

Higher Nationals in Computing

COMP1640

Enterprise Web Software Development  
GROUP REPORT

Student name: TRAN THI THANH BINH

Student ID: GCS18836

Group name: NYKAB

Assessor name: LE NGOC THANH

Class: TCS2011

Assignment due: 20 April 2021

Assignment submitted: 19 April 2021

Contents

[**Section A – Group Report 1**](#_Toc69728255)

[1 INTRODUCE 1](#_Toc69728256)

[1.1 Purpose 1](#_Toc69728257)

[1.2 Team 1](#_Toc69728258)

[2 SCHEDULE 1](#_Toc69728259)

[3 DESCRIBE REQUIREMENT SPECIFICATION 2](#_Toc69728260)

[3.1 Summarized Product Backlog 2](#_Toc69728261)

[3.2 Summarized Sprint Backlog 3](#_Toc69728262)

[4 DESIGN 3](#_Toc69728263)

[4.1 Interface design 3](#_Toc69728264)

[4.1.1 Pages of administrator 5](#_Toc69728265)

[4.1.2 Pages of student 13](#_Toc69728266)

[4.1.3 Pages of coordinator 15](#_Toc69728267)

[4.1.4 Pages of manage 17](#_Toc69728268)

[4.2 System design (UML Diagram: Use Case, Class Diagram, Sequence Diagram) 19](#_Toc69728269)

[4.2.1 Use-case Diagram 19](#_Toc69728270)

[4.2.2 Class Diagram 27](#_Toc69728271)

[4.2.3 Sequence Diagram 27](#_Toc69728272)

[4.2.4 Database design 33](#_Toc69728273)

[5 IMPLEMENT 34](#_Toc69728274)

[5.1 Operating environment 34](#_Toc69728275)

[5.2 Program Language 34](#_Toc69728276)

[5.3 Website construction structure 34](#_Toc69728277)

[5.4 Coding 36](#_Toc69728278)

[5.4.1 LOGIN (user) 36](#_Toc69728279)

[5.4.2 Captcha (User) 38](#_Toc69728280)

[5.4.3 Send Email (Student, Marketing Coordinator) 39](#_Toc69728281)

[5.4.4 Submit articles (Student) 42](#_Toc69728282)

[5.4.5 Download file 43](#_Toc69728283)

[5.4.6 Download file ZIP 44](#_Toc69728284)

[5.5 Presentation 45](#_Toc69728285)

[6 TESTING 45](#_Toc69728286)

[**SECTION B – INDIVIDUAL REPORT** 46](#_Toc69728287)

[1. EVALUATION OF PRODUCT 46](#_Toc69728288)

[2. EVALUATION OF PROCESS 47](#_Toc69728289)

[3. EVALUATION OF TEAM 47](#_Toc69728290)

[4. SELF-EVALUATION 49](#_Toc69728291)

[**5**. Conclusion 49](#_Toc69728292)

[**REFERENCES 50**](#_Toc69728293)

# Section A – Group Report

## INTRODUCE

### Purpose

This document is created to document the team's development and management in a scrum process. This is a secure role-based website system for collecting and reviewing student contributions to college journal articles every year.

### Team

Group name: NYKAB

Team members - roles

|  |  |
| --- | --- |
| MEMBERS | ROLES |
| Vo Hoang Yen | Information architect, scrum master |
| Nguyen Ngoc Nhi | Tester |
| Nguyen Dang Khoa | Programmer |
| Do Lam Hoang Anh | Web design, Information architect |
| Tran Thi Thanh Binh | Programmer |

## SCHEDULE

****

Figure 1: Schedule of project



Figure 2: Gantt chart

## DESCRIBE REQUIREMENT SPECIFICATION

### Summarized Product Backlog

Research and discuss customer requirements to come up with first user stories, receive customer requests. The whole group discuss and evaluate and select the customer requirements. Lists the functions in the system for the product owner to write user stories. From there, members will rely on use story to start doing the next work. Product backlog will be updated after each sprint review period.



Figure 3: Product Backlog

The detailed product backlog is contained at <https://drive.google.com/drive/u/0/folders/1OzmjOI1l3XNc48H5ftcOF6uDa0SKQCtL>

### Summarized Sprint Backlog

Based on the product backlog, we divide it into 3 sprints. Each sprint lasts for 2 weeks and will run about 28 points. On the last day of each sprint, the whole team will do the sprint review and retrospective. To draw out the advantages and things to improve to apply to the next sprints.

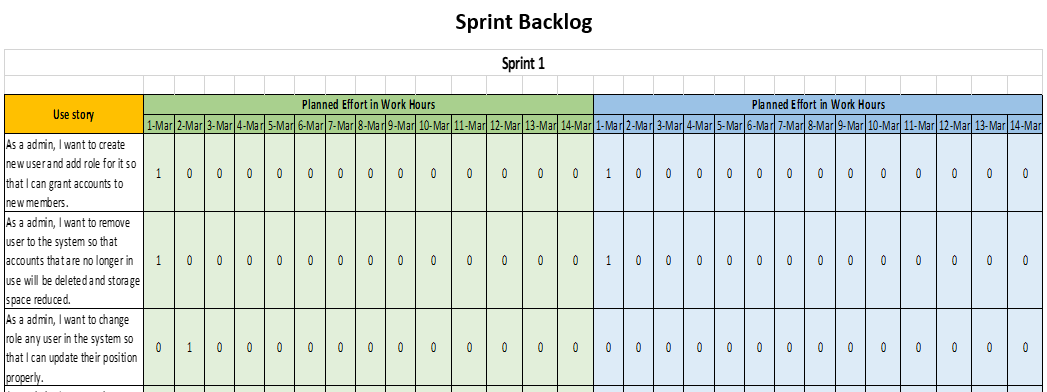


Figure 4: Sprint Backlog

The detailed print backlog is contained at <https://drive.google.com/drive/u/0/folders/1FrqBNT_FyVkIvqOr3YvPSxFBrMts8s34>

Burndown chart of each sprint is contained at <https://drive.google.com/drive/u/0/folders/1idAZ9N8ZhvTIiX5Qc6eXabXP1svZahph>

## DESIGN

### Interface design

The first job we do is draw a sitemap to know the number and pages that we need to design the interface.

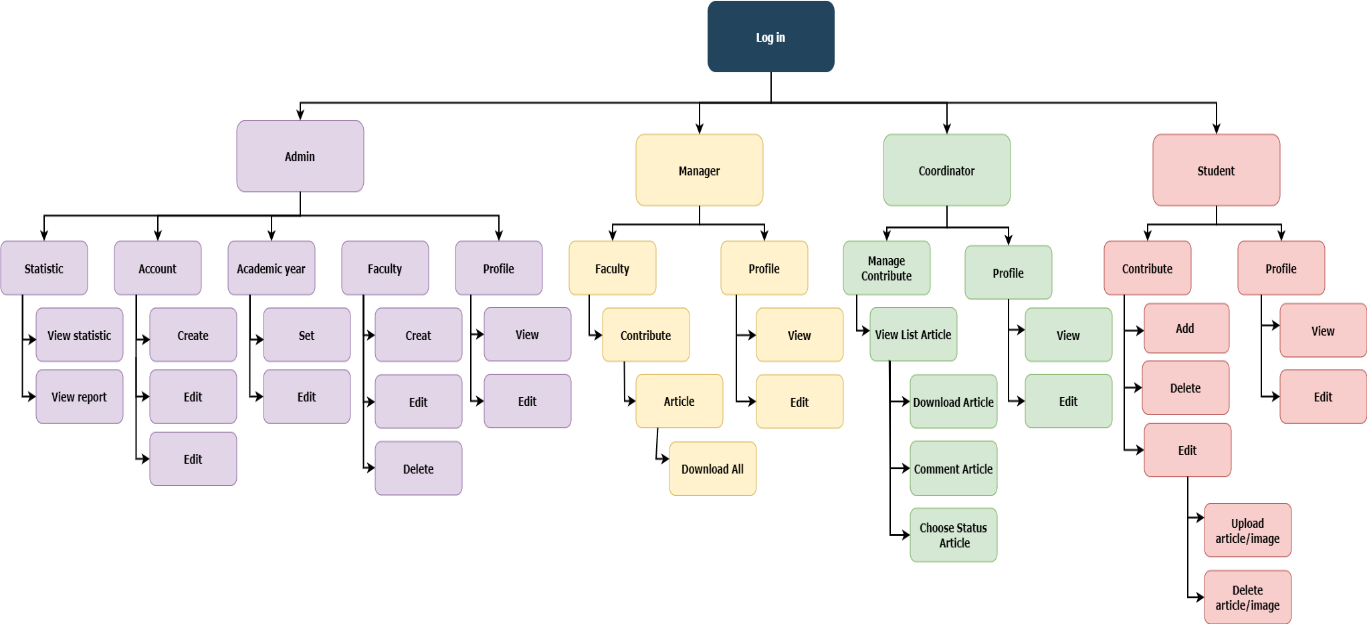


Figure 5: Sitemap the website

Before we embark on the detailed design, we conceptualized and laid out the layout for the website by drawing wireframes for desktop and mobile.

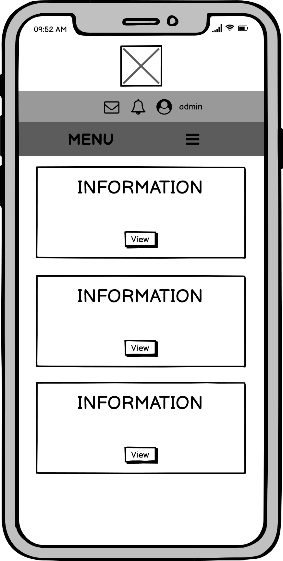
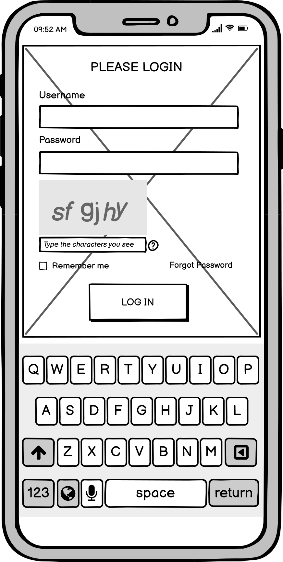
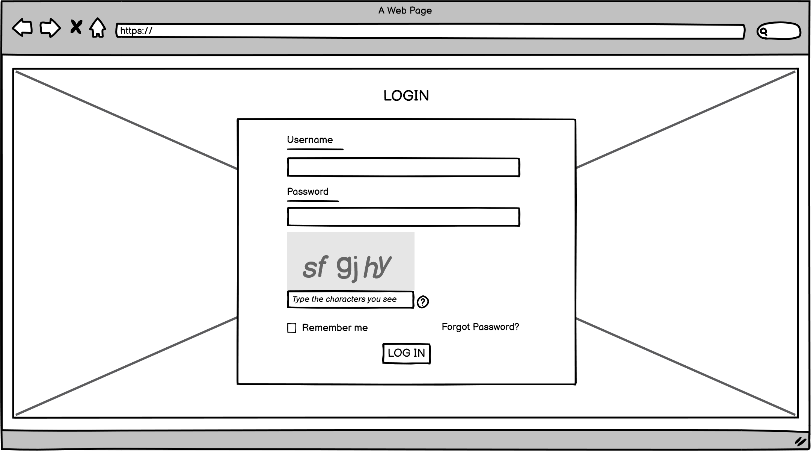
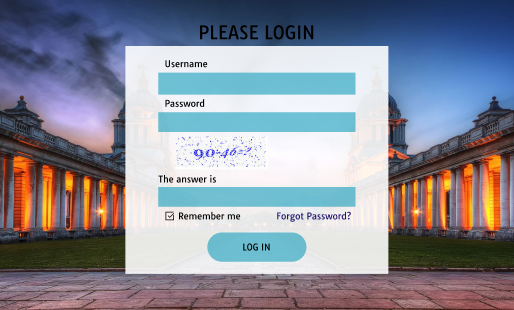


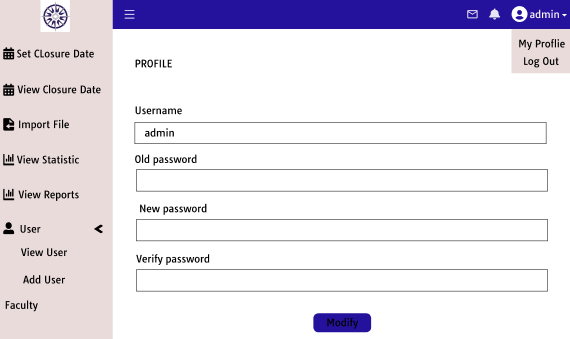
Figure 6:UI for the website

Full and detailed images of the wireframe will be saved at <https://drive.google.com/drive/u/0/folders/1YUUI-uOKtNe0270qvPx5GtebV8Q8vujj>

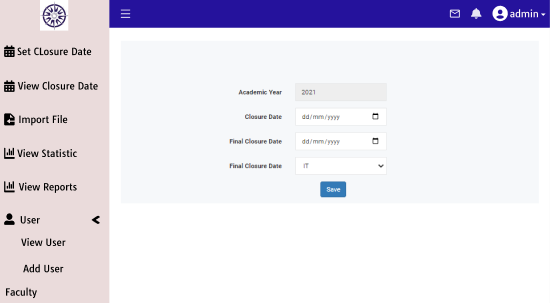
****

This is the main page of the system. Everyone needs to login to be able to do their jobs. The information to login will include the username and password created and provided by the admin. After you fill information, you need to enter a captcha to log in.

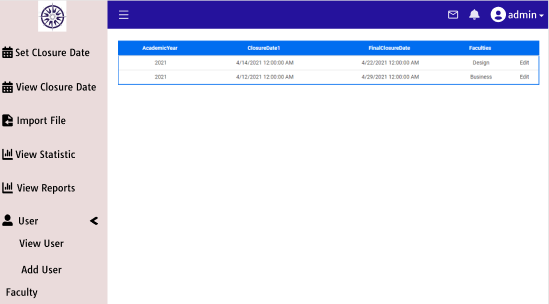
#### Pages of administrator

****

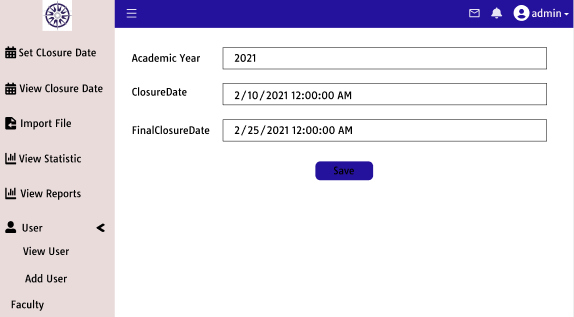
Once logged in, the admin can view the information and change the own information and password by selecting my profile in the image admin drop-down.



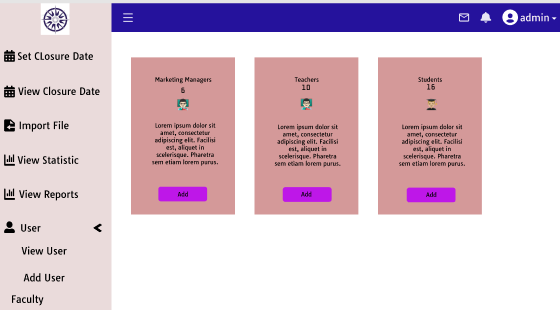
This is the page where the admin will set the deadline for submission of student submissions for each year by each department. Admin will select the closure date, the final closure date, and the department then presses save to create it.



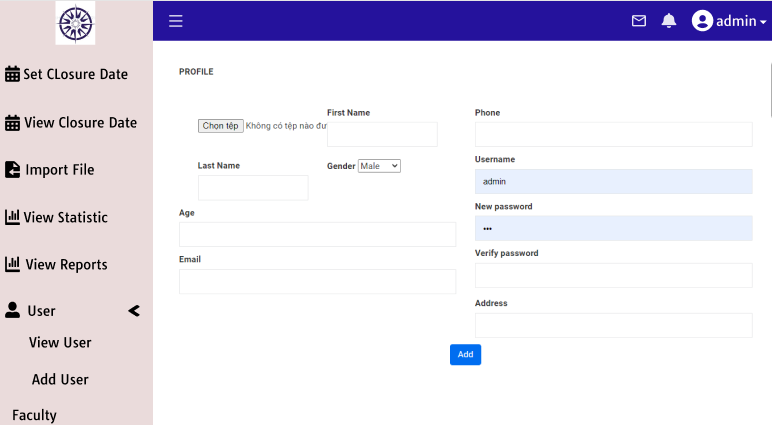
Once created, the admin can go to this page to check if the creation time has expired. Also, the admin can edit the time if necessary, by clicking the edit of the time he wants to edit.

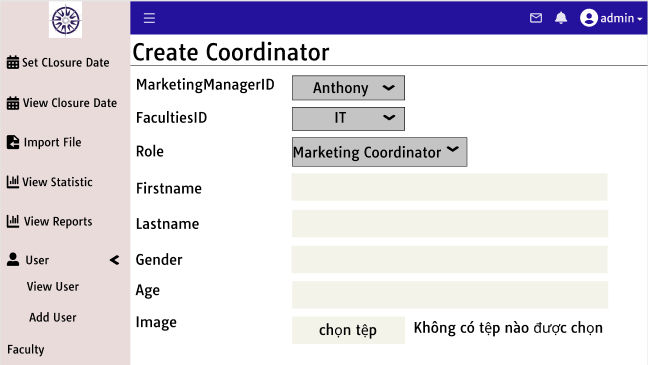


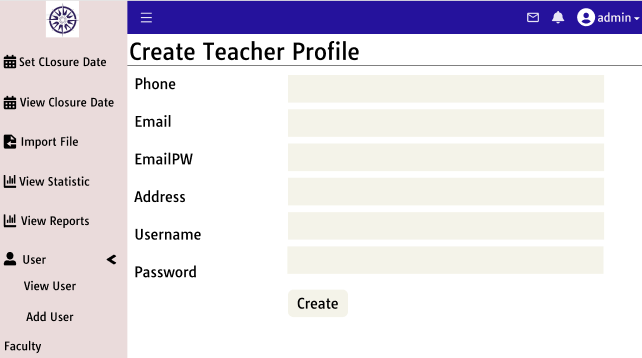
Admin will modify the date and time of the closure date or final closure date. Then press the save button to save the changed time.



To add a new user, the admin will go to the add user page and select the role of the user to be created then click on the add button. For each role, there will be different information.





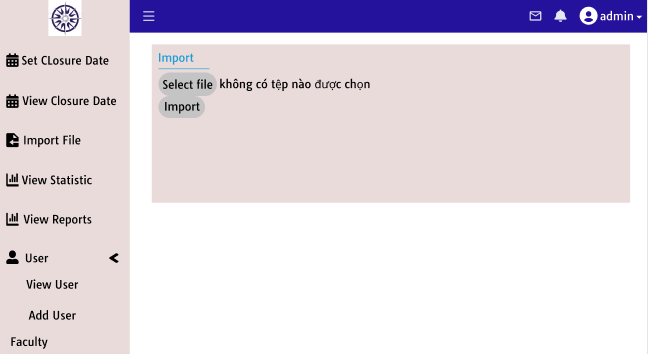


When finished filling in, the admin information will click the add button at the bottom of the page to create.

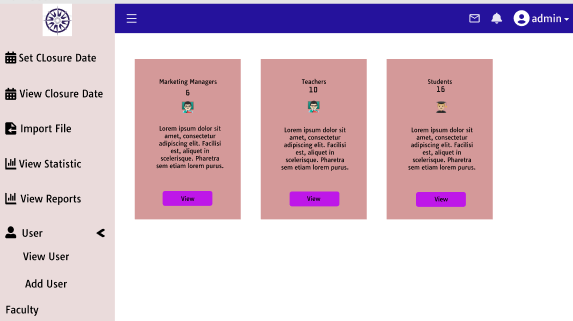
However, with students, we will be different because each school year there will be a lot of new students instead of one person. Therefore, adding each student takes admin time-consuming and the information may be wrong. The file import function saves time for this. Admin will collect student information in an excel file with complete information (Table 1).

Table 1: Information student



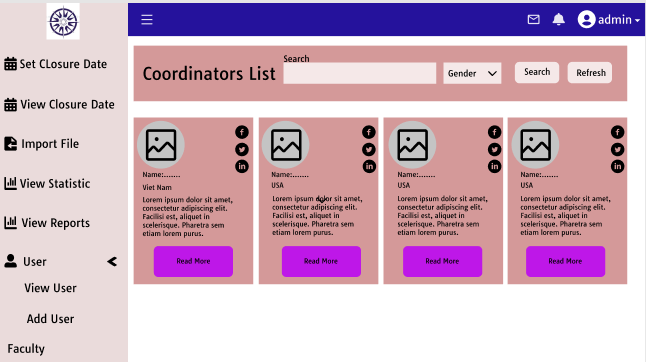


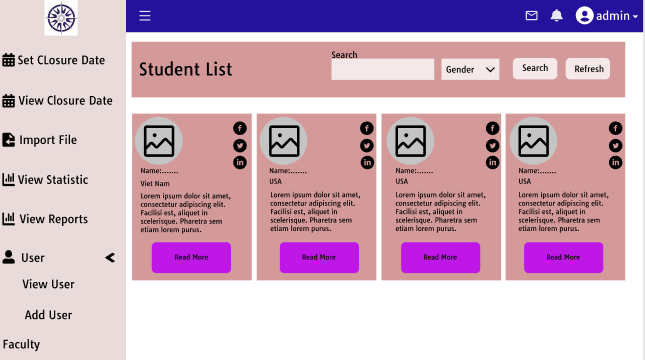
Then will go to this page, click on the select file button and select the excel file. Finally, click the import file button to add a new student. After the successful add, the new user will appear in the list.



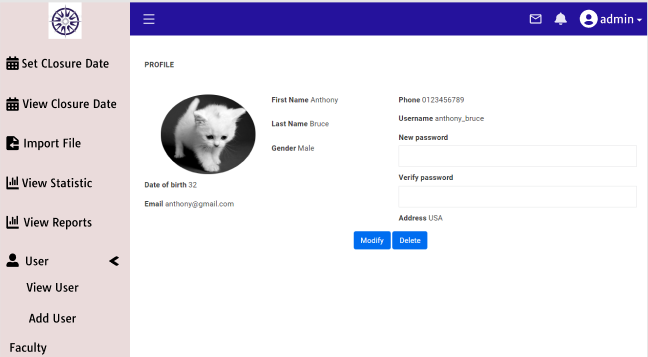
If the admin wants to see the list of users, then the admin can go to the view user in the menu. The screen will display 3 groups of users. Admin will choose which role you want to see.

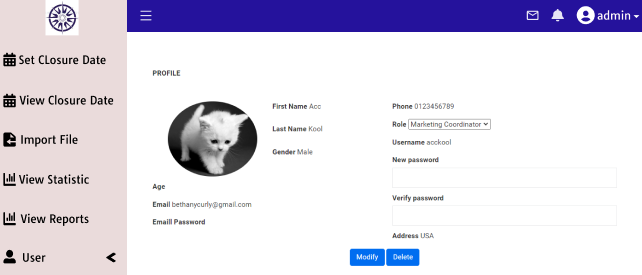


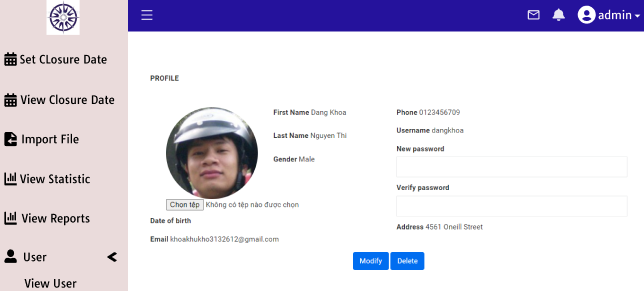




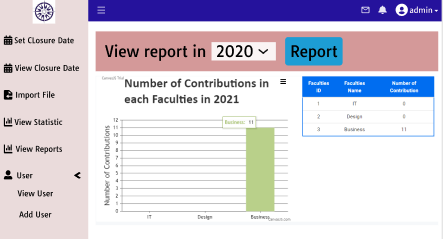
Admin can view a detailed profile, edit the password of the user and delete the user by clicking read more.



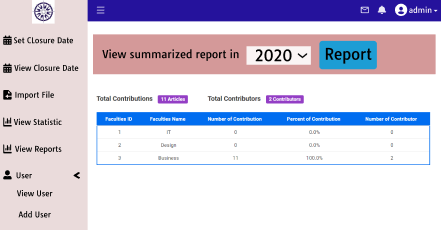




If you want to change the password, you enter a password, verify the password and click the button modify. If you want to delete a user, you click the button to delete.

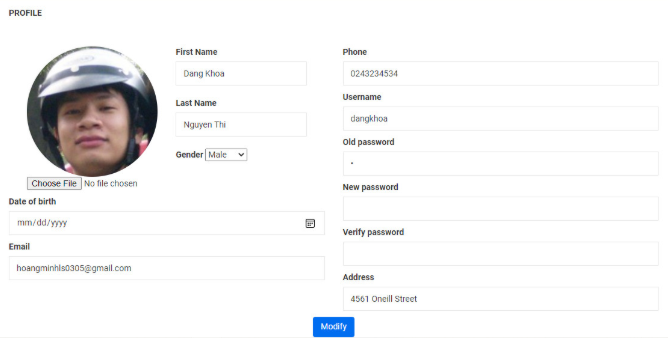


To view statistics for each year and each department, the admin will select the statistics view in the sidebar. The screen will display the charts and tables containing the statistics.

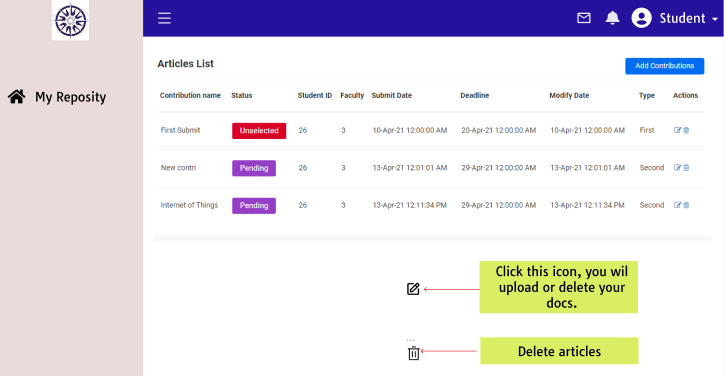


To view reports of that statistic, the admin will select the view reports in the side-bar to view.

#### Pages of student



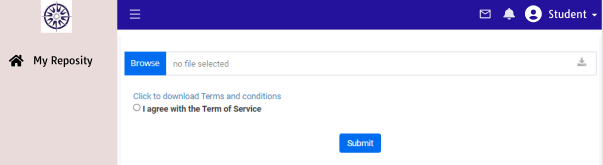
Student can go to my profile page to view information or edit the information on this page.



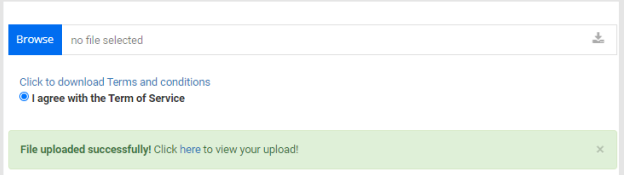
This is the page that contains all of the articles that the student has contributed. Want to contribute more new articles, students can choose the button to add contribute in the right corner of the screen.



The student will name that contribute and choose to submit to continue. Then the website will appear on a screen for students to upload files.

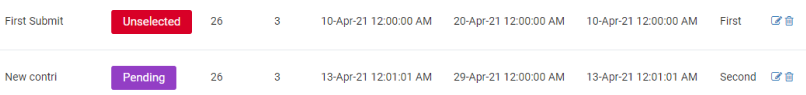


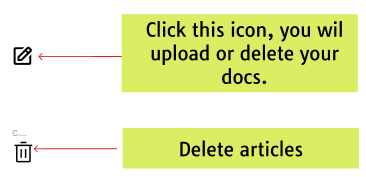
Students must choose the file they want to push and agree to the terms and conditions to upload articles. Finally, students will click the submit button to complete the creation of a new contribution.



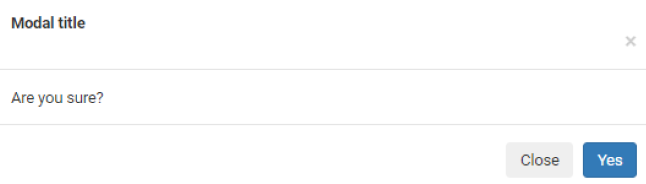
This message line will be displayed when you have done up the job upload the article. After adding the contribution, it will appear in the following table (Table 2).

Table 2: List Articles

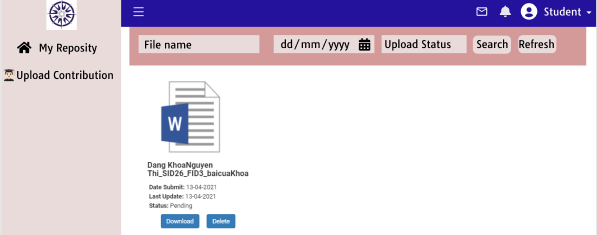




Students can delete the contribute by clicking the trash can icon. When choosing delete, a window will appear to know that you are sure about the deletion, not by accident. If you want to delete, you choose yes and vice versa.



Also, students can choose the pen icon to edit their posts. When selecting the pen icon will go to the page below.

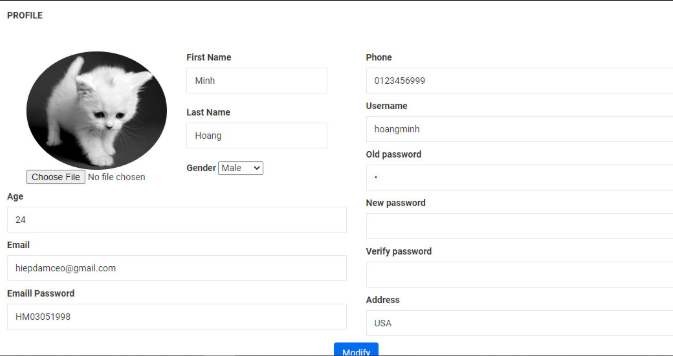


This page will contain a word file and images that students have posted. Students can change to another article by deleting the old article and uploading a new article. Also, if students lose their lessons, they can re-download the lessons to their computer by clicking the download button below each file.

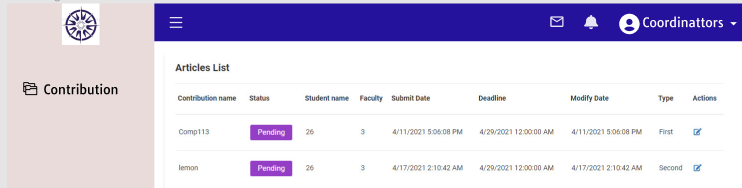


Students can reply to the coordinator to inquire about the coordinator's requirements with their articles.

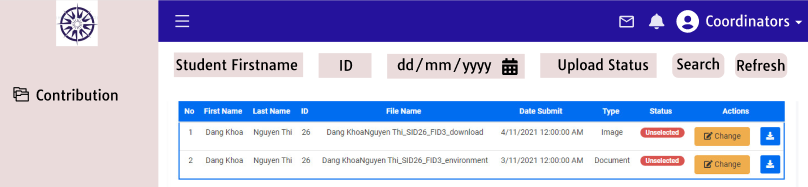
#### Pages of coordinator

****

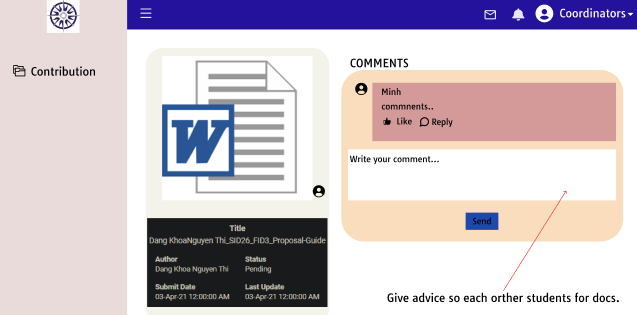
The coordinator can go to my profile page to view information or edit the information on this page.



The coordinators will see the contribute posted by the students in the faculty that they are responsible for. When the coordinator wants to see more details of contributing to it, the student will click on the pencil icon.

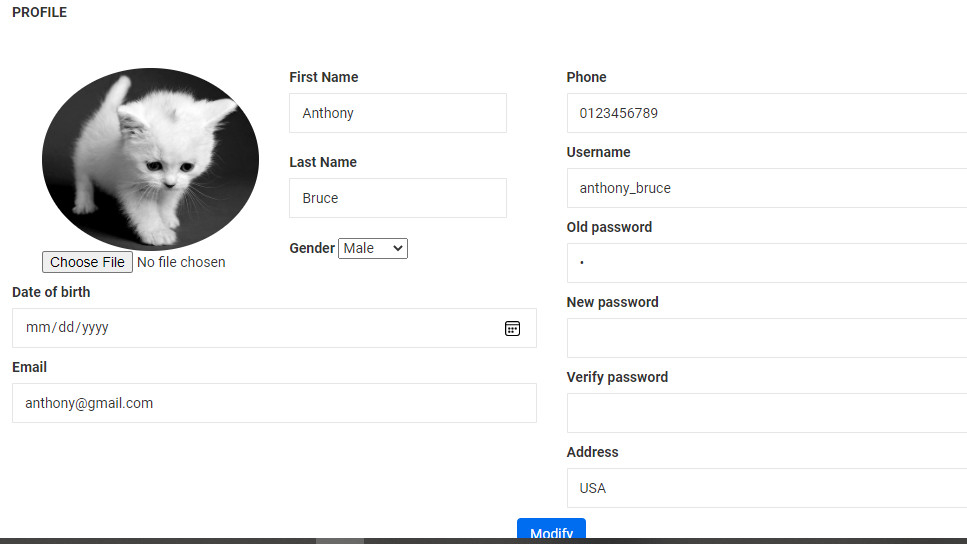


If they want to see the content, they will click on the download icon to download the article. When they feel that the post can be a journal, they will choose the status by clicking on change, select selected, and vice versa. If they feel this article is quite good, but need a little editing. They will double-click on its name to comment.

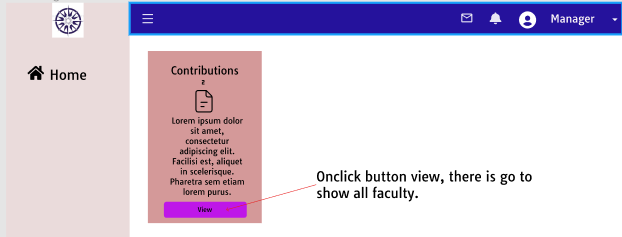


The coordinator will enter a comment and click the button send. The message will be sent to the student.

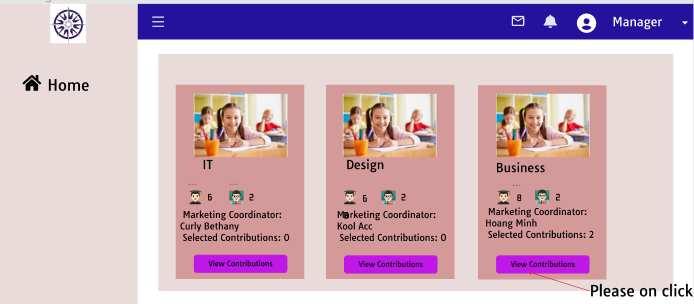
#### Pages of manage



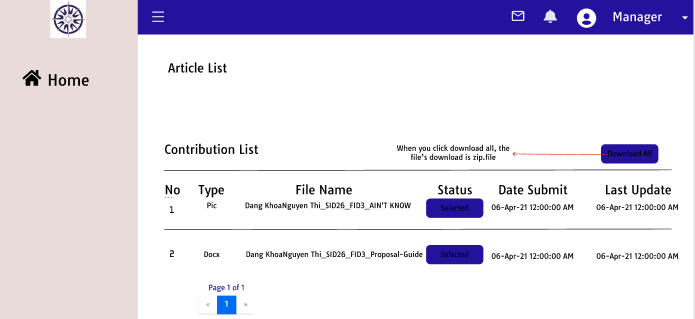
Manage can go to my profile page to view information or edit the information on this page.



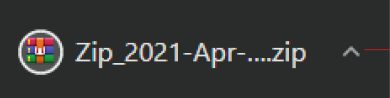
To view the contributes selected for the public, the manager will click the view button. Then the website will switch to the page below.



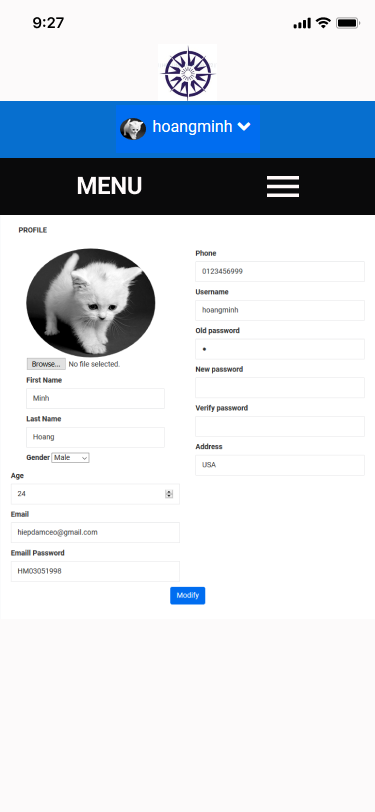
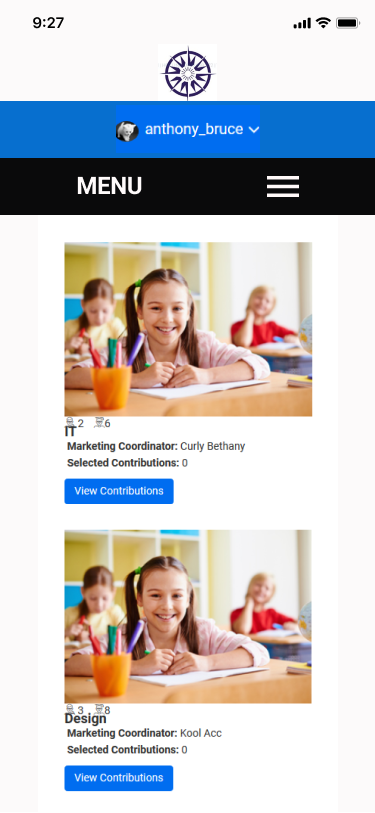
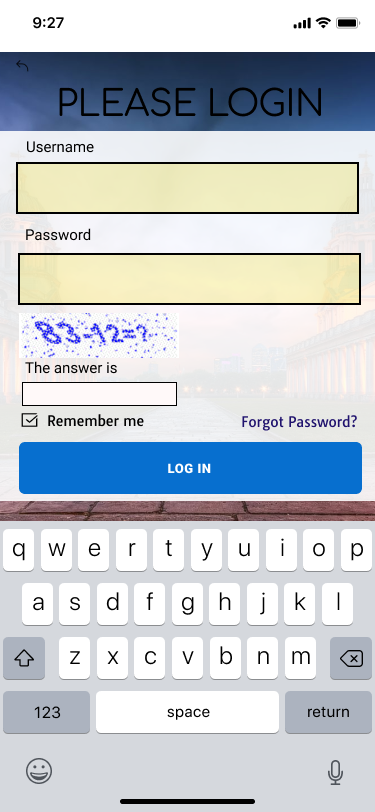
The contribution will be stored according to its faculty, so the manager needs to select a faculty to view the contributions of that department.



The manager can download all the files for journal publication. The articles will be saved to your computer as a zip file.



In addition to the design interface for the desktop device, we also design an interface for the mobile device.



More detailed images are contained in <https://drive.google.com/drive/u/0/folders/1VFh70H1l8LllT5FJ3dBA90j6G3vxlnS4>

### System design (UML Diagram: Use Case, Class Diagram, Sequence Diagram)

#### Use-case Diagram

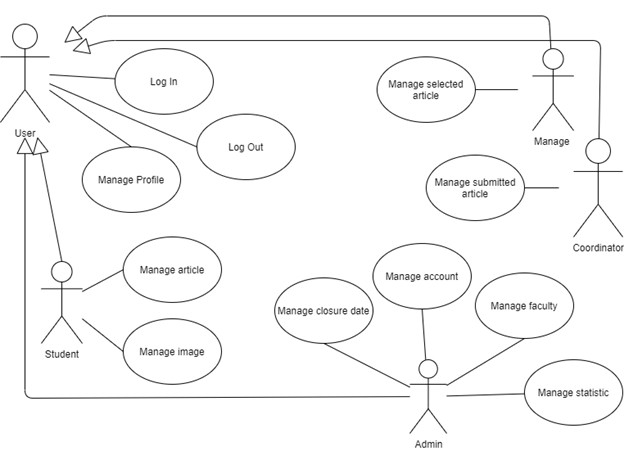


Figure 7: Use case generalization

This is a general use case that encompasses all of the website's interactive objects and the main functions of each of those objects.

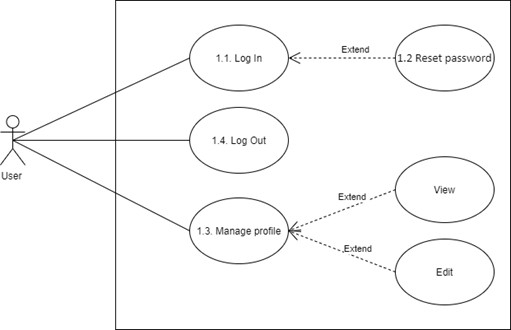


Figure 8: Use case of user

A normal user will have basic functions such as logging in, logging out, viewing profiles, changing passwords, etc. For each use case like login, logout, etc. We will write a script for it with information like use case id, name, actor of use case, basic flow, etc. Below is a scenario for a use case login.

|  |  |  |
| --- | --- | --- |
| Use Case Identification and History | | |
| **Use Case ID:** | | 1.1 |
| **Use Case Name:** | | Log in |
| **Description:** | | As the user, I want to log into the app to perform system functions |
| **User/Actor:** | | User |
| **Trigger:** | | The user wants to log into the system |
| Pre-conditions | | |
| The user already has an account.  The user’s device was connected to the internet when logging in. | | |
| **Basic Flow: The user login successfully.** | | |
| **Step** | **Actions** | |
| **1** | User access to the website . | |
| **2** | User enters information to login. | |
| **3** | User click login button. | |
| **4** | The system verifies login information and allows users to access the website. | |
| **Exception Flow: The user cannot login.** | | |
| **1** | The user entered missing information. | |
| **1a** | The system reports that the user enters missing information.  (Return step 2 in Basic Flow) | |
| **2** | The user entered incorrectly less than five times. | |
| **2a** | The system reports that the user enters incorrect information.  (Return step 2 in Basic Flow) | |
| Post conditions | | |
| The user has successfully logged in to the application | | |
| Business Rules | | |
| Users who enter incorrect login information for the 5th consecutive time will be locked out of their account. | | |

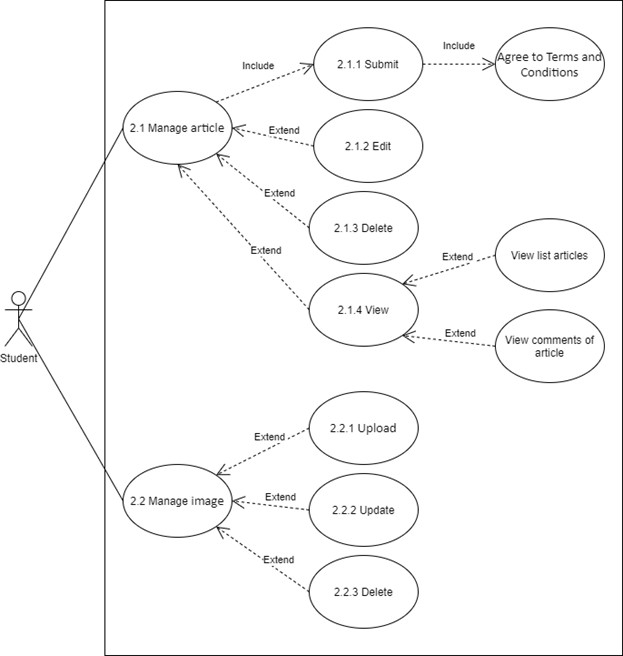


Figure 9: Use case of student

Students' tasks in this system are divided into two main functions: article management and image management of each of those articles. The management tasks are usually simple adding, deleting, and editing.

|  |  |  |
| --- | --- | --- |
| Use Case Identification and History | | |
| **Use Case ID:** | | 2.1.1 |
| **Use Case Name:** | | Submit article. |
| **Description:** | | The student will submit an article on the website. |
| **User/Actor:** | | Student |
| **Trigger:** | | Student wants to submit the article. |
| Pre-conditions | | |
| Student is logged into the system  The webpage is displayed successfully | | |
| **Basic Flow: Student submits the articles successfully.** | | |
| **Step** | **Action** | |
| **1** | Student choose upload article in menu. | |
| **2** | The system will display the upload form. | |
| **3** | Student chooses a file to upload. | |
| **4** | Student will read Term and Condition and agree with it. | |
| **5** | Student click submits button. | |
| **6** | The article display in the list. | |
| Exception Flow: Student cannot submit an article | | |
| **1** | The student cannot upload the file. | |
| **2** | Check if the file is valid. | |
| **3** | Save the file name or save it as another file  (Return step 3 in Basic flow) | |

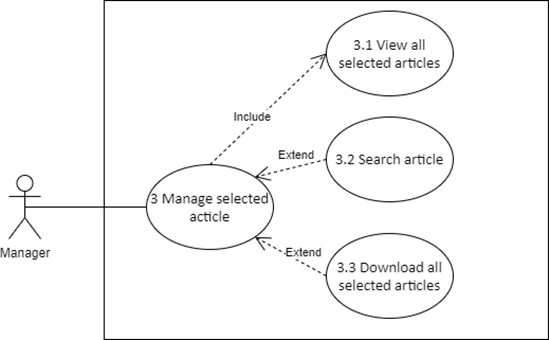


Figure 10: Use case of manager

For managers, their work only affects the posts approved by the coordinator. They can view the entire list of selected articles. They can also search for certain articles. Most importantly, they have the right to download them in their entirety.

|  |  |  |
| --- | --- | --- |
| Use Case Identification and History | | |
| **Use Case ID:** | | 3.1 |
| **Use Case Name:** | | View article |
| **Description:** | | Manage will view the selected article. |
| **User/Actor:** | | Manage |
| **Trigger:** | | Manage to want to view the article. |
| Pre-conditions | | |
| Manage is logged into the system  A webpage is displayed successfully | | |
| **Basic Flow: Manage view article successfully.** | | |
| **Step** | **Action** | |
| **1** | Manage choose view contribute. | |
| **2** | Manage will choose to contribute. | |
| **3** | The system will display the list article. | |
| Exception Flow: Manage cannot view the article. | | |
| **1** | Reload the website. | |
| **2** | Repeat all step in the basic flow. | |
| **3** | If still do not, the manager will report with the admin. | |

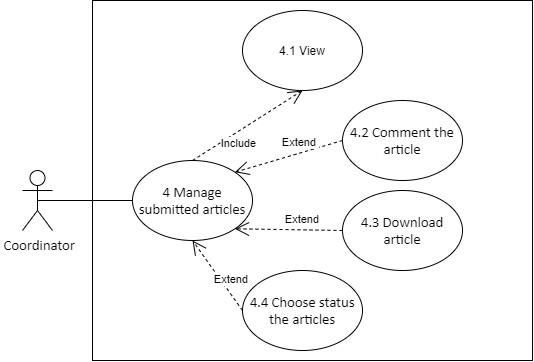


Figure 11: Use case of coordinator

The coordinator will be the one responsible for reviewing and making editing requests for the posted articles. Therefore, they can view the list of articles. Besides, to help with reading articles for the coordinator, they have the right to download articles. The coordinator can then comment under posts so that students can better edit their posts. If the coordinator finds that the article is eligible, they will choose to print it into a magazine and vice versa.

|  |  |  |
| --- | --- | --- |
| Use Case Identification and History | | |
| **Use Case ID:** | | 4.1 |
| **Use Case Name:** | | View article |
| **Description:** | | The coordinator will view the submitted article. |
| **User/Actor:** | | Coordinator |
| **Trigger:** | | The coordinator wants to view the article. |
| Pre-conditions | | |
| The coordinator is logged into the system  The webpage is displayed successfully | | |
| **Basic Flow: Coordinator view article successfully.** | | |
| **Step** | **Action** | |
| **1** | Coordinator choose contribute in menu. | |
| **2** | The system will display the list article. | |
| Exception Flow: The coordinator cannot view the article. | | |
| **1** | Reload the website. | |
| **2** | Repeat all step in the basic flow. | |
| **3** | If still do not, the coordinator will report with the admin. | |

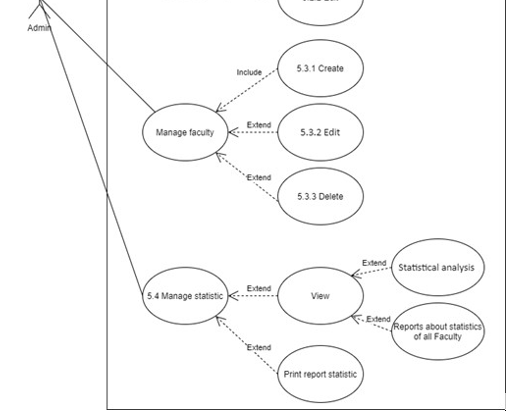
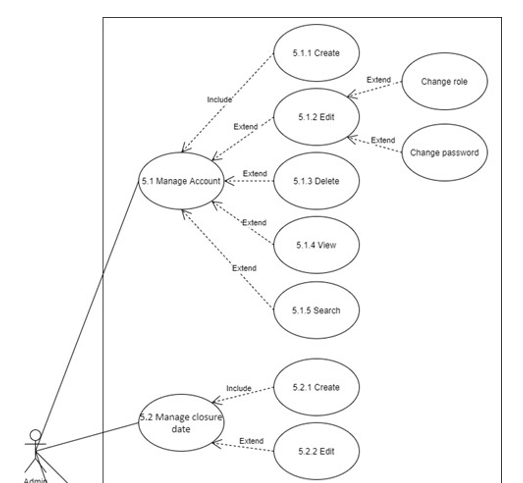


Figure 12: Use case of admin

Admin is the person who has a lot of responsibility for managing this website. Therefore, the admin has the most functions in this system. New users all have to ask the admin for their account, so the admin will be the manager of accounts including adding, deleting, searching, and viewing. Admin can also change passwords for users if required. Every year, when it comes to collecting student contributions, the admin will create a new term and set a time for it. Admin can also edit the time as needed. Also, administrators can manage faculties by adding faculties, editing faculties, or deleting faculties. Finally, the admin will be able to view the statistics and reports of those statistics.

|  |  |  |
| --- | --- | --- |
| Use Case Identification and History | | |
| **Use Case ID:** | | 5.1.1 |
| **Use Case Name:** | | Create account |
| **Description:** | | Admin will create an account. |
| **User/Actor:** | | Admin |
| **Trigger:** | | Having a new user. |
| Pre-conditions | | |
| Admin is logged into the system  The webpage is displayed successfully | | |
| **Basic Flow: Admin creates account successfully.** | | |
| **Step** | **Action** | |
| **1** | Admin will choose add user in menu | |
| **2** | Admin will select a role for a new user and click add. | |
| **3** | The system display form to create. | |
| **4** | Admin fills in information for new user. | |
| **5** | After, the admin choose button add. | |
| **6** | The user will display it in the list. | |
| Exception Flow: Account don’t display in the list. | | |
| **1** | Reload the website. | |
| **2** | Repeat all step in the basic flow. | |

For each use case, we write a scenario for it. Since the total number of use cases here is pretty much so for each actor, we just leave one representative scenario. The remaining scenarios will be recorded in the word file and can be viewed here: <https://drive.google.com/drive/u/0/folders/1n1u-L-E3TbeUTHtYiHtILQE_bqdq3HtH>

#### Class Diagram

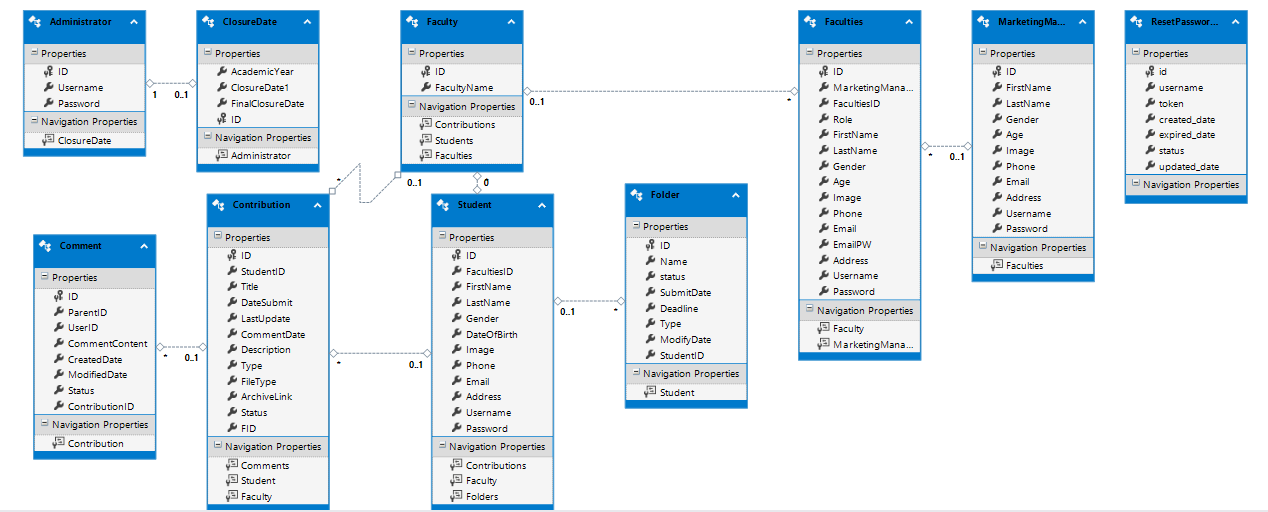
****

Figure 13: Class diagram of system

#### Sequence Diagram

Sequence diagrams show the user's interaction with the system and they are drawn based on their scenario.

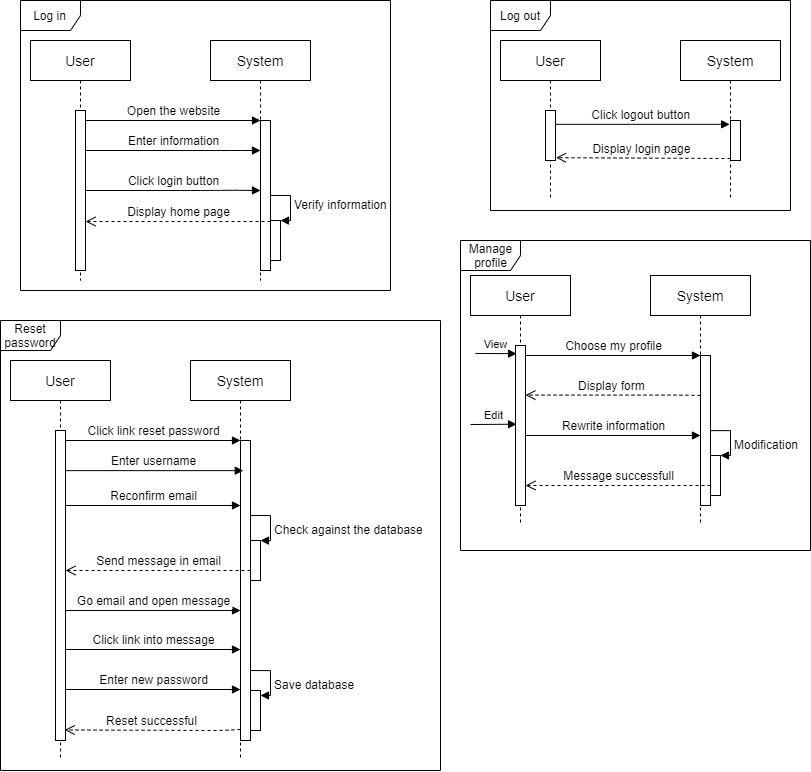


Figure 14: Sequences of user

This is the process where the user does basic things like login, logout, reset password and edit profile. Each drawing will represent a function and user interaction with the system and vice versa.

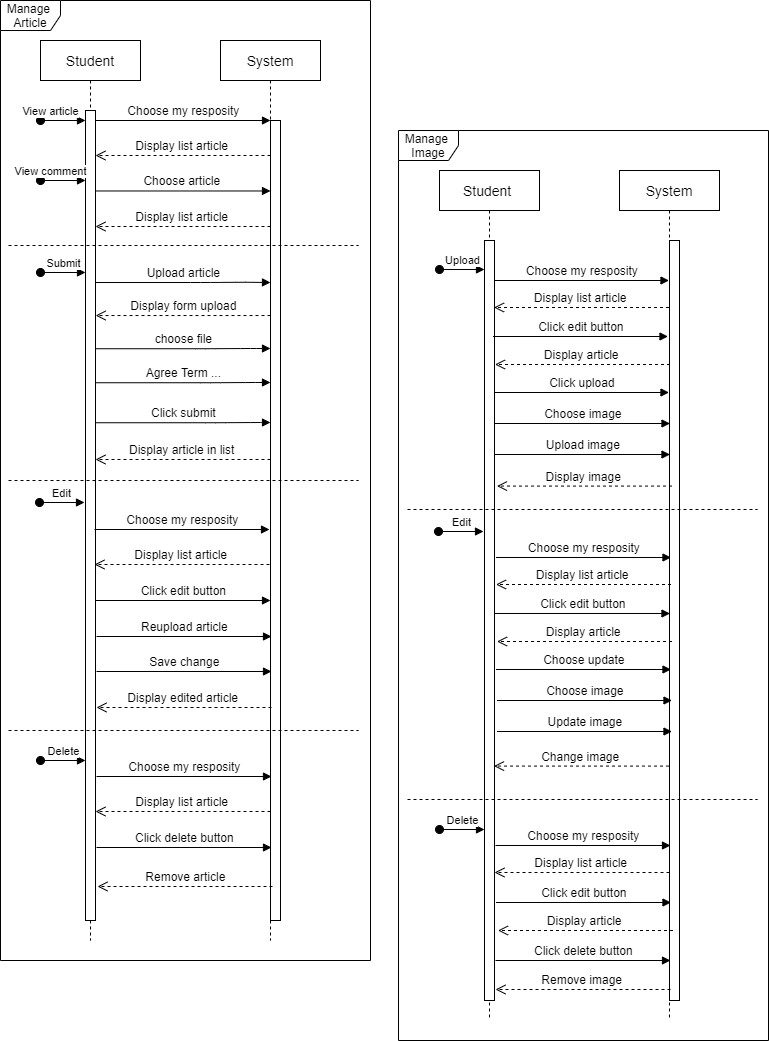


Figure 15: Sequences of student

For students only the functions of article management and image management. Although it sounds simple because it is usually just simple tasks such as adding, deleting, and editing. But in the situation of this website when we draw the sequence, we will see different features and important details. That makes the coding process easier.

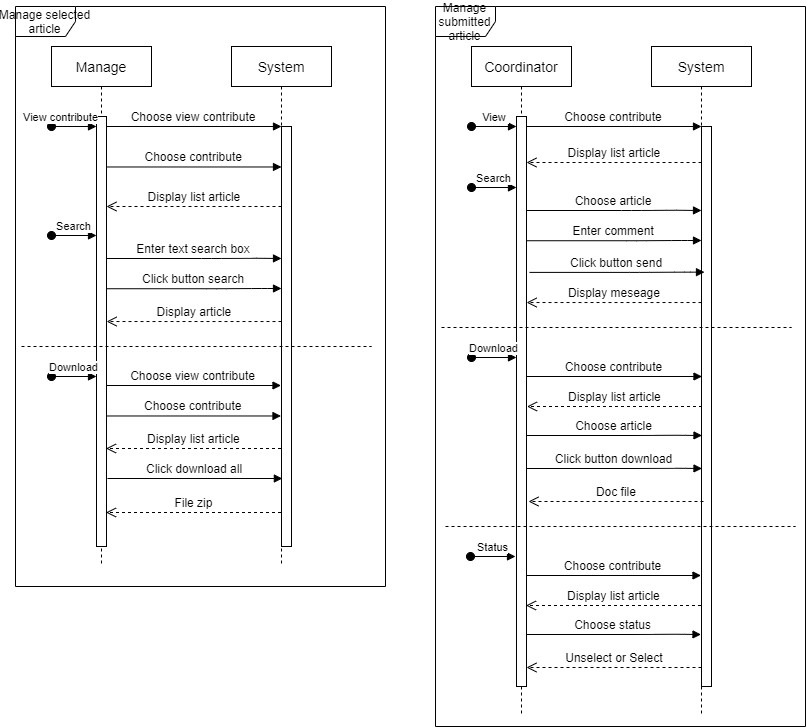


Figure 16: Sequences of coordinator and manager

Although the functions of the coordinator and manager are quite small, it also requires clear steps so that when implementing the code, the developer will not be idle and easier to perform.

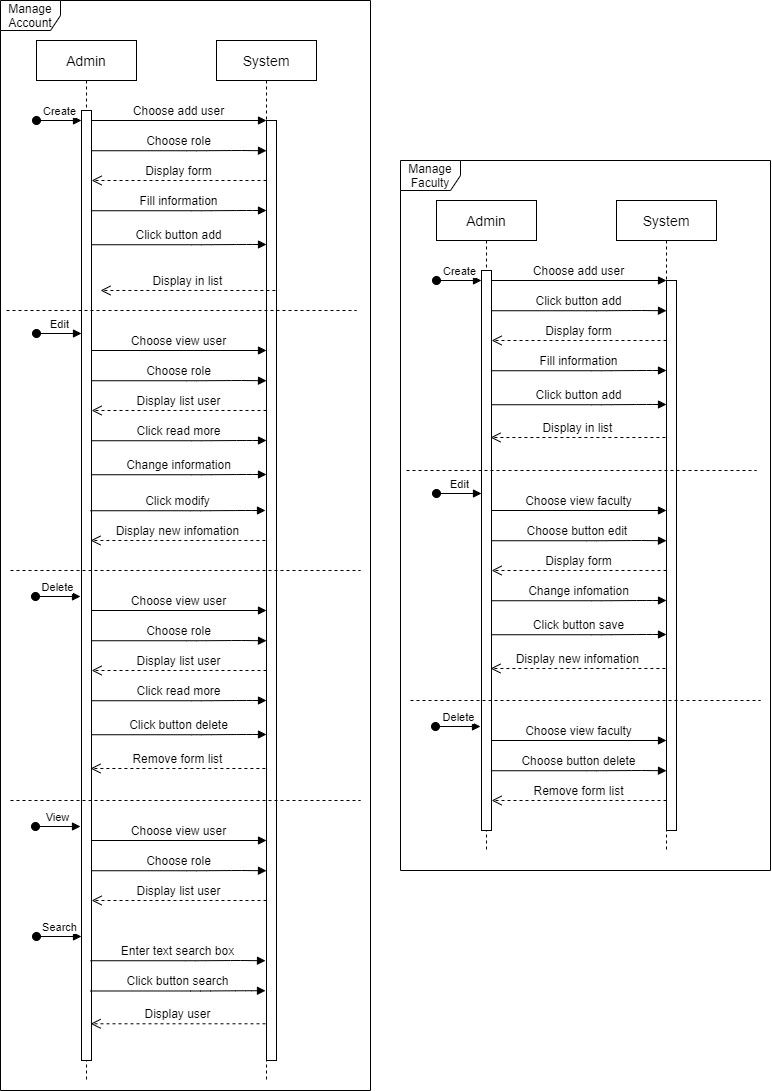


Figure 17: Sequences of admin

On the part of admin account management, we can think at first glance. However, since it has too much information, each account will be assigned its own role. Therefore, there is a need for a sequence that demonstrates the interaction of the admin with the system to better support the coder. As for the faculty management function, it is a function that is rarely used and has little information, so its interaction is quite simple.

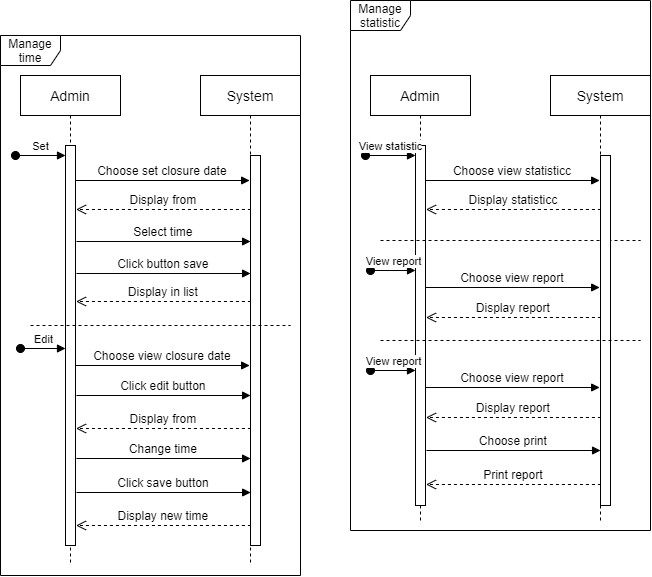


Figure 18: Sequences of admin

The function of setting time periods for submission is quite important in the system so I need to take steps properly and make the interaction easier. Although the functions related to statistics are not very important, sequences are still required for the programmer to know how the flow is.

#### Database design

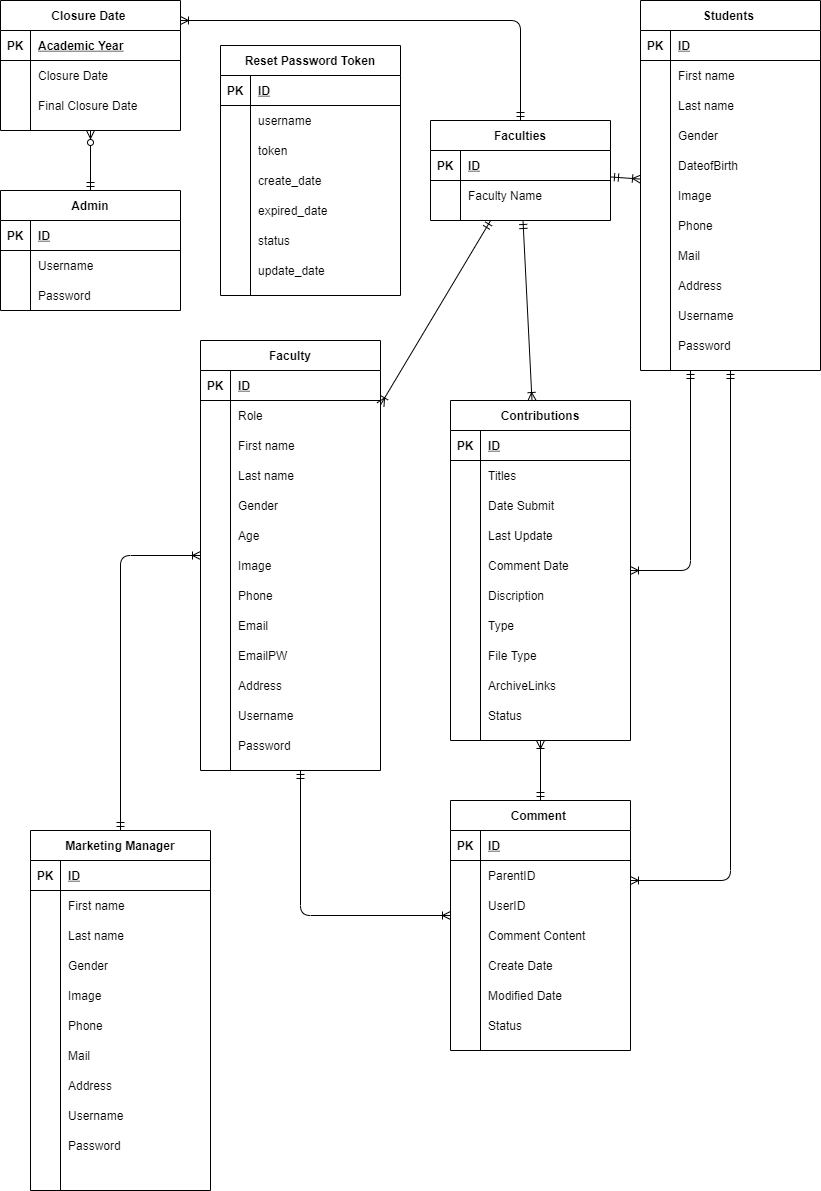
****

Figure 19: Database diagram of the system

The system's data map has 9 tables of Marketing Manager, Amin, Faculty, Student, Faculties, Contributions, Comments, Closure Date and Reset Password Token. Each table represents the components in the system and the specifics of each table are as follows:

* Marketing Manager: ID, First name, Last name, Gender, Image, Phone, Mail, Address, Username, Password
* Admin: Username, Password
* Faculty: ID, Role, First name, Last name, Gender, Age, Image, Phone, Email, EmailPW, Address, Username, Password
* Students: ID, First name, Last name, Gender, DateofBirth, Image, Phone, Mail, Address, Username, Password
* Faculties: ID, Faculty Name
* Contributions: ID, Titlles, Date Submit, Last Update, Comment Date, Description, Type, File Type, ArchiveLinks, Status
* Comment: ID, ParentID, UserID, Comment Content, Create Date, Modifiled Date, Status
* Closure Date: Academic Year, Closure Date, Final Closure Date.
* Reset Password Token: ID, username, token, create\_date, expired\_date, status, update\_date.

## IMPLEMENT

### Operating environment

* Operating System: Window
* Database: SQL Server
* Platform: ASP.NET

### Program Language

* Front-end: HTML, CSS, JavaScript and Bootstrap
* Back-end: C# and framework MVC

### Website construction structure

We choose the ASP.Net MVC framework to do the website development. So, when using MVC, it will divide the pattern into 3 main parts: model, controller and view. We will explain the structured flow for using MVC in detail as follows:

* The model is where we create the classes of the database to store information linked together in the website.
* Controller is the brain that manages the user interactions on the website. It will contact the model to get information from the updated data
* View will be the working place of the website display interface when users visit the website to see data.

Besides, the main component also needs support from other subsections to make up a complete website such as

* Folder “Content” contains CSS files for better website layout design.
* Folder "Script” will create more professional web effects, attract more eyes.
* Folder "Avatars" contains images to create highlights for websites such as logos.

****

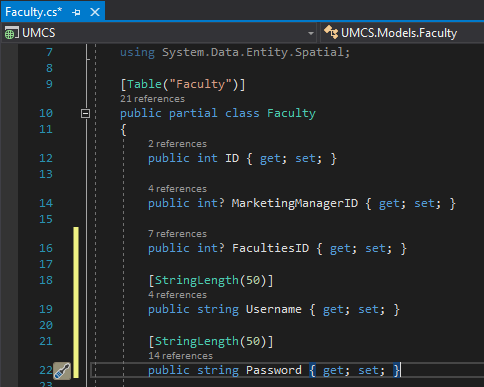
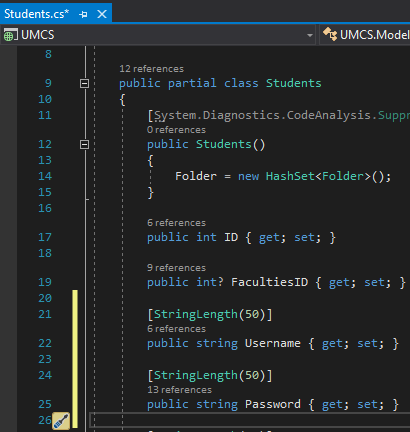
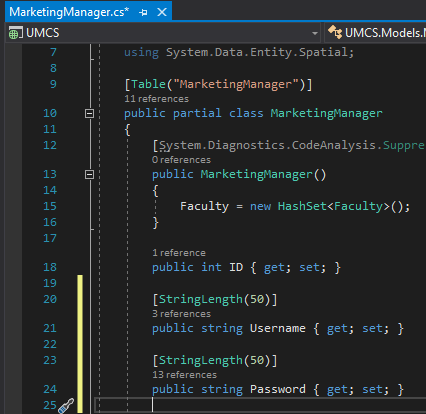
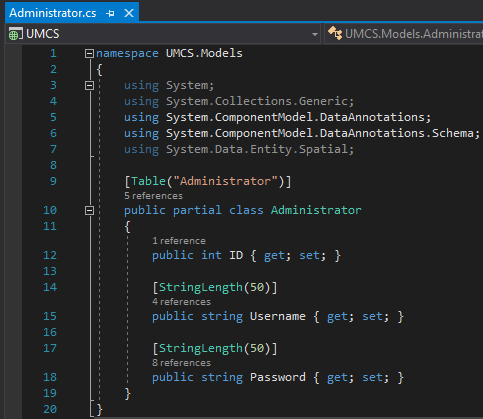
Figure 20: Website construction structure

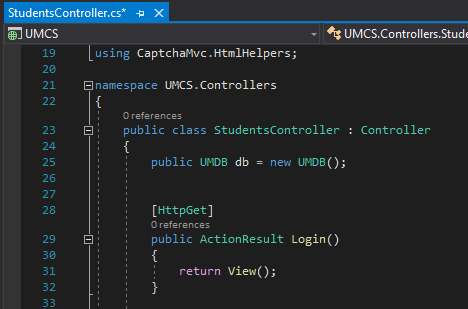
### Coding

#### LOGIN (user)

According to the requirements of the system, the website has 4 web users, namely admin, Marketing Manager, Marketing Coordinator, and Student, corresponding to different functions in the system. To not wrong between the permissions on the login is the first.

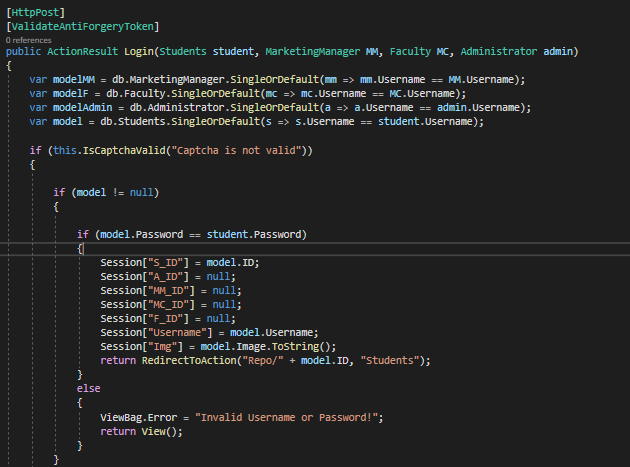
In Model, we have created class objects that all have the ID, username, and password of the field to distinguish the different objects.



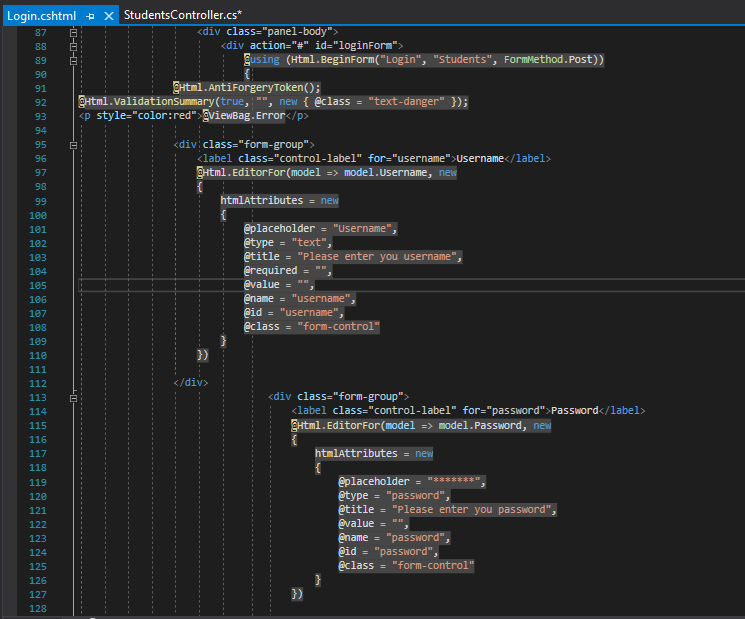


Continue we configure the settings of functions such as login, logout, and authorization at **StudentsController.cs.** When the user enters account information from **View/Students/Login.cshtml**, the system will pass the value to **StudentsController**.

Login () will check the Username and Password in the data, if correct, will attach the session to the user and redirect to the user's pre-installed home page.



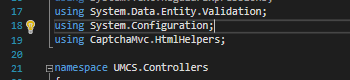
We call the data from the object tables to check if the account already exists in the data. Use *SingleOrDefault()* to specify during the query only a single result (username). Here we are working on the student object, in case the password in the data matches the user (student) password, the system will switch to the page that is visible to the student object and information such as the student's username and photo will appear using the Session command. And vice versa, if the password is wrong or captcha will display the message "Invalid Username or Password!" or "Error: captcha is not valid."



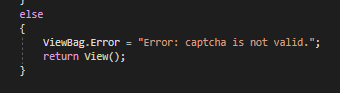
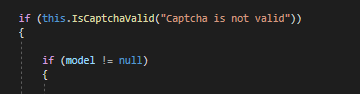
At **View/Students/Login.cshtml**, we create a login form interface. This is the page that will appear on the website interface, where users enter account information.

#### Captcha (User)

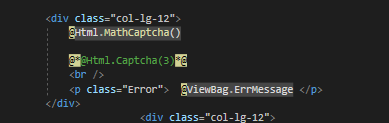
To enhance the security of the website, we use the captcha to avoid automatic programs or software using the system. We use the ***CaptchaMVC*** library (1). Captcha is used for login functionality so we test the captcha in ***StudentController.cs*** and use the library with the following command line.



At **StudentController.cs**, captcha will be checked with two cases, enter correct captcha, username and password, the system will switch to the specified page to display. And vice versa, enter the wrong captcha will show " captcha is not valid."



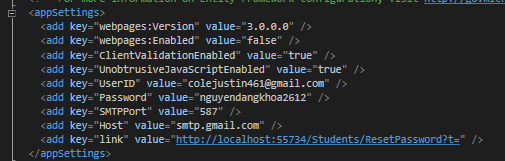
At **View/Students/Login**, display Captcha in the login form, ***@Html.MathCaptcha()*** will create an image of the math, *@Html.Captcha(3)* will create a length of 3 characters.



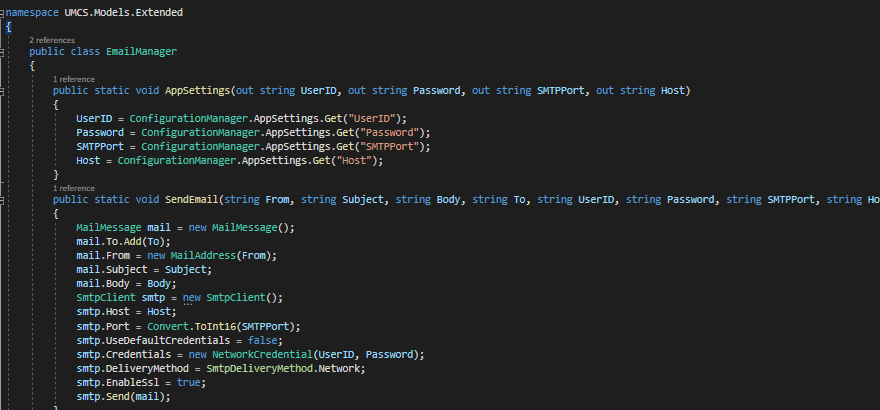
#### Send Email (Student, Marketing Coordinator)

Send mail in the system to inform website users about the tasks to be done in the system. Objects with the function of sending emails are Marketing Coordinator and Student.

First, to implement this function, we will edit the configuration in **Web.config** to edit email information such as email display, admin email, host, etc.

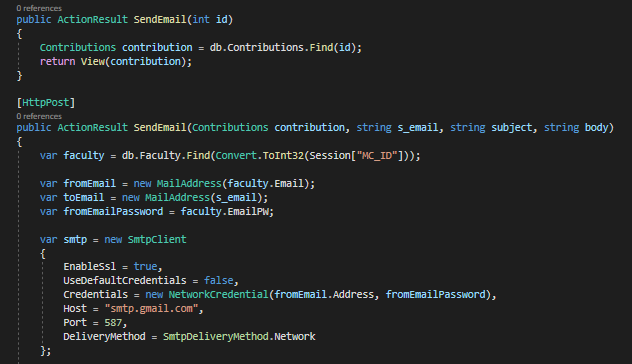


Then we create the **EmailManager.cs** class, edit the content and layout of the outgoing mail such as the mail header (mail.Subject), whether the email information is used hmtl (mail.Body), etc.



* For Marketing Coordinator

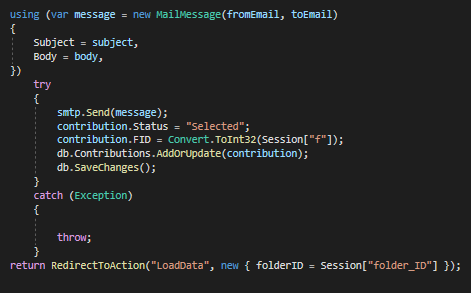
The Marketing Coordinator their job is to select and comment on student articles. Therefore, the mail announcing the number of articles to be commented on will cause the Marketing Coordinator to allocate suggestion time to the students.



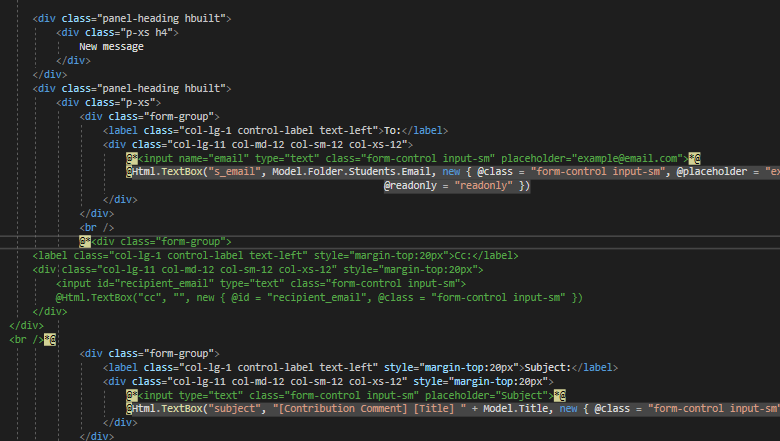
The Marketing Coordinator will send an email to the student to let her know if the student's articles have been selected. We get data from database information such as email sender, receiver.

* “*fromEmail”* will get the email address of the Coordinator
* “*toEmail”* will get the student's email.
* At the same time, we have also tweaked **smtp** for successful mail delivery.
* *EnableSsl* means secure mail sending
* *Credentials* authenticate the connection by providing an account (google email address) and a password.

Use the try-catch command to use if the mail is sent successfully, it will return to the LoadData page. In the event of a failure, it will stop immediately.

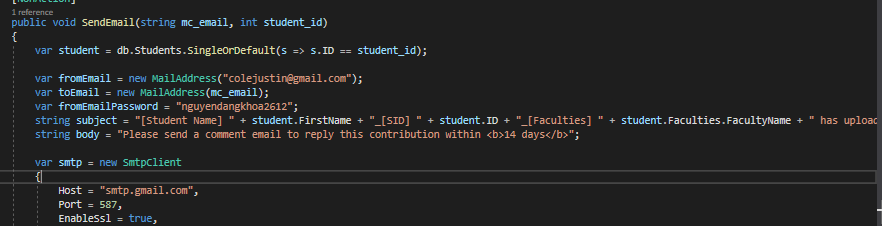


**View/MarketingCoordinators/SendEmail.cshtml** will display the form to send email to Student by Coordinator



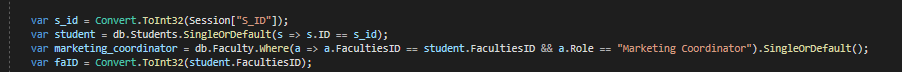
* For Student

After students submit articles, the system will **automatically send** an email to the Coordinator-email of the administrator (nguyendangkhoa2612) with Student information from the database. The title will be assigned with *FirstName, ID, Faculties* with the text "has uploaded a file to the system!". Contents "Please send a comment email to reply to this contribution within".

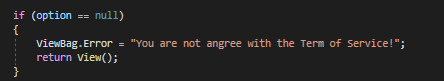


#### Submit articles (Student)

Here, we will pull data from the Student table and Faculty table. Because of the links between the tables, in the faculty there will be student information submit articles and the student's marketing coordinator.



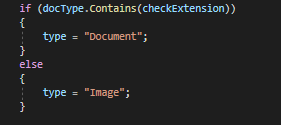
Students, after creating the topic name folder, we use the if statement. If the student does not accept the condition, the submission action cannot be performed.



We use the ***allowedExtensions*** command to control the files that can be uploaded to the website. The website accepts students to submit files with the extension ".doc", ".docx", ".jpeg", ".jpg", ".png", ".gif". And check whether the file condition is valid or not.



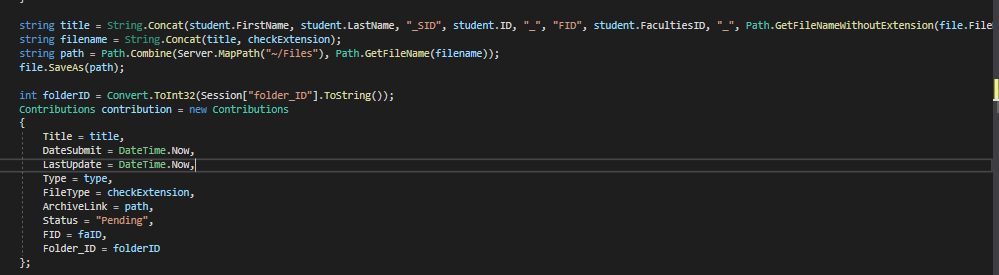
The command ***checkExtension*** to check if the file extension matches the request or not. If the file submitted is not correct, there will be a message saying "Invalid file type!"



***Server.Mappath*** will convert a virtual path that makes the file system on the server understandable. "~" is the root directory for the page.

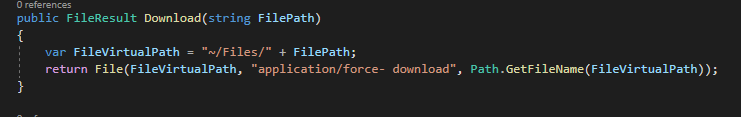


After successfully submitting the file, the file name will be changed to another name and displayed according to the following information string that has been installed. Then the system will send emails automatically to the Coordinator. And the status of articles after being submitted will be "Pending".

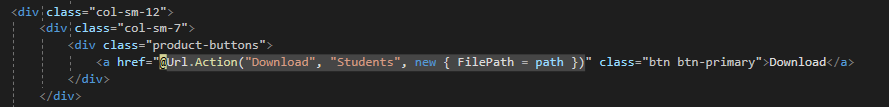


#### Download file

Calling the ***FileVirtualPath*** variable is a link that will associate the file location with the virtual location. Then, call the file back from its path based on the filename.

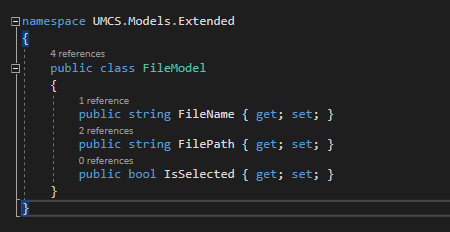


At View, implement the interface of the download button



#### Download file ZIP

We use the DotNetZip library, then create class FileModel with properties that perform ZIP file functionality (2).



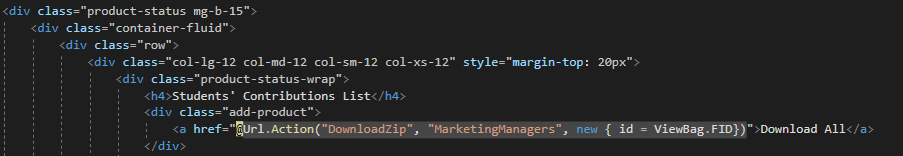
The file will be read from the Directory including the File details items and the path to the FileModel class. The *foreach loop* will be executed against the files according to the selectedContributes list of articles.



After the user executes the command, the files in the list will become objects of the DotNetZip library. The files will be downloaded in ZIP format, showing the actual date ZIP.



At View, make a button Download All



We have explained some of the main code snippets in the article and the full code of this website can be viewed without this link: <https://drive.google.com/drive/u/0/folders/10FM_p3Zw0l1Lflsv1KCm-CG1OM1bctnT>

We also went back to web hosting and its functionality. Click this link to see the clip: <https://drive.google.com/drive/u/0/folders/1isFve9Xax24gy8oywYa90BTE1_7c9WwF>

### Presentation

After completing the project, we have a lecture for the class and the teachers. During that session, we introduced our group. The work is done by each person and the project process follow the scrum process. Here is our presentation: <https://drive.google.com/drive/u/0/folders/1y1yjjgCRauriALlrqbYEn9W77bOMKWpd>

## TESTING

For the testing process, my team has fulfilled all the requirements such as test plan, test log, and proof of test finding error. The function pass rate of the product is 80-90%, our team has tested according to the process when the code has failed functions above 40%, then re-code and retest so the test situation is fully updated according to the test case. My team checked to follow the goals outlined in the test plan and recorded who did it. During the testing process, our team had some faulty functions, but only some errors were fixed and some errors did not have enough time to fix. After testing, my team saw some advantages about my test such as linking functionality to the product backlog, having test cases, testing steps, desired results, and actual results. On the downside, our team only tested according to the requirements set out in the test plan but lacked the process of finding bugs in the system. Through the testing process, our team learned how to test the system and how to perform testing. Although, there are some checks that my team doesn't have enough time to do such as test data, the bug of the system. But our team tried very hard to achieve the goal set out by the test plan. From taking the test, my team gained more experience to implement future projects and knew their strengths and weaknesses, and overcome them for the better. The link is down below <https://drive.google.com/drive/u/0/folders/1aDVM717mzH1RIXgkAxAaEU0zXWjJaz66>

# SECTION B – INDIVIDUAL REPORT

## EVALUATION OF PRODUCT

Overall, the system has been well designed and developed to respond to almost complete application requirements already out of the system.

Regarding the database system, full of data for the objects using the system and the functions of the system. Table data is named, easy to understand; the field data in the table details is almost absolutely accurate. Building systems between tables aims to link information between tables very well. The proof is that the speed of querying data between tables is very fast and closely. Database documentation is also prepared in detail from ERD design drawings by draw.io software to building models for the system. However, the current issue of security is still weak, if we have more time, we will fix it immediately.

Regarding the process of building the interface, the link between UI and UX is very tight. First, we implement a sitemap so we can use the functionality of the objects in the system. Then, conceptualize the interfaces of individual objects (UI) to detailed design using Figma tool design. Besides, during the design process, we made the system manual song and below each figure cut from the design, there is an additional explanation. All functions must be designed and designed in different dimensions. During the design implementation, problems that arise are inevitable such as the design lacks a function in an object, page confusion, etc. But in the end, after much discussion, the problems were resolved. to handle.

The development of the system's functions is also quite stable. Based on the requirement listed in the Product backlog, some of the functions that we perform are being used by nearly 90% of the total listed functionality. Besides functional development, we are also interested in functions such as improving security. We may use additional image validation when performing login and adding a function in the report. And feedback is configured on the screen of the form. At the end of each sprint, the commissioning begins and the results are given, if corrections are needed, it is put into the next sprint or run. At the sprint end, we test the system and calculate its success rate. The standard document is for too a running try to be done from the first date for the console interface. Most are on the list of running successful tests, but there are also a few sub-functionalities rated as unstable.

## EVALUATION OF PROCESS

The system build is pretty good for a team that started using Scrum after learning how Scrum works. System development and required documents are systematically linked and carried out in parallel. Members are free to comment or add comments. Analyzing the system requirements, assigning the members' work to the interface design stage is fine. However, at the functional development stage of the system, the initial tentative time we proposed hasn't ended. The actual feature building process is risky, and issues like the new feature need to be explored more and debugging takes longer than expected. However, using Scrum allows us not to worry about adding requirements or making adjustments to improve the system. We hope to be able to implement Scrum in the next project so we can focus on the project completely. As we have other topics, it is difficult to arrange a free time to meet, so we can only meet twice a week. Also, the documentation of the product creation process is stored in certain folders and saved by the meetings for retrieval later.

## EVALUATION OF TEAM

* Vo Hoang Yen: the analyst, responsible for the documentation in the process. Continuously support the remaining members when they have questions about the functional requirements of the system in her understanding. She is present in all group meetings.
* Nguyen Ngoc Nhi: Testing the system. She participates in all meetings and constantly takes care of people, supporting other members when they have problems in the project. She is a capable tester
* Nguyen Dang Khoa: A capable programmer. He is diligent and is always looking for the fastest way to build functions and suggest additional functions in the system. He is present at almost all meetings.
* Do Lam Hoang Anh: A good listener and patient who combines very well with requests from the other members. Skills to use quick, neat drawing tools. Streamlined and edited interface code reasonably. And is the one who records the daily meetings of the group. However, she's late, so one of the members will record it and send it to her later.
* Tran Thi Thanh Binh: She is in charge of database construction and functional development for the system. Ability to concentrate, communicate and explain work to other members. And she is present in all meetings.

All members make great contributions to the overall efforts of the group. They do very well their assigned tasks and provide great support to other members. On the other hand, there are still some people who make mistakes but still strive to complete the task.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Factors | Weight | Yen | Nhi | Khoa | Anh | Binh |
| Attendance | 10 | 10 | 10 | 9 | 7 | 10 |
| Attitude | 9 | 9 | 9 | 9 | 9 | 9 |
| Contribute | 9 | 8 | 8 | 8 | 7 | 7 |
| Communication | 10 | 10 | 9 | 10 | 9 | 10 |
| Complete the task | 10 | 10 | 10 | 10 | 10 | 9 |
| Total | 9.6 | 9.4 | 9.2 | 9.2 | 8.4 | 9 |

Figure 21: Factor Comparision

## SELF-EVALUATION

My task on the team is to design the database and program the functions in the system. After the first meeting and analyze the functions and objects with the whole group. I do a database design (ERD), and after discussion, we will modify the database to avoid missing fields in the data table. Then I build the database in the Model of the system. The second task is to develop functions in the system together with the Faculty. When I have a problem, I will ask for the help of other members. During the development of the system, I completed the assigned tasks on time. Participate in all group meetings. My strength is communication, so during meetings, I try to help the members of the group feel more comfortable, communicate with each other more by setting up a group chat on Zalo. At the same time, I learned how to document and how Scrum works, how to listen and synthesize opinions. There is some unsatisfactory functionality on the last testing, if there is more time I will correct the unsatisfactory functions.

1. **Conclusion**

The result of our team project was the magazine web system. Most of the functions in the system respond well to the requirements set out for the system. We use Asp.Net to do system development, allowing it to run on different platforms. And using the Scrum method to build a web system, has helped us complete the work on schedule and the communication between members is still very good. And understanding the process and implementation in Scrum, after this project has helped me gain more experience on how to work in the Scrum method, which will be useful for me in the future.

## REFERENCES

1. Nuget.org. *CaptchaMvc.Mvc4 1.5.0*. [online] Available at: <https://www.nuget.org/packages/CaptchaMvc.Mvc4/> [Accessed 17 April 2021].
2. Nuget.org. *DotNetZip 1.15.0*. [online] Available at: <https://www.nuget.org/packages/DotNetZip/> [Accessed 17 April 2021].
3. Colorlib.com. Dashboard V.1 | Kiaalap - Kiaalap Admin Template. [online] Available at: <https://colorlib.com/polygon/kiaalap/index.html> [Accessed 18 April 2021].
4. CanvasJS. Beautiful HTML5 Charts & Graphs | 10x Fast | Simple API. [online] Available at: <https://canvasjs.com/> [Accessed 18 April 2021].
5. Guru99.com. 2021. Test Plan Template: Sample Document with Web Application Example. [online] Available at: <https://www.guru99.com/test-plan-for-project.html#3> [Accessed 18 April 2021].
6. Softwaretestingmaterial.com. 2021. [online] Available at: <https://www.softwaretestingmaterial.com/test-case-template-with-explanation/> [Accessed 18 April 2021].
7. 2021. [online] Available at: <https://www.softwaretestinggenius.com/test-case-log-and-its-sample-template/> [Accessed 18 April 2021].
8. Tutorialspoint.com. 2021. STLC - Overview - Tutorialspoint. [online] Available at: <https://www.tutorialspoint.com/stlc/stlc_overview.htm#:~:text=STLC%20stands%20for%20Software%20Testing,the%20software%20or%20the%20product.&text=As%20soon%20as%20the%20development,cases%20and%20start%20with%20execution.> [Accessed 18 April 2021].
9. Guru99.com. 2021. TEST PLAN: What is, How to Create (with Example). [online] Available at: <https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html> [Accessed 18 April 2021].
10. Daily meeting report: <https://drive.google.com/drive/u/0/folders/1-1aCIaxQGRiSjodHfreMBZTmUVh0f0-C>