**GROUP PROJECT TEAM MANAGER WEB APPLICATION**

**SAMAD AFOLABI**

**1910611**

**INTRANET SYSTEMS DEVELOPMENT**

**CMM007**

**1.0 PROJECT DESCRIPTION**

**1.1 OVERVIEW**

This project is a web application to help in the creation, assigning and management of student groups involved in projects. It is being done to automate or reduce the manual work of assigning students to groups and managing the groups. It will be available online which will allow students and administrators to have access everywhere.

**1.2 PURPOSE**

To design and develop a functional web application with a good user interface to aid team managers and students involved in group projects with creating, assigning and monitoring of groups without much user interference.

**1.3 SCOPE**

* Creation of two types of users: Administrators and Students.
* Confirmation of group members by providing access to the group account.
* Creation of group accounts.
* Provision of email notification to notify group members of their statuses.
* Provision of upload feature for lists of students to be assigned into groups.
* Provision of forms for students to request for a group change.
* Random assignment of students to groups.
* Provision of comments boxes for group members to leave comments in group accounts.
* Provision of upload feature for Group members to share files in group accounts.

The project is designed to help create student groups and assign students to them automatically. There are two major users who would be involved in the use of this project: Administrators and Students.

The administrator: He/she is responsible for responsible for registering and creating group accounts. A list of actions that can be performed by the administrator include:

* Registering and logging into the application.
* Creation of New groups.
* Uploading of student lists or typing in of students in a class.
* Deletion of groups.
* Viewing of available groups.
* Updating of groups with students without groups.
* Reassignments of students into other groups upon request.

Students: He/she is the user of the system. A list of actions that can be performed by the student includes:

* Registering and Logging into the application.
* Commenting in the group accounts areas.
* Requesting for a new group if needed.
* Sharing of files in the group accounts areas.
* Viewing of all groups in a class.

**2.0 CONSTRAINTS**

The Web Application must have all its code available on GitHub. This involves creating a repository on a GitHub account already available or to be newly created if not available. The Integrated development environment being used will be set up to allow for committing and pushing of code to the repository without accessing the GitHub website.

The Web application must be hosted on a web server. This will be achieved through hosting for free on cloud service providers such as Amazon Web Services (AWS). This service offers free tiers which serves the purpose of demonstrating a fully functional application without incurring the additional costs associated with paid hosting providers.

The Web application must have a front-end (client side) to allow users to navigate through, register, sign in, sign out, comment and perform other actions that the user is allowed to do. The front end will allow users to input data which will be sent to the backend. The interface must be clear about what each page of the web application is developed for. The front end will be constructed using HTML, CSS, JavaScript, Bootstrap and jQuery (if necessary) which are all front-end technologies.

The Web application must have a backend (server side) which involves databases and the code to connect the front-end and the database at the backend. This code will be written in the PHP programming language. The code will allow for user’s data inputted at the frontend to be validated with the information available in the database.

The Web application must have a login system which will allow a user to input their credentials to be validated or to register as a new user. The data inputted in the login system will be validated by the backend. The login system will be developed using both the frontend and backend technologies defined to make it fully functional.

**3.0 FUNCTIONAL REQUIREMENTS**

|  |  |
| --- | --- |
| **S/N** | **REQUIREMENT DESCRIPTION** |
| 1. | The student shall be able to check a list of all the groups available. |
| 2. | The student shall be able to comment in student group account areas. |
| 3. | The student shall be able to request for a change of groups if not satisfied with the current grouping. |
| 4. | The student shall be able to the student shall be able to sign in to the web application. |
| 5. | The student shall be able to sign out of the web application. |
| 6. | A user shall be able to register for a new account on the web application. |
| 7. | The administrator shall be able to create new groups. |
| 8. | The administrator shall be able to delete groups. |
| 9. | The administrator shall be able to upload lists of students. |
| 10. | The administrator shall be able to change a student s group if requested. |
| 11. | The administrator shall be able to delete student comments if inappropriate. |
| 12. | A user shall be able to register as an administrator on the web application. |

**4.0 NON-FUNCTIONAL REQUIREMENTS**

**4.1 SECURITY AND PRIVACY**

* The web application will request for passwords with a length above 6 characters to guarantee security of accounts.
* In the event of a security breach, an email notification is to be sent to all administrators and students to change account details such as password to prevent further damage.
* A backup of the web applications data such as the database and the code written at the backend will be backed up on a weekly or monthly basis depending on the number of users will be developed in the event of loss or corruption of data.
* To enhance further security, a second pin will be requested of users at registration to add further protection.
* During registration, new users will be asked whether they are administrators or students to clearly define their roles.

**4.2 RELIABILITY**

* The web application must be able to handle all users request at the same time.
* The web application must notify its users before it goes offline for any potential upgrades to be made.
* A separate email service will be set up to notify users in the event of the web application not being online.

**4.3 RECOVERABILITY**

* In the event of the web application being unavailable to users because of web application failure, the failure must be repaired within a minimum of 48 hours.
* The database must be capable of being restored to its optimum condition in no more than 24 hours after the corruption or failure took place.

**4.4 AVAILABILITY**

* The web application must be available to users Monday through Friday between the hours of 8:00 a.m. and 9:00 p.m.
* The application will be unavailable between the hours of 10:00 p.m. and 7:00 a.m.

**4.5 CAPACITY**

* The web application must be able to handle a growing number of users.

**4.6 ERROR HANDLING**

* In the event of errors, the user will be notified of errors with instructions provided if it is a simple error such as invalid input.
* In the event of system errors that cannot be resolved by the user, the web application administrator is to be notified.

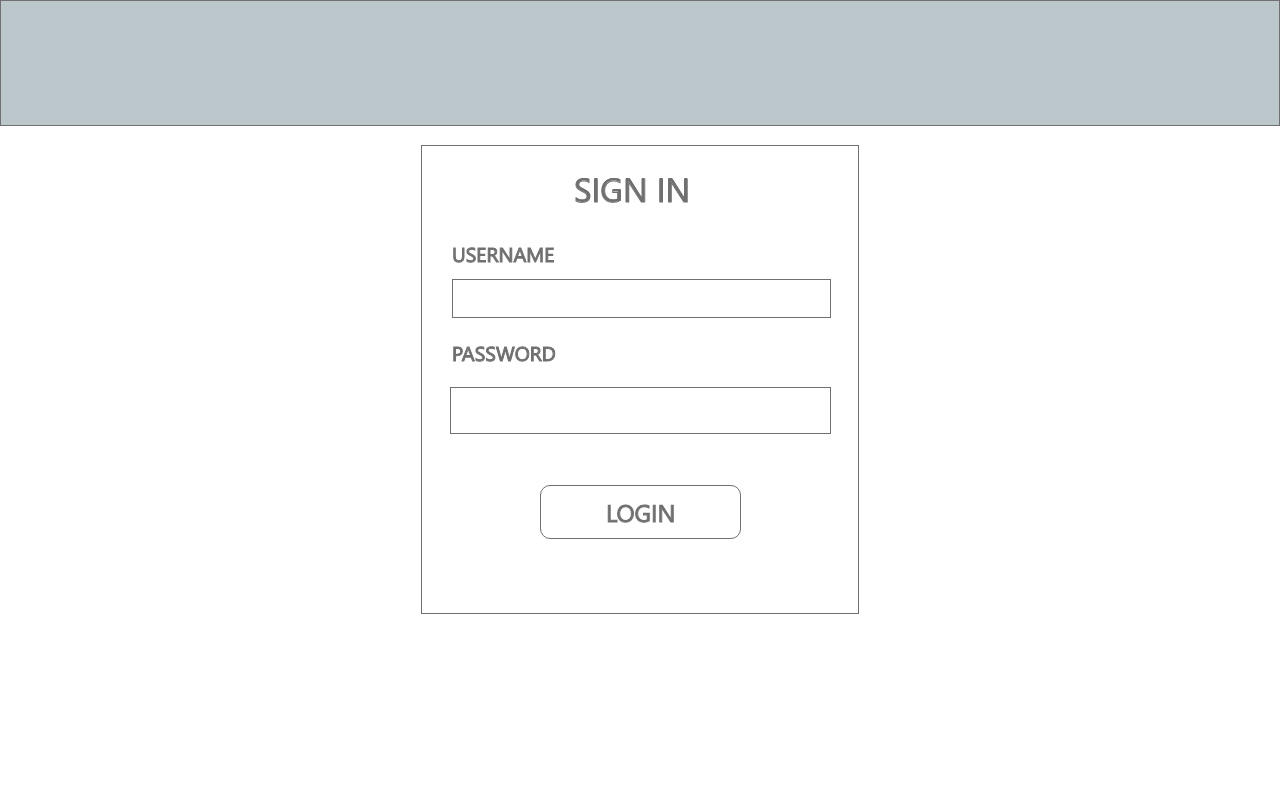
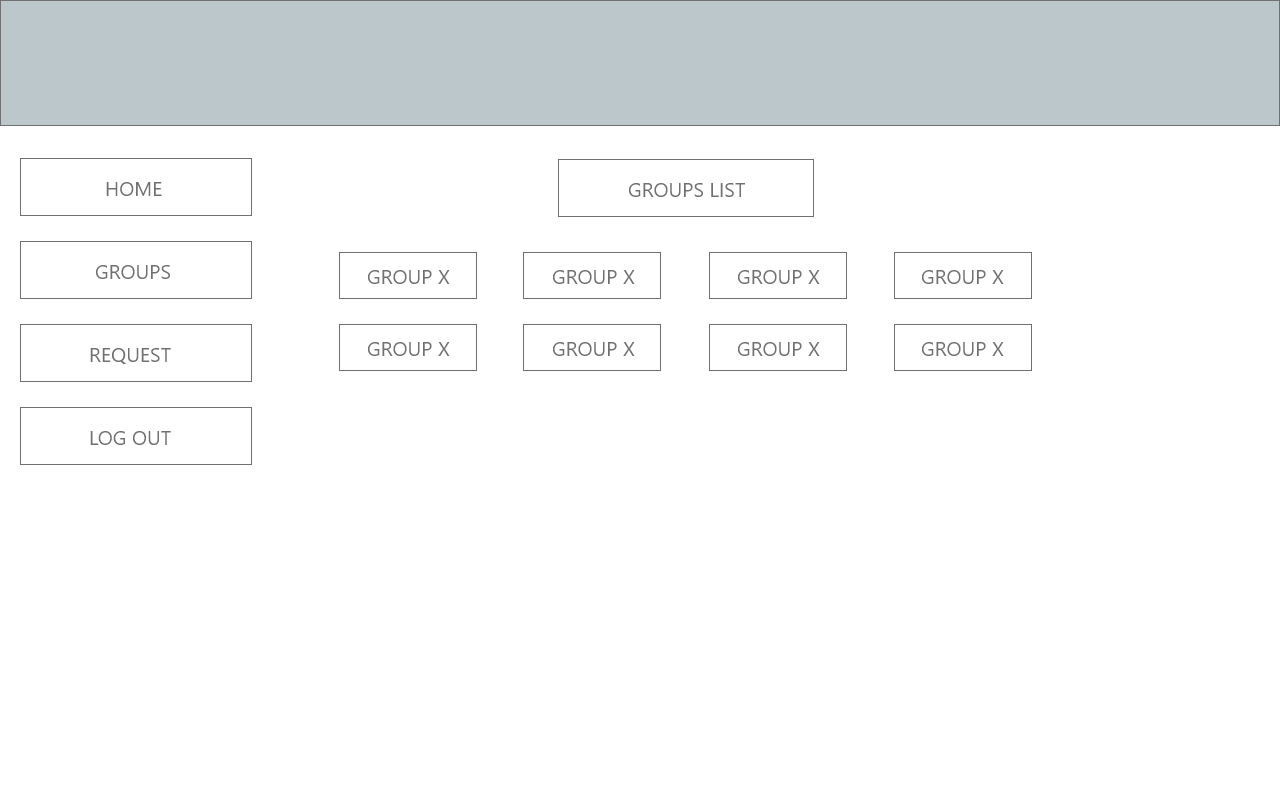
**WIREFRAMES**

Figure 1: Wireframe Sign in Page

Figure 2: Wireframe Groups List Page