****

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**

* Kế hoạch phát triển ban đầu sẽ theo hướng nào, tại sao
* Nêu ra: có những tasks nào cần thực hiện, liên quan như thế nào đến đề bài
* Phân công các thành viên trong nhóm vào các vị trí : Database design, web design, programmer, tester – mỗi thành viên phải carry ít nhất 3 roles khác nhau (đổi role khi qua sprints mới)
* Meeting khi nào, dự kiến ở đâu, mục đích của các cuộc meeting đó là gì

# **AGILE METHODS FOLLOWED**

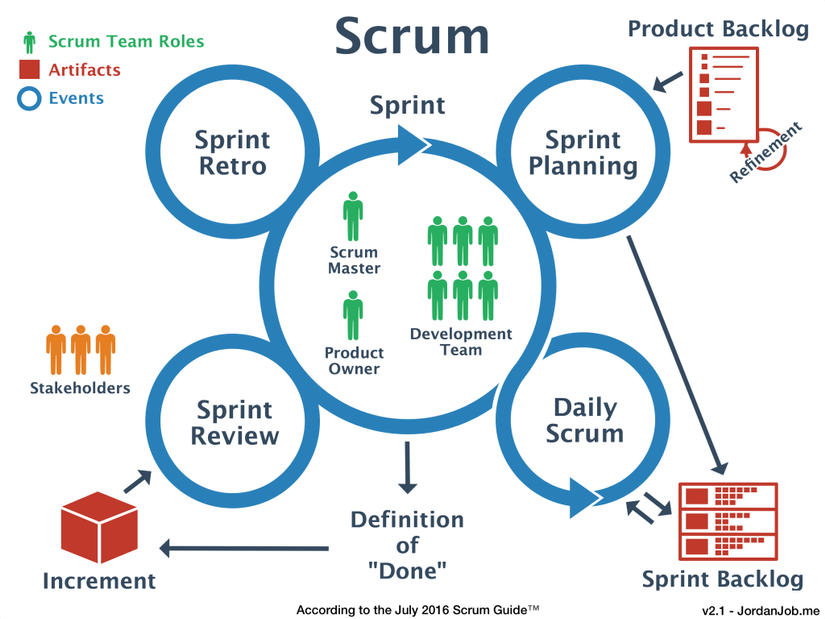


Figure 1: Scrum Method

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.

* Hình burndown chart của project hiện tại + đưa ra 1 file riêng để gửi riêng cho trường
* Giải thích ý nghĩa từng cột
* Giải thích một số số liệu bất thường (ví dụ ngày 12 thì số lượng công việc hoàn thành ít, do có thành viên bị ốm)

## *Minutes of Meetings*

* Định nghĩa, giải thích sơ minutes of meetings trong scrum
* 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*

* Định nghĩa sơ sprints trong scrum
* 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*

* Định nghĩa sơ product backlogs trong scrum
* 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:

* Giai thich thuan loi trong viec tao database nhu the nay (ví dụ: truy xuất nhanh)
* Hinh anh minh hoa 1

### Cons:

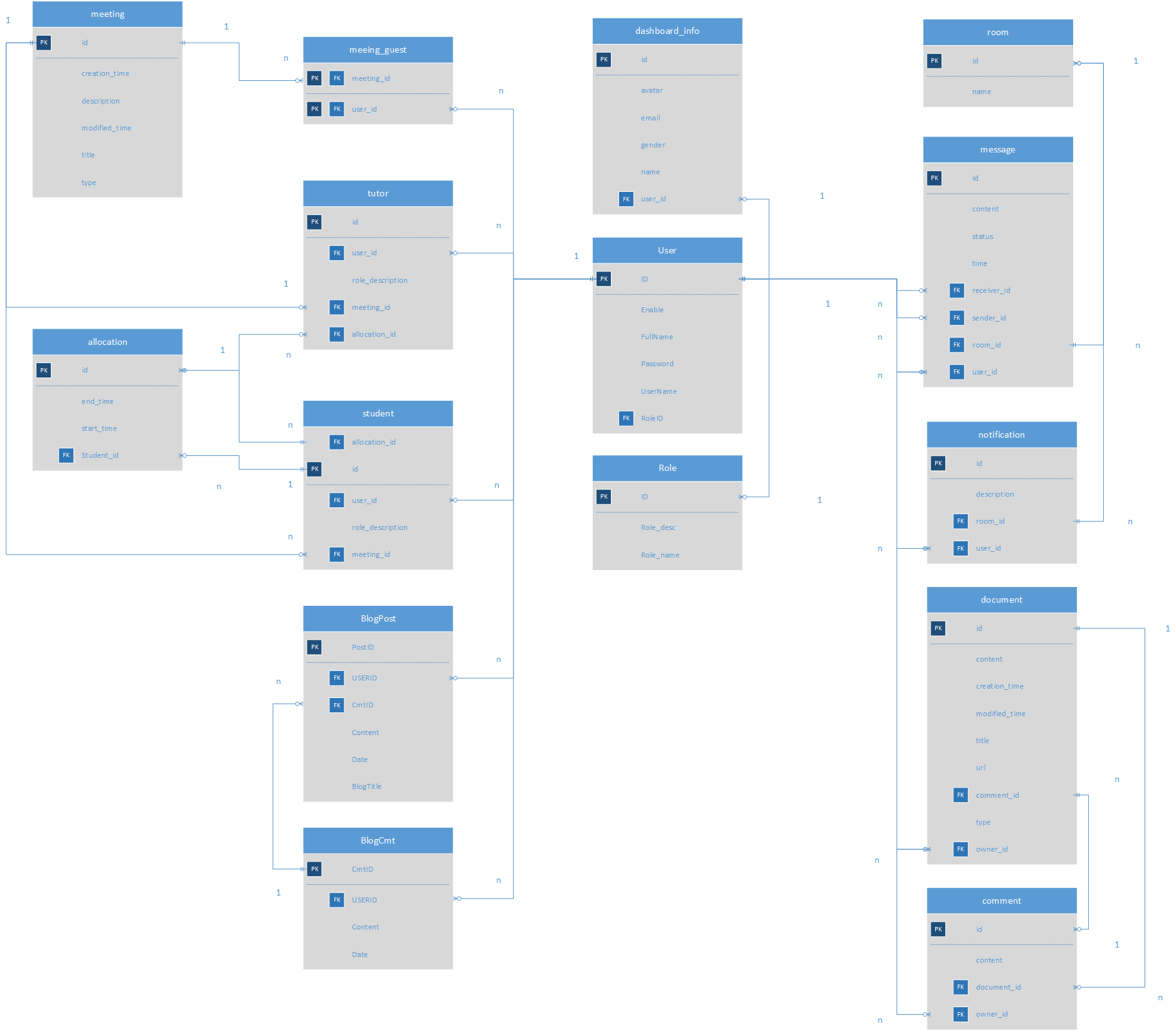
* Neu mot so bat loi trong viec tao database nhu the nay (vi du: lap du lieu...)
* Hinh anh minh hoa 2

## *Validation & Datatypes*

* Neu mot so datatype được dùng trong CSDL
* Giải thích tại sao dùng datatype đó
* Validation như nào

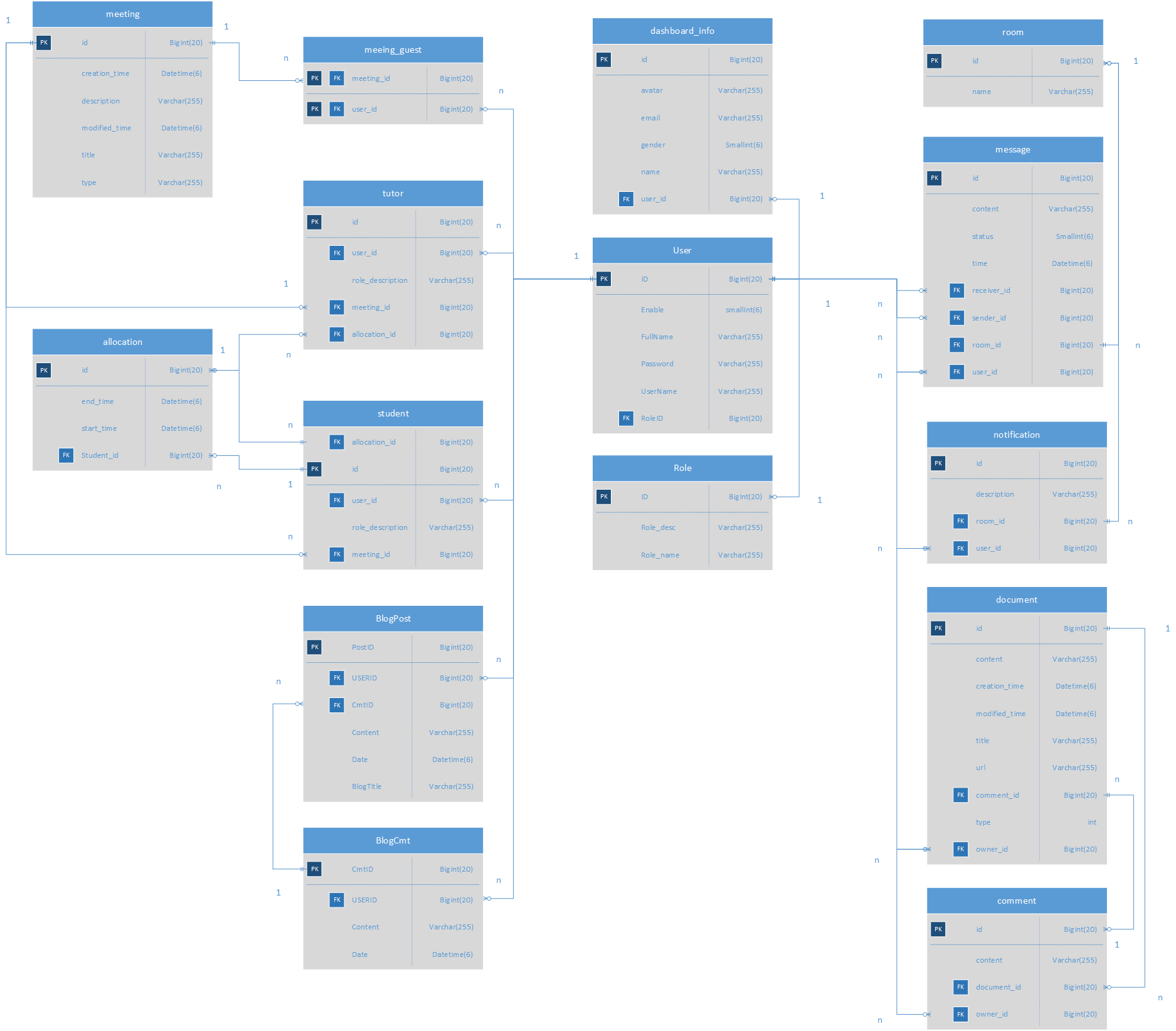
## *ERD*:

Entity relationship Diagram (without datatype version)



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)



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:*

* Giải thích các mối quan hệ trong CSDL (chụp ảnh từng bảng có quan hệ để giải thích)

Role base function

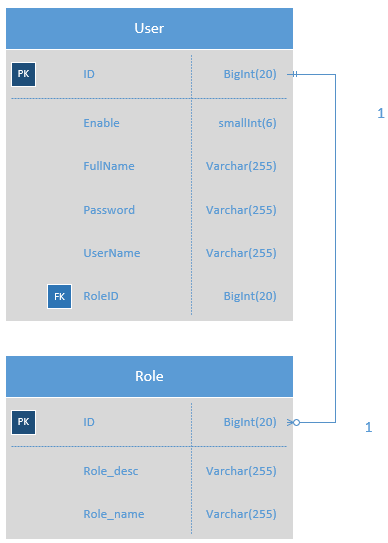
****

Figure 2: 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

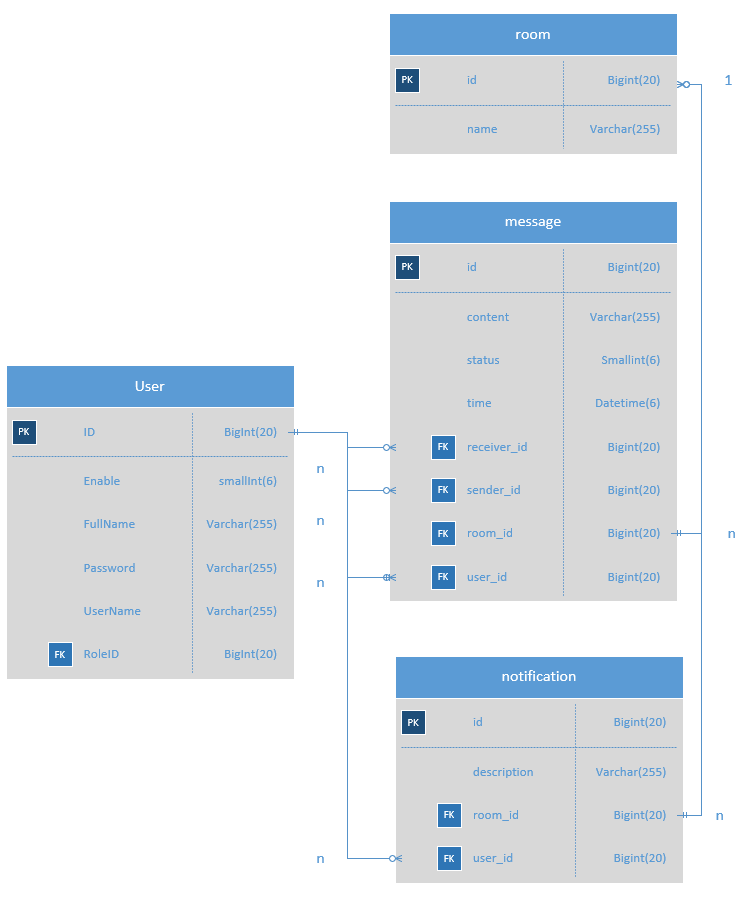


Figure 3: Message function

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

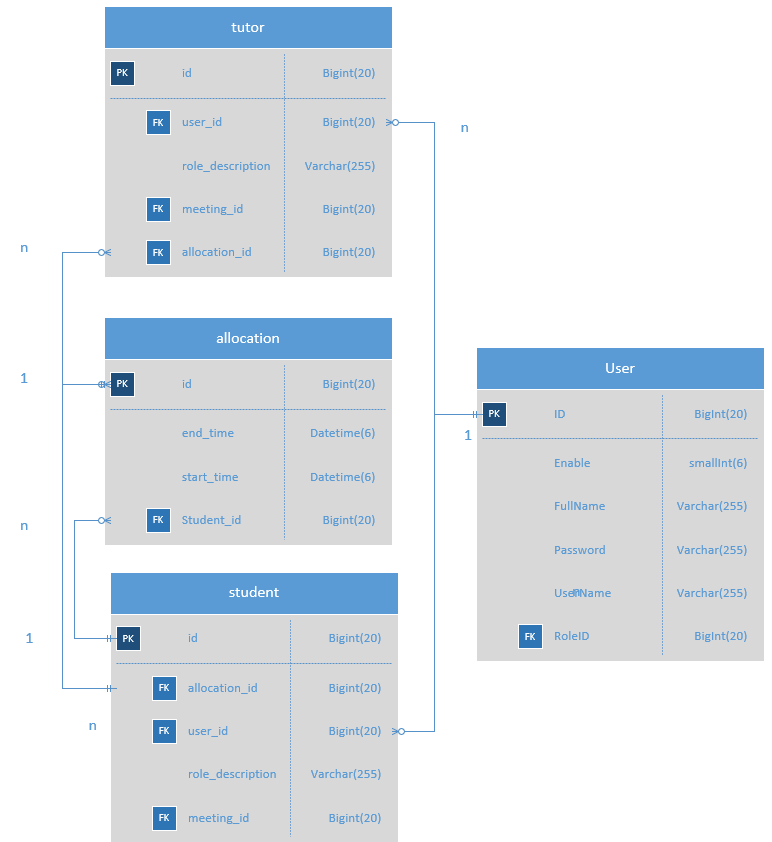


Figure 4: Allocating function

It’s a bit complex but with 4 tables: User, Tutor, Student, and Allocation, that will be come 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

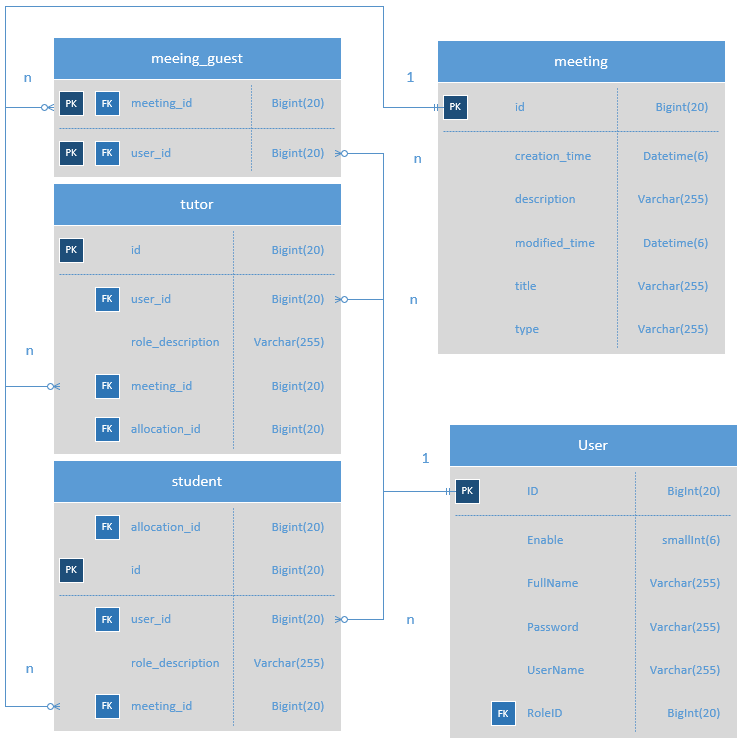


Figure 5: Meeting function

Document and Comment Function

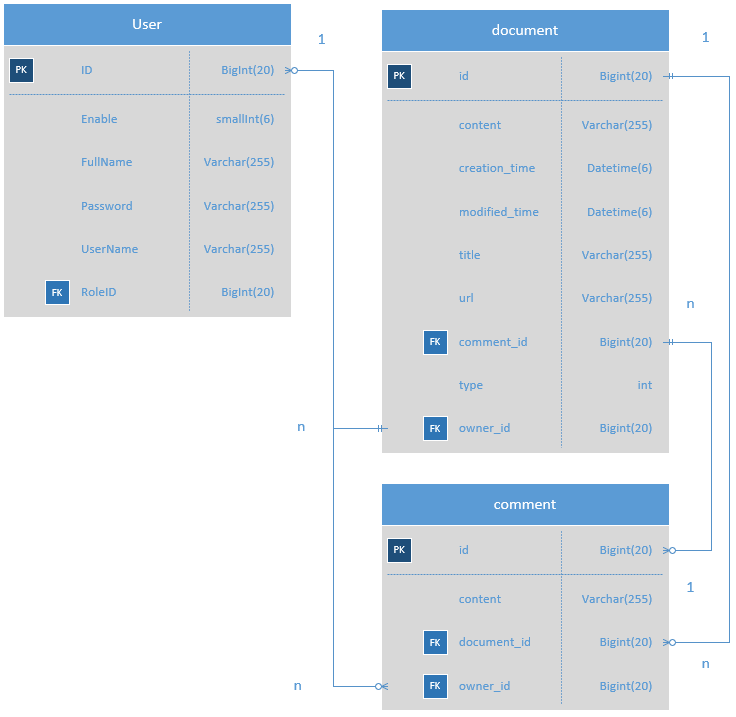


Figure 6: Upload funtion

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 7: Blog function

# **SITE DESIGN**

## *What is Responsive Design*

* Định nghĩa sơ Responsive Design

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

* Hình chụp code minh họa
* Giải thích mình đã dùng những gì trong frontend để nó trở thành Responsive Design

## *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

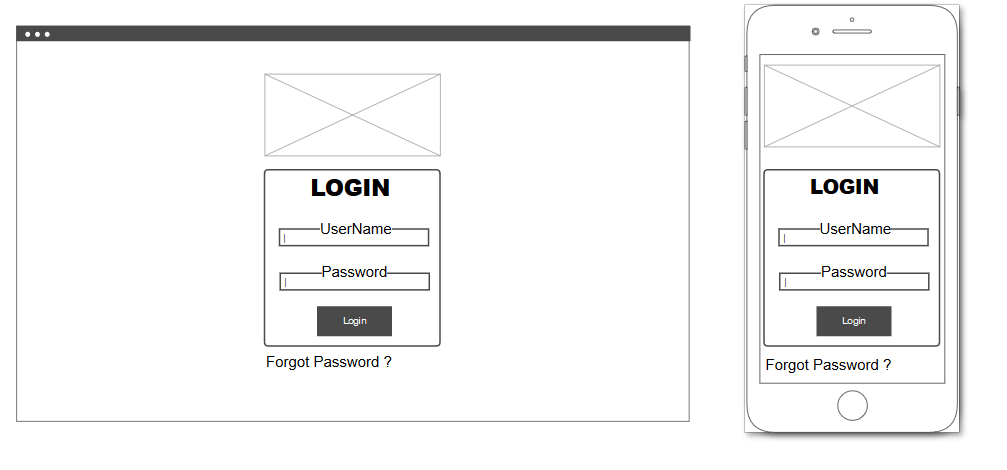


Figure 8: Home Page

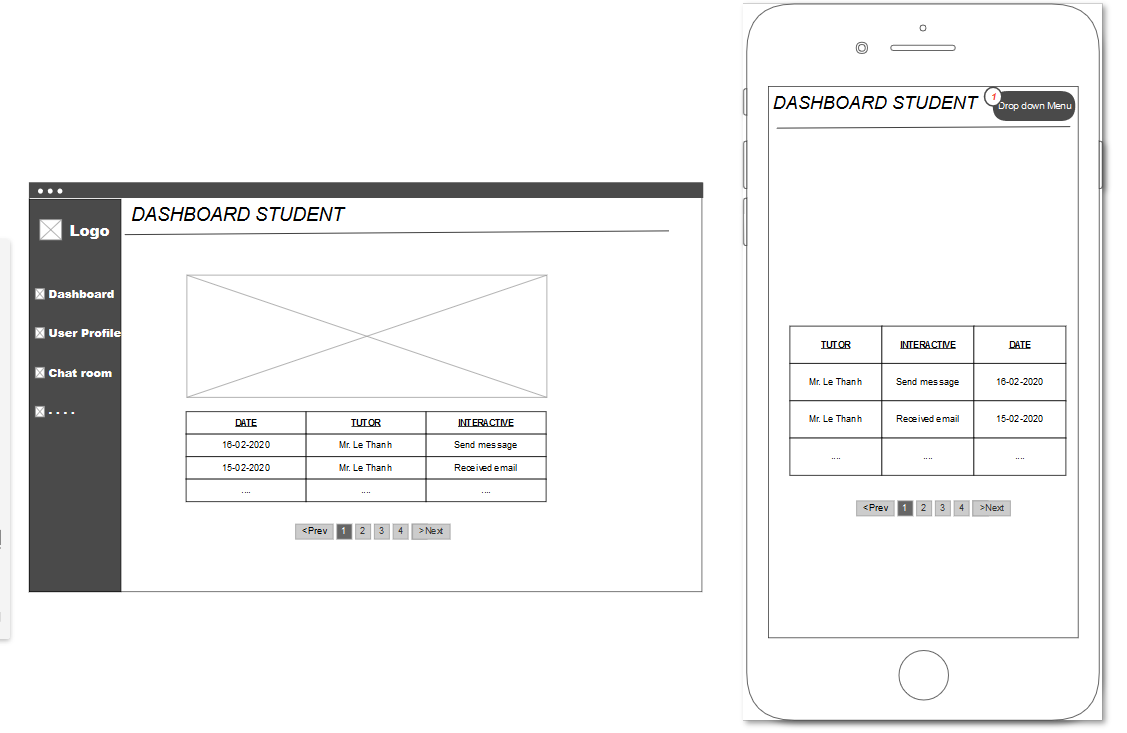


Figure 9: Student Dashboard

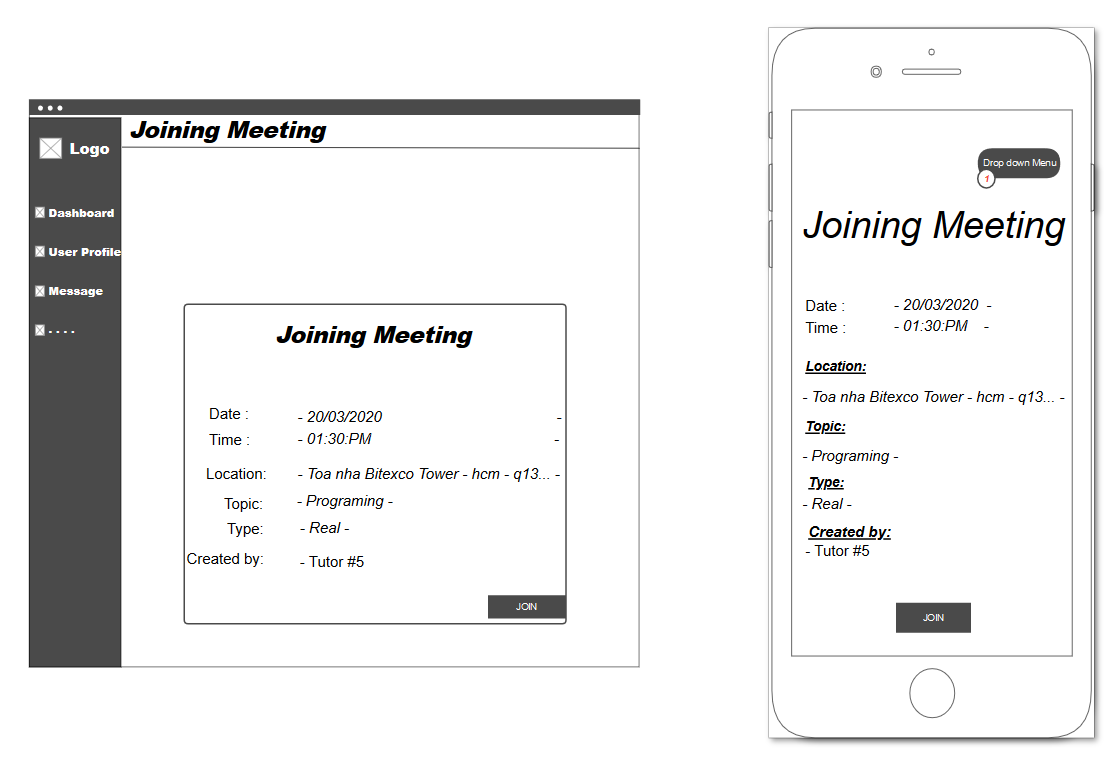


Figure 10: Student Joining Meeting Page

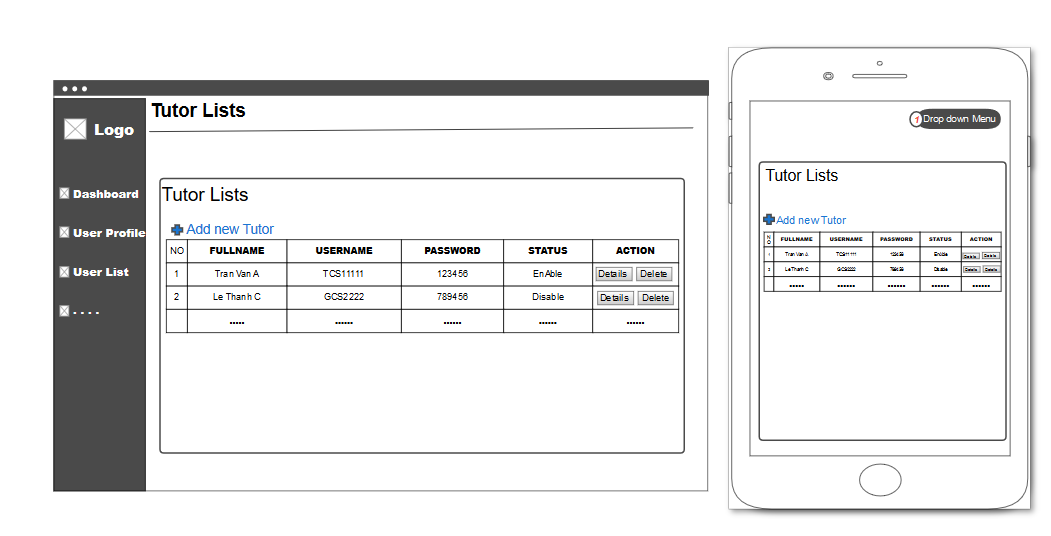


Figure 11: Tutor List Page

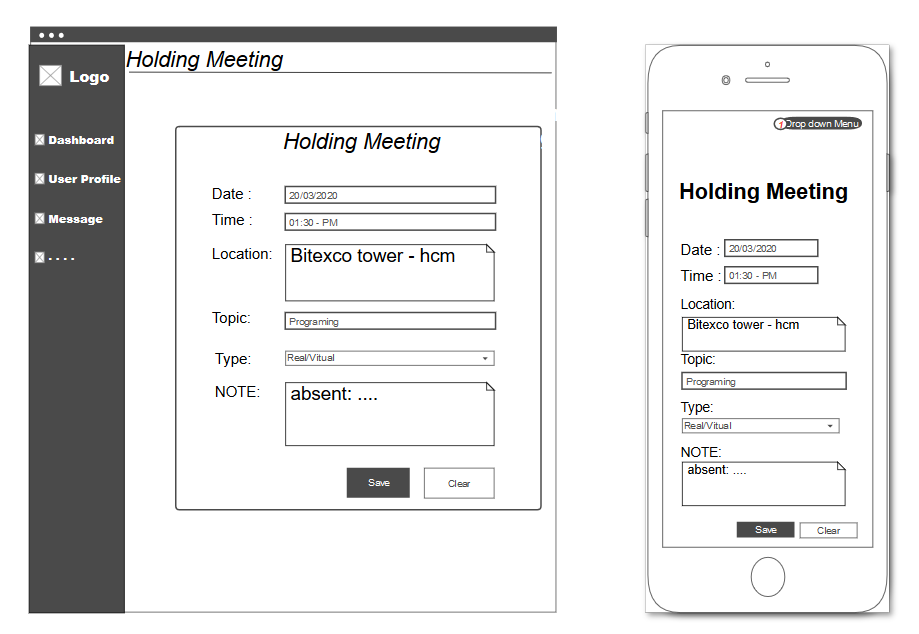


Figure 12: Tutor Holding Meeitng Page

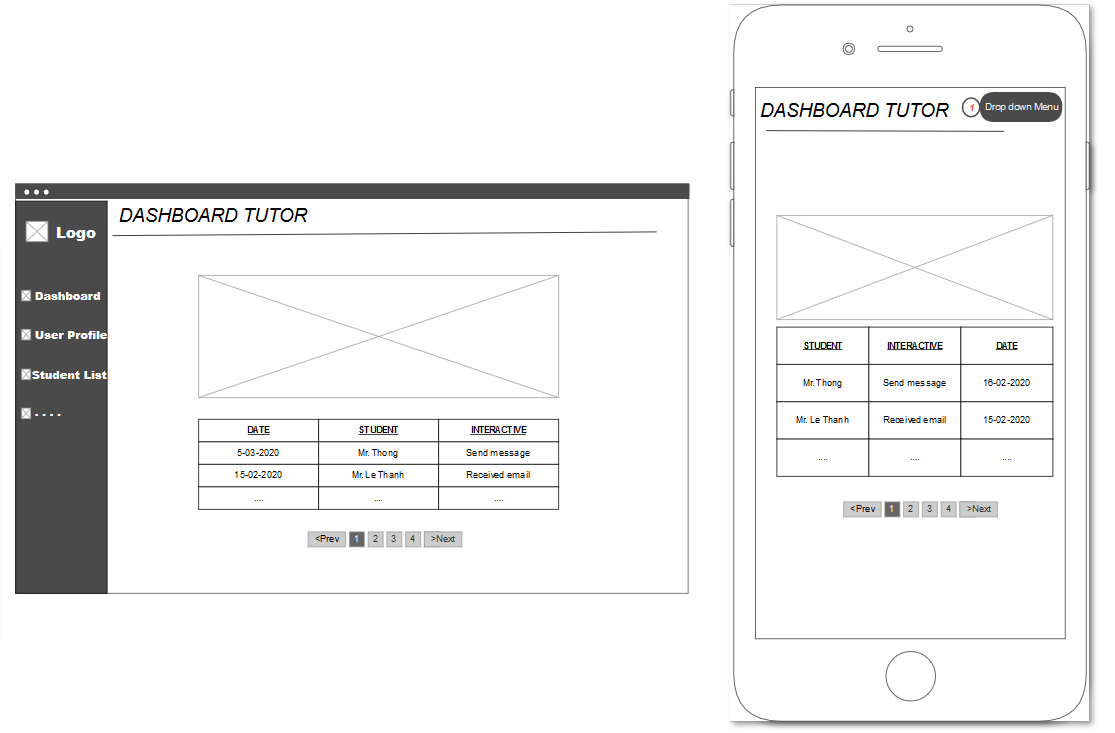


Figure 13: Tutor Dashboard

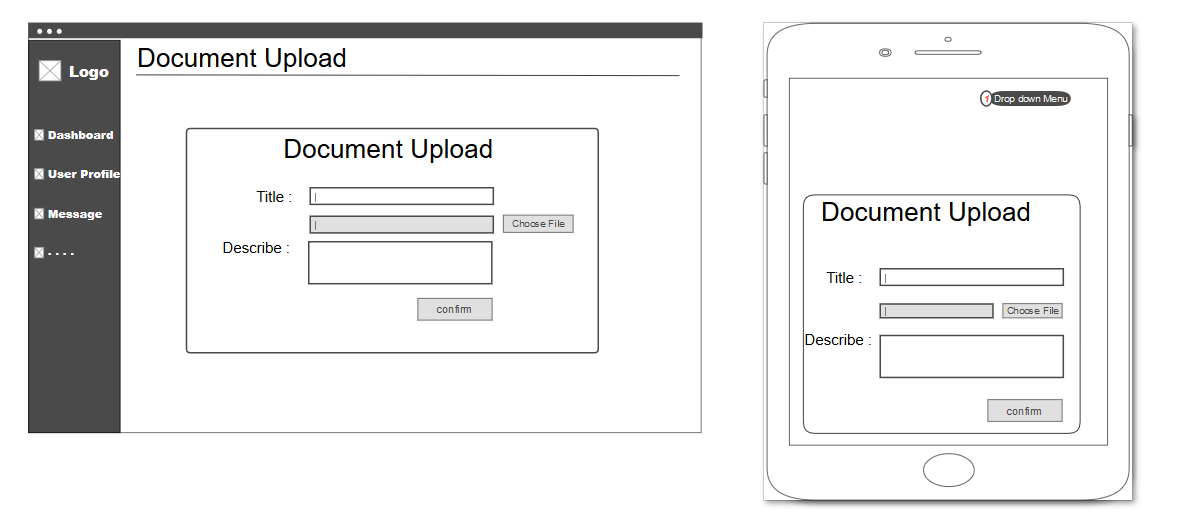


Figure 14: Document Upload Page

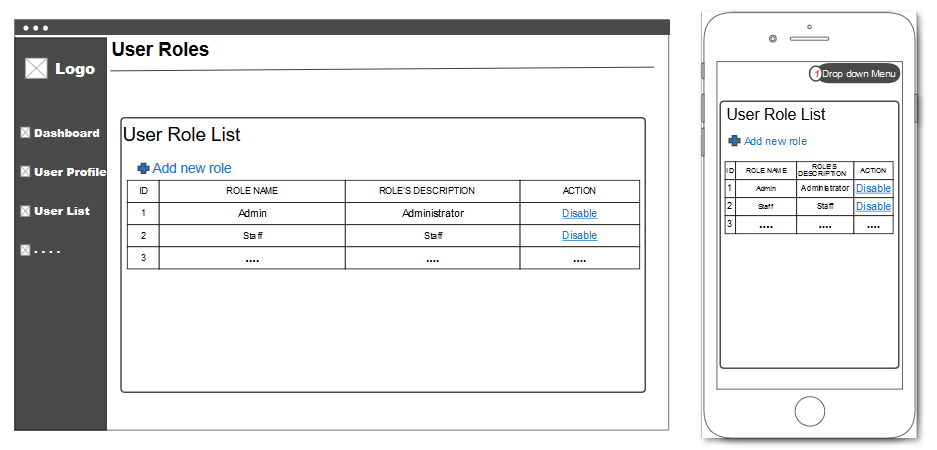


Figure 15: User Role Management Page

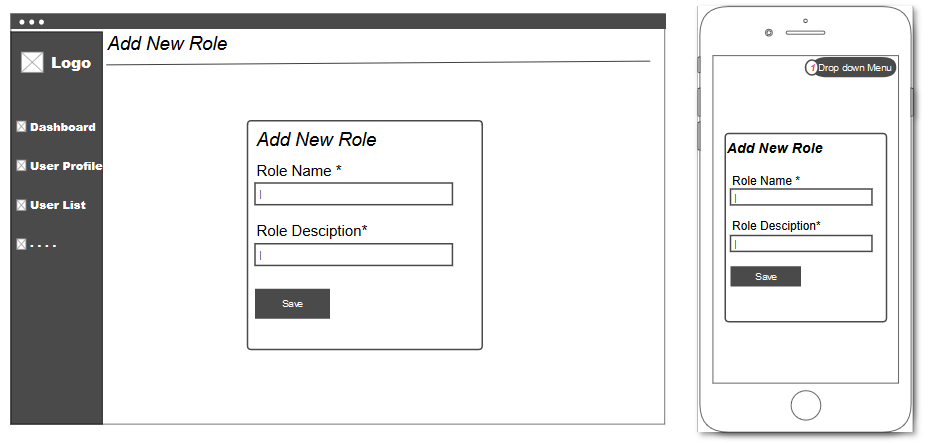


Figure 16: Add New Role Page

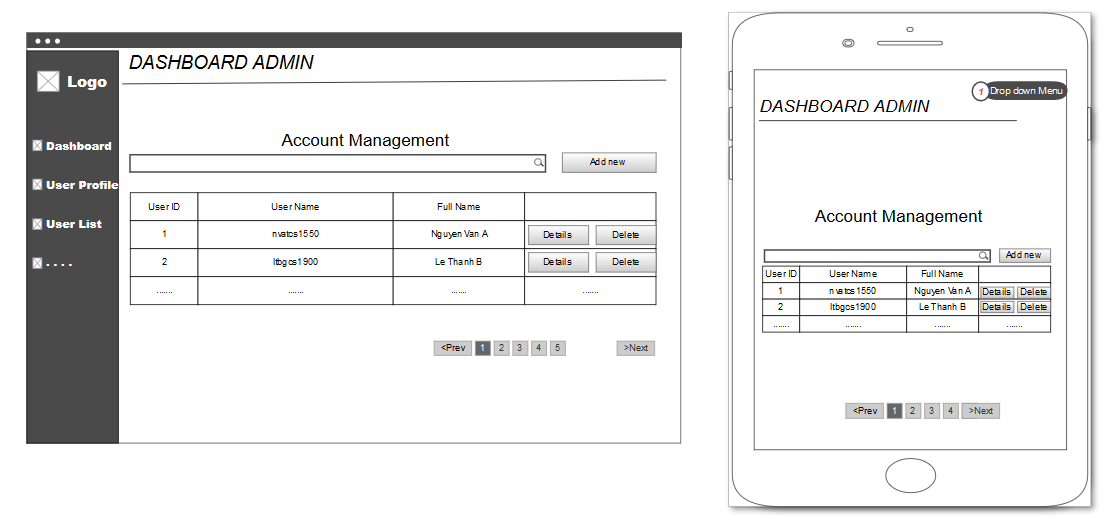


Figure 17: Admin Dashboard

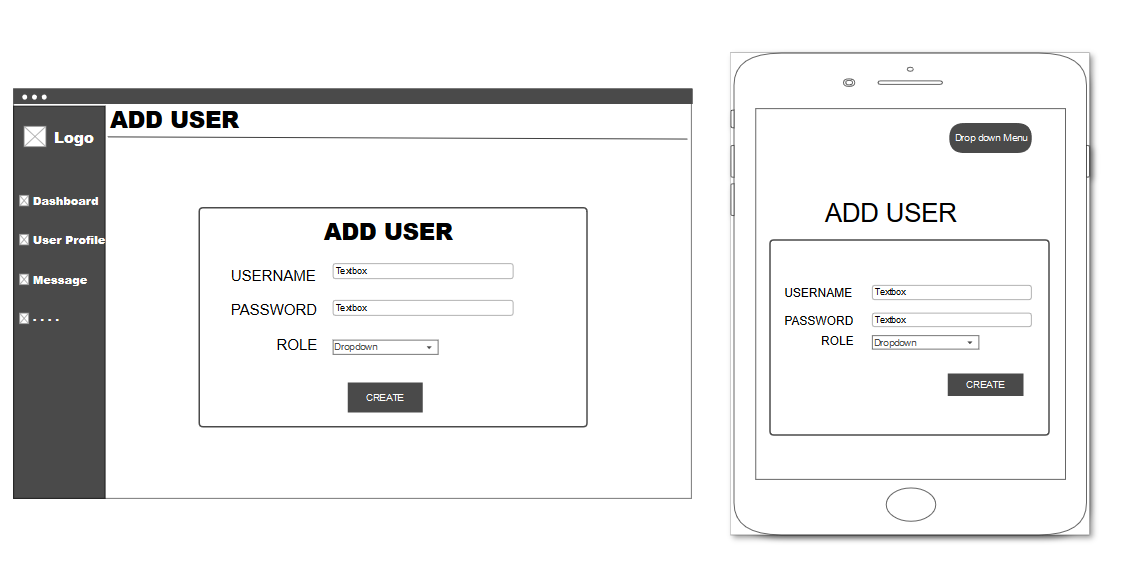


Figure 18: Admin Add User Page

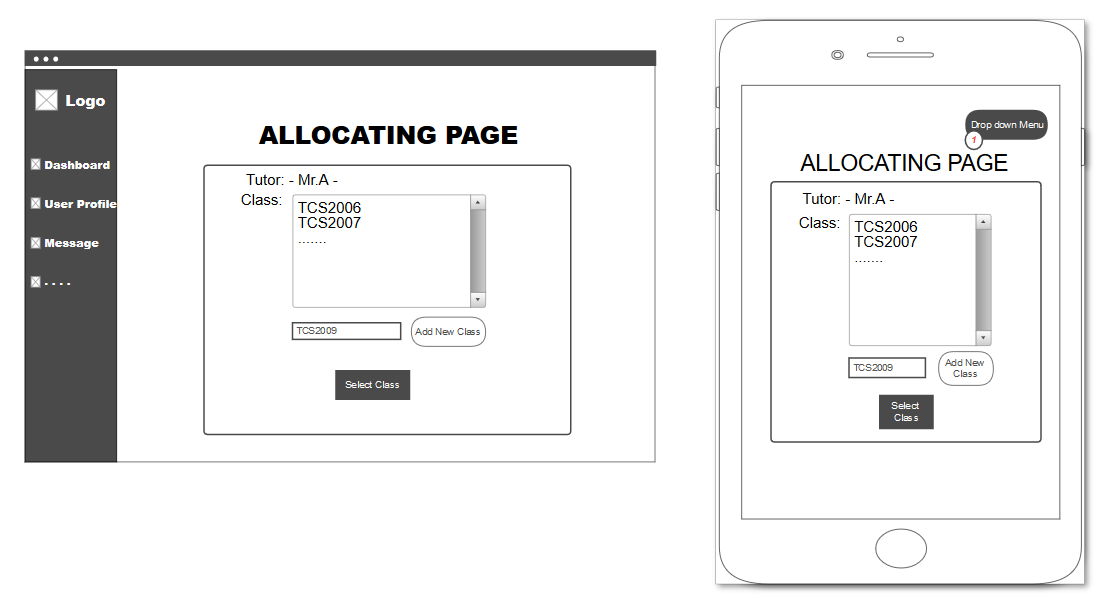


Figure 19: Allocating Class Page

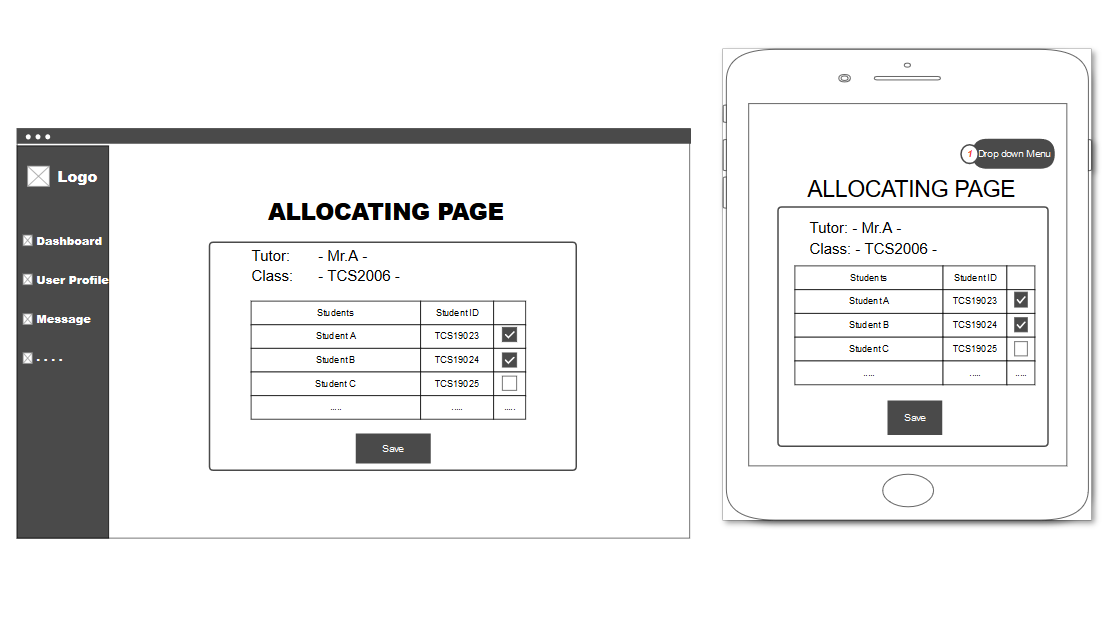


Figure 20: Allocating Student Page



Figure 21: Allocating Select Tutor Page



Figure 22: Blog Comment Page

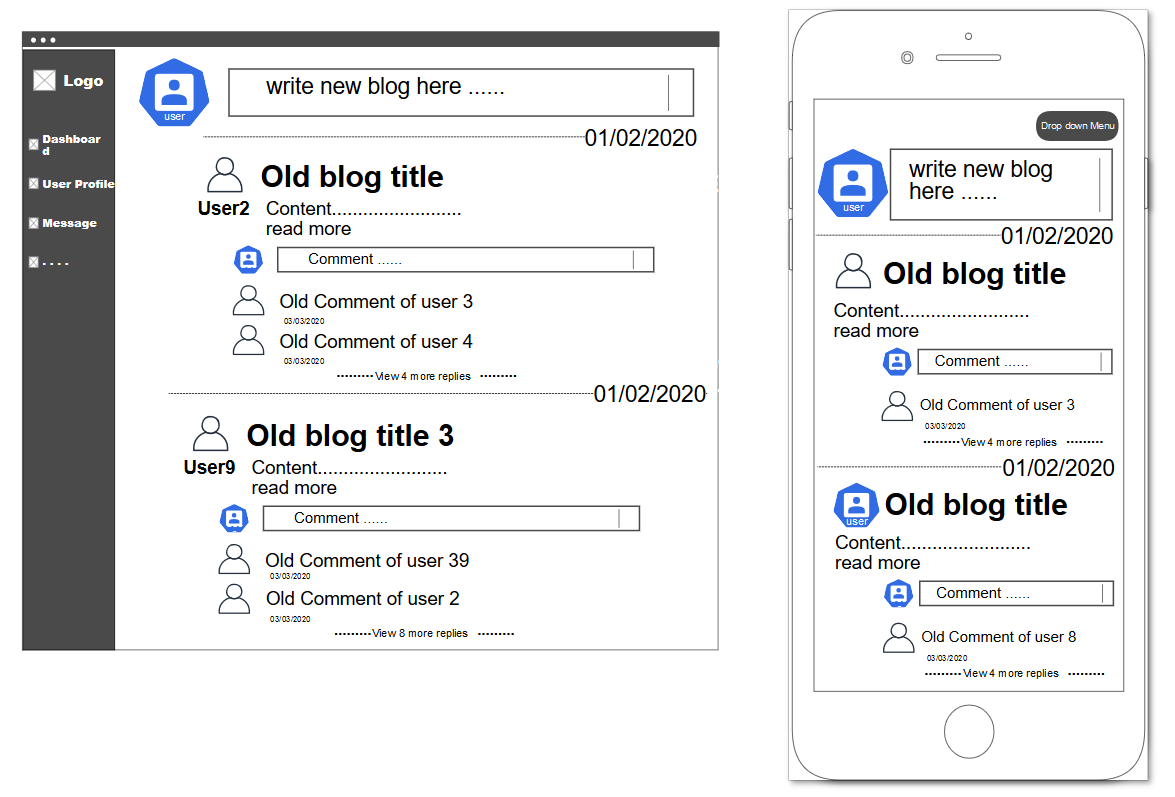


Figure 23: Blog Main Page

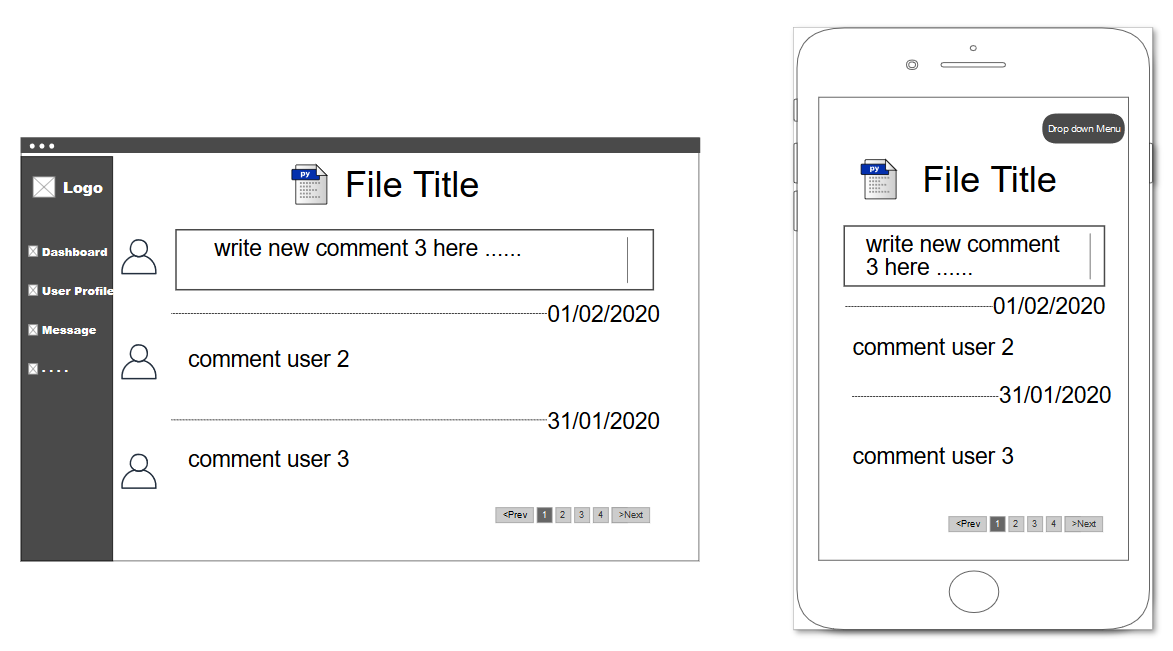


Figure 24: Upload Comment Page

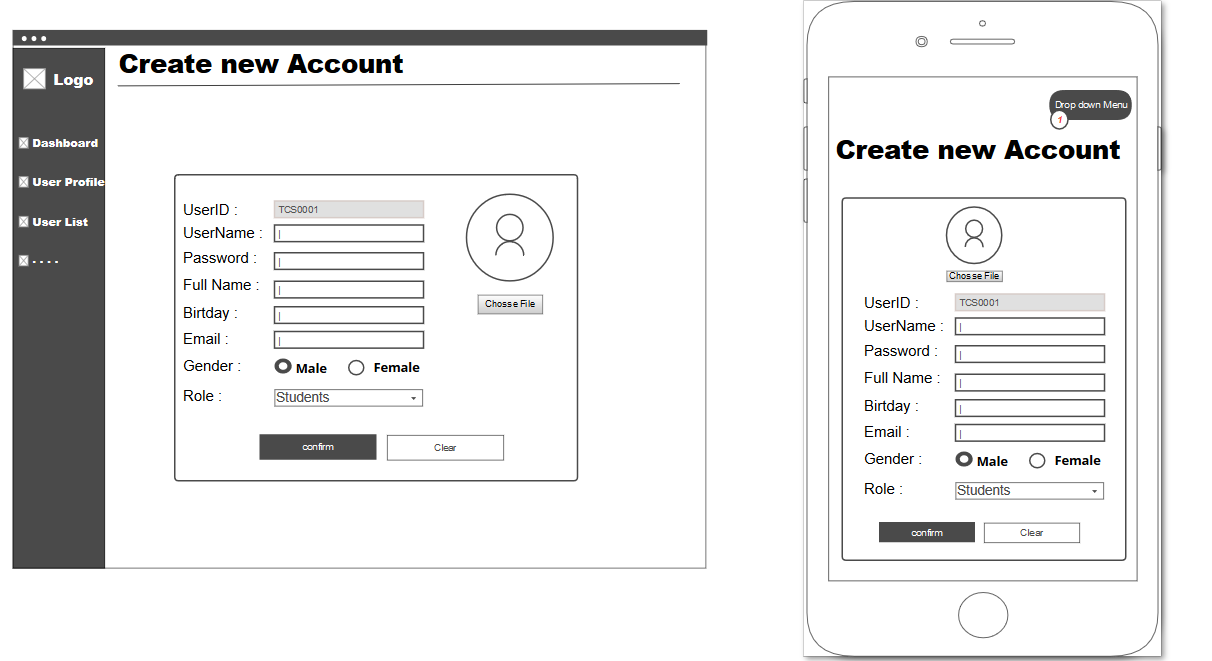


Figure 25: Staff Create new account page

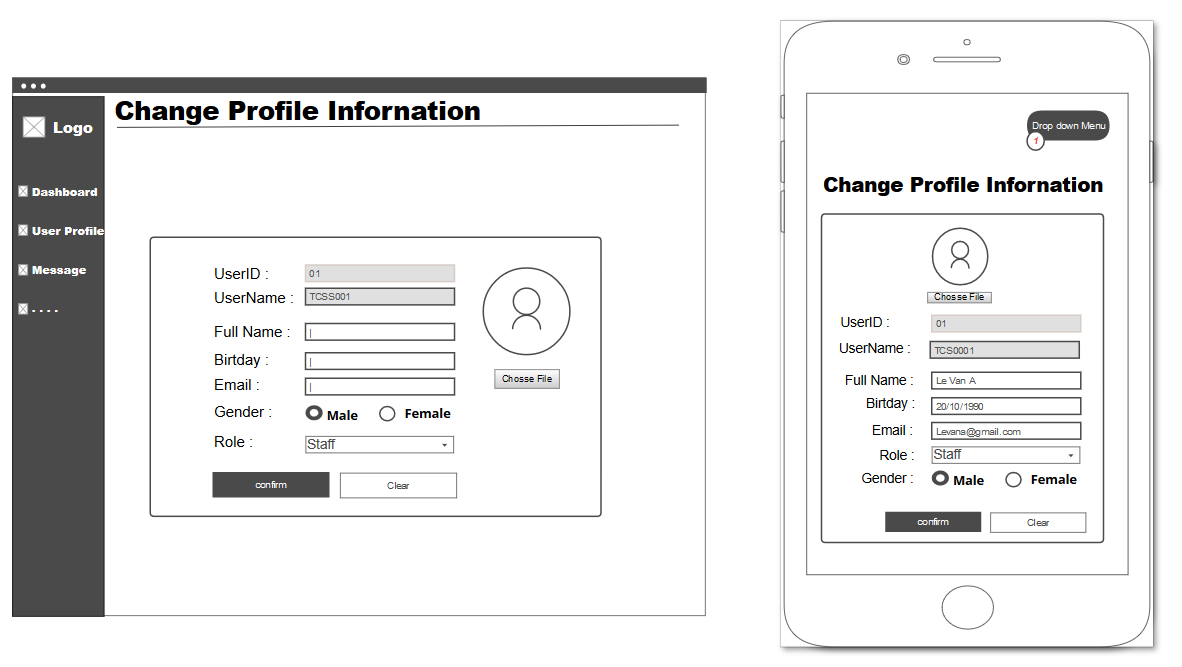


Figure 26: Admin Change user Profile page

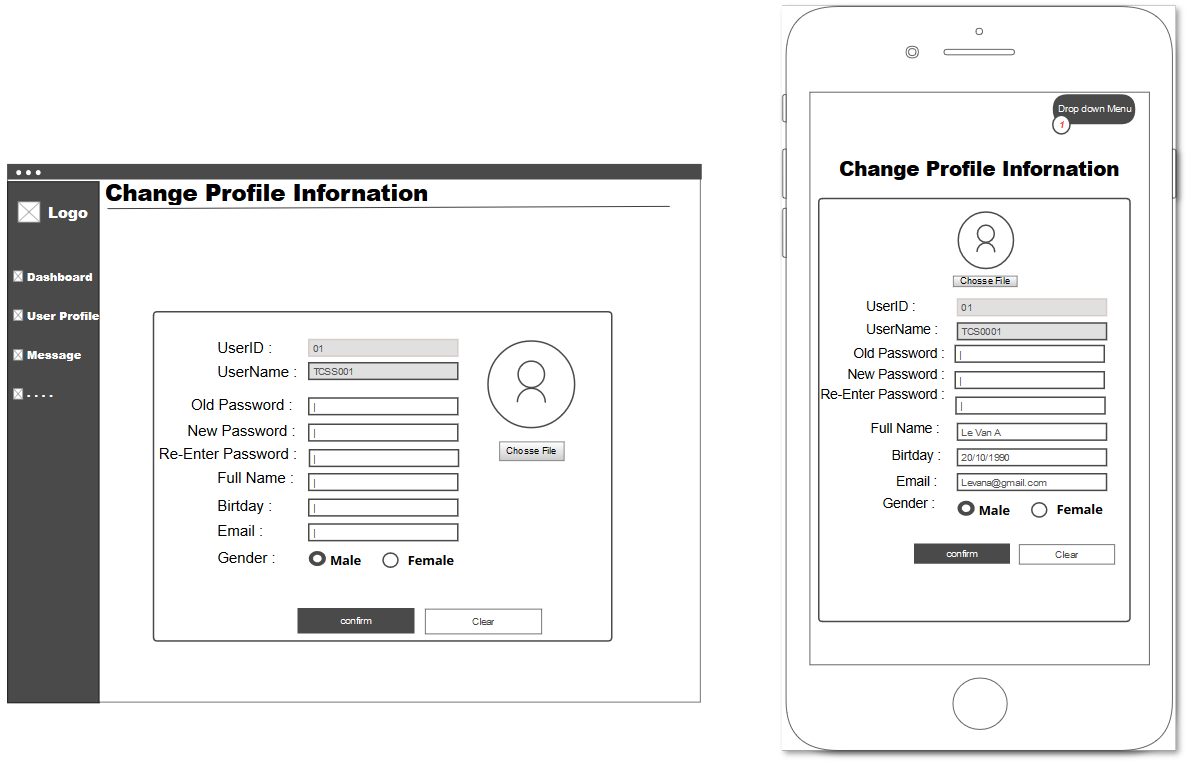


Figure 27: Users Change Profile page

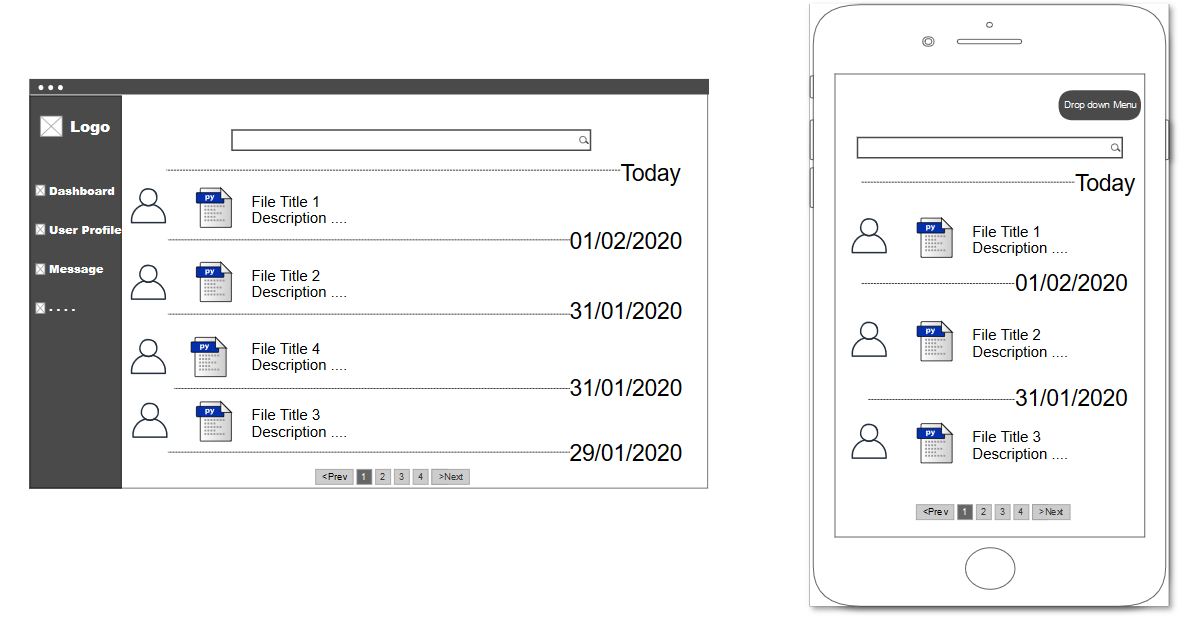


Figure 28: Upload Main page

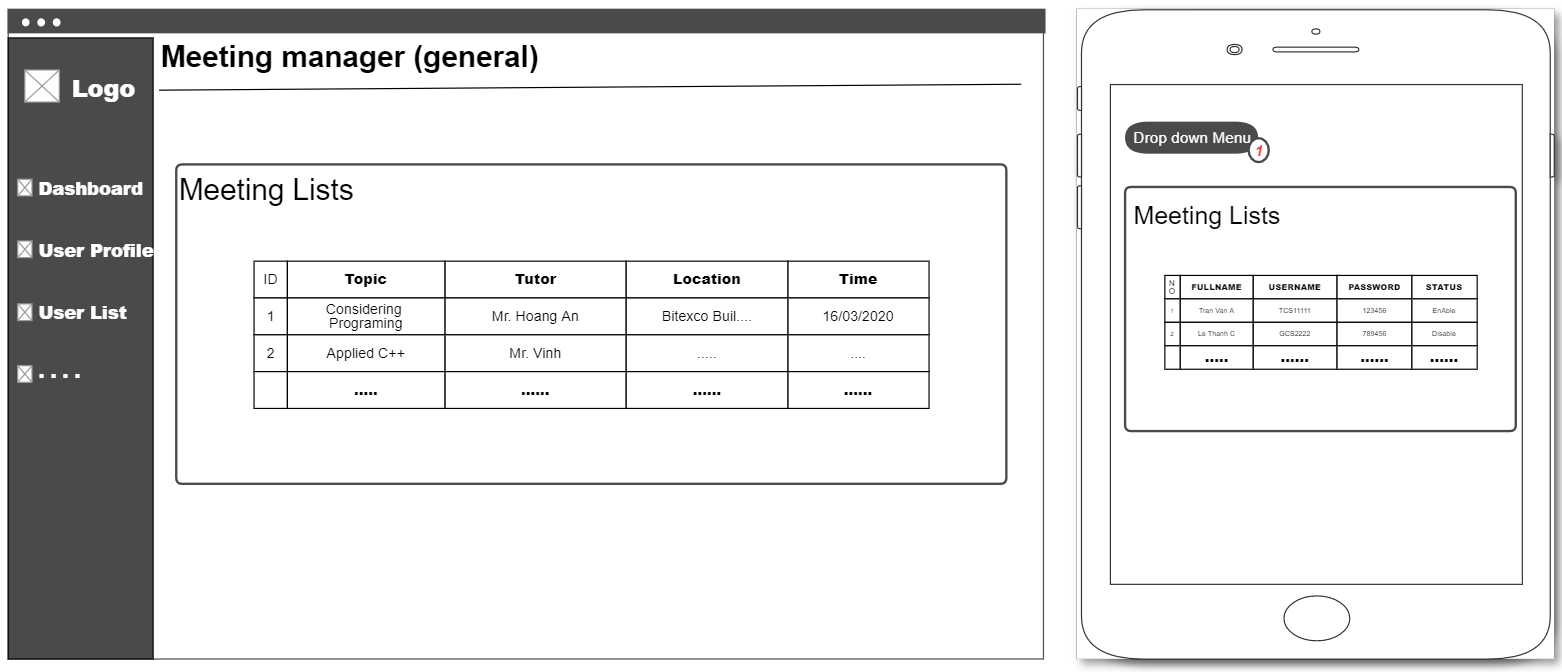


Figure 29: General Meeting Manager



Figure 30: Message Board

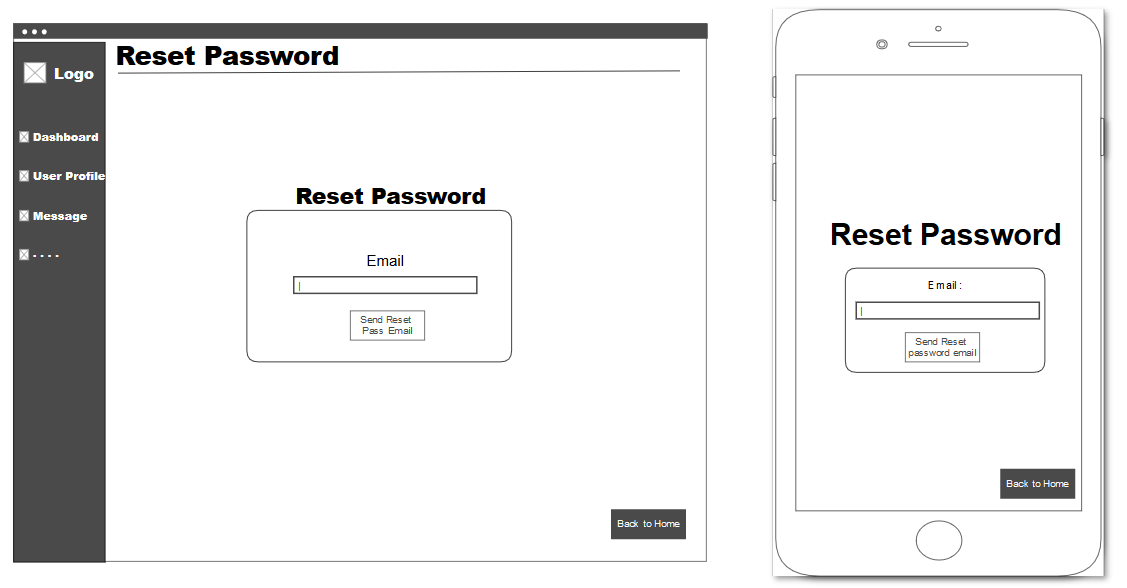


Figure 31: Reset Password Page



Figure 32: Admin Dashboard

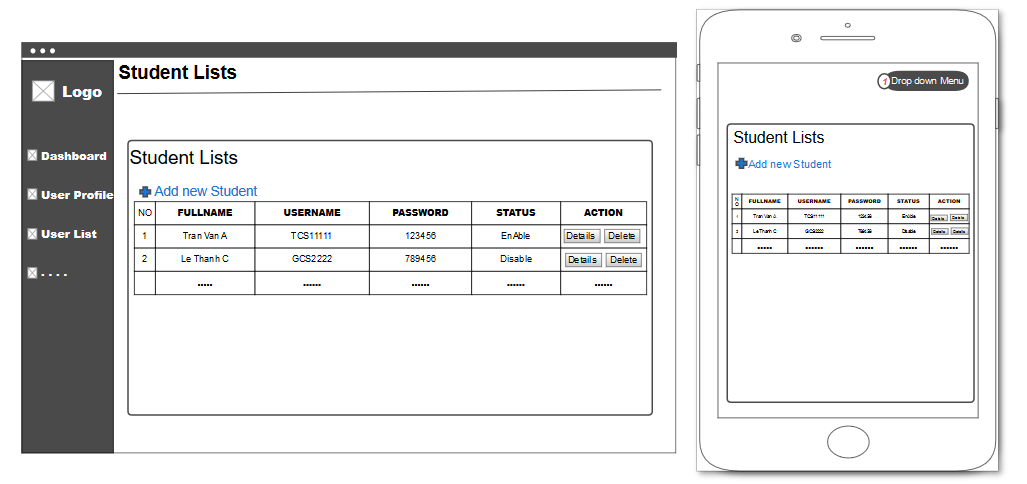


Figure 33: Student List Page

## *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*

* Giải thích tại sao chia role lại tốt cho việc bảo mật
* Hình vài đoạn code minh họa cho việc chia role sẽ dễ phân loại người dùng hơn

## *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 Email from the system*

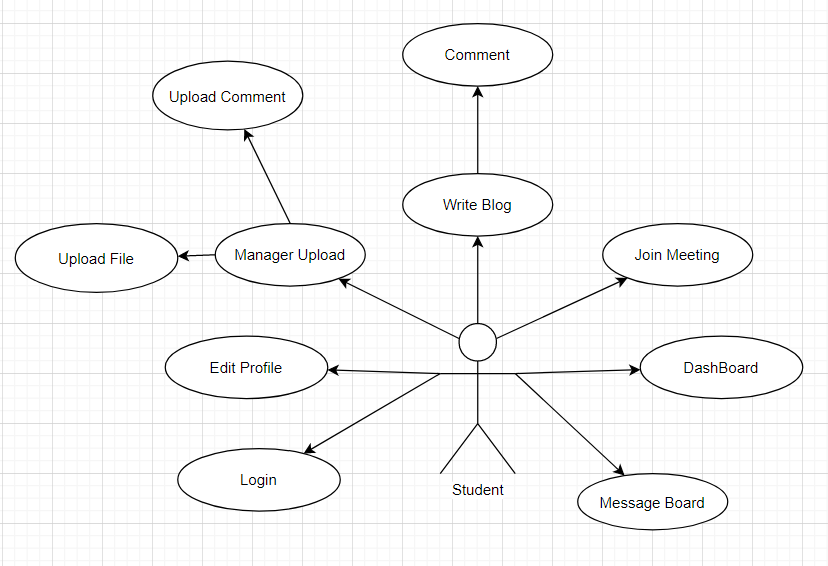
* Giải thích lại yêu cầu của đề bài về email
* Hình một đoạn code minh họa về gửi notification khi có email
* Giải thích đoạn code đó
* Hình về một cái notification khi có email trên trang web
* Giải thích, đánh giá cái notification đó có ý nghĩa thế nào với tutor/student

## *UML diagram*

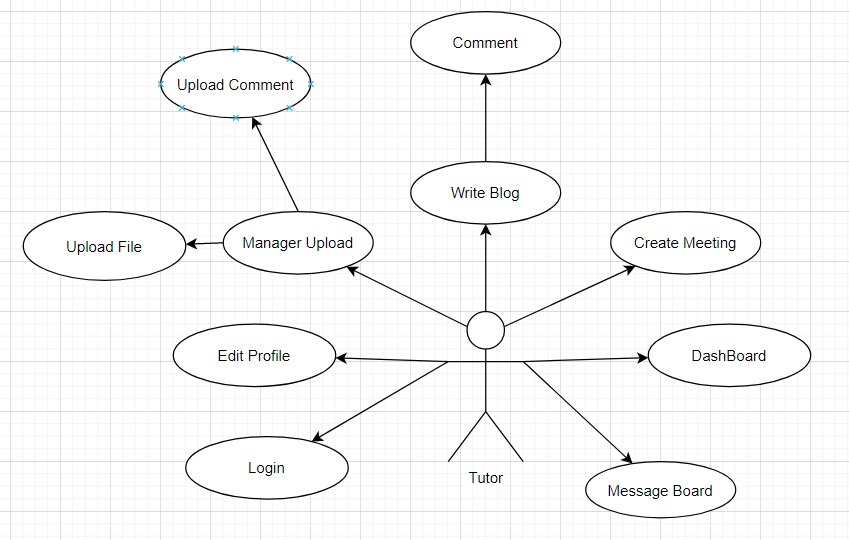
UML – Use Case Diagram is the primary form of system and software requirements for a new software program under development. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual presentation. A key concept of use case modelling is that it helps us to design a new system from the end user’s perspective. It is an effective technique for communicating system behavior in user’s terms by specifying all externally visible system behaviors.

A use case diagram is usually simple. It does not show all of the detail of use cases:

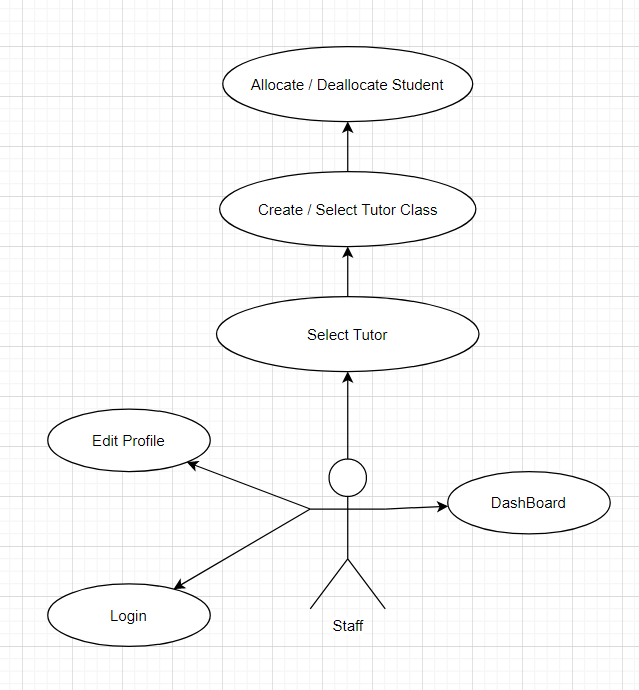
* It only **summarizes some of the relationships** between use cases, actors and systems.
* It **does not show the order** in which steps are performed to achieve the goals of each use case.
* **Student Use Case Diagram**



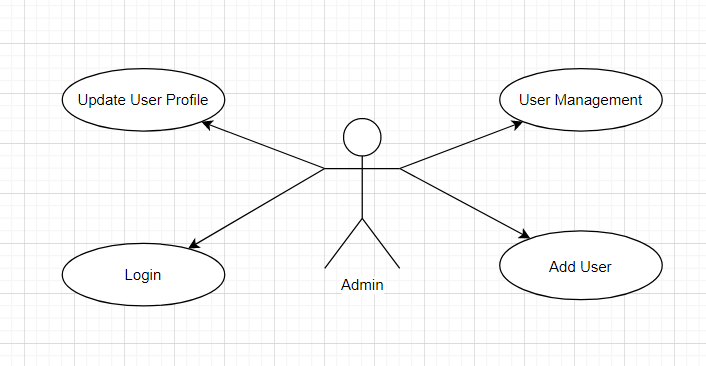
* **Tutor Use Case Diagram**



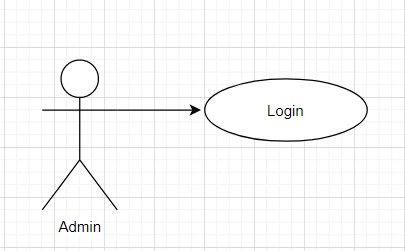
* **Staff Use Case Diagram**



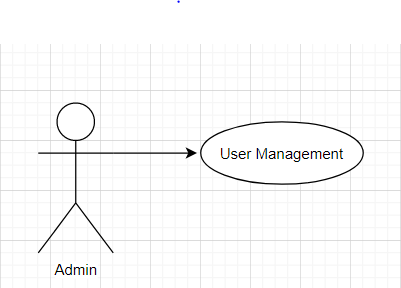
* **Admin Use Case Diagram**



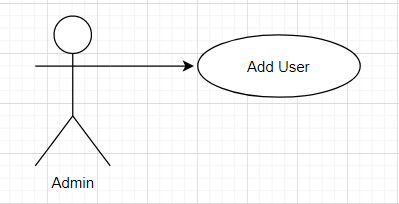
* **Use Case Breakdown**



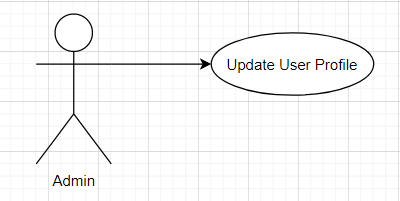
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Admin Login Page |
| **Actors** | Admin |
| **Description** | Login into the specific admin manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | 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 |



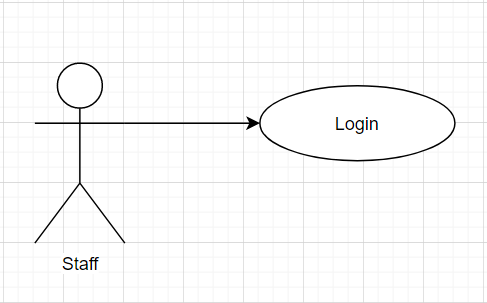
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | User Management DashBoard |
| **Actors** | Admin |
| **Description** | Manage Users |
| **Goal** | Reach the manager page of Admin DashBoard & view Users Statistics |
| **Pre-condition** | Display Login Successful Page |
| **Trigger** | 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 |



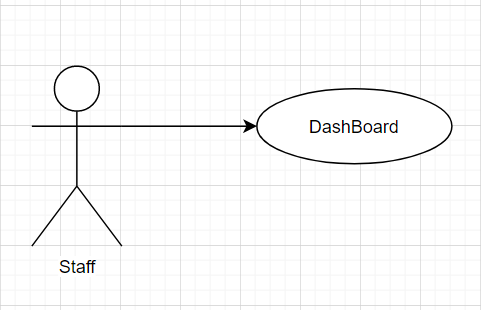
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Add User Page |
| **Actors** | Admin |
| **Description** | Add new users into the system |
| **Goal** | Perform add user function successful |
| **Pre-condition** | Display Add User Page |
| **Trigger** | Press “Add” button |
| **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 |



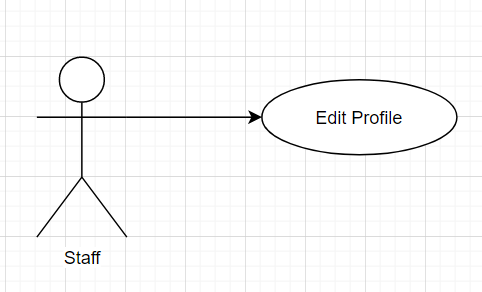
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Update User Profile Page |
| **Actors** | Admin |
| **Description** | Update users information |
| **Goal** | Perform update users function successful |
| **Pre-condition** | Display Update User Profile Page |
| **Trigger** | 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 |



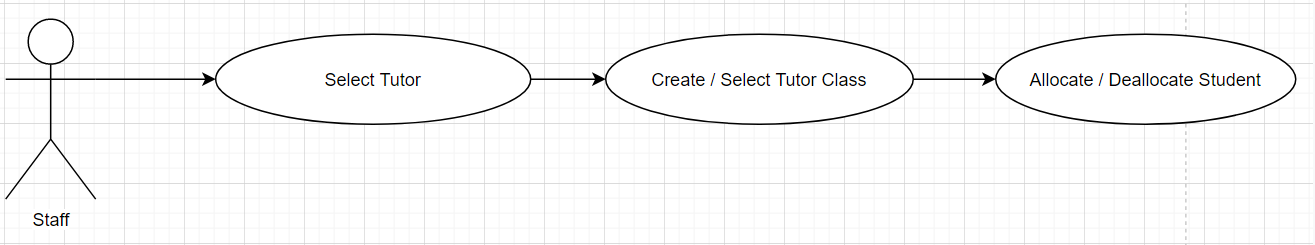
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Staff Login Page |
| **Actors** | Staff |
| **Description** | Login into the specific staff manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | 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 |

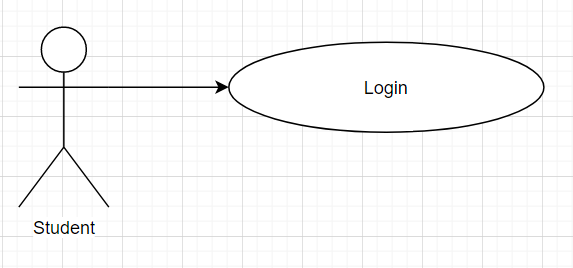


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Create New User Account Page |
| **Actors** | Staff |
| **Description** | Input new user’s information to create new user account |
| **Goal** | New user account creation |
| **Pre-condition** | Successful Display Staff Create 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 |

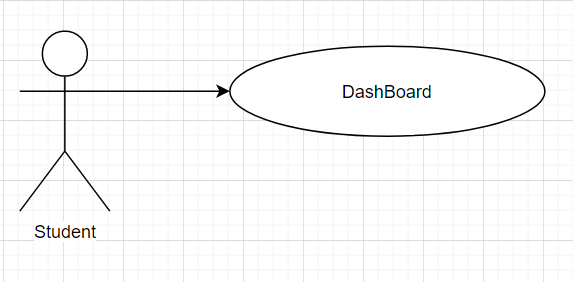


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Class Room Creation & Select Tutor Page |
| **Actors** | Staff |
| **Description** | Create new classroom & select tutor for corresponding classroom |
| **Goal** | Perform create & select functions successful |
| **Pre-condition** | Display Class Room Creation & Select Tutor Page |
| **Trigger** | Create Classroom |
| **Post-condition** | Show message, update new class room field & tutor class in system database |
| **Normal flow** | 1. Create a new class  2. Select tutor for that class  3. 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 |

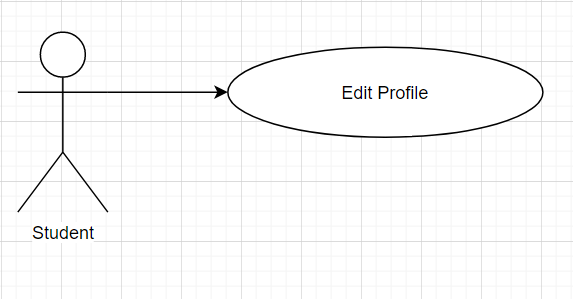
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Allocation page |
| **Actors** | Tutor |
| **Description** | Allocate / Deallocate student & tutor |
| **Goal** | Perform Allocation / Deallocation action successful |
| **Pre-condition** | Display Student & Tutor Manager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



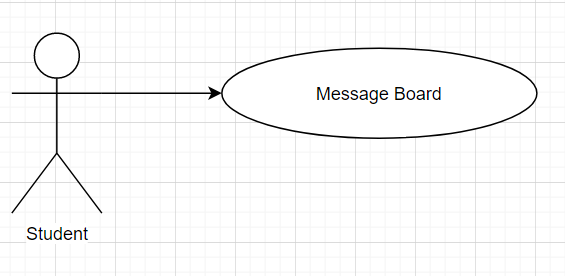
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Login Page |
| **Actors** | Student |
| **Description** | Login into the specific student manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | 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 |



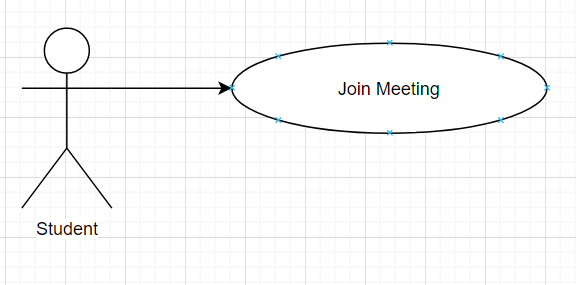
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student DashBoard |
| **Actors** | Student |
| **Description** | View & read student information on DashBoard |
| **Goal** | Reach the manager page of Student DashBoard & view student information |
| **Pre-condition** | Display 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 |



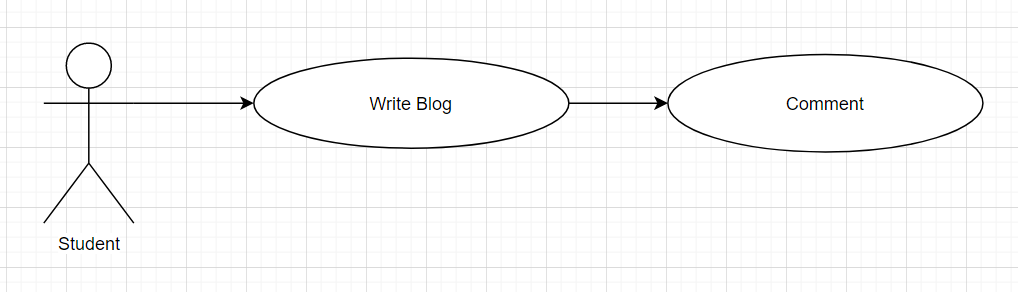
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Edit Profile Information Page |
| **Actors** | Student |
| **Description** | Input student data to update new information with the database |
| **Goal** | Update new user information into system database |
| **Pre-condition** | 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 |

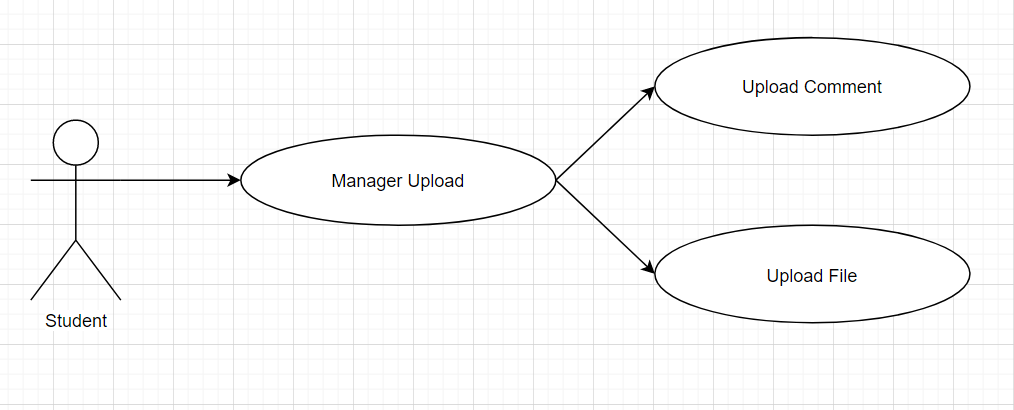


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Join Meeting Page |
| **Actors** | Student |
| **Description** | Perform join meeting function to allocate student to a group meeting |
| **Goal** | Student successfully joins a meeting |
| **Pre-condition** | Successful 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 |

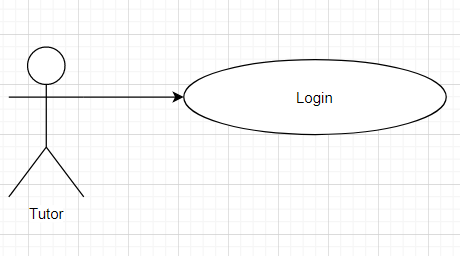


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Write Blog Page |
| **Actors** | Student |
| **Description** | Create new blog post from student |
| **Goal** | Successfully create new blog post |
| **Pre-condition** | Successful Display Write Blog Page |
| **Trigger** | Press “Post” button (missing from the mockup write blog image) |
| **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, Inappropriate content]  Show the error page  Return to the Student Write Blog Page |

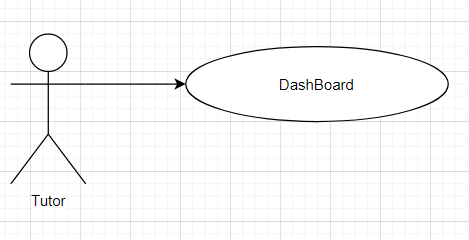
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Blog Comment Page |
| **Actors** | Student |
| **Description** | Student uses this page to send comment on blog post |
| **Goal** | Successfully send comment on blog post |
| **Pre-condition** | Successful Display Blog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



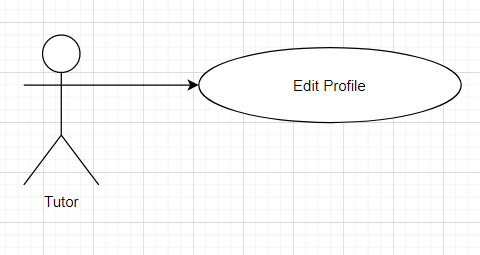
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Student Upload Page |
| **Actors** | Student |
| **Description** | Upload student files |
| **Goal** | Successfully upload new files, update new data in system database |
| **Pre-condition** | Successful Display Student Upload Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message |
| **Normal flow** | [Upload file flow]  1. Student click on “Choose File” button  2. Student select appropriate file to upload and confirm  3. Input Document title (required)  4. Input Document description (optional) |
| **Alternative flow** | [Failed to upload file from Student Upload Page / Website unresponsive]  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 |
| **Key components** | **Explanation** |
| **Name** | Student Upload Comment Page |
| **Actors** | Student |
| **Description** | Student use this page to send comment on upload files |
| **Goal** | Successfully send upload comments |
| **Pre-condition** | Successful Display Student Upload Comment Page |
| **Trigger** | Press “Send” button |
| **Post-condition** | Show success message |
| **Normal flow** | [Upload comment flow]  1. Student click on “Write your comment” field  2. Student inputs new comment (required)  3. Student presses “Send” button |
| **Alternative flow** | [Failed to send comment from Student Upload Comment 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 |



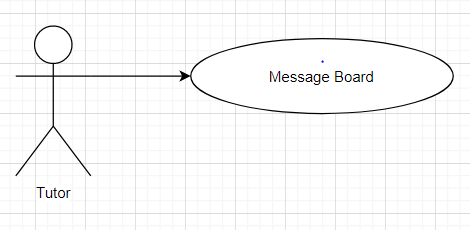
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Login Page |
| **Actors** | Tutor |
| **Description** | Login into the specific tutor manager page |
| **Goal** | Reach the login success page |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User presses “Enter” or “Login” button |
| **Post-condition** | Success page appears, user has been redirected to DashBoard page of Tutor |
| **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 |



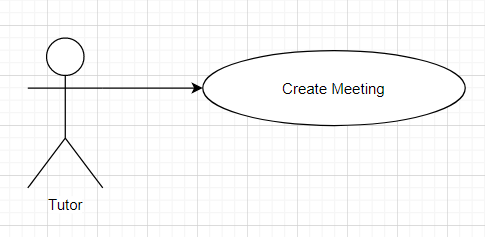
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor DashBoard |
| **Actors** | Tutor |
| **Description** | View & read tutor specific information on DashBoard |
| **Goal** | Reach the manager page of Tutor DashBoard & view tutor statistics information |
| **Pre-condition** | Display Tutor 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 Tutor |
| **Normal flow** | [Tutor logged in]  1. User reach Success page  2. User has been redirected to Student DashBoard page |
| **Alternative flow** | [Tutor 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 |



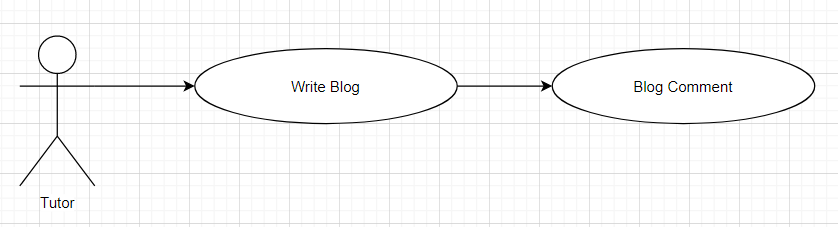
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Edit Profile Page |
| **Actors** | Tutor |
| **Description** | Input tutor data to update new information with the database |
| **Goal** | Update new user information into system database |
| **Pre-condition** | Successful Display Tutor Edit Profile Information Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message & update new user data in system database |
| **Normal flow** | 1. Tutor 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 Tutor Edit 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 to the Tutor Edit Profile Page again |



|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Message Board Page |
| **Actors** | Tutor |
| **Description** | Deliver tutor’s message across the system (classroom, group, private conversation) |
| **Goal** | Successfully send tutor’s message |
| **Pre-condition** | Successful Display Tutor Message Board Page |
| **Trigger** | Press “Enter” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Tutor input message content  2. Press “Enter” button  3. Successful message display  4. Redirected back to / Stay on Tutor Message Board Page |
| **Alternative flow** | [Failed to send tutor’s message from Tutor 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 Tutor Message Board Page |

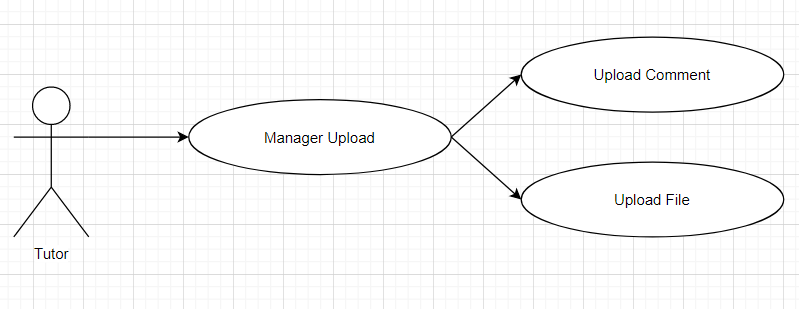


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Create Meeting Page |
| **Actors** | Tutor |
| **Description** | Perform create meeting function to create a new group meeting entry on school system’s schedule |
| **Goal** | Tutor successfully creates a new meeting |
| **Pre-condition** | Successful Display of Create Meeting Page |
| **Trigger** | Press “Create” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Tutor selects specific meeting date, environment, topic details for a new meeting  2. Click on meeting icon on the system calendar  3. Press “Create” button  4. “Created Successful” message display |
| **Alternative flow** | [Failed to create message from Create 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 |



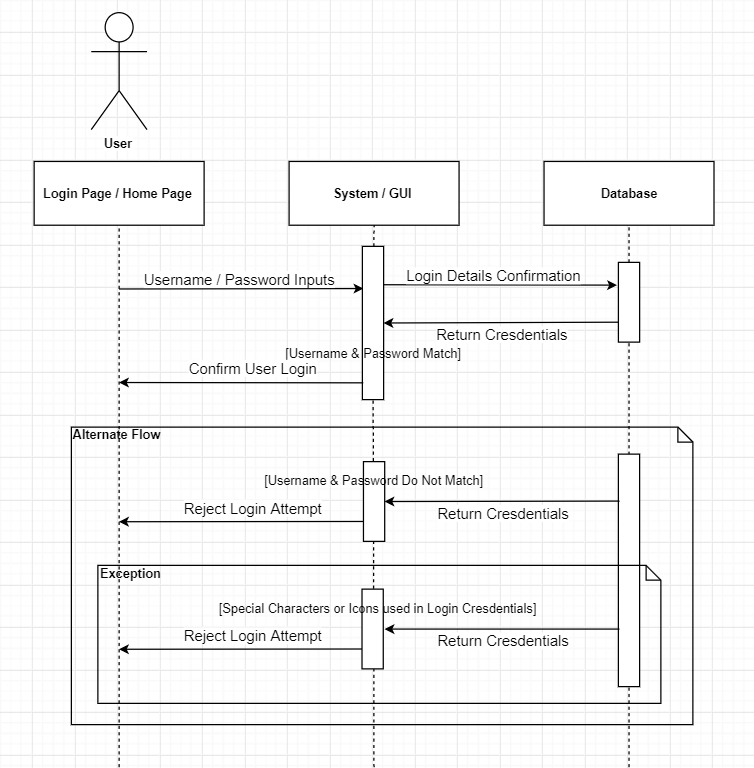
|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Write Blog Page |
| **Actors** | Tutor |
| **Description** | Create new blog post from tutor |
| **Goal** | Successfully create new blog post |
| **Pre-condition** | Successful Display Write Blog Page |
| **Trigger** |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

|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Blog Comment Page |
| **Actors** | Tutor |
| **Description** | Tutor uses this page to send comment on blog post |
| **Goal** | Successfully send comment on blog post |
| **Pre-condition** | Successful Display Blog Comment Page |
| **Trigger** | Press “Send” button |
| **Post-condition** | Show success message |
| **Normal flow** | 1. Tutor types in comment field  2. Press “Send” button  3. “Send Successful” message display |
| **Alternative flow** | [Failed to send comment from Tutor Blog Comment Page]  Show the error page  Repeat steps in normal flow |
| **Exception flow** | [Special characters used when writing the comment, Inappropriate content]  Show the error page  Return to the Student Write Blog Page |

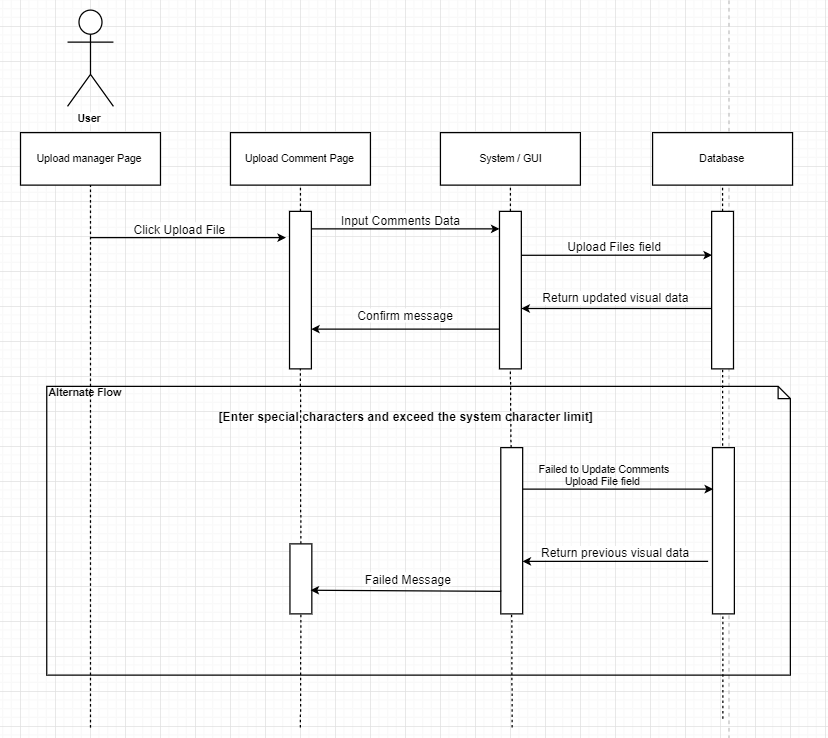


|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Tutor Upload Page |
| **Actors** | Tutor |
| **Description** | Upload tutor files |
| **Goal** | Successfully upload new files, update new data in system database |
| **Pre-condition** | Successful Display Student Upload Page |
| **Trigger** | Press “Confirm” button |
| **Post-condition** | Show success message |
| **Normal flow** | [Upload file flow]  1. Tutor click on “Choose File” button  2. Tutor select appropriate file to upload and confirm  3. Input Document title (required)  4. Input Document description (optional) |
| **Alternative flow** | [Failed to upload file from Tutor Upload Page / Website unresponsive]  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 |
| **Key components** | **Explanation** |
| **Name** | Tutor Upload Comment Page |
| **Actors** | Tutor |
| **Description** | Tutor use this page to send comment on upload files |
| **Goal** | Successfully send upload comments |
| **Pre-condition** | Successful Display Tutor Upload Comment Page |
| **Trigger** | Press “Send” button |
| **Post-condition** | Show success message |
| **Normal flow** | [Upload comment flow]  1. Tutor click on “Write your comment” field  2. Tutor inputs new comment (required)  3. Tutor presses “Send” button |
| **Alternative flow** | [Failed to send comment from Tutor Upload Comment 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 |

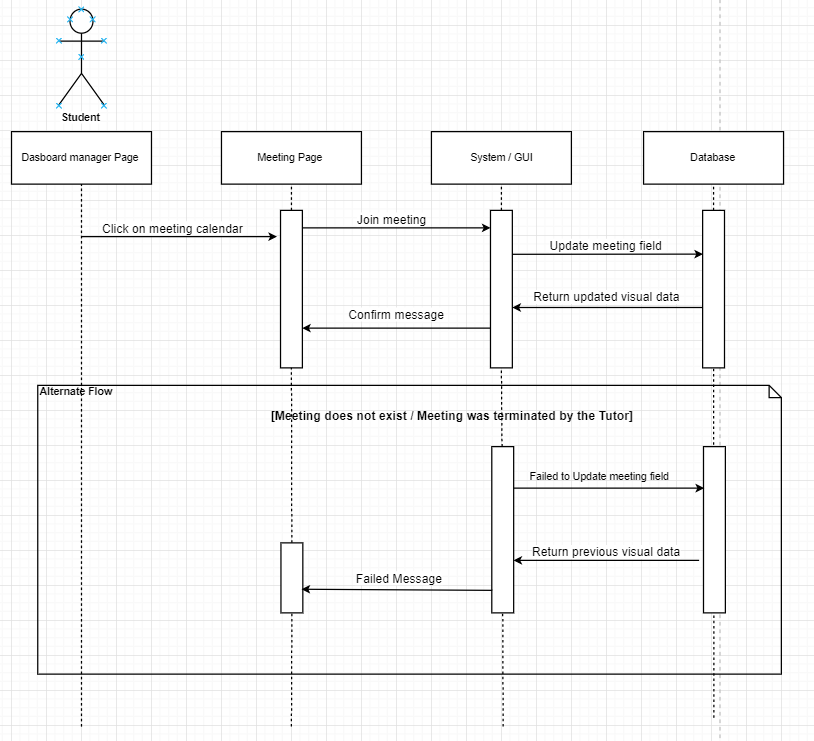
* **Login Sequence Diagram**



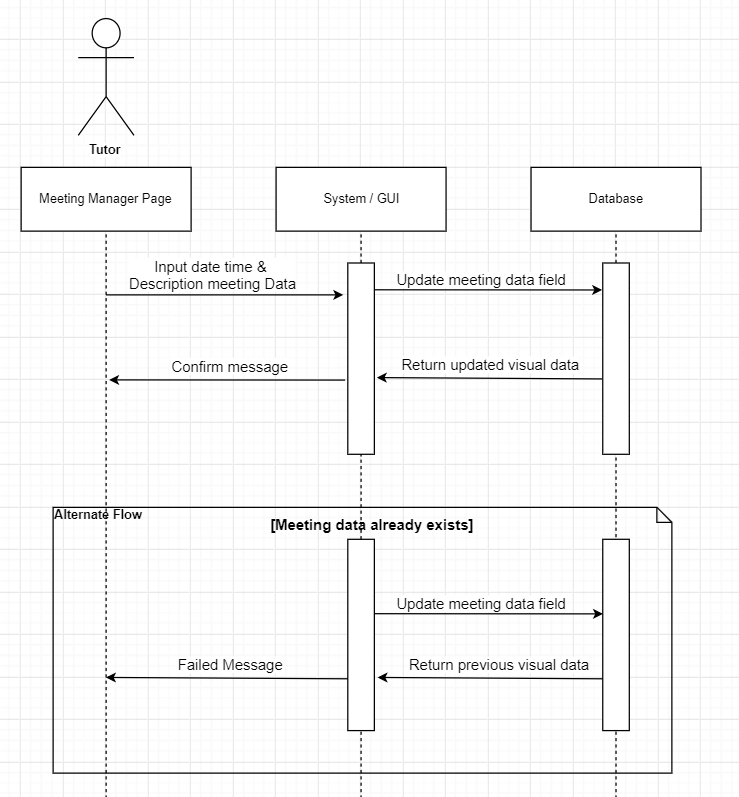
* **Upload Comment Sequence Diagram**



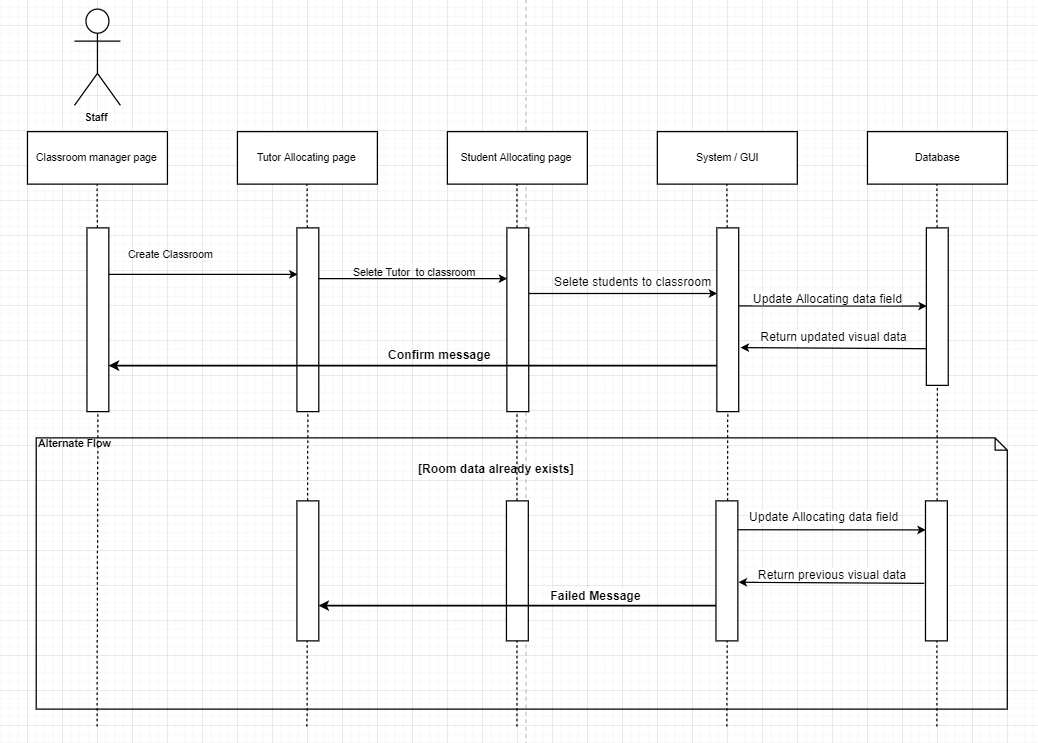
* **Student’s Join Meeting Sequence Diagram**



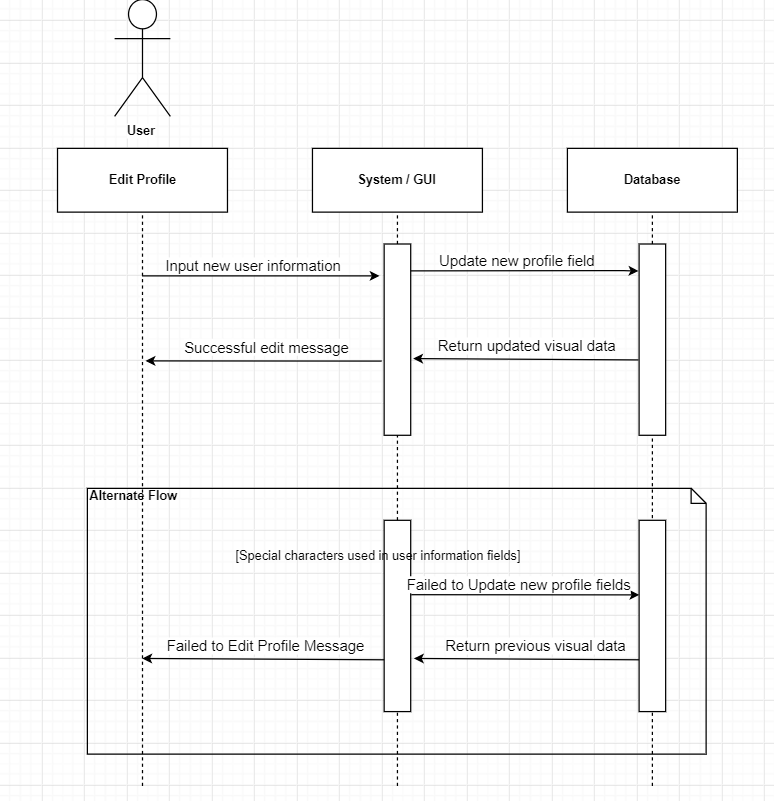
* **Tutor’s Create Meeting Sequence Diagram**



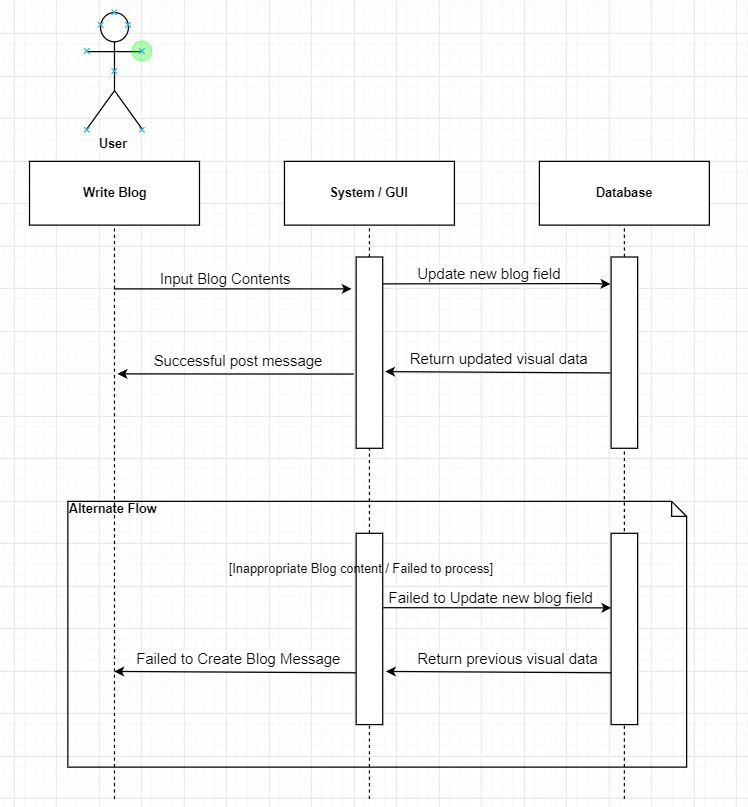
* **Staff’s Allocating Sequence Diagram**



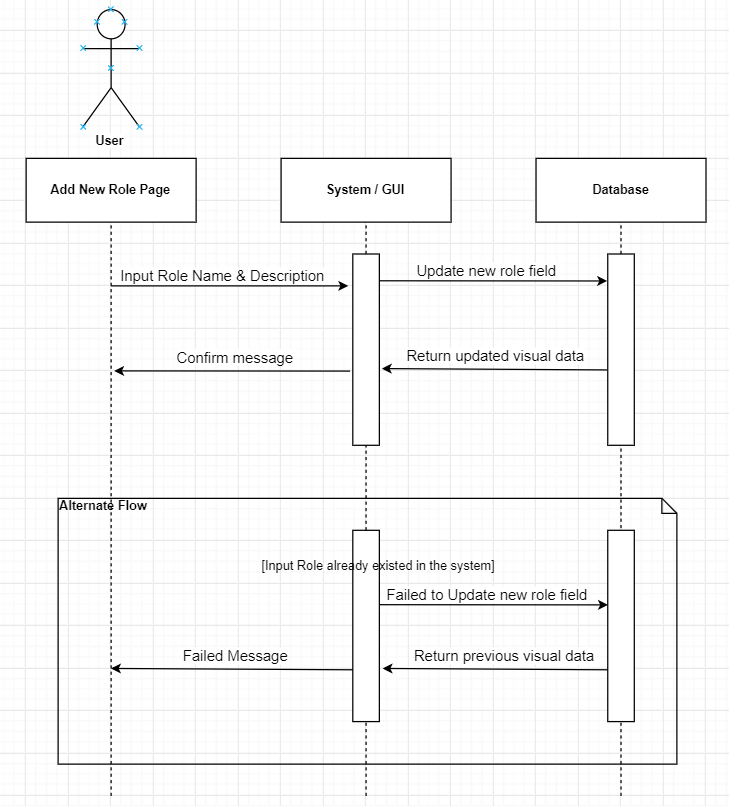
* **Edit Profile Sequence Diagram**



* **Write Blog Sequence Diagram**



* **Staff’s Add New Role Sequence Diagram**



# **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].

Phải vẽ hình giải thích dataflow của cái report, nhập dữ liệu như thế nào, nó chạy querry như nào, xuất ra như nào (có slide), nói về hệ thống MIS của message thôi

Chỉnh lại format 1.5 space, 12, calibri, căn tất cả cho dễ nhìn (cái thư 4 trên căn) . giải thích từng cái mockup

Cái tổng của use case ?

Product backlog file excel bỏ vô

Sequence diagram cho mỗi use case

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

chỉnh lại thêm thắt delete edit các thứ ...

tạo thêm trang meeting management – chuyển calendar thành dòng ngày giờ