributed level of authorities, we can easily control how the workflow works for each user by categories, and then distributed data for it better.



Message is another function of the application. This function help the users have a little chat with other, may the room is the group of two or more people, the message that has been sent to the room will be carry out the record of the user with them, like time post, who post, on what room, and how is the content. And also, we have Notification function that notify user each and every time there is something new inside the room.

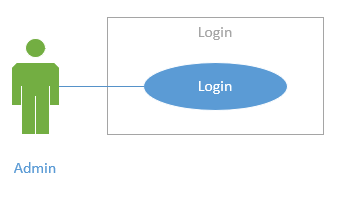


The Upload Document Function is just something similar to Blog function, the only different is that instead of posting a blog, now user will post/upload the file to the server, and everyone can leave the comment on it. Those table be designed so that it can record who post this, on which time, the url, the title, and also the comment.

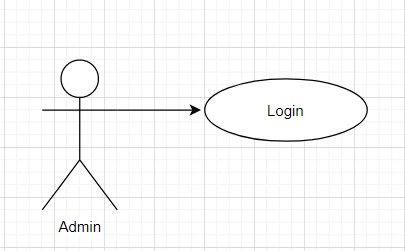


Another function is Meeting and Allocation functions. Those functions will need a sub-table such as Tutor and Student to divide the group of user in to separate authorities, because some user such as Tutor have greater responsibilities and also power to create the Meeting/Allocation room for other Students to join in.

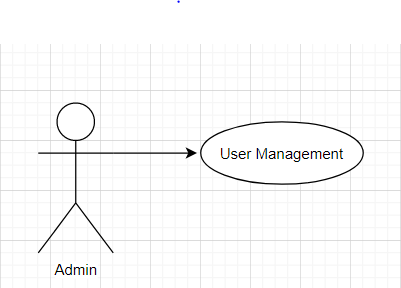
# USE CASE diagram

* 

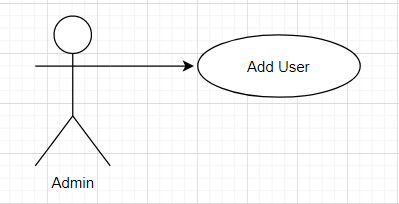
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Login |
| **Actors** | Admin |
| **Description** | Login into the specific manager page |
| **Goal** | Reach the manager page of Admin DashBoard |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User press Enter or Login button |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Admin |
| **Normal flow** | [User logged in]  1. User reach main page  2. User input username/password  3. User press Enter or Login button  4. User reach Success page  5. User has been redirected to Manager page of Admin |
| **Alternative flow** | [Wrong Username/Password]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special character on Username/Password]  Show the error page  Return the main page again |



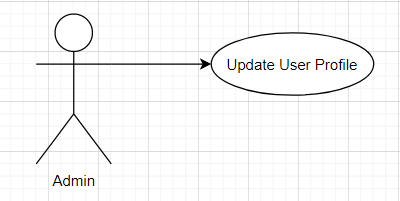
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Admin Login Page |
| **Actors** | Admin |
| **Description** | Login into the specific admin manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User press “Enter” or “Login” button |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Admin |
| **Normal flow** | [User logged in]  1. User reach main page  2. User input username/password  3. User press Enter or Login button  4. User reach Success page  5. User has been redirected to Manager page of Admin |
| **Alternative flow** | [Wrong Username/Password]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return the main page again |



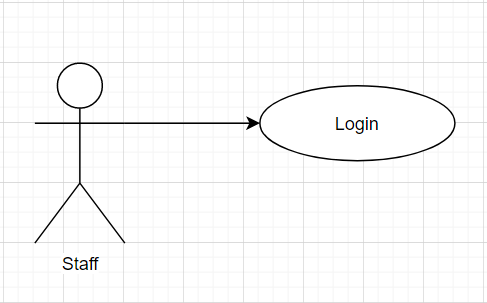
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | User Management DashBoard |
| **Actors** | Admin |
| **Description** | Manage Users |
| **Goal** | Reach the manager page of Admin DashBoard & view Users Statistics |
| **Pre-condition** | Display Login Successful Page |
| **Trigger** | Automatically transfer to DashBoard page after 5 seconds of login process |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Admin |
| **Normal flow** | [Admin logged in]  1. User reach Success page  2. User has been redirected to User Management DashBoard of Admin |
| **Alternative flow** | [Admin redirected to wrong page]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Website non-responsive / hacked page]  Show the error page  Return the main page again |



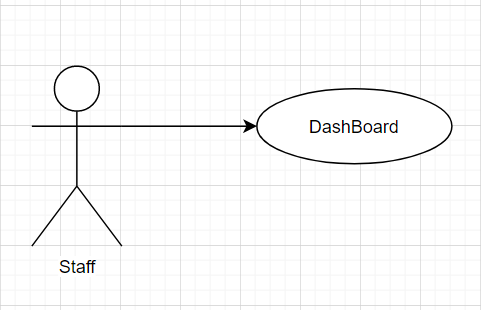
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Add User Page |
| **Actors** | Admin |
| **Description** | Add new users into the system |
| **Goal** | Perform add user function successful |
| **Pre-condition** | Display Add User Page |
| **Trigger** | Press “Add” button |
| **Post-condition** | Show message & new user data has been created in system database |
| **Normal flow** | 1. Admin type in new user’s information  2. Choose new user’s role  3. Press “Add” button  4. Successful page display  5. Redirected back to DashBoard page |
| **Alternative flow** | [Admin inputted existing user data in database]  Show the “Data already existed” message and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return the main page again |



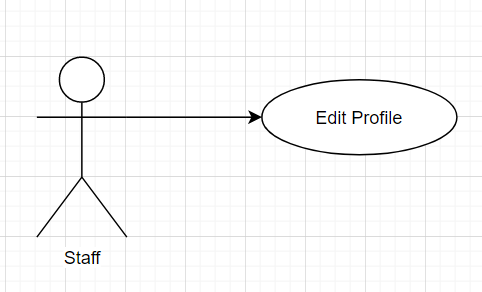
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Update User Profile Page |
| **Actors** | Admin |
| **Description** | Update users information |
| **Goal** | Perform update users function successful |
| **Pre-condition** | Display Update User Profile Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show message & update user data in system database |
| **Normal flow** | 1. Admin change user’s information  2. Press “Confirm” button  3. Successful page display  4. Redirected back to Update User Profile page |
| **Alternative flow** | [Cannot access User Profile Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return the Update User Profile page again |



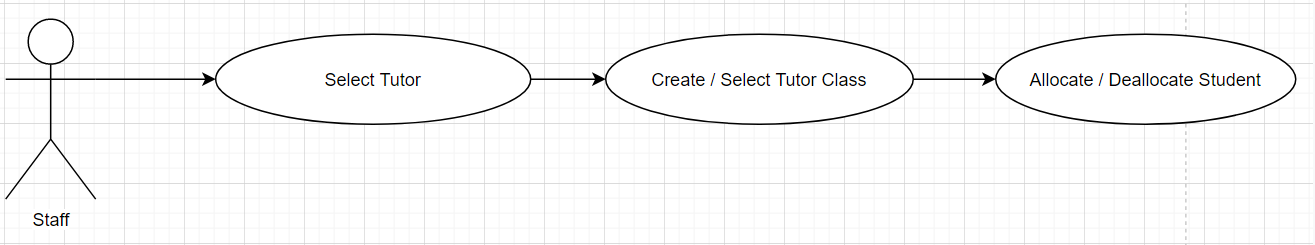
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Staff Login Page |
| **Actors** | Staff |
| **Description** | Login into the specific staff manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User presses “Enter” or “Login” button |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Staff |
| **Normal flow** | [User logged in]  1. User reach main page  2. User input username/password  3. User press “Enter” or “Login” button  4. User reach Success page  5. User has been redirected to Manager page of Staff |
| **Alternative flow** | [Wrong Username / Password]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return the main page again |



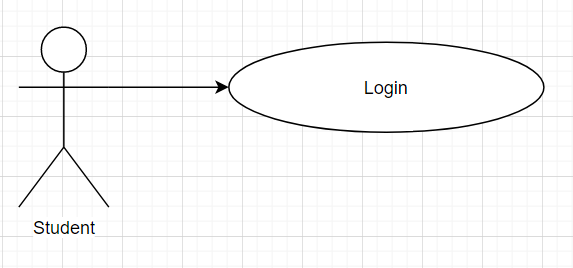
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Staff DashBoard |
| **Actors** | Staff |
| **Description** | View Statistics Report on DashBoard |
| **Goal** | Reach the manager page of Staff DashBoard & view system users statistics |
| **Pre-condition** | Display Login Successful Page |
| **Trigger** | Automatically transfer to DashBoard page after 5 seconds of login process |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Staff |
| **Normal flow** | [Staff logged in]  1. User reach Success page  2. User has been redirected to User Management DashBoard page of Staff |
| **Alternative flow** | [Staff redirected to wrong page]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Website non-responsive / hacked page]  Show the error page  Return the main page again |



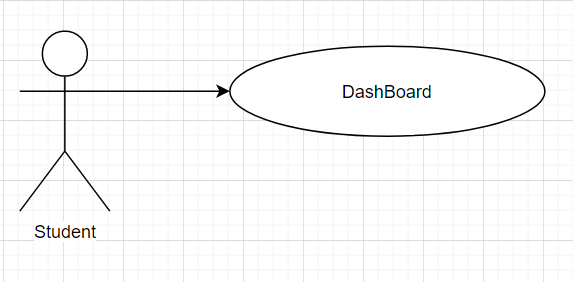
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Create New User Account Page |
| **Actors** | Staff |
| **Description** | Input new user’s information to create new user account |
| **Goal** | New user account creation |
| **Pre-condition** | Successful Display Staff Create New User Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message & update new user data in system database |
| **Normal flow** | 1. Staff input new user’s information  2. Press “Confirm” button  3. Successful message display  4. Redirected back to Create New User Account Page |
| **Alternative flow** | [Cannot access Create New User Account Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return to the Create New User Account Page again |



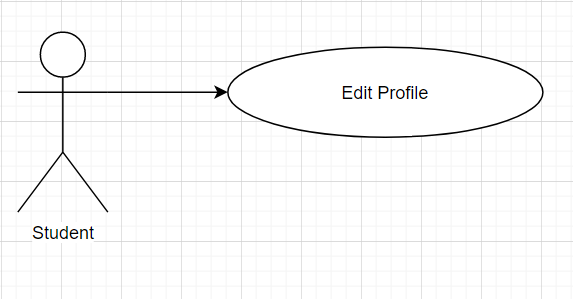
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student & Tutor Manager Page |
| **Actors** | Staff |
| **Description** | Allocate / Deallocate student & tutor |
| **Goal** | Perform Allocation / Deallocation action successful |
| **Pre-condition** | Display Student & Tutor Manager Page |
| **Trigger** | Select Tutor for a class |
| **Post-condition** | Show message, update student & tutor class in system database |
| **Normal flow** | 1. Create a new class  2. Select tutor for that class  3. Drag & add student for class  4. Successful page display |
| **Alternative flow** | [Staff inputted existing class data in system database]  Show the “Class already existed” message and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Website non-responsive / hacked page]  Show the error page  Return the main page again |



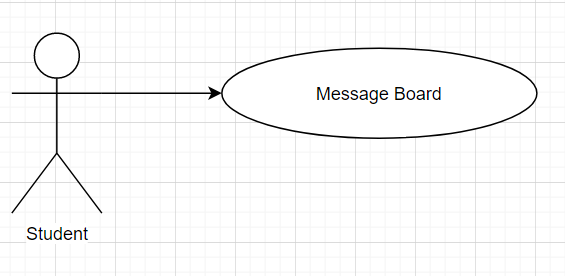
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Login Page |
| **Actors** | Student |
| **Description** | Login into the specific student manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User presses “Enter” or “Login” button |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Student |
| **Normal flow** | [User logged in]  1. User reach main page  2. User input username/password  3. User press “Enter” or “Login” button  4. User reach Success page  5. User has been redirected to Manager page of Student |
| **Alternative flow** | [Wrong Username / Password]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return the main page again |



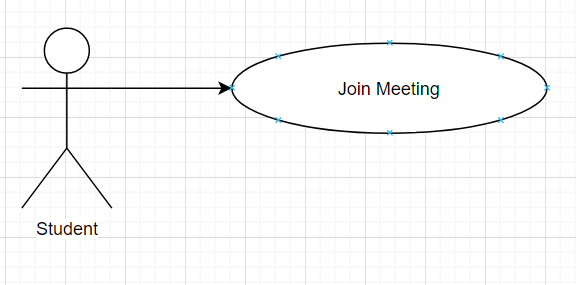
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student DashBoard |
| **Actors** | Student |
| **Description** | View & read student information on DashBoard |
| **Goal** | Reach the manager page of Student DashBoard & view student information |
| **Pre-condition** | Display Student DashBoard page |
| **Trigger** | Automatically transfer to DashBoard page after 5 seconds of login process |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Student |
| **Normal flow** | [Student logged in]  1. User reach Success page  2. User has been redirected to Student DashBoard page |
| **Alternative flow** | [Student redirected to wrong page]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Website non-responsive / hacked page]  Show the error page  Return the main page again |



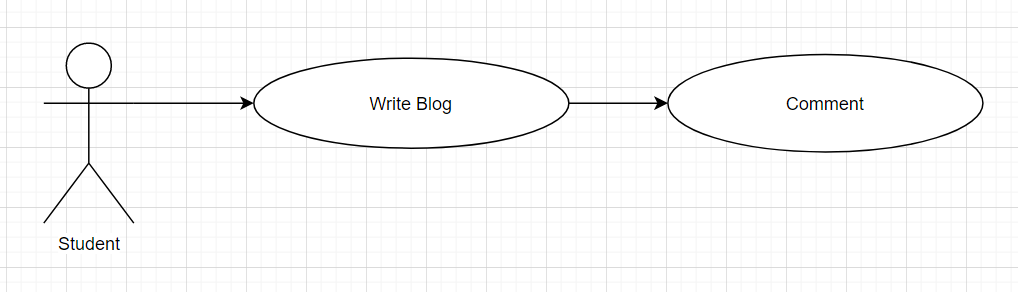
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Edit Profile Information Page |
| **Actors** | Student |
| **Description** | Input student data to update new information with the database |
| **Goal** | Update new user information into system database |
| **Pre-condition** | Successful Display Student Edit Profile Information Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message & update new user data in system database |
| **Normal flow** | 1. Student input new user’s information  2. Press “Confirm” button  3. Successful message display  4. Redirected back to Student Edit Profile Information Page |
| **Alternative flow** | [Cannot access Student Edit Profile Information Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting Username & Password]  Show the error page  Return to the Student Edit Profile Information Page again |



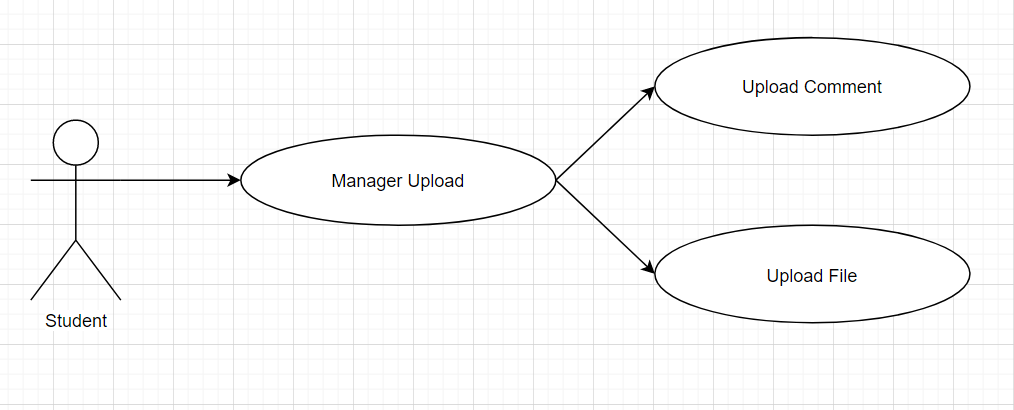
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Message Board Page |
| **Actors** | Student |
| **Description** | Deliver student’s message across the system (classroom, group, private conversation) |
| **Goal** | Successfully send student’s message |
| **Pre-condition** | Successful Display Student Message Board Page |
| **Trigger** | Press “Enter” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Student input message content  2. Press “Enter” button  3. Successful message display  4. Redirected back to / Stay on Student Message Board Page |
| **Alternative flow** | [Failed to send student’s message from Student Message Board Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when inputting the message / Message exceed characters limit – 250 words]  Show the error page  Return to the Student Message Board Page |



|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Join Meeting Page |
| **Actors** | Student |
| **Description** | Perform join meeting function to allocate student to a group meeting |
| **Goal** | Student successfully joins a meeting |
| **Pre-condition** | Successful Display Student Join Meeting Page |
| **Trigger** | Press “Join” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Student selects a specific meeting that they want to join  2. Click on meeting icon on the system calendar  3. Press “Join” button  4. “Joined Successful” message display |
| **Alternative flow** | [Failed to join message from Student Join Meeting Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Meeting does not exist / Meeting was terminated by the Tutor]  Show the error page  Return to the Student Join Meeting Page |



|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Write Blog Page |
| **Actors** | Student |
| **Description** | Create new blog post from student |
| **Goal** | Successfully create new blog post |
| **Pre-condition** | Successful Display Write Blog Page |
| **Trigger** | Press “Post” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Student types in blog post content  2. Press “Post” button  3. “Posted Successful” message display |
| **Alternative flow** | [Failed to create blog post from Student Write Blog Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when writing the blog post, unsupported content]  Show the error page  Return to the Student Write Blog Page |



|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Upload Page |
| **Actors** | Student |
| **Description** | Upload student files and comments |
| **Goal** | Successfully files and comments, update new data in system database |
| **Pre-condition** | Successful Display Student Upload Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message |
| **Normal flow** | [Upload document flow]  1. Student click on “Choose File” button  2. Student select appropriate file to upload and confirm  3. Input Document title (Must have)  4. Input Document description (optional) |
| **Alternative flow** | [Failed to upload file from Student Upload Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Inappropriate / Unsupported file type]  Show the error page  Return to the Student Upload Page |

(I will add Tutor’s pages later, mostly same as student with the only difference is Create Meeting Page)

* Sequence Diagram của toàn bộ hệ thống + use case tổng ?

# **TESTING**

## *Test Plan*

* Định nghĩa sơ test plan là gì
* Toàn bộ test plan (dưới dạng table) + user stories của mỗi phần test

## *Test log*

* Toàn bộ test log (dưới dạng table) + đưa ra 1 file riêng để gửi riêng cho trường

## *ERROR*

* Hình một số phần test bị fail (frontend)
* Các test bị fail (dưới dạng table)
* Hướng giải quyết các test fail

# **SCREENCAST & PRESENTATION**

* Link Slide giới thiệu sơ về toàn bộ dự án
* Link Đoạn quay phim đang presentation của toàn bộ các thành viên (từng người 1, cắt ghép video) + đưa ra 1 file riêng để gửi riêng cho trường
* User Guide cho toàn bộ các function chính của sản phẩm (hướng dẫn dùng bằng hình ảnh) + đưa ra 1 file riêng để gửi riêng cho trường

# **LOCATION REPOSITORY**

* Link để download toàn bộ bài

# **REFERENCES**

Phần này cái nào ở trên có định nghĩa thì mới có references – 10 references ít nhất, dòng nào dùng reference thì phải đánh số của reference vô trước dòng đó

1. Courses.lumenlearning.com. (2019). *What is a Feasibility Study/Report | Technical Writing*. [online] Available at: https://courses.lumenlearning.com/alamo-technicalandbusinesswriting/chapter/unit-4-b\_feasibility-report\_readings-2/ [Accessed 20 Aug. 2019].
2. Smartsheet. (2019). *What's the Difference? Agile vs Scrum vs Waterfall vs Kanban*. [online] Available at: https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban [Accessed 20 Aug. 2019].
3. softwaretestinghelp. (2019). *what-is-stlc-v-model*. [online] Available at: https://www.softwaretestinghelp.com/what-is-stlc-v-model/ [Accessed 20 Aug. 2019].
4. Software Testing Books. (2019). *What is Spiral Model? Advantages and Disadvantages of Spiral Model - Software Testing Books*. [online] Available at: http://softwaretestingbooks.com/what-is-spiral-model [Accessed 20 Aug. 2019].
5. Tutorialspoint.com. (2019). *SDLC - RAD Model*. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc\_rad\_model.htm [Accessed 20 Aug. 2019].
6. model, S. (2019). *Spiral model*. [online] IONOS Startupguide. Available at: https://www.ionos.com/startupguide/productivity/spiral-model/ [Accessed 20 Aug. 2019].
7. Simplilearn.com. (2019). *Why a Feasibility Study is Important in Project Management*. [online] Available at: https://www.simplilearn.com/feasibility-study-article [Accessed 20 Aug. 2019].
8. Startrungrow.com. (2019). *Purpose of Feasibility | StartRunGrow - your one-stop site for everything business*. [online] Available at: http://www.startrungrow.com/information/business/1,2315,purpose-of-feasibility.htm [Accessed 20 Aug. 2019].
9. Khan, M. (2019). *A Comparative Review of Shared Hosting vs Dedicated Hosting vs Cloud Hosting (2019)*. [online] The Official Cloudways Blog. Available at: https://www.cloudways.com/blog/wordpress-shared-vs-dedicated-vs-cloud-hosting/ [Accessed 20 Aug. 2019].
10. Automation.com. (2019). *Understanding Microsoft's .NET Technology & Its Impact on Automation Application Development*. [online] Available at: https://www.automation.com/library/articles-white-papers/manufacturing-intelligence-industrial-information-management/understanding-microsofts-.net-technology-its-impact-on-automation-application-development [Accessed 29 Aug. 2019].
11. Medium. (2019). *Java vs .NET: Factors to Consider*. [online] Available at: https://codeburst.io/java-vs-net-factors-to-consider-cde1d22b06a7 [Accessed 29 Aug. 2019].
12. Slant, E., Code, V., IDEA, I., Rider, J. and IDE, N. (2019). *Slant - Eclipse vs Visual Studio detailed comparison as of 2019*. [online] Slant. Available at: https://www.slant.co/versus/1957/12045/~eclipse\_vs\_visual-studio [Accessed 29 Aug. 2019].
13. GeeksforGeeks. (2019). *Difference between Waterfall Model and Spiral Model - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/difference-between-waterfall-model-and-spiral-model/ [Accessed 29 Aug. 2019].

<https://viblo.asia/p/scrum-la-gi-va-tai-sao-nen-dung-scrum-1Je5Ed6LlnL>

<https://www.geeksforgeeks.org/sql-creating-roles/>