

**Group 4 Report**

**Enterprise Web Software Development**

**Comp 1640**

Team members:

Nguyễn Thanh Phong

Trần Lâm Bảo Trân

Huỳnh Hoàng Nam

Đỗ Phú Thịnh

Nguyễn Hoàng Vỹ

Nhan Khánh Đình

Bùi Cao Nguyên

**Group report**

# Project Introduction

The eTutoring system is a web-based secure role-based application designed for a large university. The system aims to facilitate communication and documentation between students and their personal tutors. It includes features such as messaging, meeting arrangements, document uploads, blogging, and notification emails. The project follows Agile Scrum methodologies, and roles are assigned within the development team to ensure an organized and efficient workflow.

The project team consists of key roles including a Scrum Master who oversees Agile processes, a Product Owner acting as the liaison between the development team and the tutor (client), a Database Designer responsible for schema development and integration with the university’s MIS system, an Information Architect who structures system information, multiple Programmers developing system functionalities, a Web Designer ensuring responsive UI for various devices, and a Tester conducting rigorous testing for functionality, security, and usability.

The Agile Scrum workflow includes sprint planning where user stories and acceptance criteria are defined, daily stand-ups where progress and challenges are shared, sprint reviews for demonstrating completed features and gathering feedback, and sprint retrospectives for continuous improvement.

The system provides secure access control with defined user roles including students, personal tutors, authorized staff, and administrators. Key functionalities include student-personal tutor allocation, bulk allocation of up to 10 students, and email notifications for allocation changes. Messaging and meeting scheduling allow secure in-system communication and tracking of real and virtual meetings. Document management supports uploading, sharing, and commenting on documents, while the blogging feature enables students and tutors to maintain academic blogs.

The notification system ensures that email notifications are only sent for event updates with no direct communication via email. Personal dashboards are provided for students and tutors, summarizing interactions, and authorized staff can access all dashboards with sorting and filtering functionalities. Data is integrated securely with the university’s MIS system to maintain student and staff information. The interface is designed to be responsive across mobile, tablet, and desktop devices.

The technology stack includes frontend frameworks such as React.js or Vue.js, backend technologies like Node.js, Django, or Flask, and databases such as PostgreSQL or MySQL. Authentication mechanisms include OAuth or SAML, messaging is implemented with WebSockets, and hosting is managed on cloud platforms like AWS, Azure, or Google Cloud.

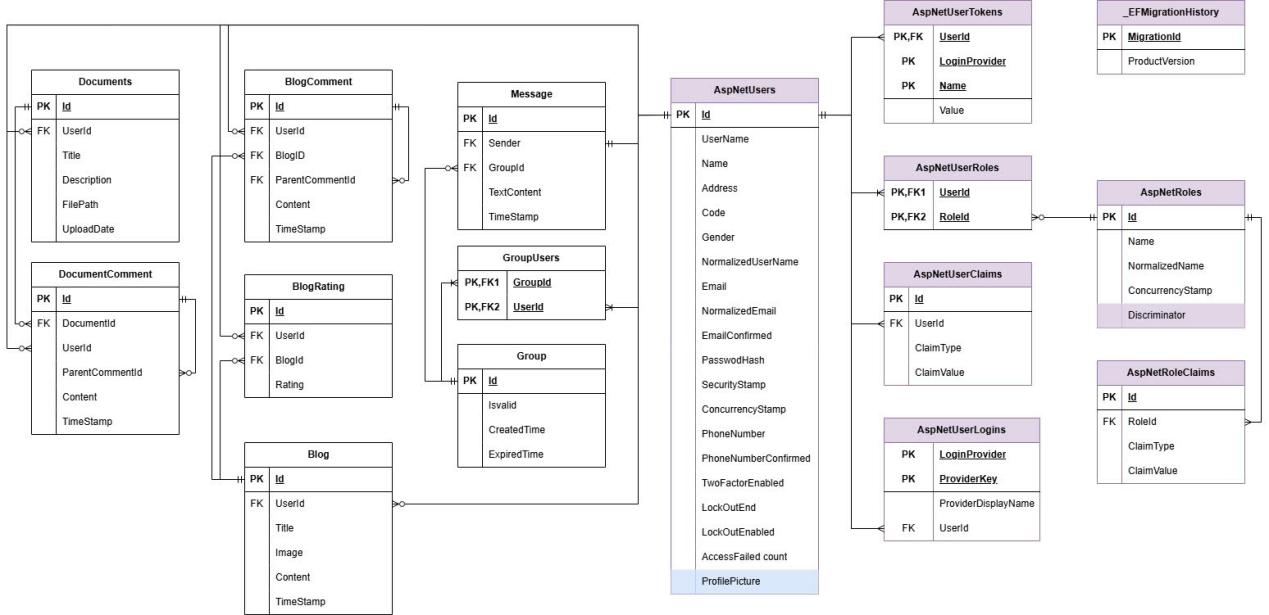
The development process is structured into milestones, beginning with requirement gathering and UI wireframing, followed by database design, API development, authentication, and dashboard features. Subsequent sprints focus on messaging, meeting functionalities, document uploads, blogging, and testing, culminating in deployment within twelve weeks.

Testing and deployment strategies include unit testing of individual components, integration testing to verify module interactions, user acceptance testing (UAT) for usability validation, security testing for role-based access control and data protection, and cloud deployment for secure hosting.

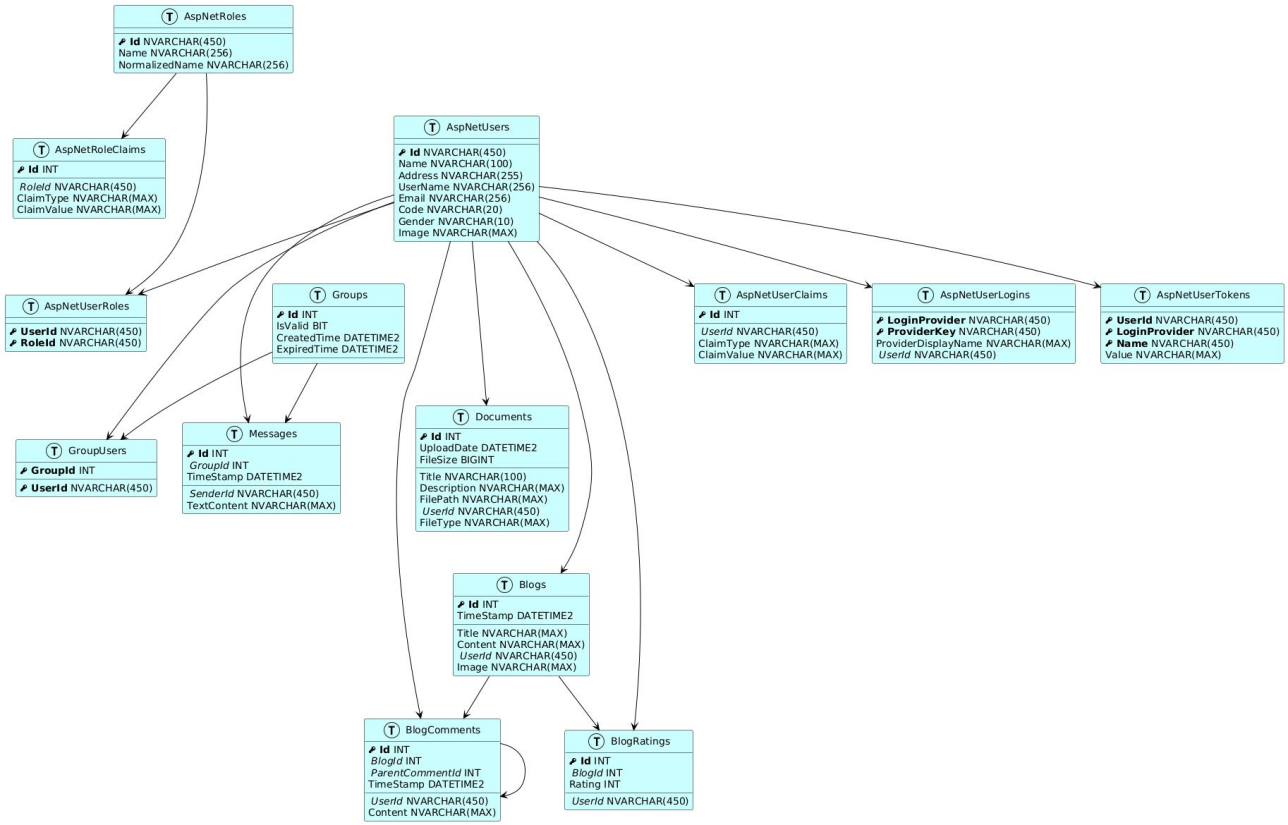
Documentation and reporting are essential, including recorded meeting minutes for all Scrum ceremonies, detailed code documentation, weekly progress reports, and a final project report summarizing individual contributions and outcomes.

In conclusion, the eTutoring system is designed to enhance communication and organization between students and tutors while ensuring a secure and efficient platform. By adopting Agile Scrum practices, the project ensures continuous improvement and responsiveness to requirements. The development team collaborates to deliver a robust and scalable solution that meets university needs.

# ERD diagram



# Relation Schema



# Data dictionary

**AspNetUsers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | nvarchar(450) | PK | Primary key identifier |
| Name | nvarchar(100) | NOT NULL | User's full name |
| Address | nvarchar(255) | NULLABLE | User's address |
| UserName | nvarchar(256) | NULLABLE | Username for login |
| NormalizedUserName | nvarchar(256) | NULLABLE | Normalized username for searching |
| Email | nvarchar(256) | NULLABLE | User's email address |
| NormalizedEmail | nvarchar(256) | NULLABLE | Normalized email for searching |
| EmailConfirmed | bit | NOT NULL | Flag indicating if email is confirmed |
| PasswordHash | nvarchar(max) | NULLABLE | Hashed password |
| SecurityStamp | nvarchar(max) | NULLABLE | Security stamp for authentication |
| ConcurrencyStamp | nvarchar(max) | NULLABLE | Concurrency stamp for optimistic concurrency |
| PhoneNumber | nvarchar(max) | NULLABLE | User's phone number |
| PhoneNumberConfirmed | bit | NOT NULL | Flag indicating if phone is confirmed |
| TwoFactorEnabled | bit | NOT NULL | Flag for two-factor authentication |
| LockoutEnd | datetimeoffset(7) | NULLABLE | End time for account lockout |
| LockoutEnabled | bit | NOT NULL | Flag indicating if lockout is enabled |
| AccessFailedCount | int | NOT NULL | Count of failed access attempts |
| Code | nvarchar(20) | NOT NULL | User code identifier |
| Gender | nvarchar(10) | NOT NULL | User's gender |
| Image | nvarchar(max) | NULLABLE | Path to user's profile image |

**AspNetRoles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | nvarchar(450) | PK | Primary key identifier |
| Name | nvarchar(256) | NULLABLE | Role name |
| NormalizedName | nvarchar(256) | NULLABLE | Normalized role name for searching |
| ConcurrencyStamp | nvarchar(max) | NULLABLE | Concurrency stamp for optimistic concurrency |

**AspNetUserRoles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| UserId | nvarchar(450) | PK, FK | User ID reference to AspNetUsers |
| RoleId | nvarchar(450) | PK, FK | Role ID reference to AspNetRoles |

**Groups**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| IsValid | bit | NOT NULL, DEFAULT(1) | Flag indicating if group is valid |
| CreatedTime | datetime2(7) | NULLABLE | Time when group was created |
| ExpiredTime | datetime2(7) | NULLABLE | Time when group expires |

**GroupUsers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| GroupId | int | PK, FK | Group ID reference to Groups |
| UserId | nvarchar(450) | PK, FK | User ID reference to AspNetUsers |

**Messages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| SenderId | nvarchar(450) | NOT NULL, FK | Sender ID reference to AspNetUsers |
| GroupId | int | NOT NULL, FK | Group ID reference to Groups |
| TextContent | nvarchar(max) | NOT NULL | Message content |
| FileCount | int | NOT NULL, DEFAULT(0) | Count of attached files |
| TimeStamp | datetime2(7) | NOT NULL | Time when message was sent |

**Blogs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| Title | nvarchar(max) | NOT NULL | Blog title |
| Content | nvarchar(max) | NOT NULL | Blog content |
| TimeStamp | datetime2(7) | NOT NULL | Time when blog was created |
| UserId | nvarchar(450) | NOT NULL, FK | User ID reference to AspNetUsers |
| Image | nvarchar(max) | NULLABLE | Path to blog image |

**BlogComments**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| BlogId | int | NOT NULL, FK | Blog ID reference to Blogs |
| UserId | nvarchar(450) | NOT NULL, FK | User ID reference to AspNetUsers |
| ParentCommentId | int | NULLABLE, FK | Parent comment ID for nested comments |
| Content | nvarchar(max) | NOT NULL | Comment content |
| TimeStamp | datetime2(7) | NOT NULL | Time when comment was created |

**BlogRatings**

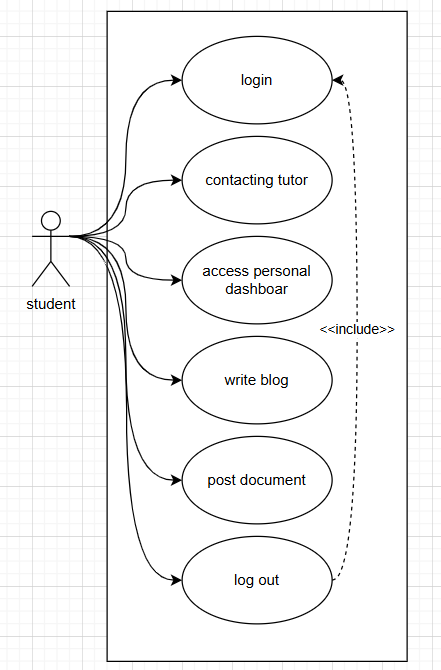
|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| UserId | nvarchar(450) | NOT NULL, FK | User ID reference to AspNetUsers |
| BlogId | int | NOT NULL, FK | Blog ID reference to Blogs |
| Rating | int | NOT NULL | Rating value |

**Documents**

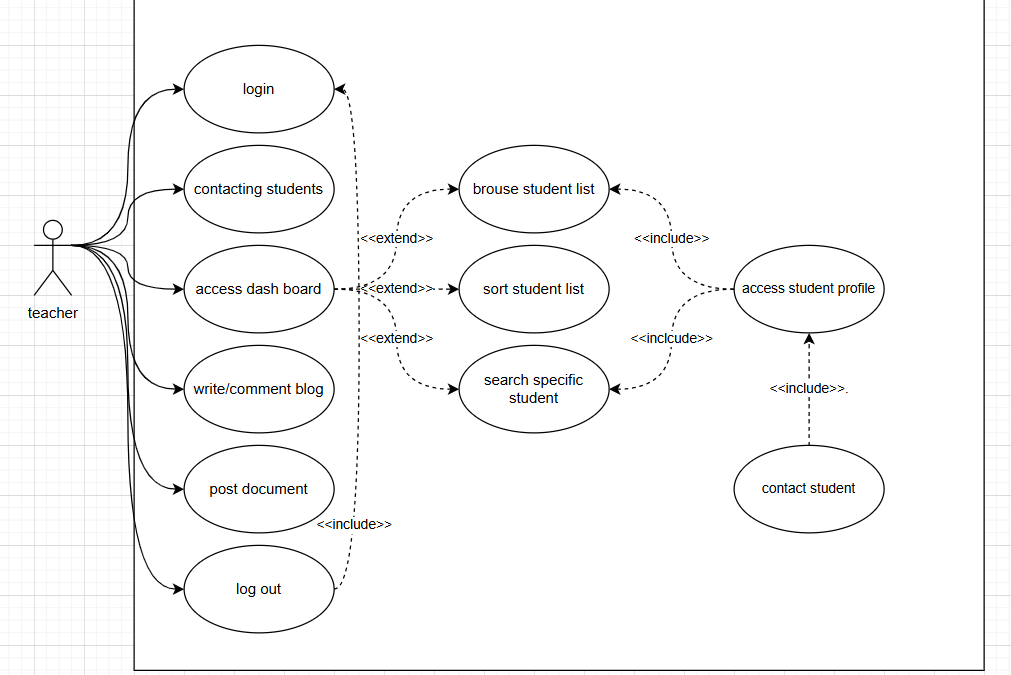
|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | int | PK, IDENTITY | Primary key identifier |
| Title | nvarchar(100) | NOT NULL | Document title |
| Description | nvarchar(max) | NOT NULL | Document description |
| FilePath | nvarchar(max) | NOT NULL | Path to document file |
| UserId | nvarchar(450) | NOT NULL, FK | User ID reference to AspNetUsers |
| UploadDate | datetime2(7) | NOT NULL | Date when document was uploaded |
| FileType | nvarchar(max) | NOT NULL | Type of document file |
| FileSize | bigint | NOT NULL | Size of document file |

# Use-case diagram

Use-case of Student:



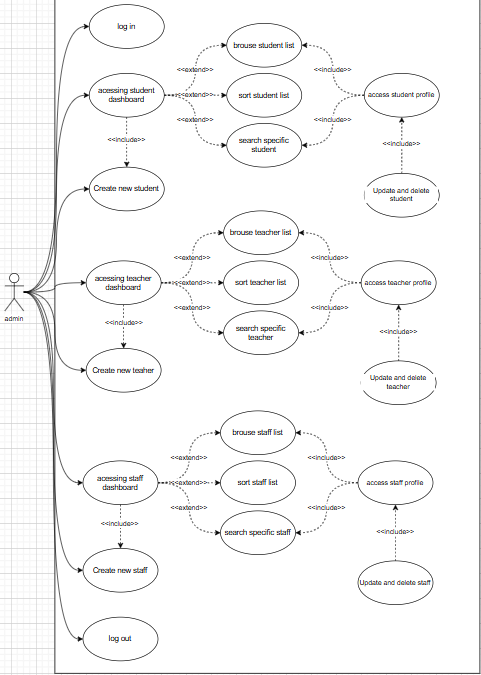
Use-case of Teacher



Use-case of Staff

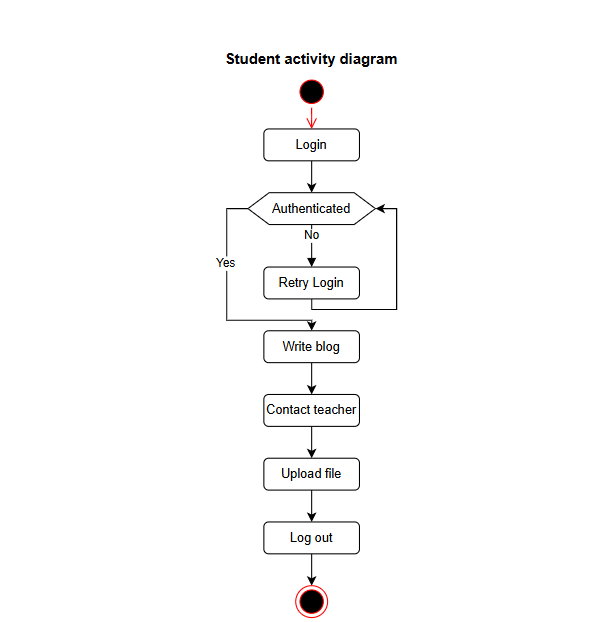


Use-case of Admin

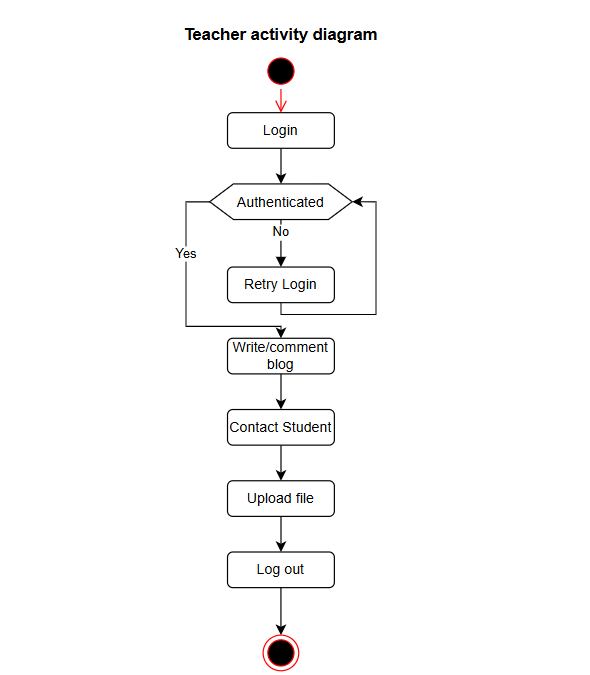


# Activity diagram

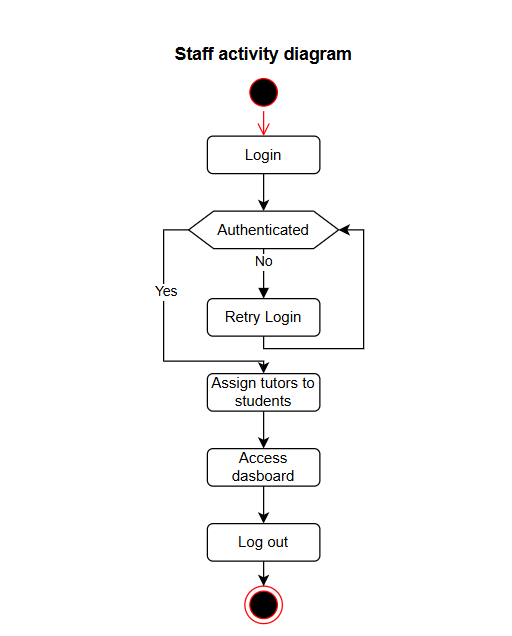
Activity diagram of student :



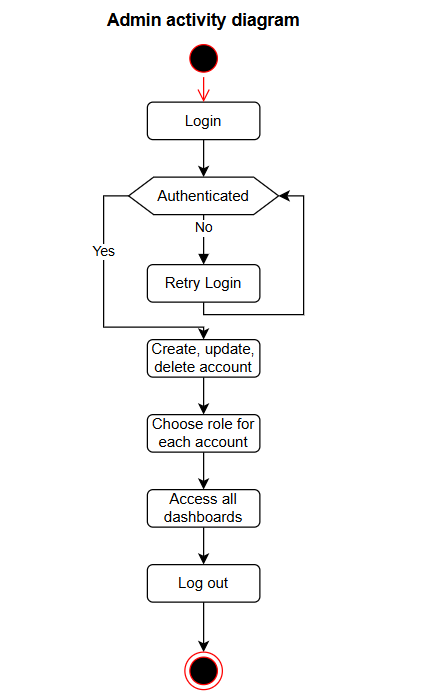
Activity diagram of teacher:



Activity diagram of staff :

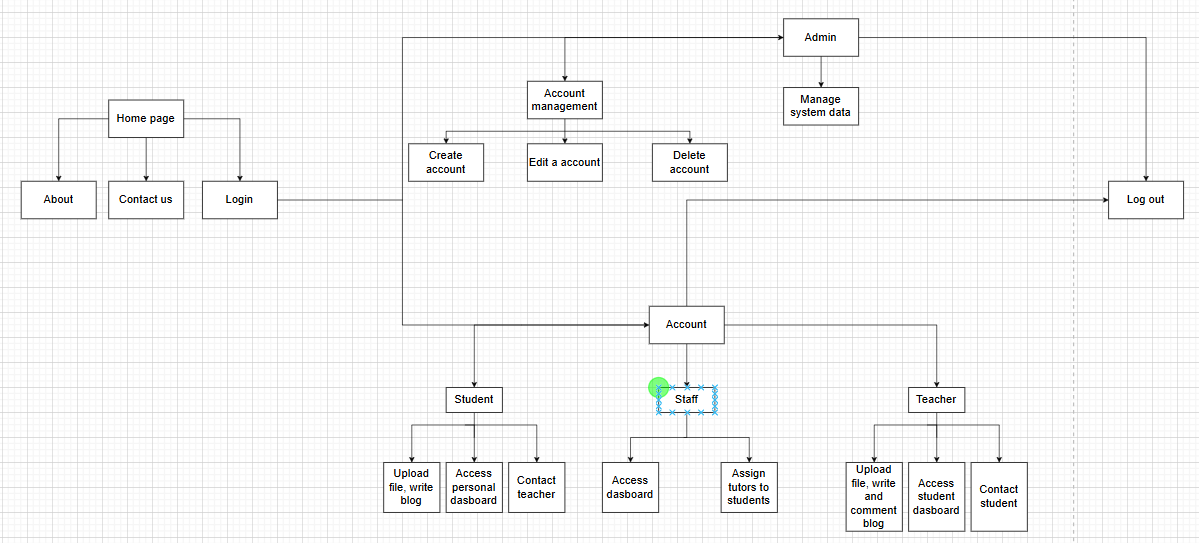


Activity diagram of Admin:

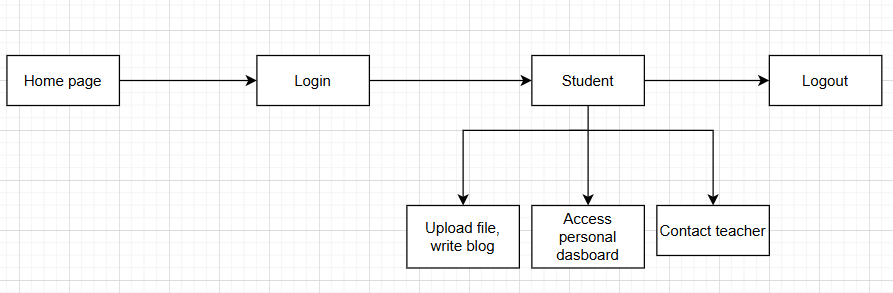


# Sitemap

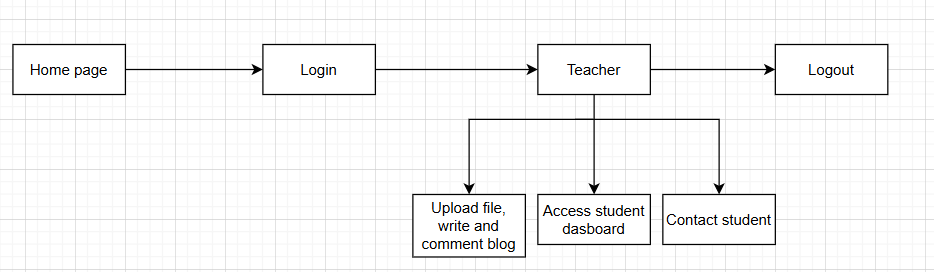
An access-controlled educational system design appears through this sitemap. The homepage contains standard page links that include About, Contact Us and Login. After users authenticate successfully they gain access to manage accounts along with managing system data as part of their admin privileges. The system distinguifies between Users who fall under the roles of Students, Teachers and Staff. After authenticating their login users can upload files, write blogs and access their dashboard while contacting teachers. Blog management together with the ability to monitor student dashboards and send messages to students remains part of the teaching features. Staff personnel execute administrative work that includes tutor allocation for student cases. Compatible with all users is the logout feature featuring in the platform interface.



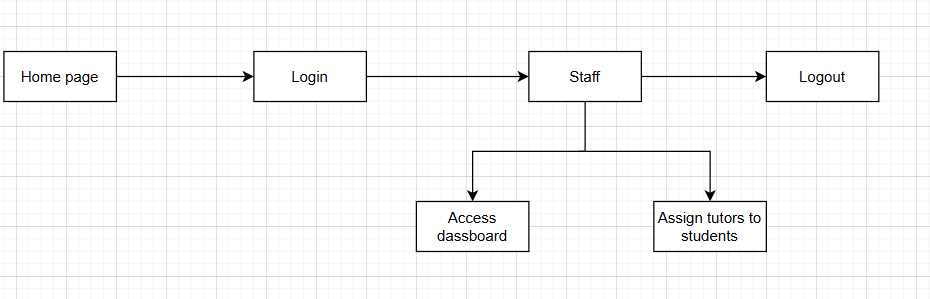
Sitemap of Student:



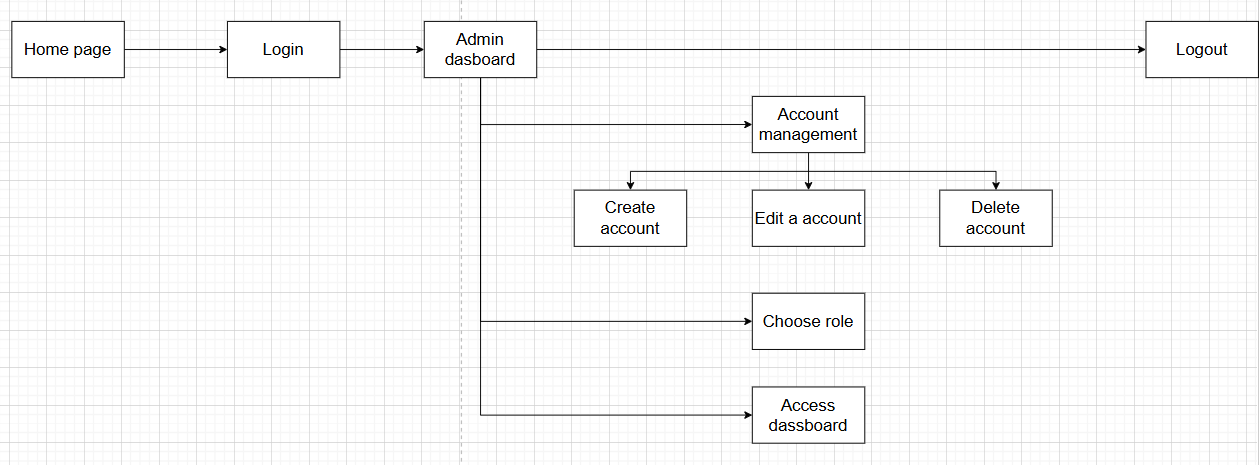
Sitemap of Teacher:



Sitemap of Staff:

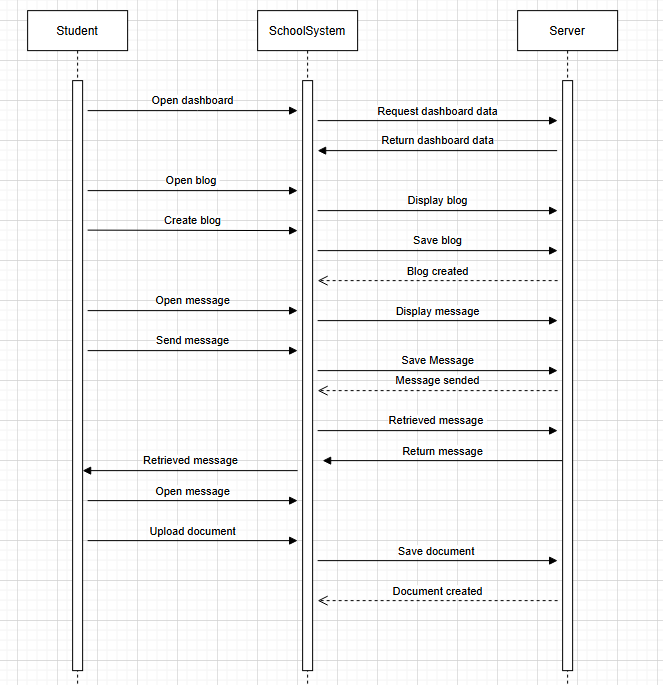


Sitemap of Admin:

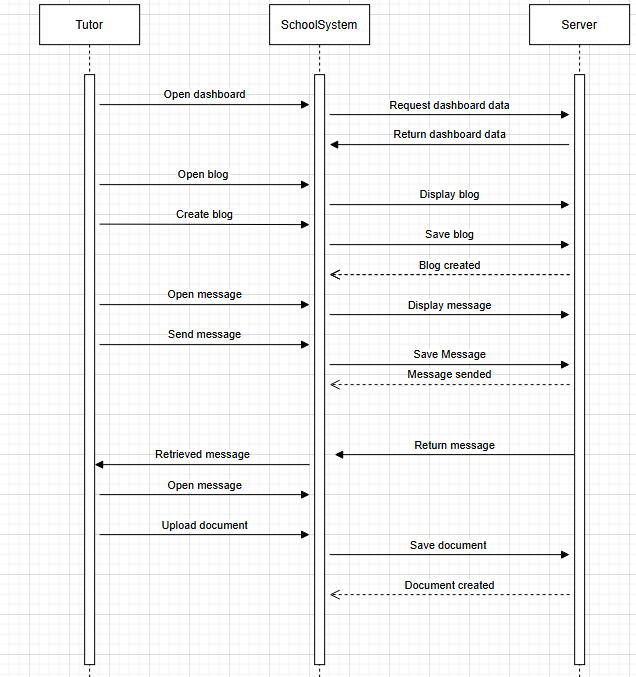


# Sequence diagram

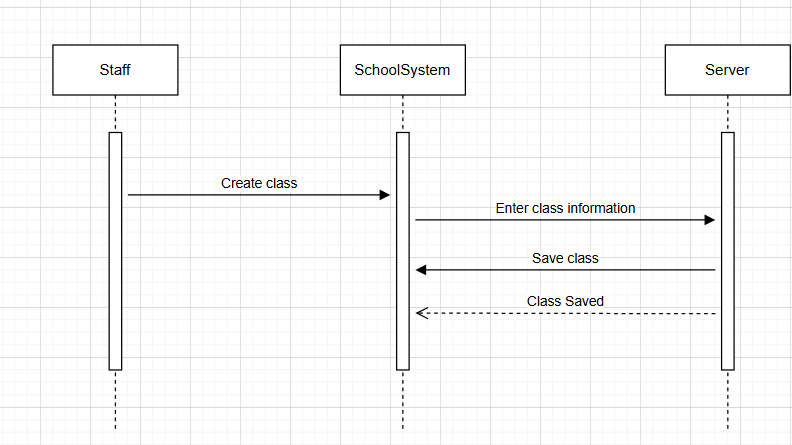
Student sequence diagram:



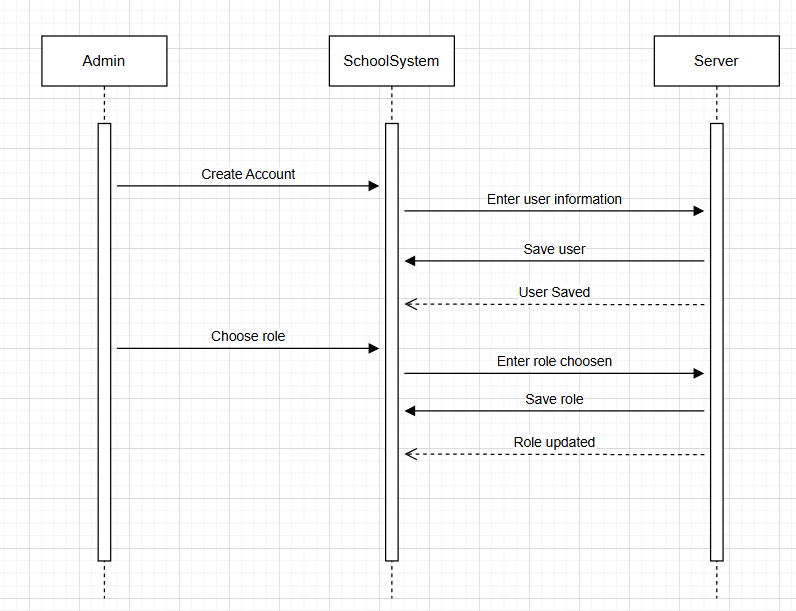
Tutor sequence diagram:



Staff sequence diagram:



Admin sequence diagram:



# Testing

# Agile Method: Sprint Planning, Sprint Backlogs, Schedule, Daily Meetings, Sprint Review & Retrospective Meetings, Burndown Charts

# 10.1 Role in sprint:

|  |  |
| --- | --- |
| Full name | Role |
| Nguyễn Thanh Phong | Backend developer, Scrum master, Product owner |
| Huỳnh Hoàng Nam | Database designer, Co designer of the use case diagram, Backend developer |
| Đỗ Phú Thịnh | Backend developer, Use case designer |
| Nguyễn Hoàng Vỹ | Frontend developer, Website design (Figma) |
| Trần Lâm Bảo Trân | Frontend developer, Website design (Figma) |
| Bùi Cao Nguyên | Backend developer, Tester |
| Nhan Khánh Đình | Backend developer, JavaScript for Document Features, Relational Schema Design, Responsive Design |

**Nguyen Thanh Phong (Backend developer, Scrum master, Product owner)**

As Product Owner & Scrum Master:

The software development activities followed Scrum methodology to provide efficient project guidance throughout the entire development process.

The development team obtained focus and alignment with project goals through the development of Sprints that originated from the Product Backlog together with task prioritization.

The team received project domain knowledge from me while I provided essential contributions to form the Product Owner’s vision.

A leader conducted all Scrum events starting with Sprint Planning followed by Daily Stand-ups and Sprint Reviews and Sprint Retrospectives to maintain effective team cohesion and project progress assessment.

I attended all Agile Scrum meetings to check the product status of each completed sprint cycle.

As Backend Developer and Analyst:

The development of fundamental backend functionalities supported the Student and Tutor use of Dashboard features.

I added sort and filter tools to the Tutor Dashboard to allow better student assignment data viewing abilities.

The project implementation included creation of system-level diagrams which included Activity Diagram as well as Sitemap and Sequence Diagram.

The team author compiled the entire project report documenting deliverables alongside implementation details and project progress summary.

**Huynh Hoang Nam ( Database designer, Co designer of the use case diagram, Backend developer)**

The project depended on him to design its primary databases. He took full responsibility for designing the system database which required him to produce a detailed Entity Relationship Diagram (ERD) that showed data structures and application relationships properly.

Apart from his database work he helped build the Use Case Diagram by specifying system functionality organization through user interactions.

As a Backend Developer he delivered all functions of the Group module by developing its Group User management and Messaging capabilities. He created necessary database models and controllers and database tables with an emphasis on producing strong and efficient backend logic.

**Do Phu Thinh (Backend developer, Use case designer)**

He was essential in designing the Admin Use Case Diagram to identify administrative user essential functionalities through proper structure design.

He was responsible for writing code in the backend to create the account management system with user authentication and role-based access. He wrote all backend programming logic for the Blog platform which contained abilities to make blogs and keep them current and to erase them together with commenting choices for both document and blog pages.

Through frontend work he constructed HTML page frameworks for both administrative sections and blog pages as well as implemented interactive JavaScript-based comment functions.

**Nguyen Hoang Vy (Frontend developer, Website design)**

The developer developed every interface section that students needed to use in the website. The student-facing sections of the website were developed by him resulting in the creation of the Home Page and Blog Page and complete functions within the Student Dashboard that include chat capabilities and document sharing features.

Through Figma he imposed design consistency on the platform while producing interface prototypes as a UI/UX designer. One of his tasks involved creating email templates which maintained professional presentation alongside brand identity.

His implementation of responsive design made the student interface run without issues on computers and tablets and smartphones simultaneously.

**Tran Lam Bao Tran (Frontend developer, Website design)**

The developer created multiple essential user interface features dedicated to different system roles. The Login Page alongside Admin Dashboard, Register User, User Listing, Role Creation and Listing, and Tutor Dashboard were among her system development initiatives.

Through Figma interface design work she helped maintain accessible user interfaces which also served users seamlessly. The system experienced seamless functionality across multiple devices and screen resolutions because she established responsive design criteria on every page.

**Bui Cao Nguyen (Backend developer, Tester)**

His work concentrated on developing major functionality behind the email notification platform. The development of Email Controller together with EmailService enabled the system to deliver various notifications such as event alerts for registration confirmations.

He fulfilled test duties by examining every backend controller professionally to ascertain their correct operation in every possible situation.

**Nhan Khanh Dinh (Backend developer, JavaScript for Document Features, Relational Schema Design, Responsive Design)**

**Documents Functionality (Backend):**

The development of backend features for document management included all aspects such as:

Listing documents

Viewing document details

Uploading files

Editing document listings

Downloading files

Updating files

Deleting files

**JavaScript for Document Features:**

The system includes JavaScript-enabled interactive elements which enhance document-related user interactions for the document section.

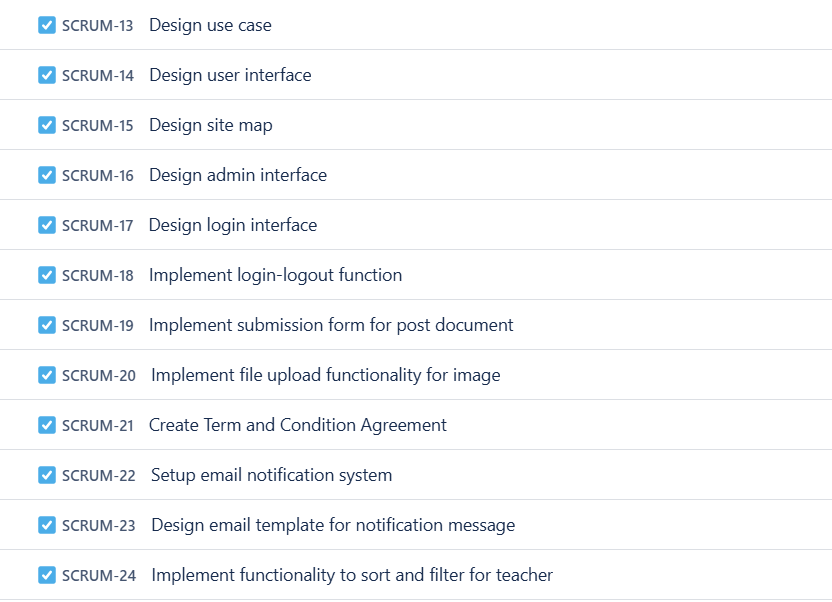
**Relational Schema Design:**

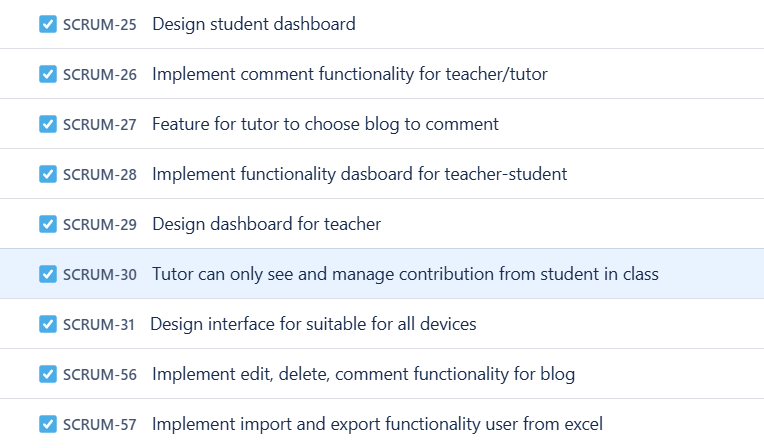
I designed a relational database schema for supporting the document module structure along with its functionality.

**Responsive Design (Student Dashboard):**

Together with the frontend team I worked on making sure the student dashboard especially its document section displayed correctly on every device type for responsive performance.

# 10.2 Product Backlog



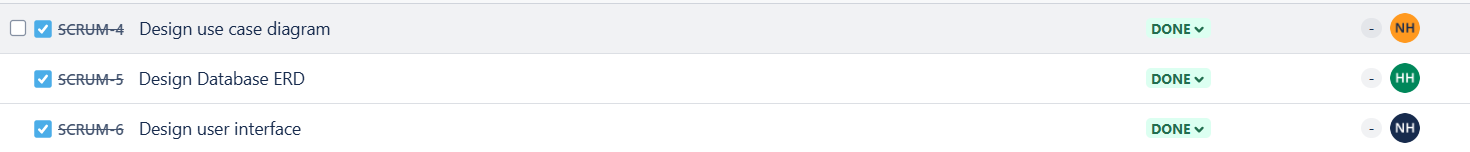


# 10.3 Sprint planning :

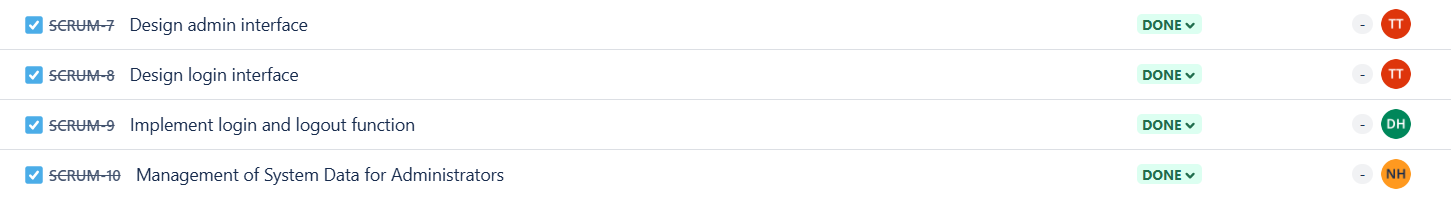
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Role | Duration | Timescale |
| Sprint Planning - Sprint 1 | Scrum Master | 1-2 | Week 1 |
| During Sprint 1 from January 26th to February 1st the team designs the use case diagram and database ERD as well as the user interface to create a base for future development stages. | Product Owner, Development Team | 1-2 | Week 1 |
| Sprint Planning - Sprint 2 | Scrum Master | 1-2 | Week 2-3 |
| Development during this phase focuses on creating the use case diagram and database ERD together with developing user interface design which forms a basis for upcoming phases. | Product Owner, Development Team | 1-2 | Week 2-3 |
| Sprint Planning - Sprint 3 | Scrum Master | 1-2 | Week 4-5-6 |
| This sprints concentrates on user-system function enhancement by developing interaction patterns through site maps and activity diagrams and by establishing capabilities for file submission with image uploading and setting terms of service and email alerts as well as giving users the ability to create blog content. | Product Owner, Development Team | 1-2 | Week 3 |
| Sprint Planning - Sprint 4 | Scrum Master | 1-2 | Week 7-8-9 |
| The upcoming sprint emphasizes dashboard creation for teachers and students together with blog control features along with adaptive design elements and sorting/filtering and user data import/export capabilities to improve the educational platform system usability. | Product Owner, Development Team | 1-2 | Week 7-8-9 |

# 10.4 Sprint backlog

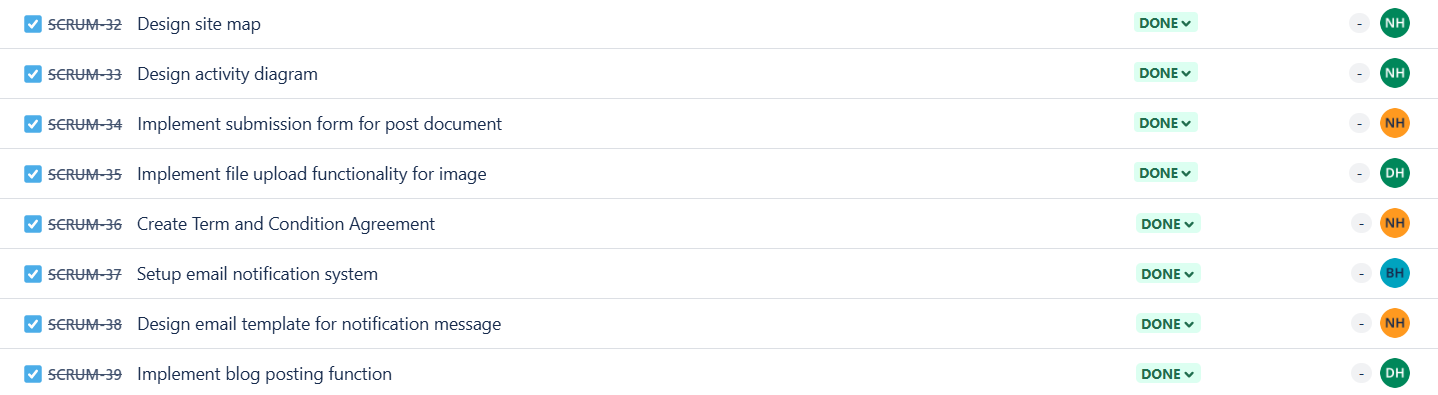
Sprint 1 :



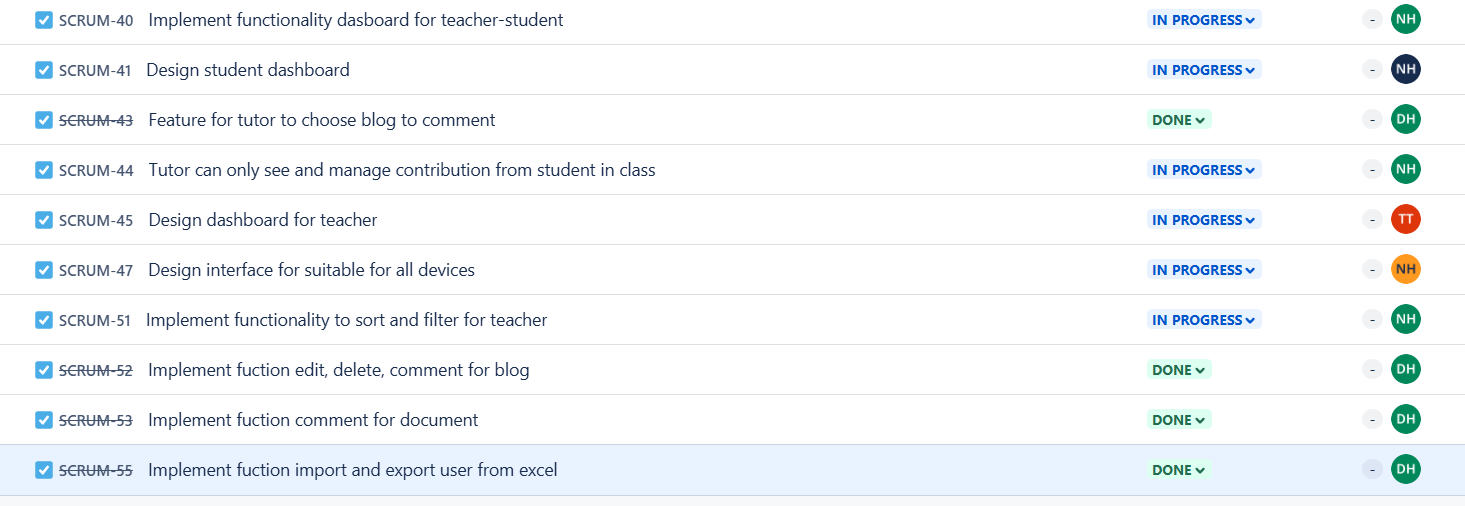
Sprint 2 :



Sprint 3 :

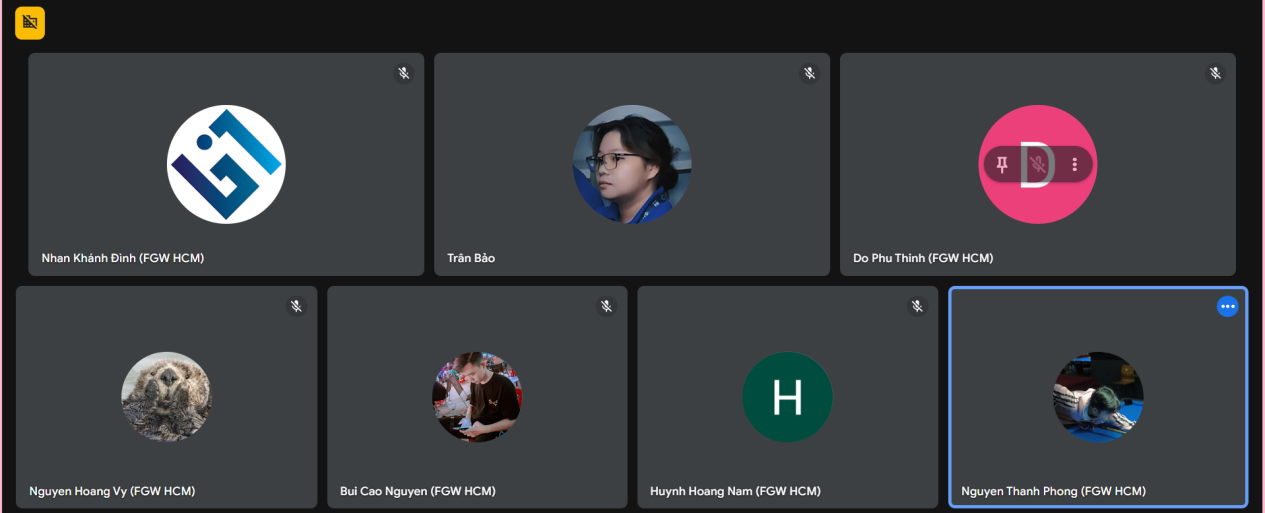


Sprint 4 :



# 10.5 Daily Standup

The group conducts online and offline meetings as part of their daily stand-up. Workers at meetings will frequently discuss accomplished jobs together with ongoing tasks as well as their work-related challenges. The duration of every scheduled remote meeting typically stays at 15 minutes.



# 10.6 Sprint review

Sprint review - Sprint 1:

The sprint successfully executed every intended goal to provide essential fundamental elements for project development. The three main tasks were finalized by the team.

Use Case Diagram – This diagram converted system requirements into user-task interactions which demonstrated how users would interact with the core features. The established reference serves as excellent guidance for developers.

Database ERD – A final version of the Entity-Relationship Diagram with a clear structure designed the database model connections to create an effective database structure.

User Interface Design – The first user interface design applied usability thinking to create mockups that followed project objectives before frontend development started.

Sprint review - Sprint 2

Sprint 2 added crucial authentication features together with administrative control interfaces for the COMP1640 system. All four of the planned issues were successfully implemented by the team which created essential user access and system management components.

Completed Work

SCRUM-7: Design Admin Interface

The team designed an interface for administrative system management which served as a clean functional solution for management needs.

The system development included role-based access controls as part of the design framework.

Reviewers from all stakeholder groups approved the design through one round of prototype review during iteration development.

SCRUM-8: Design Login Interface

The development of responsive login pages supported brand guidelines.

The system includes simple error notifications together with validation feedback mechanisms.

Five company employees participated in testing the system usability during our research phase.

SCRUM-9: Implement Login/Logout Function

The system adopts JWT tokens for its secure authentication process.

Established session management system

Integrated with backend user database

SCRUM-10: Admin Data Management

The system received CRUD functionalities to administer system data.

Implemented basic audit logging functionality

Data export capabilities through CSV format were developed in the system.

Sprint review : Sprint 3

The development team successfully improved document management capabilities as well as their notification system and basic blogging features after completion of Sprint 3. The team delivered seven out of eight planned issues except for site map design that remained pending because requirements changed.

SCRUM-32: Site Map Design

Finalized comprehensive navigation structure

The development follows the newly modified set of features.

Integrated with existing UI components

SCRUM-33: Activity Diagram Design

The team used visualized business process diagrams to establish better development directions.

The stakeholders approved the system during the first review session.

SCRUM-34: Document Submission Form

Implemented responsive form with validation

The system contains an integration of backend document storage functionality.

SCRUM-35: Image Upload Functionality

A secure file management system with size constraints (up to 2MB) and image authenticity checks runs throughout the application.

Added preview functionality before submission

SCRUM-36: Terms & Conditions Agreement

Created legally-reviewed template

Implemented mandatory acceptance flow

SCRUM-37: Email Notification System

Configured SMTP service integration

Established event-based triggering system

SCRUM-38: Email Template Design

Developed 5 responsive email templates

Incorporated branding guidelines

SCRUM-39: Blog Posting Function

Built rich text editor integration

Implemented draft autosave feature

Sprint review : Sprint 4

Sprint 4 delivered its full dashboard suite for teachers and students as designed with a total of eight built user stories. The implementation of full responsive design and every specified acceptance criterion was completed by the team.

SCRUM-40: Teacher-Student Dashboard Functionality

Implemented dual-view dashboard system

Established secure role-based access controls

SCRUM-41: Student Dashboard Design

Created intuitive progress tracking interface

Included submission status visualization

SCRUM-43: Tutor Blog Selection

Developed blog filtering by course/student

Implemented comment assignment system

SCRUM-44: Class Contribution Management

Built class-specific activity views

Added participation metrics

SCRUM-45: Teacher Dashboard

Designed comprehensive grading interface

Integrated deadline management

SCRUM-47: Responsive Design

Achieved full mobile compatibility

Passed WCAG 2.1 AA accessibility standards

SCRUM-51: Sorting/Filtering

Implemented multi-criteria filtering

Added saveable view presets

SCRUM-52/53: Content Management

Developed rich text editing tools

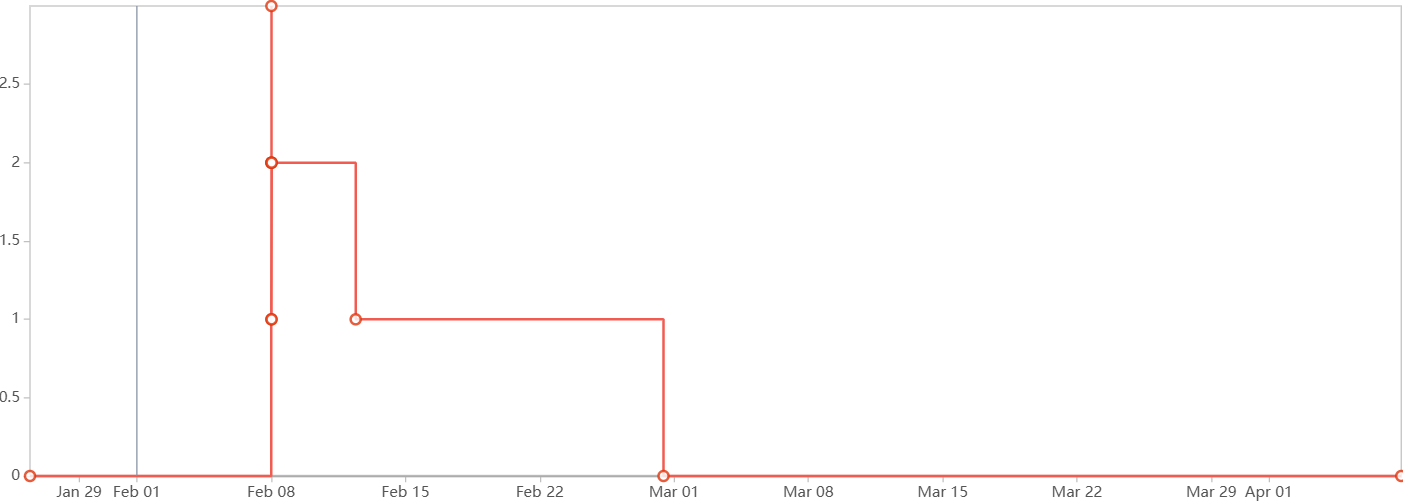
Built threaded comment system

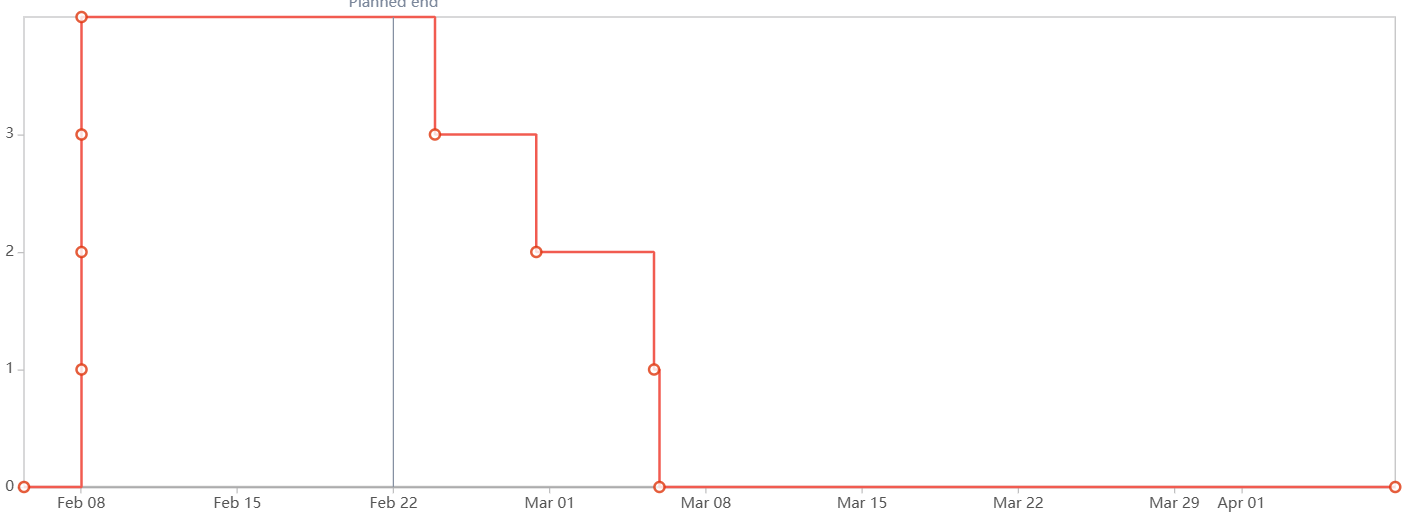
SCRUM-55: User Import/Export

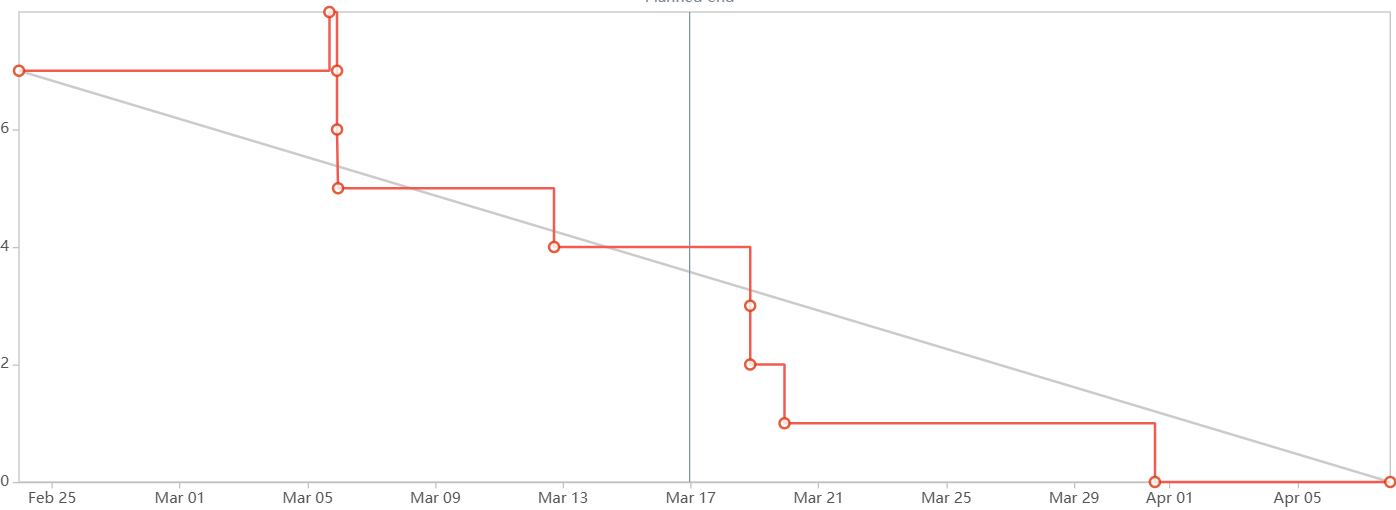
Created Excel data handler

Implemented bulk operation controls

# 10.7 Burndown chart

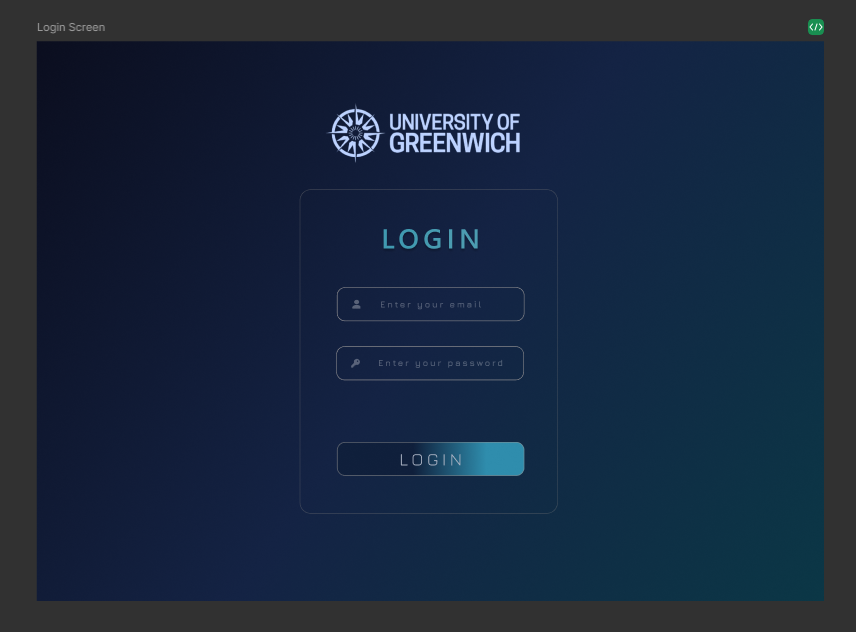






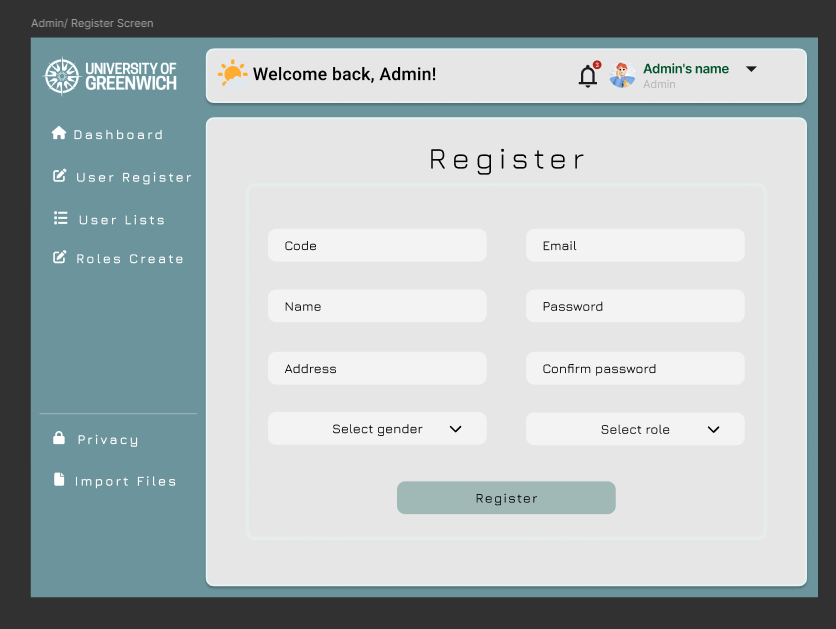
# Wireframe

**Login page:**

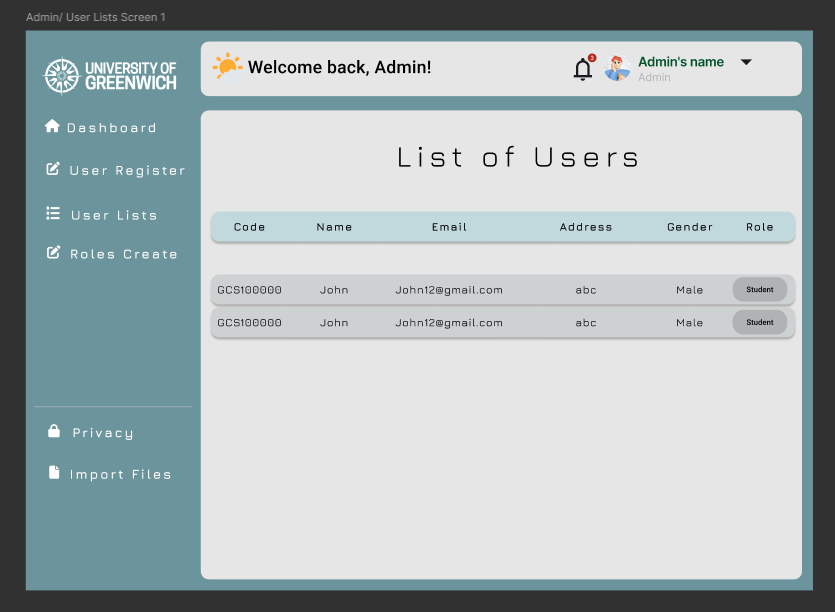


**Admin Page:**

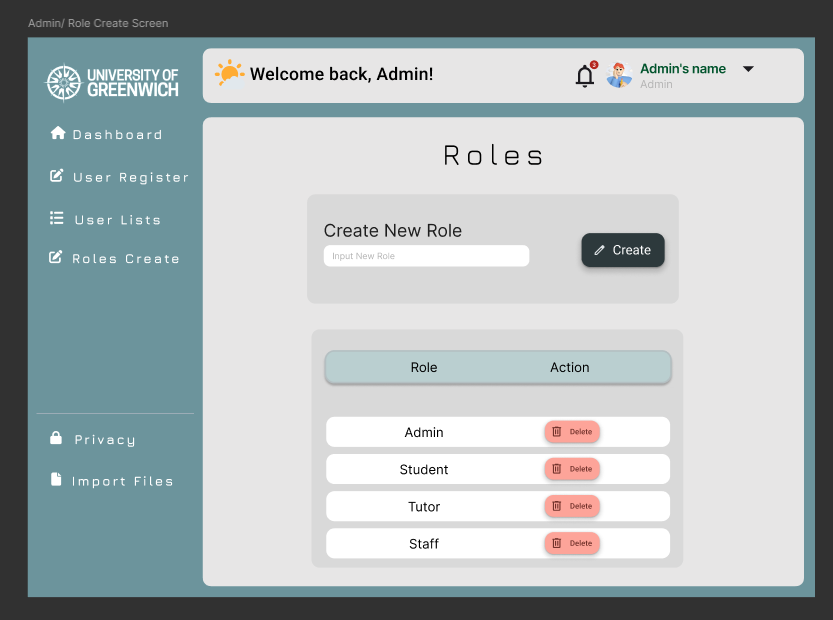
**Register**



**List user:**



**Create role and list role**

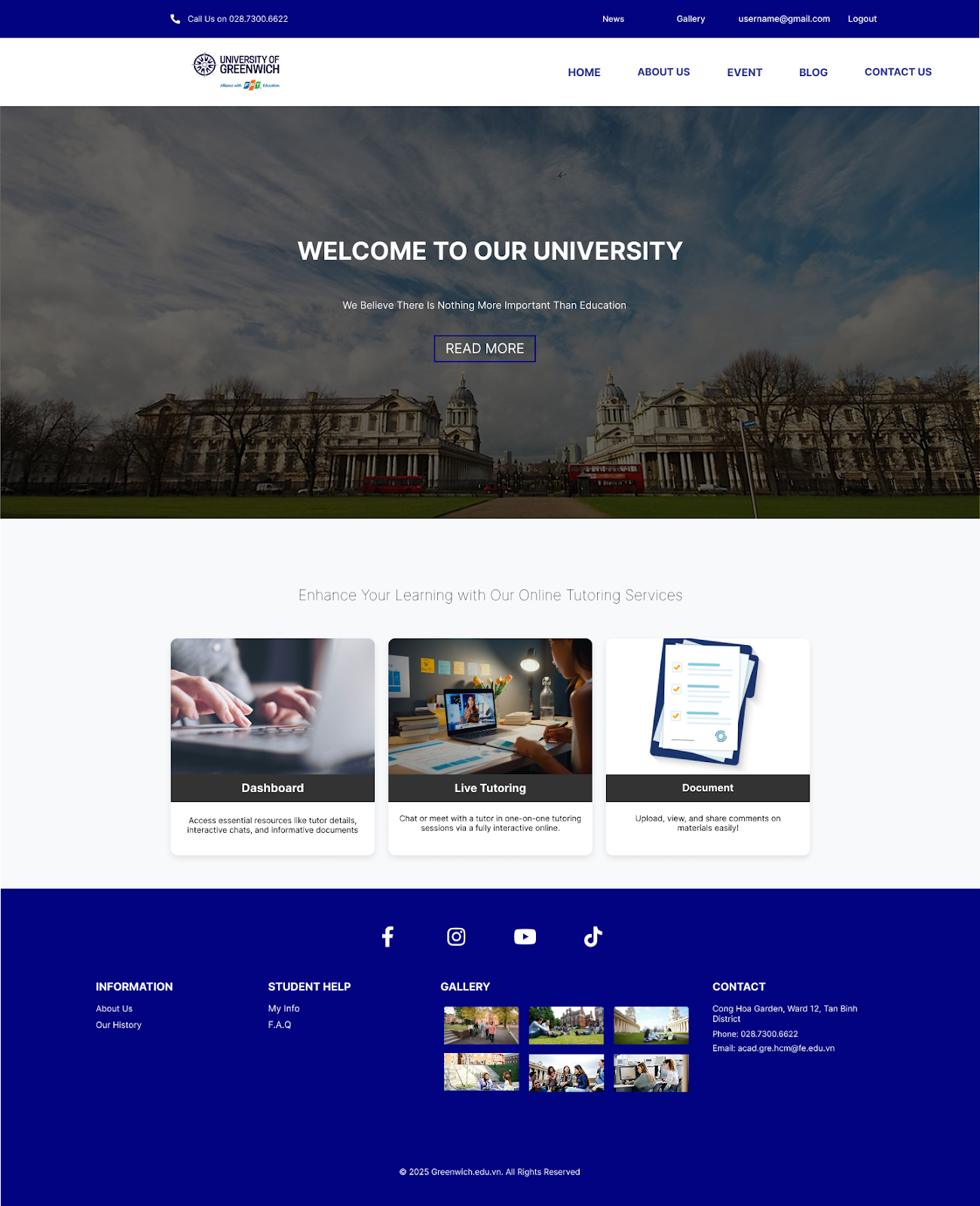


**Tutor :**

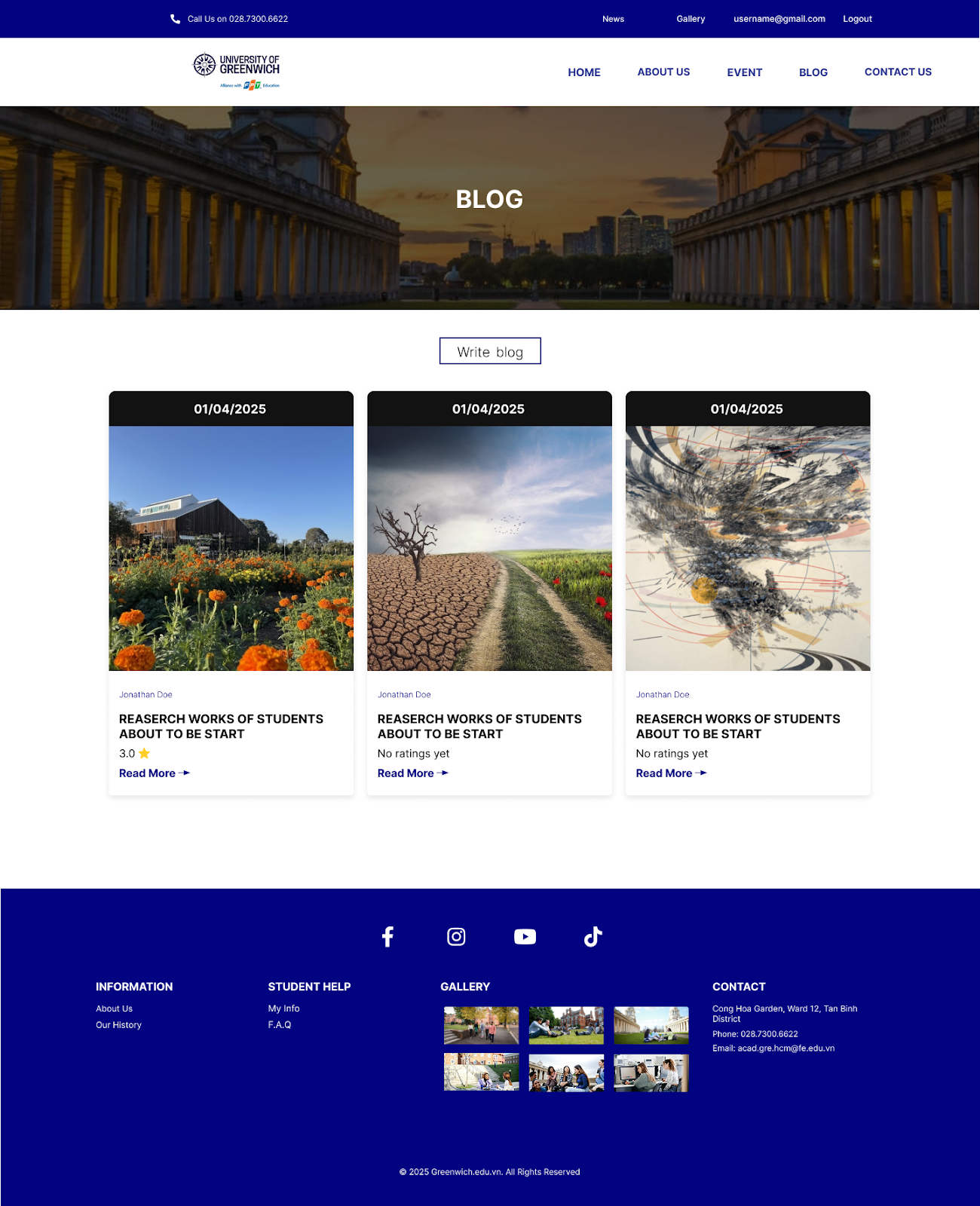
**Dashboard**

\****

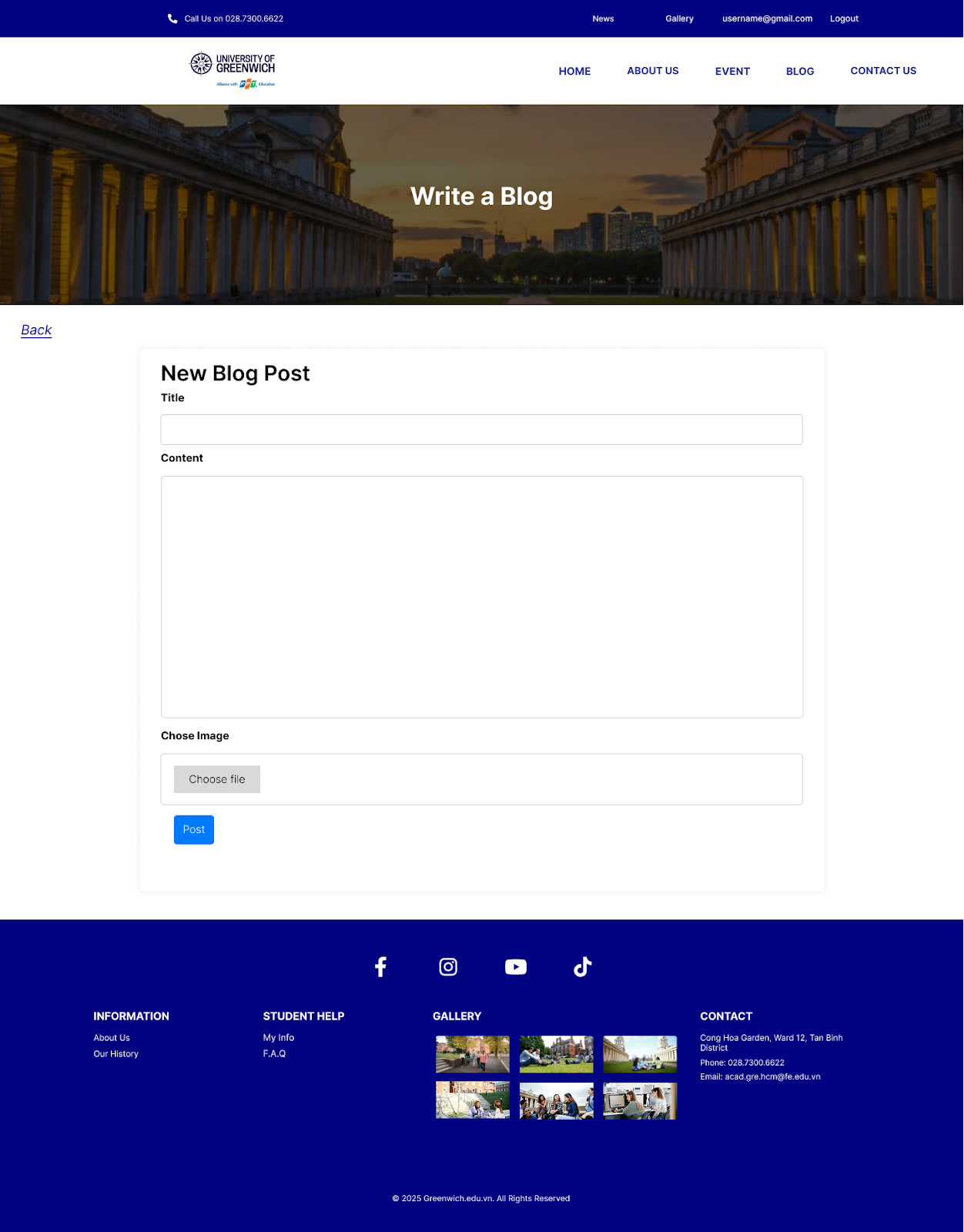
**User Index:**



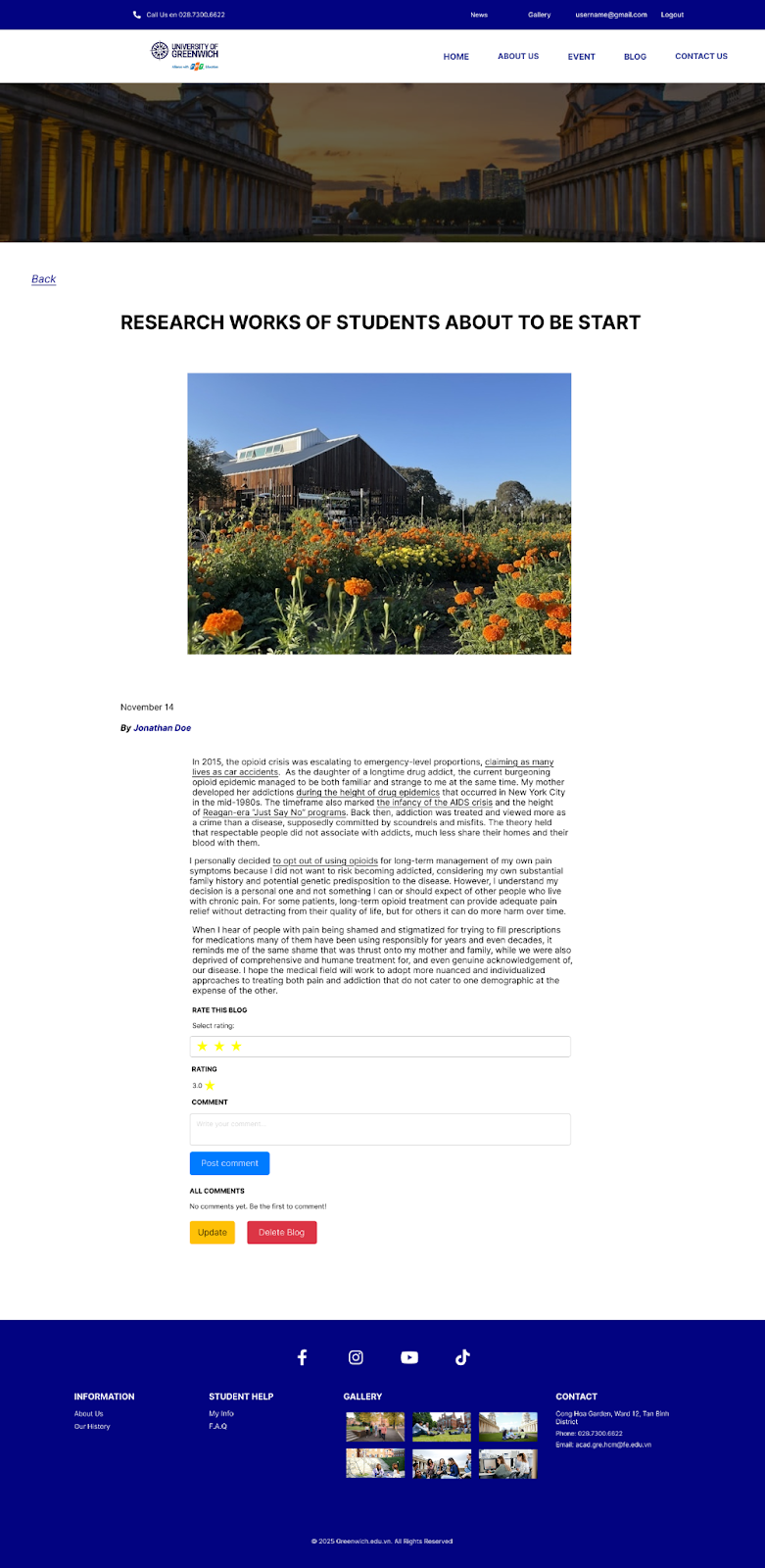
**Blog Page:**



**Writting Blog Page:**

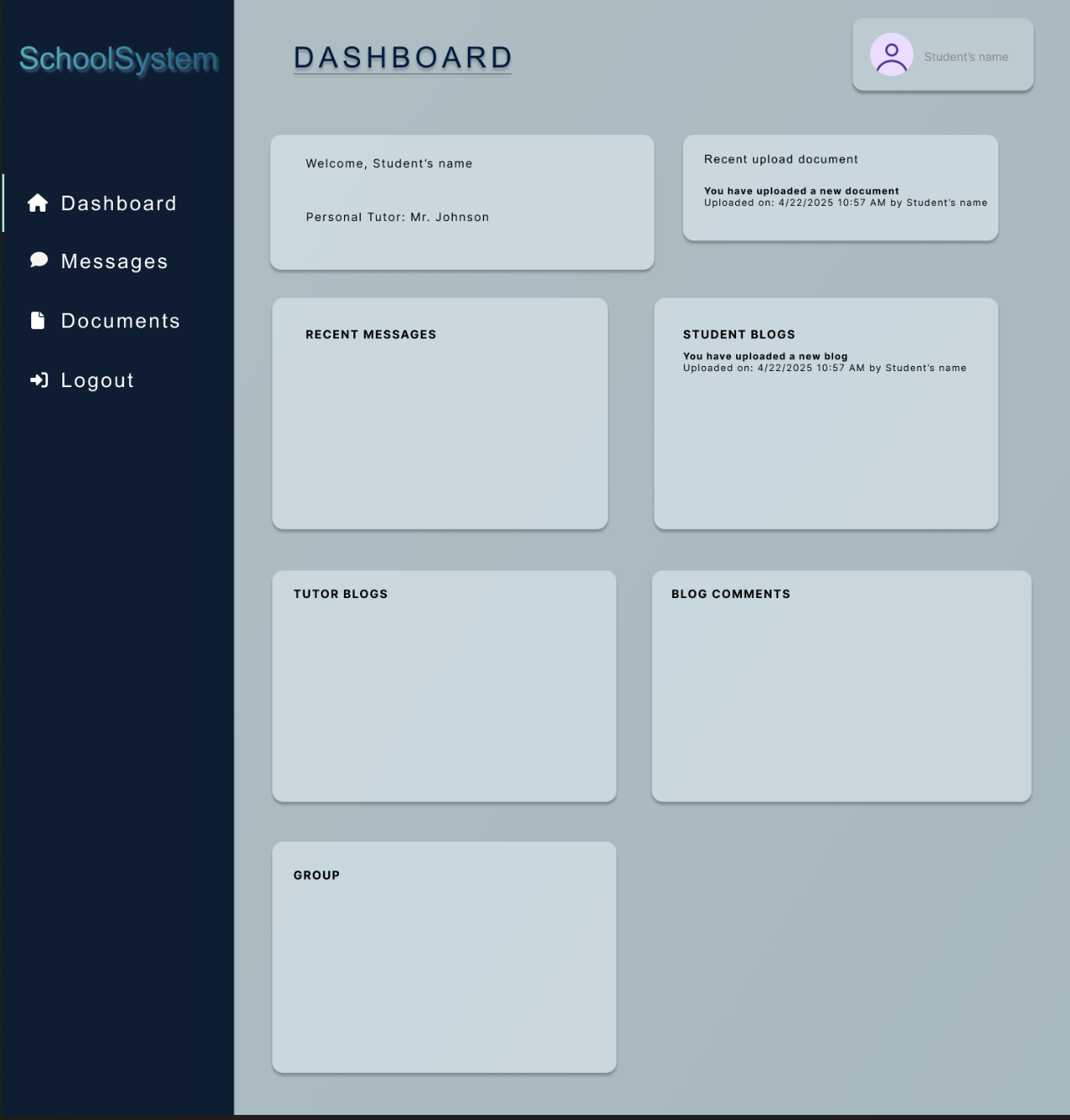


**Blog Detail:**



**Student:**

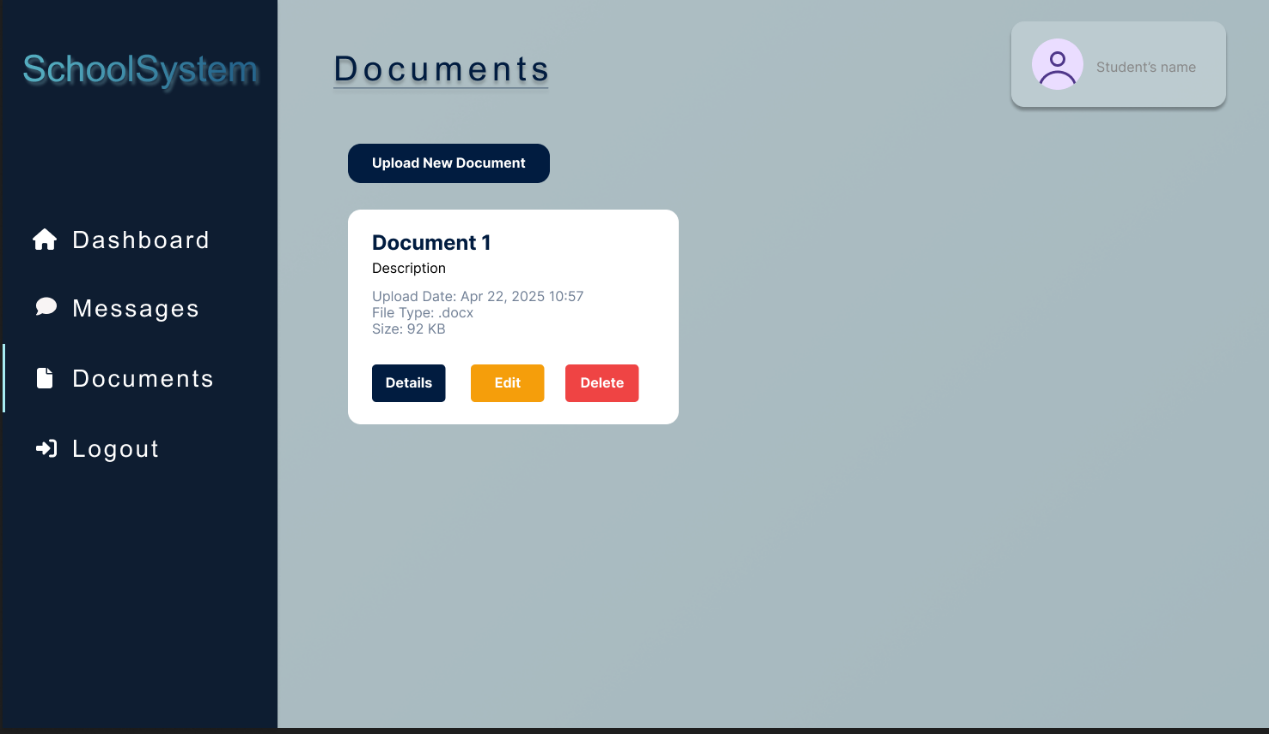
**Dashboard:**



**Messages:**

****

**Documents:**



# 12. Product screenshots of your actual application and its functions

# 13. Links: Source Codes (Github), Screencast (Youtube/Google Drive/Onedrive)