VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

AN ONLINE-TEACHING MANAGEMENT SYSTEM

By
Lê Quốc Vũ

A thesis submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Information Technology/Computer Science/Computer Engineering

Ho Chi Minh City, Vietnam
2022

AN ONLINE-TEACHING MANAGEMENT SYSTEM

APPROVED BY:

_____

Huỳnh Khả Tú

THESIS COMMITTEE
(Whichever applies)

ACKNOWLEGMENTS

First and foremost, I would want to express my heartfelt thanks to my supervisor, Dr. Huynh Kha Tu. I have gained considerable information and experiences as a result of her ideas and activities that drive me to face obstacles and make an effort to solve them. Due to her consistent support and tolerance, I have done my hardest to learn and enhance my talents along my journey to complete this Thesis. Her knowledge and pleasant demeanor are worthy of a master.

My second source of gratitude is the Department of Computer Science and Engineering, which has assisted me in grasping the basis through relevant courses. My Thesis work began shortly after the coronavirus (COVID-19) epidemic broke out. Students' health is constantly valued at International University, notably by the staff of the School of Computer Science, and online instruction is run effectively at the same time.

Last but not least, I would want to express my gratitude to all of my IU friends as well as my team: Ngan Chau, Phuong Linh, Le Thanh, Viet Hung, Hoang Quyet, Minh Ngoc, Mai Duyen for their support and friendship. They are all kind people who have not hesitated to share their experiences when I have asked for help. My family is another significant aspect that I would like to address. Thank you for making it possible for me to study in such nice surroundings, for lifting my spirits, and for constantly conveying a happy mood.

TABLE OF CONTENTS

ACKNOWLEGMENTS 3

TABLE OF CONTENTS 4

LIST OF FIGURES 7

LIST OF TABLES 9

ABSTRACT 10

CHAPTER 1 11

INTRODUCTION 11

1.1. Background 11

1.2. Problem Statement 11

1.3. Scope and Objectives 12

1.4. Assumption and Solution 12

1.5. Structure of thesis 12

CHAPTER 2 14

LITURATURE REVIEW/RELATED WORK 14

2.1. Techniques and Tools: 14

2.1.1. jQuery [1] 14

2.1.2. Datatables [2] 14

2.1.3. Bootstrap [3] 15

2.1.4. Microsoft SQL Server [4] 15

2.1.5. NetBeans [5] 15

2.2. Related Work 16

2.2.1. IU Edusoftweb [6] 16

2.2.2. IU BlackBoard [7] 17

CHAPTER 3 19

IMPLEMENTATION 19

3.1. Overview 19

3.2. User requirement analysis 19

3.2.1. Use case diagram 19
3.2.2. Sequence diagram 28
3.3. Database Design 30
3.3.1. Entity Relational Diagram 30
3.3.2. Database diagram 32
3.4. System Design 32
3.4.1. Class Diagram 32
3.4.2. Project Structure 33
CHAPTER 4 35
IMPLEMENT AND RESULTS 35
4.1. Implement 35
4.2. Results 35
4.2.1. Login and sign up page 36
4.2.2. Home page and edit user's information 36
4.2.3. Post a new post 38
4.2.4. Check class information 39
4.2.5. Student homepage 41
4.2.6. Add/delete course 42
4.2.7. Grant user's permission 44
4.2.8. Add course for student 44
CHAPTER 5 45
DISCUSSION AND EVALUATION 45
5.1. Discussion 45
5.2. Comparison 45
5.3. Evaluation 45
CHAPTER 6 47
CONCLUSION AND FUTURE WORK 47

6.1. Conclusion 47
6.2. Future work 47
REFERENCES 48

LIST OF FIGURES

Figure 1: jQuery 14
Figure 2: DataTables 14
Figure 3: Bootstrap 15
Figure 4: Microsoft SQL Server 15
Figure 5: User interface of NetBeans 16
Figure 6: IU Edusoftweb page 17
Figure 7: IU BlackBoard home page 18
Figure 8: IU BlackBoard Content page 18
Figure 9: Use case Diagram 20
Figure 10: SQ1 Post a new post 29
Figure 11: SQ2 Check a post 29
Figure 12: SQ3 Create new course 30
Figure 13: ERD Diagram 31
Figure 14: Database Schema 32
Figure 15: Class Diagram 33
Figure 16: Project Structure 34
Figure 17: Login page 36
Figure 18: Sign up page 36
Figure 19: Home page of instructor 37
Figure 20: Profile modal 37
Figure 21: Edit profile 38
Figure 22: Post a new class information declaration 38
Figure 23: Overview the post 39
Figure 24: Edit the post 39
Figure 25: Check homepage 40
Figure 26: Post verification 40
Figure 27: Choose instructor, choose subject 41
Figure 28: Show all posts of a subject 41
Figure 29: Student homepage 42
Figure 30: Admin function 42
Figure 31: View all courses 43
Figure 32: Add new course 43
Figure 33: Delete courses 43

Figure 34: Show all users 44
Figure 35: Grant user's permission 44
Figure 36: Add course for student 44

LIST OF TABLES

Table 1: Assumption and Solution 12
Table 2: UC1 Login 21
Table 3: UC2 Create an account 21
Table 4: UC3 Grant account access 22
Table 5: UC4 Post a new post 22
Table 6: UC5 Update a post 23
Table 7: UC6 Delete a post 24
Table 8: UC7 View posts 24
Table 9: UC8 Check a post 25
Table 10: UC9 Update profile 26
Table 11: UC10 Create a course 26
Table 12: UC11 View all courses 27
Table 13: UC12 Delete courses 27
Table 14: UC13 View all users 28
Table 15: UC14 Add student course 28

ABSTRACT

Nowadays, people are dealing with the covid epidemic; everyone is advised to go out only when absolutely necessary; many companies have also allowed employees to work from home; and schools have also been closed, causing students to be unable to study at school but to do so online through computer screens. It gets tough to learn when you are unable to study face to face. Many approaches have been used to manage the online education system today, but very few of them are efficient and well-built owing to a lack of specifics and the inability to enhance further as some areas of the education system are constantly updated. This problem necessitates a more effective solution. Traditional approaches, however, have failed this mission, resulting in the failure and inefficiency of this system in current times. Many groups and companies have been doing research to solve this problem by applying various technological approaches to the system; nevertheless, not every solution can be employed intelligently or acceptably owing to incompleteness or unproductiveness. The research will establish a system to handle weekly lessons from instructors, faculty secretaries, and students based on that to acquire information, manage, and examine as a solution to enhance the education system as well as manage the quality of online teaching.

CHAPTER 1

INTRODUCTION

2.1. Background

Nowadays, the advancement of information technology (IT), particularly the Industrial Revolution 4.0, has altered our way of life. People can enhance their life quality by using familiar tools or automatic applications. Especially in today's world of covid, it is critical to use information technology to handle online education. How can we handle thousands of online classes at a large institution in the field of university management? That is an issue that can be readily handled by employing computer software.

To solve these problems, I propose a system to manage all lessons posted by instructors at the institution called "Online-teaching management system" (OTM) which I have been researching at the School of Computer Science and Engineering at the International University of VNUHCM. The OTM is built using Web Application Development, Database System, and Information System Management methodologies.

The OTM offers services for managing all lectures that posted by instructors. Before each lesson, the instructor will declare the information of the class on the system, such as today's lesson content, lesson slides as well as the link to attend the online class. Students and faculty secretaries may quickly access the system to learn about lesson material, download lectures, and administer online classrooms. As a result, the OTM can store and manage all lessons information that instructors and faculty secretaries can communicate with the system to perform their tasks. It is easy to use and reuse for many sorts of customers and so on.

2.2. Problem Statement

There are many problems that instructors, as well as faculty, are faced when evaluating an online course through the traditional method. In general, the problem that they are facing is that they have difficulty managing online classes. Because everyone is studying and teaching at home, they cannot go to every classroom to check if the instructor is there to teach. Furthermore, it is difficult to determine whether the lesson content is truly beneficial and relevant to the subject being taught. Besides, the issue is in the data structure, which means that the recorded by using manual storage and requires a huge number of papers to store instructional information and performance. The solution to these problems is handled by creating a system to manage instructors' lectures as well as directly participating in online classes to manage students, manage teaching content, etc, and storing all the data in the database for future enhancement. This system should help academic units such as faculties

and departments to actively follow and monitor the quality of teaching as well as the teaching history and profile of lecturers.

2.3. Scope and Objectives

This project will cover all OTM functions that help instructors, faculty, and students, especial faculty to manage online class.

- Instructor: Posting weekly class information, check and manage posted content.
- Student: Check the weekly class content, get the slides of each lesson and join the online class.
- Faculty Secretary: Check the content of the posts posted by the teacher is correct, join the online class quickly to check the teacher's teaching, take attendance of students... Manage all the content of the teacher's courses that have been posted.

The main point of this project is: Upload the content of each lesson and manage that content.

2.4. Assumption and Solution

Assumption	**Solution**
Delete information wrong	Do not delete actual data, just set isDelete column to 1 to keep track
Transition to a new semester	Hide old semester data, update new semester data
No longer teaching online, but instead teaching offline	Can still be used to post class and lecture information

Table 1: Assumption and Solution

2.5. Structure of thesis

This thesis project will be divided into six major sections:

- Chapter 1 Introduction: learn about the present school management system's background issue.
- Chapter 2 Literature Review: a summary of the techniques, framework, tools, and algorithms that will be used in the project.
- Chapter 3 Methodology: determine the system architecture, create the use case diagram, sequence diagram, and database diagram for this management system.
- Chapter 4 Implementation and Results: the picture will be given to the analyst once the results are met and the search aim is reached.
- Chapter 5 Discussion and Evaluation: a summary and discussion of the entire project.

- Chapter 6 Conclusion and further work: the conclusion summarizes the whole project and identifies the next work that has to be done to improve the project in the future.

CHAPTER 2

LITURATURE REVIEW/RELATED WORK

2.6. Techniques and Tools:

In this chapter, I will go through the techniques and tools that I use to create my web application. It explains why I chose these strategies and technologies.

2.1.1. jQuery [1]

The jQuery library is a JavaScript library. It was developed to assist developers in creating modules using JavaScriptsimpler. jQuery has a number of modules, ranging from an animation module to a selection module. jQuery encapsulates many typical operations that require multiple lines of JavaScript code into methods that can be called with a single line of code. Ajax, Attributes, Effect, Event, Form, DOM, and Selector are the main modules of jQuery. There are many alternative JavaScript frameworks available, but it appears that jQuery is the most popular and adaptable. This framework is used by several well-known websites, including Google, Netflix, and Microsoft.

Figure 1: jQuery

2.1.2. Datatables [2]

Datatables is a jQuery Javascript library plug-in. It is an excellent tool for completing operations involving tables shown on HTML pages. Datatables provide several advantages, including pagination, fast search, and multi-column sorting. It can leverage practically any data source, including DOM, Ajax, and server-side processing, and is therefore readily compatible with jQuery UI and Bootstrap.

Figure 2: DataTables

2.1.3. Bootstrap [3]

Bootstrap is a framework that makes website design easier and faster. Bootstrap comes with HTML templates, CSS templates, and JavaScript to let you design typography, forms, buttons, tables, navigation, and more. Furthermore, Bootstrap is compatible with all browsers, including Chrome, Firefox, Safari, and Opera, as well as mobile device interfaces, because Responsive is supported by this framework. Furthermore, because it is built on fundamental HTML, CSS, and Javascript understanding, it is incredibly simple to use.

Figure 3: Bootstrap

2.1.4. Microsoft SQL Server [4]

Microsoft SQL Server is a relational database management system that was created by Microsoft. As a database server, it is a software product that stores and retrieves data as requested by other software programs, which may operate on the same computer or on another computer over a network (including the Internet). Microsoft SQL Server is available in at least a dozen distinct versions, each geared at a particular audience and workload, ranging from simple single-machine applications to huge Internet-facing systems with many concurrent users.

Figure 4: Microsoft SQL Server

2.1.5. NetBeans [5]

NetBeans IDE is a free and open-source integrated development environment (IDE) for creating desktop, mobile, and online applications. The IDE facilitates the building of

applications in a variety of languages, including Java, HTML5, PHP, and C++. The integrated development environment (IDE) supports the whole development cycle, from project creation to debugging, profiling, and deployment. The IDE is compatible with Windows, Linux, Mac OS X, and other UNIX-based operating systems. The IDE fully supports JDK 7 technology as well as the most recent Java improvements. It is the first IDE to include JDK 7, Java EE 7, and JavaFX 2 capabilities. The IDE fully supports Java EE by utilizing the most recent Java, XML, Web services, and SQL standards, as well as the GlassFish Server, the Java EE reference implementation.

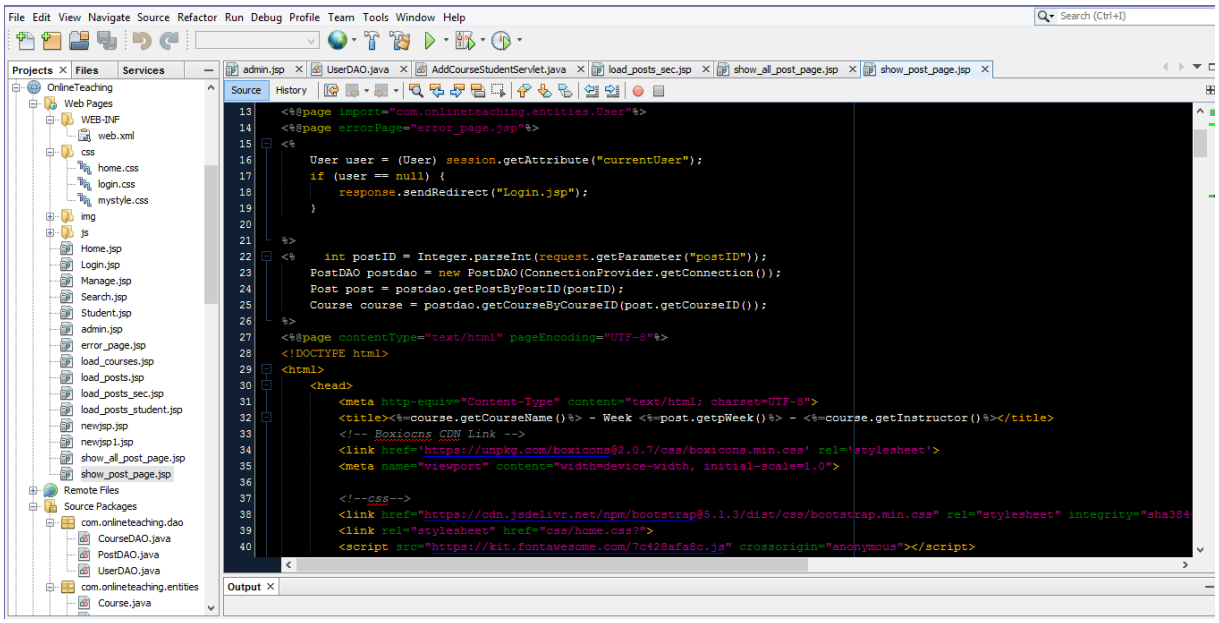

Figure 5: User interface of NetBeans

2.7. Related Work

2.7.1. IU Edusoftweb [6]

Edusoftweb is a website system related with Edusoft training management software, which is extensively utilized in many university systems and has the following key functions: View the whole school calendar, view the lecturer schedule, consider each student's specific timetable, research students' test schedules and re-examination schedules, sign up for online credit system training classes, re-enroll the student in the topic for the year, grading score assigned by instructors, etc.

Edusoftweb has the following advantages: First of all, Edusoftweb, which is utilized in IU and is based on webbase. It is simple to use in all stages, frameworks, and devices. The second advantage is that it may be utilized at any time of day. The third benefit is that it supports all basic functions for students and lecturers. Finally, a payment feature has already been included, making it simple to pay the tuition cost.

The disadvantage of this web service is that instructors cannot update lesson content and accompanying documents for students or faculty secretary to grasp the information. With the current function, faculty secretaries cannot manage online classes without using other tools.

Figure 6: IU Edusoftweb page

2.7.2. IU BlackBoard [7]

The Blackboard Learning Management System (LMS) is well-known as one of the best academic solutions that provide actual learning activities across the board, as well as extensive support with course content. There are some Backboard system's advantages: Easy to use, available across all platforms (Desktop, Tablets, Mobile, etc.), Allow students to update and download all submitted class materials, connecting students and teachers, doing quizzes and submitting assignments through the internet, easily adhering to the plans/deadlines stated in your timetable, having a variety of handy tools (Speed Grader, Turn-it-in, Chat box, etc.) However, this website only serves interaction for lecturers and students, in addition, the faculty secretary cannot check the status of the lessons as well as ensure that the organization of online classes is taking place according to the correct process or not.

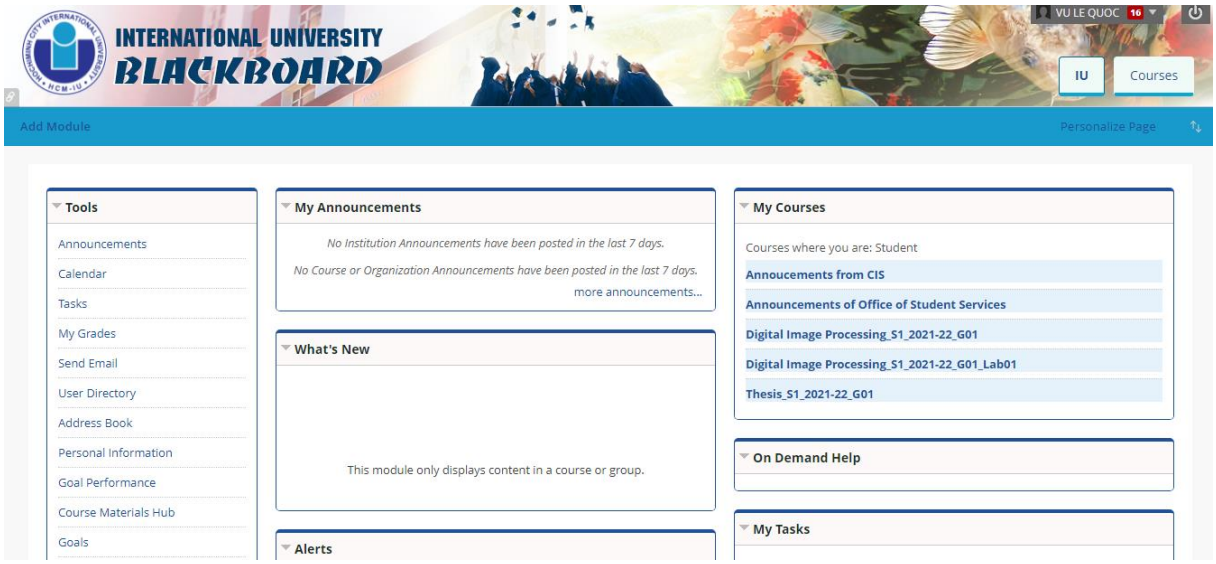

Figure 7: IU BlackBoard home page

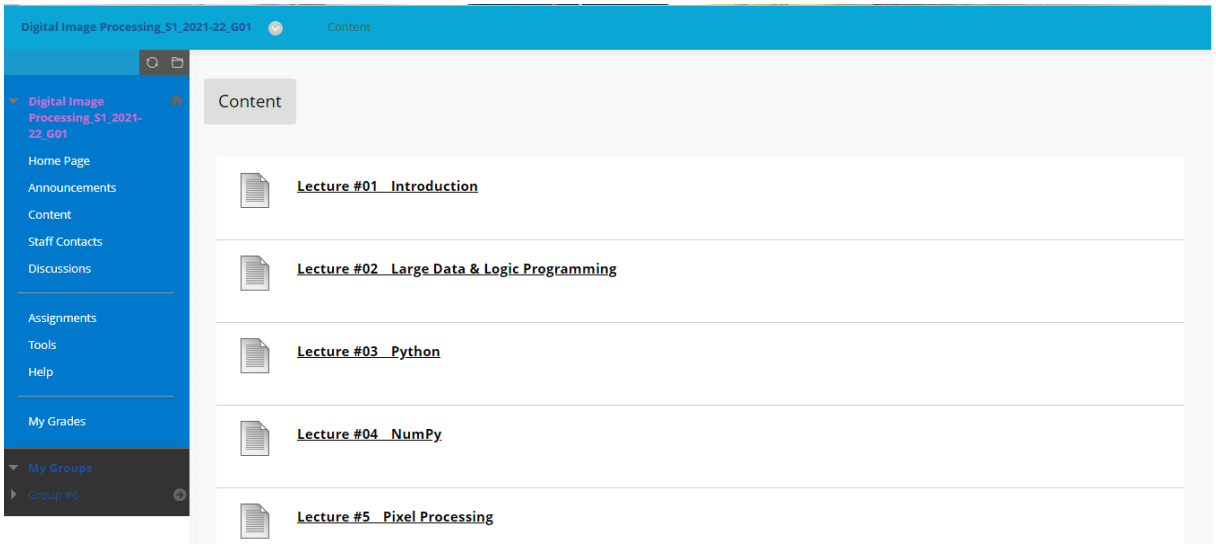

Figure 8: IU BlackBoard Content page

CHAPTER 3

IMPLEMENTATION

3.1. Overview

This chapter covers all aspects of methodology. The use case diagrams are created depending on the needs of the users. Use Cases clearly outline how actors interact with objectives. It also includes Sequence Diagrams, which depict the interplay of items to achieve system function. Finally, there is an Entity Relationship Diagram and a Database Schema, which show the design of the system and how it connects to all of its aspects.

3.2. User requirement analysis

Firstly, OTM will make it easier for the faculty secretaries to manage online classes. They can manage information about lessons, online rooms, manage students' class attendance, as well as check to see if the teacher shows up to the class fully and on time.

Second, the website allows users to examine and change their personal information. In addition, faculty can also review their own posts and edit and delete them, and check if the faculty secretary has checked the post.

Third, the system will display all the content by the week of each course, thereby helping the faculty secretary to easily collect data and use it for other purposes.

Finally, everyone can create an account to use this system. Each position is assigned to the account by the admin.

3.2.1. Use case diagram

Use cases are used in this section to identify and arrange the requirements. The application's four primary components are Faculty Secretary, Instructor, Student, and Administration... In addition, this section will demonstrate the database structure in order to improve the project.

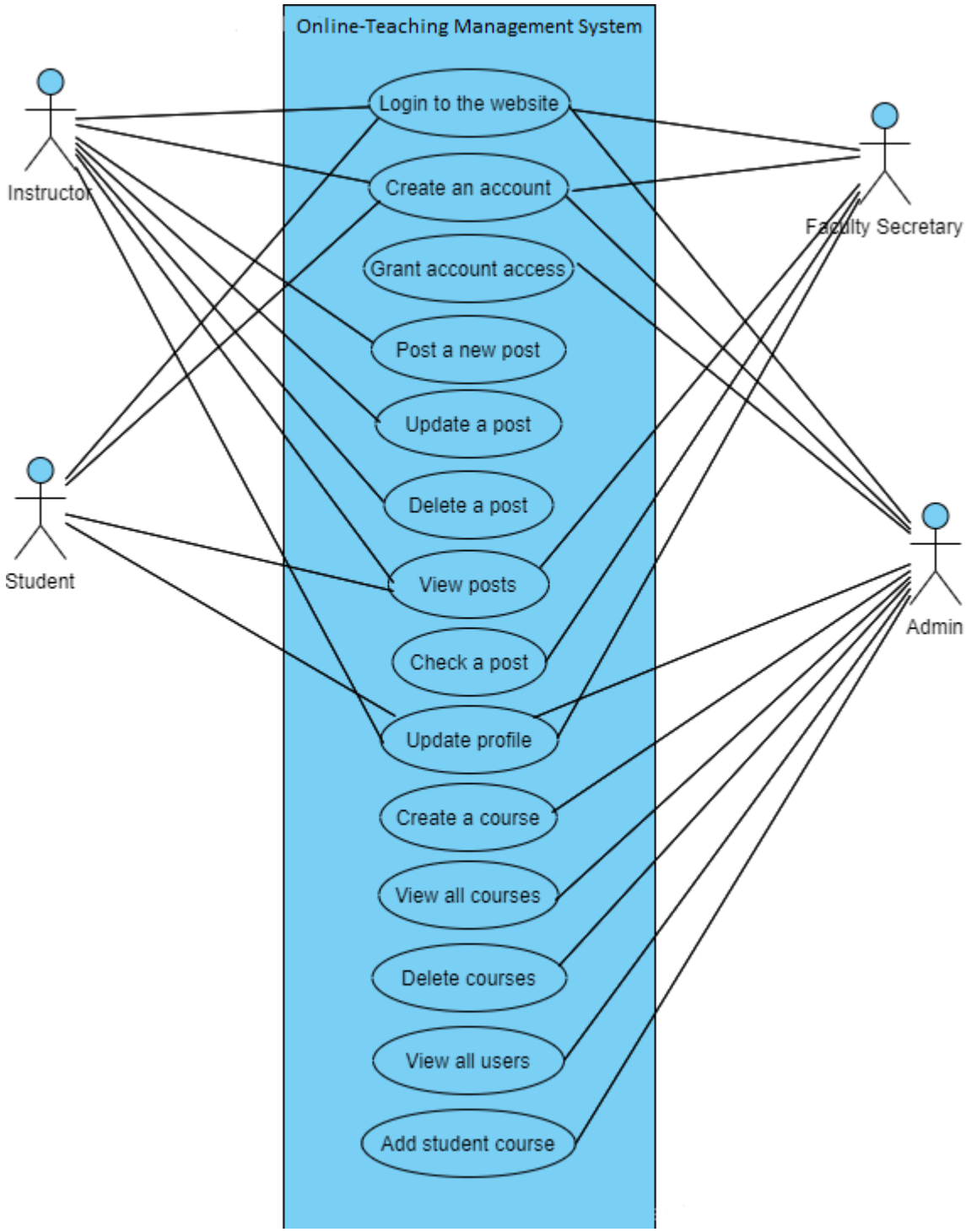

Figure 9: Use case Diagram

USE CASE 1

Name: Login to the website

Identifier: UC1

Inputs: username, password

Outputs: Login successful into home page

Step	Action	Expected Response
1	Input invalid username, password	
2	Click on ‘Login’ button	Show error message: ‘Username or password is wrong’
3	Input valid username, password	
4	Click on ‘Login’ button	The system displays the Home page with different role access. (admin/ instructor/ student/ faculty secretary)

Table 2: UC1 Login

Precondition: User must have an account to login into the website

Post condition: Login into the website successful

USE CASE 2

Name: Create an account

Identifier: UC2

Inputs: name, username, password

Outputs: Create account successful

Step	Action	Expected Response
1	Input name, exist username, password	
2	Click on ‘Register’ button	Show error message: ‘Create account not successful’
3	Input name, valid username, password	
4	Click on ‘Register’ button	Show success message: ‘Create account successful’ and move to the login page

Table 3: UC2 Create an account

Precondition: None

Post condition: Create new user account on database

USE CASE 3

Name: Grant account access

Identifier: UC3

Inputs: userID, role

Outputs: Grant account access successful

Step	Action	Expected Response
1	Login with Admin account	The system shows Admin homepage
2	Click on ‘User Access’ on the navbar	The system shows the User Access modal
3	Input UserID, select Role in Select Option (Instructor, Faculty Secretary, Student, Admin)	
4	Click on ‘Update’ button	Show success message: ‘Grant usser access successful’ and move to the login page

Table 4: UC3 Grant account access

Precondition: Login with admin account, account that need to grant permissions must have been created

Post condition: Update Role attribute of Users entity on the database

USE CASE 4

Name: Post a new post

Identifier: UC4

Inputs: courseID, week, content, link join class online, file slide pdf

Outputs: Post successful

Step	Action	Expected Response
1	Login with Instructor account	The system shows Instructor homepage
2	Click on ‘Post’ on the navbar	The system shows the Class information declaration modal
3	Select course, input week, content, class online link and upload file slide pdf	
4	Click on ‘Post’ button	The system check if input data wrong then show error message, else show success message and move to the login page

Table 5: UC4 Post a new post

Precondition: Login with instructor account, courses have been added by admin

Post condition: Add new post to the database

USE CASE 5

Name: Update a post

Identifier: UC5

Inputs: content, link join class online, file slide pdf

Outputs: Edit post successful

Step	Action	Expected Response
1	Login with Instructor account	The system shows Instructor homepage
2	Choose a course on the homepage	The system shows all posts of this course to the homepage
3	Click on 'Edit' button on the post that need to be edited	The system shows the Edit Post modal
4	Click on 'Close' button	Close the Edit Post modal
5	Click on 'Save' button	The system check if input data wrong then show error message, else show success message and move to the login page

Table 6: UC5 Update a post

Precondition: Login with instructor account, posted a post

Post condition: Edit post information to the database

USE CASE 6

Name: Delete a post

Identifier: UC6

Inputs: none

Outputs: Delete post successful

Step	Action	Expected Response
1	Login with Instructor account	The system shows Instructor homepage
2	Choose a course on the homepage	The system shows all posts of this course to the homepage
3	Click on trash icon button on the post that need to be deleted	The system shows the warning message
4	Click on 'No' button	The system closes the delete modal

5	Click on ‘Yes’ button	Show success message: ‘Delete post successful’

Table 7: UC6 Delete a post

Precondition: Login with instructor account, posted a post

Post condition: Change attribute isDelete of Post entity on the database

USE CASE 7

Name: View posts

Identifier: UC7

Inputs: none

Outputs: Show posts

Step	Action	Expected Response
1	Login with Instructor, Faculty secretary or Student account	The system shows Instructor/Faculty secretary/Student homepage
2	Choose a course on the homepage with Instructor/Student account	The system shows all posts of this course to the homepage
3	Click on ‘Slide’ button	Download slide into computer
4	Click on ‘Join online class’ button	Move to the link that join this online class
5	Click on ‘View post by Instructor’ on navbar with Faculty Secretary account	The system moves to the search page
6	Choose an Instructor on the screen	Show all courses that the instructor is teaching
7	Click on ‘View’ button on the course that need to show all posts	The system moves to the ‘show all posts’ page and show every posts of this courses

Table 8: UC7 View posts

Precondition: None

Post condition: View all posts

USE CASE 8

Name: Check a post

Identifier: UC8

Inputs: none

Outputs: Check post successful

Step	Action	Expected Response
1	Login with Faculty secretary account	The system shows Faculty secretary homepage
2	Click on ‘More’ button on the post that need to check	The system moves to the ‘show post’ page and shows every information of this post
3	Click on ‘Slide’ button	Download slide into computer
4	Click on ‘Join online class’ button	Move to the link that join this online class
5	Click on ‘Check’ button	Show success message: ‘Check post successful’ and move to the homepage

Table 9: UC8 Check a post

Precondition: Instructor must post a new post and that post have not been checked by another Faculty Secretary

Post condition: Change attribute isCheck of Post entity on the database

USE CASE 9

Name: Update profile

Identifier: UC9

Inputs: name, email, gender, IU code, department, password, file image

Outputs: Update profile successful

Step	Action	Expected Response
1	Login	The system shows the homepage
2	Click on the avatar of user in the navbar	Show the profile modal that contain every personal information
3	Click on ‘Close’ button	Close the profile modal
4	Click on ‘Edit’ button	Show the edit profile form
5	Input name, email, gender, IU code, password; select department, upload file image for avatar	
6	Click on ‘Save’ button	The system check if input data wrong then show error message, else show success

		message and move to the login page

Table 10: UC9 Update profile

Precondition: Login with any account

Post condition: Edit user profile on the database

USE CASE 10

Name: Create a course

Identifier: UC10

Inputs: courseName, department, courseCourse, weekDay, room, instructor, start slot, numbers of slots, classID, semester, school year, group, class type, lab group, userID

Outputs: Create a new course successful

Step	Action	Expected Response
1	Login with Admin account	The systems show the Admin homepage
2	Click on 'Add Course' in the navbar	The systems show the 'Add New Course' modal
3	Input courseName, courseCourse, room, instructor, start slot, numbers of slots, classID, school year, group, lab group, userID; select department, weekDay, semester and class type	
4	Click on 'Add' button	The system check if input data wrong then show error message, else show success message and move to the login page

Table 11: UC10 Create a course

Precondition: Login with Admin account

Post condition: Add new course to the database

USE CASE 11

Name: View all courses

Identifier: UC11

Inputs: none

Outputs: Show all courses of the system

Step	Action	Expected Response
1	Login with Admin account	The system shows the Admin homepage
2	Click on ‘All Courses’ in the navbar	The system shows data table of all courses

Table 12: UC11 View all courses

Precondition: Login with Admin account

Post condition: Show all courses

USE CASE 12

Name: Delete courses

Identifier: UC12

Inputs: semester, school year

Outputs: Delete courses successful

Step	Action	Expected Response
1	Login with Admin account	The system shows the admin homepage
2	Click on ‘Delete Courses’ in the navbar	The system shows ‘Delete Courses’ modal
3	Select semester (1,2,3), input school year	
4	Click delete button	Show success message: ‘Delete courses successful’ and move to the login page

Table 13: UC12 Delete courses

Precondition: Login with Admin account, the system has courses on database

Post condition: Change attribute isDelete of Courses entity on the database

USE CASE 13

Name: View all users

Identifier: UC13

Inputs: none

Outputs: Show all users of this system

Step	Action	Expected Response
1	Login with Admin account	The system shows the admin homepage

2	Click on ‘All Users’ in the navbar	The system shows data table of all Users

Table 14: UC13 View all users

Precondition: Login with Admin account

Post condition: Show all users

USE CASE 14

Name: Add student course

Identifier: UC14

Inputs: userID, courseID

Outputs: Add student course successful

Step	Action	Expected Response
1	Login with Admin account	The system shows the admin homepage
2	Click on ‘Add Course’ on Student tab in the navbar	The system shows the ‘Add Course for Student’ modal
3	Input userID, courseID	
4	Click on ‘Add’ button	Show success message: ‘Add course successful’ and move to the login page

Table 15: UC14 Add student course

Precondition: Login with Admin account, must have student account and course on database

Post condition: Add student course into database

3.2.2. Sequence diagram

The illustrations below demonstrate how the functions work in each use situation. It depicts item interactions in a temporal sequence. The diagrams below illustrate some of the functions of this project.

- **Sequence Diagram 1 (SQ1)**: Post a new post

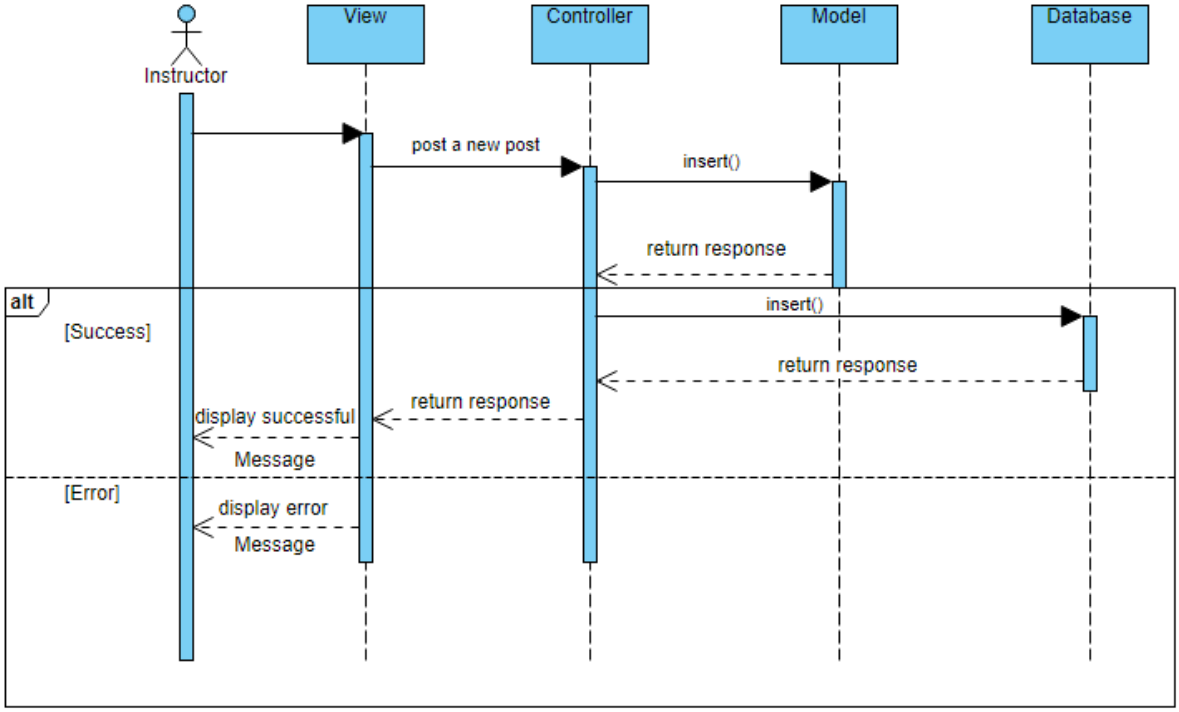

Figure 10: SQ1 Post a new post

When Instructor click on 'Post' button from View, a post function will active to Controller class. The Controller class insert in model and the model return the response. In Alternative form, Model will insert to database class and it will send the response through all class.

- **Sequence Diagram 2 (SQ2)**: Check a post

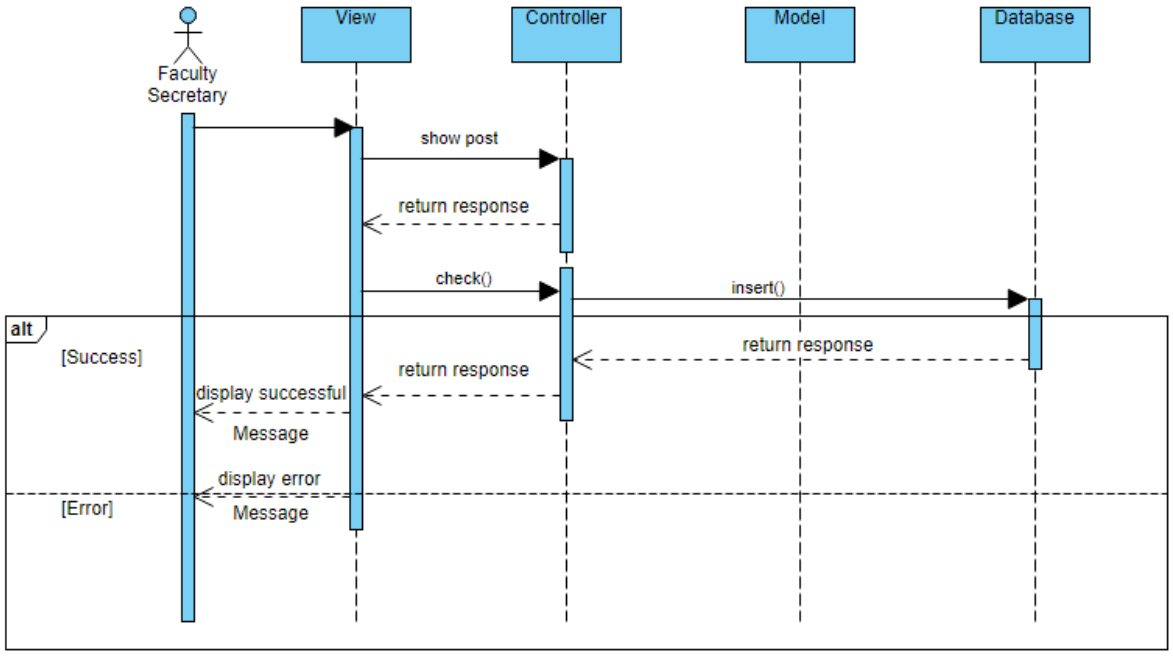

Figure 11: SQ2 Check a post

When Faculty Secretary click on ‘More’ button from View, a show post function will active to Controller class. The Controller return the response, then User click on ‘Check’ button from view, the Controller class insert in model and the model return the response. In Alternative form, Model will insert to database class and it will send the response through all class.

- **Sequence Diagram 3 (SQ3)**: Create a new course

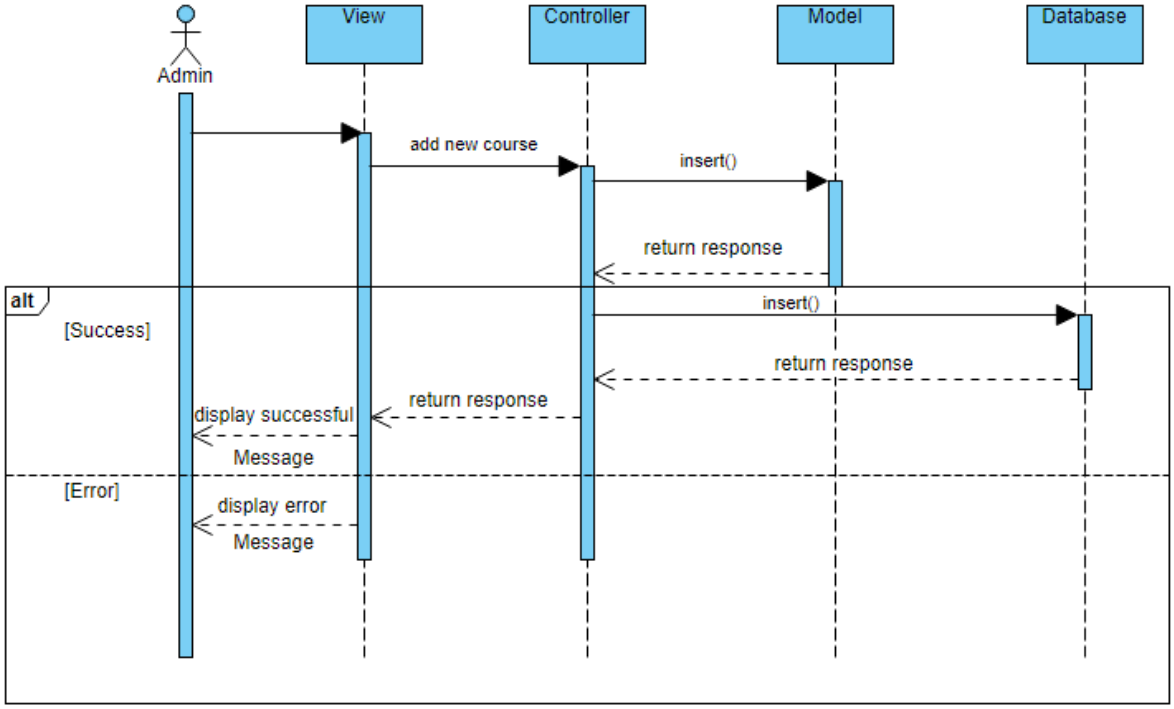

Figure 12: SQ3 Create new course

When Instructor click on ‘Add Course’ button from View, a post function will active the Controller class. The Controller class insert in model and the model return the response. In Alternative form, model will insert to database class and it will send the response through all class.

3.3. Database Design

In the database design section, we will concentrate on creating a database for this application. We will utilize SQL and the ER model to validate all of the entities and relationships in this system. We can simply construct a database structure based on this.

3.3.1. Entity Relational Diagram

We can govern the workflow of the application by creating an Entity Relationship Diagram that analyzes the application structure as well as the relationship between each actor.

This application has 5 entities: users, courses, department, post and student. Each entity is related to the others (Figure 13, 14).

First, each users' account is divided into many different roles to match the display of content on the system.

Second, it has various relationships with different entities in the user, which supports properly defining information of the user. An user can have many posts, only one department, and many courses.

Furthermore, the student entity is created to manage how a student can have multiple courses and view multiple posts.

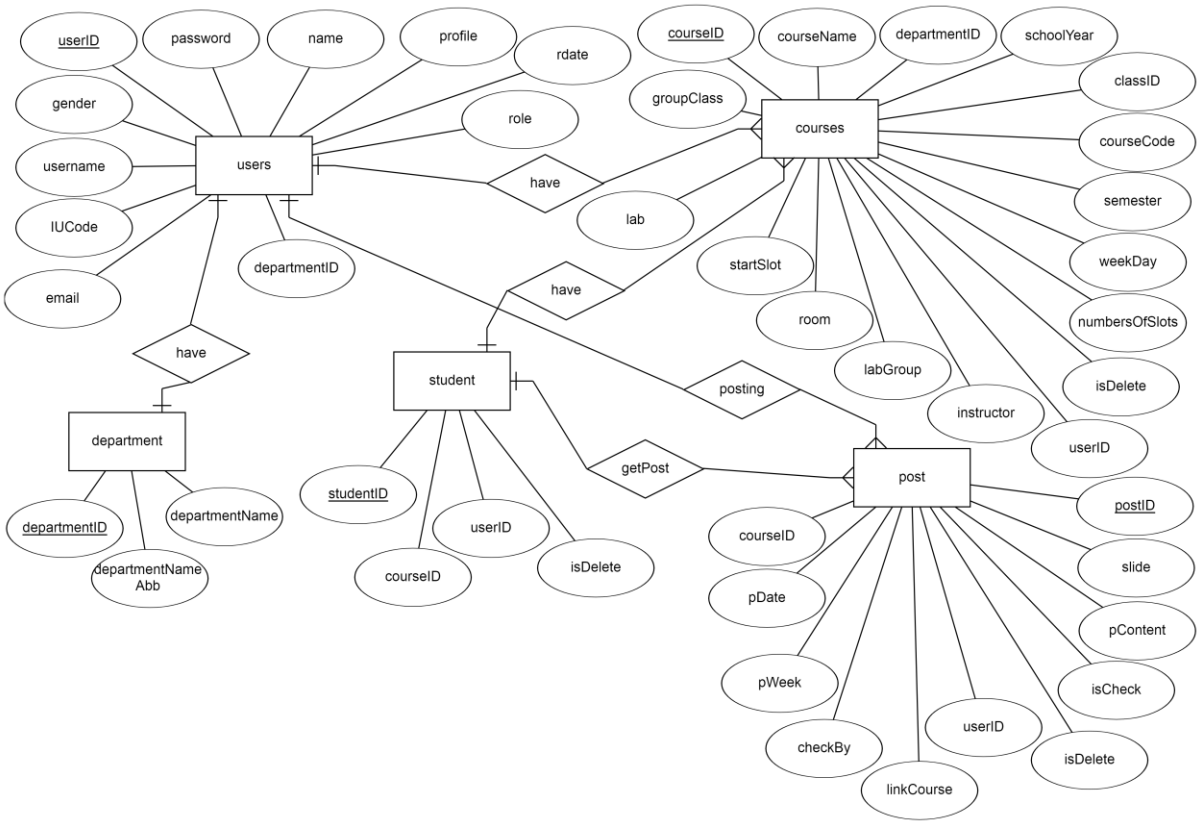

Figure 13: ERD Diagram

3.3.2. Database diagram

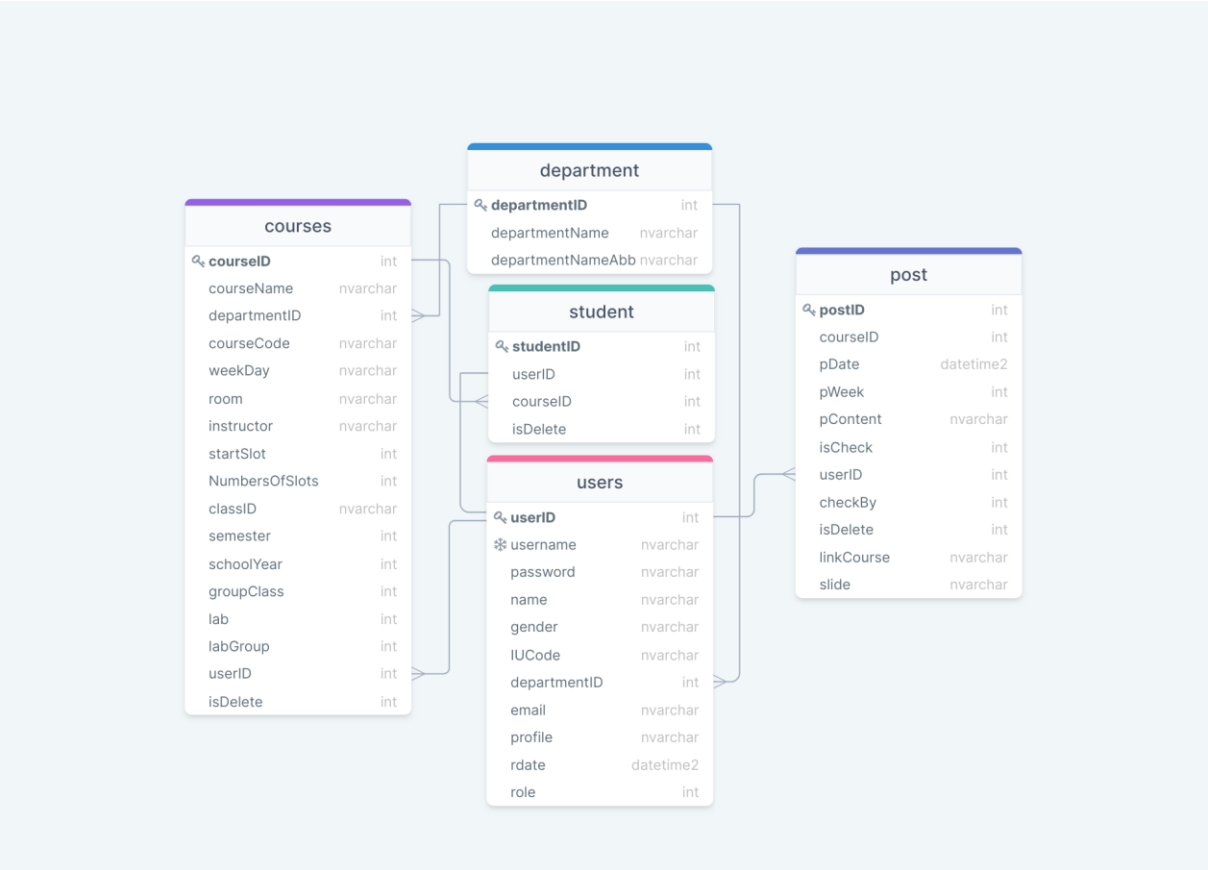

Figure 14: Database Schema

3.4. System Design

3.4.1. Class Diagram

This diagram (Figure 15) shows various classes in the system as well as their relationships. The class has attributes and methods that explain the elements and operations that the class may do.

First of all, there are four main elements of the system include: User, Course, Department and Post. Each element has a relationship with different elements.

User and Course have n-n relationship, because one Instructor can teach many Courses, Student and Faculty can get many Courses as well, and one Course can be used for many user. User and Post have 1-n relationship, one Instructor can post many posts. Course and Department have 1-n relationship, one Department can have many Courses, but one Course only be in one Department.

In addition, there are a number of support elements for handling file uploads as well as connecting to the database system.

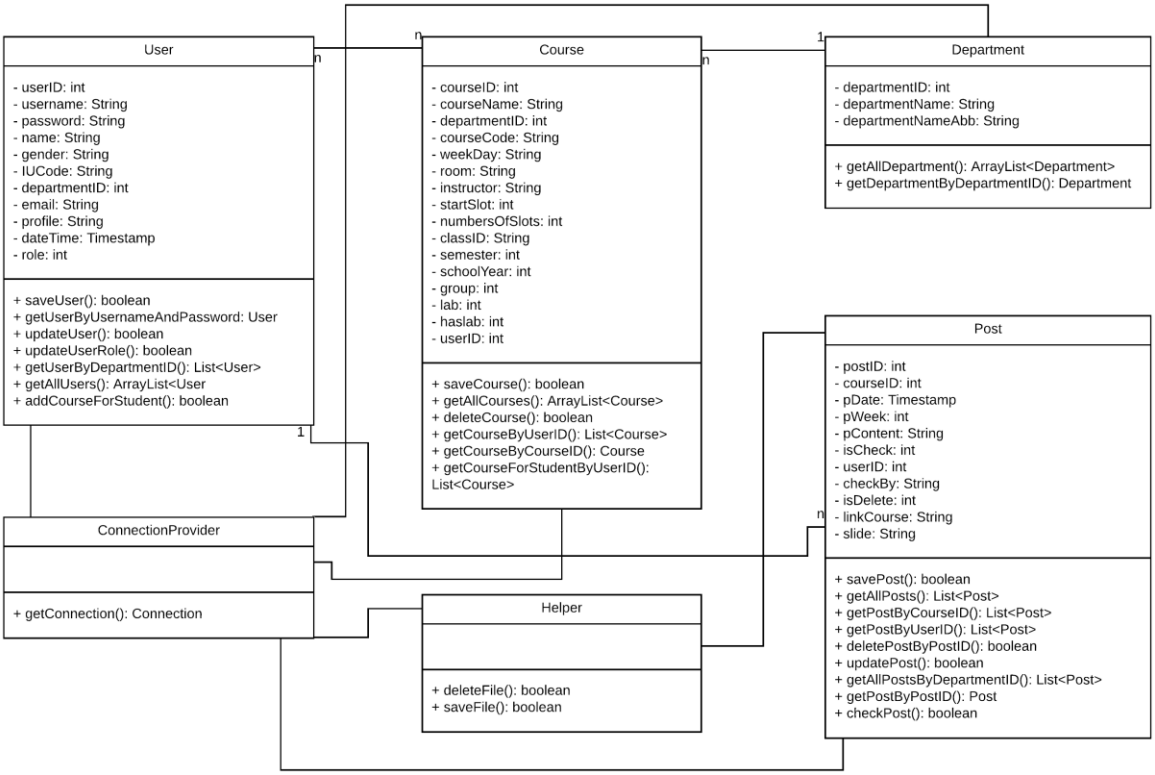

Figure 15: Class Diagram

3.4.2. Project Structure

This web application was built using the MVC model, which includes the Model, View, and Controller. Controllers employ the Model to handle user requests and data access. Models communicate with the database and retrieve data from your object. Views for displaying pages as in Figure 16.

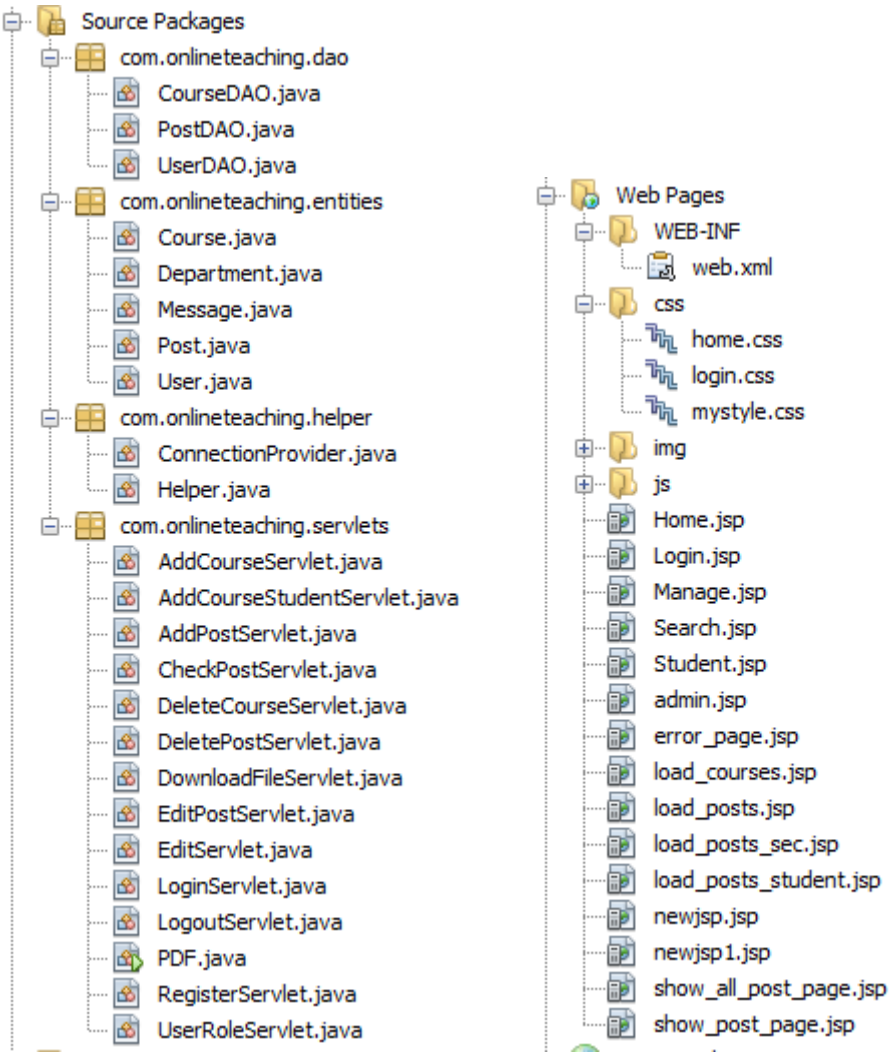

Figure 16: Project Structure

CHAPTER 4

IMPLEMENT AND RESULTS

Afterwards, investigate the technology and determine the methods necessary for developing an online-teaching management system. This chapter will describe the implementation procedure in this report.

4.1. Implement

There are 4 roles active in this aplication: Admin, Instructor, Faculty Secretary and Student. Each role has their authorizations and functions.

- Admin: The online-teaching management system is created for managing online courses in university. The admin have the right to add new courses for instructors and students, delete all courses by semester and year school, views all courses from database. Besides that, admin can grant access account for instructor or faculty secretary, because when you first create an account, the system will default to a student account.
- Instructor: User login to the website with instructor account will see all courses that they are teaching, Users can post a new article with content including the subject name, class week, summary of the lesson that day, accompanying study materials and a link to attend the class. In addition, users can review posted articles, edit or delete posts, check if the faculty secretary has checked or not.
- Faculty Secretary: With a faculty Secretary account, when user login to the website, they will see all the latest articles that the lecturers have posted, thereby checking the content of the articles is correct or not, in addition they can see all the posts of a subject to management and control.
- Student: For student accounts, after logging in, they will see the posts of the subjects they have registered to study, thereby capturing lesson information, downloading documents and participating in online classes.

4.2. Results

The end product is the website's graphic user interface, complete with explained functionality and screen shots.

4.2.1. Login and sign up page

This page allows user to log into the website by using their username and password (Figure 17). All information are stored in the database. Each account has different roles and different functions. There is also a sign-up account section on the sid (Figure 18), currently anyone can create an account but in the future I will remove the public account creation section here and only allow admins to create accounts.

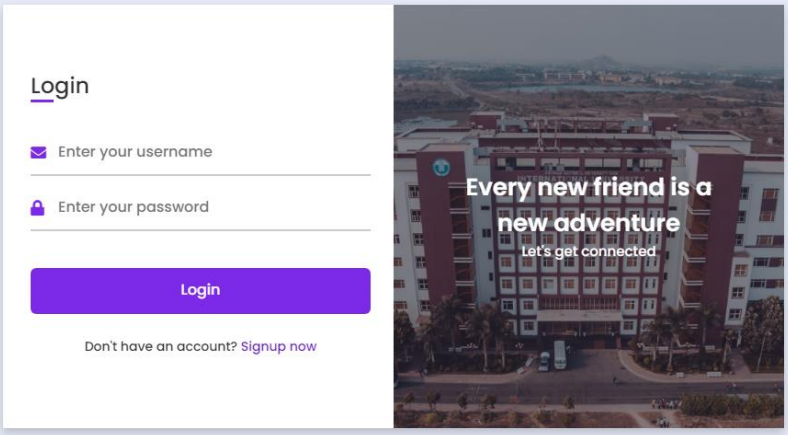

Figure 17: Login page

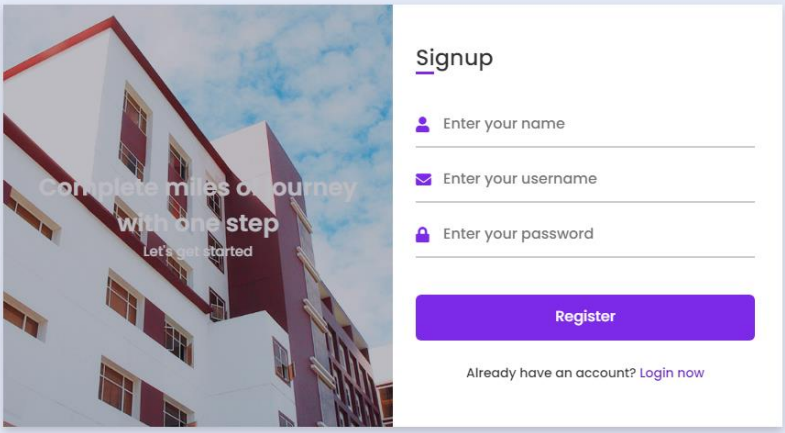

Figure 18: Sign up page

4.2.2. Home page and edit user's information

After login successful, user will move to the homepage (Figure 19), each role of users will have a different homepage, but all of them have a navbar on the left side, and it can be collapsed. In addition, when clicking on the avatar, the user can see his/her personal information (Figure 20) and edit them (Figure 21).

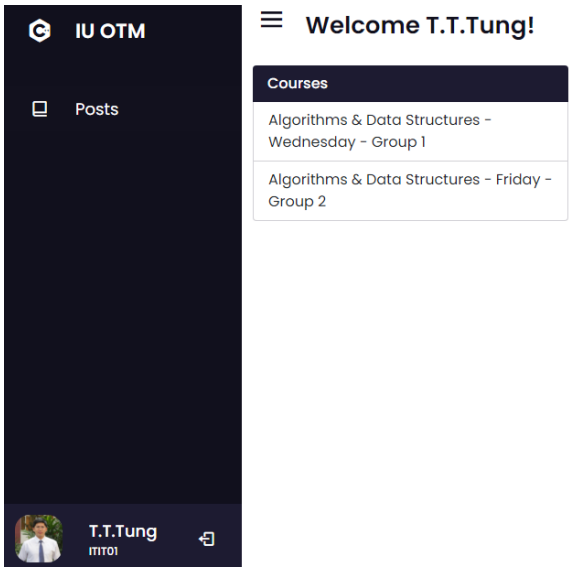

Figure 19: Home page of instructor

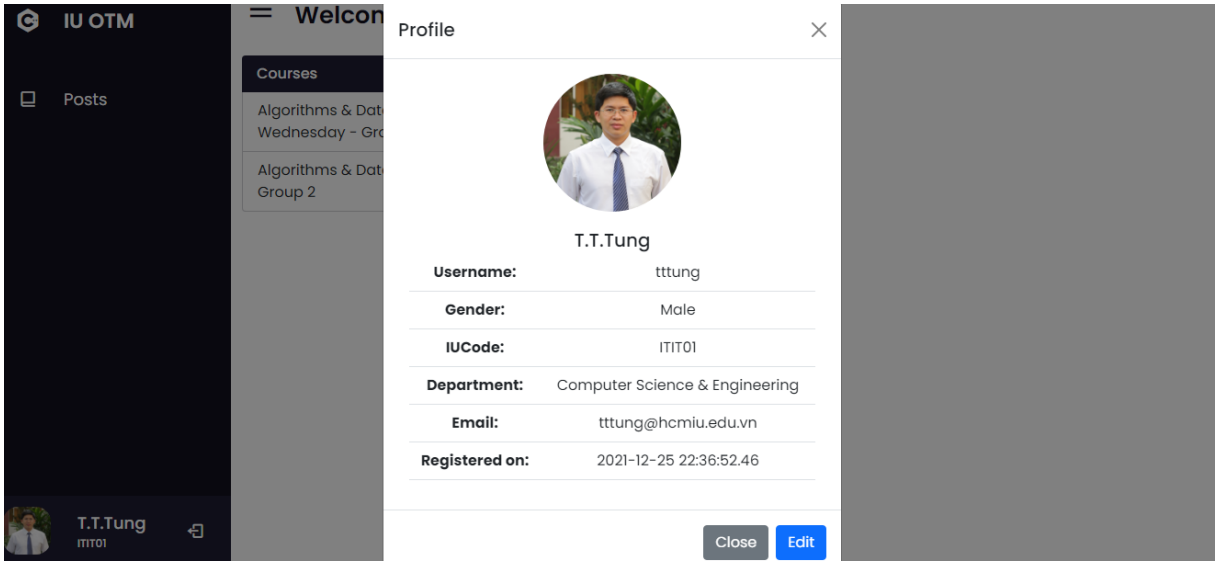

Figure 20: Profile modal

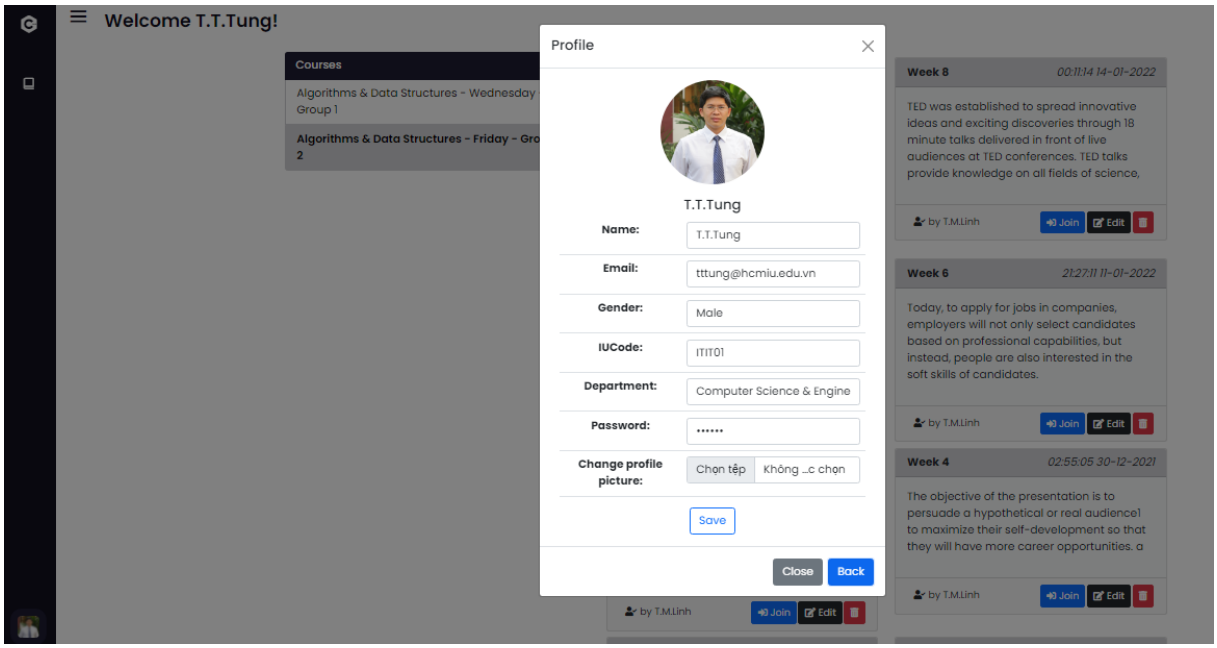

Figure 21: Edit profile

4.2.3. Post a new post

In the instructor's account, user can post a new post about today's class content so that the faculty secretary as well as the students can get the information of the class quickly.

The information that the instructor must enter includes the class name, the week of the class, the summary of the content of the lesson, the lecture slides and the link to attend the online class (Figure 22).

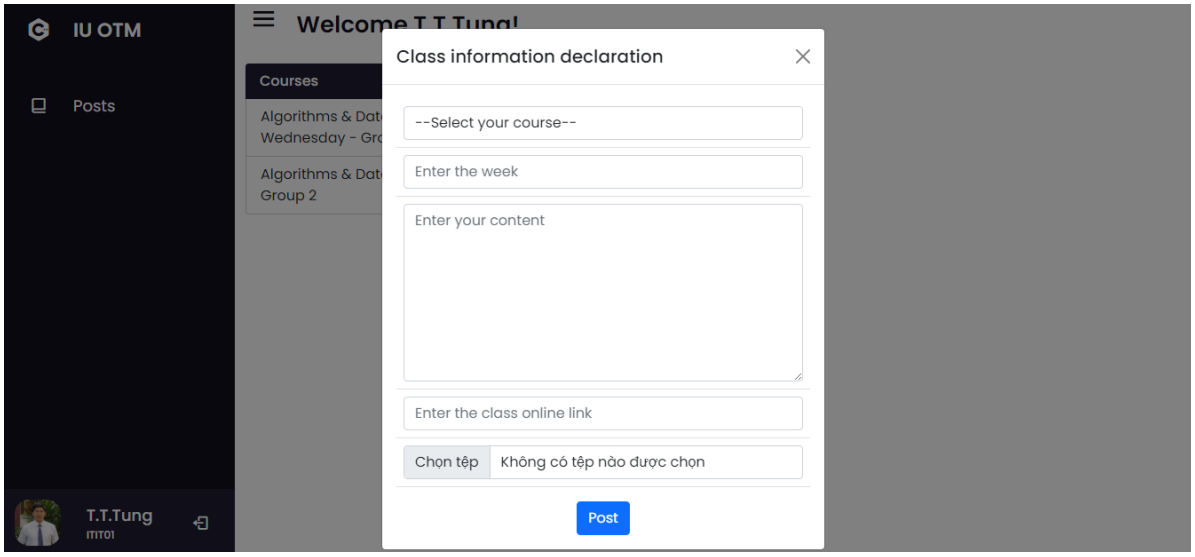

Figure 22: Post a new class information declaration

After successfully posting the new post, the instructor can check the post by clicking on the subject on the homepage, the website will show all the information that the instructor has posted (Figure 23). In addition, the instructor also can edit or delete that post (Figure 24).

Besides that, in the bottom left corner of each post, the lecturer can check if the faculty secretary has checked the post or not.

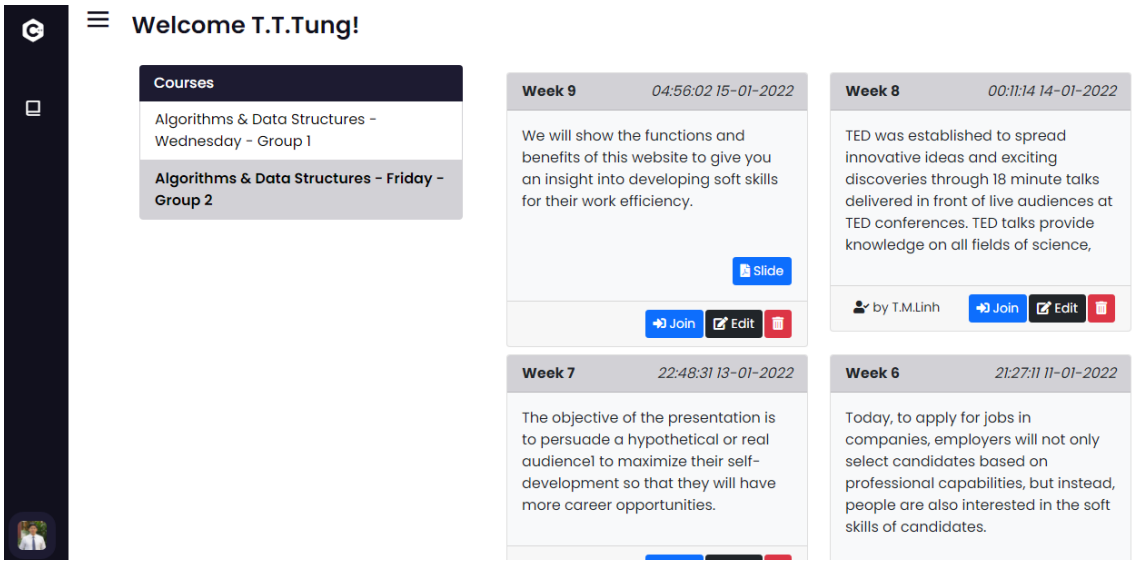

Figure 23: Overview the post

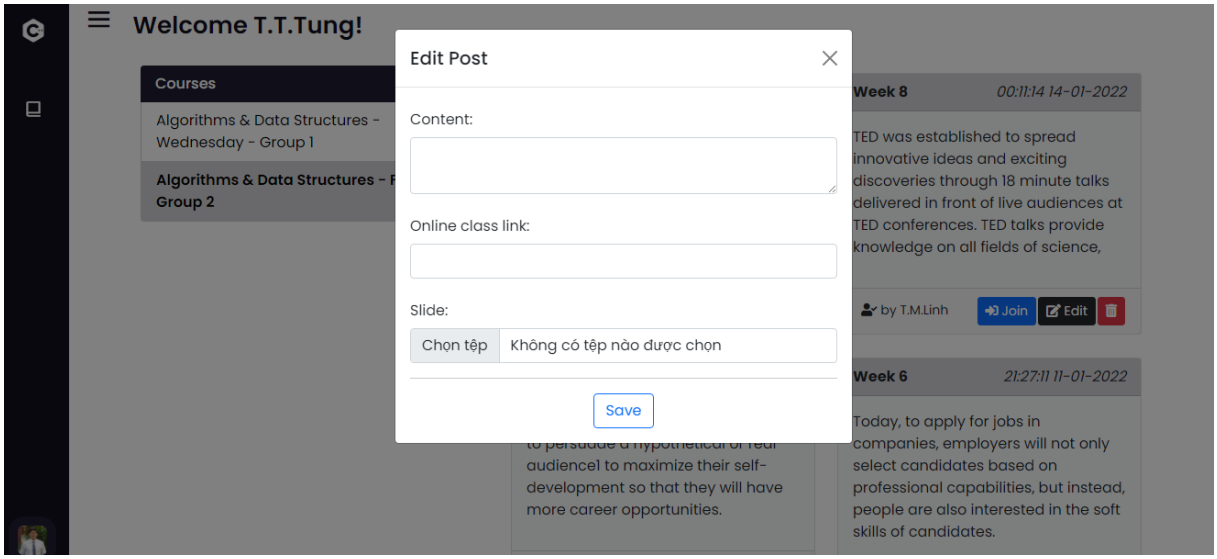

Figure 24: Edit the post

4.2.4. Check class information

The function of the faculty secretary in this website is to check if the posts posted by the instructor are correct, use the link to join the online class to participate in and check the quality of the class, take attendance....

Faculty secretary's account after logging in will go to the check homepage (Figure 25), here will display all the latest unchecked posts that faculty instructors have posted. Faculty secretary will click on each post and do the post verification (Figure 26), after done, they will click the check button and those posts will disappear on the check homepage and will display the tester's name on the posts of the instructor's account page.

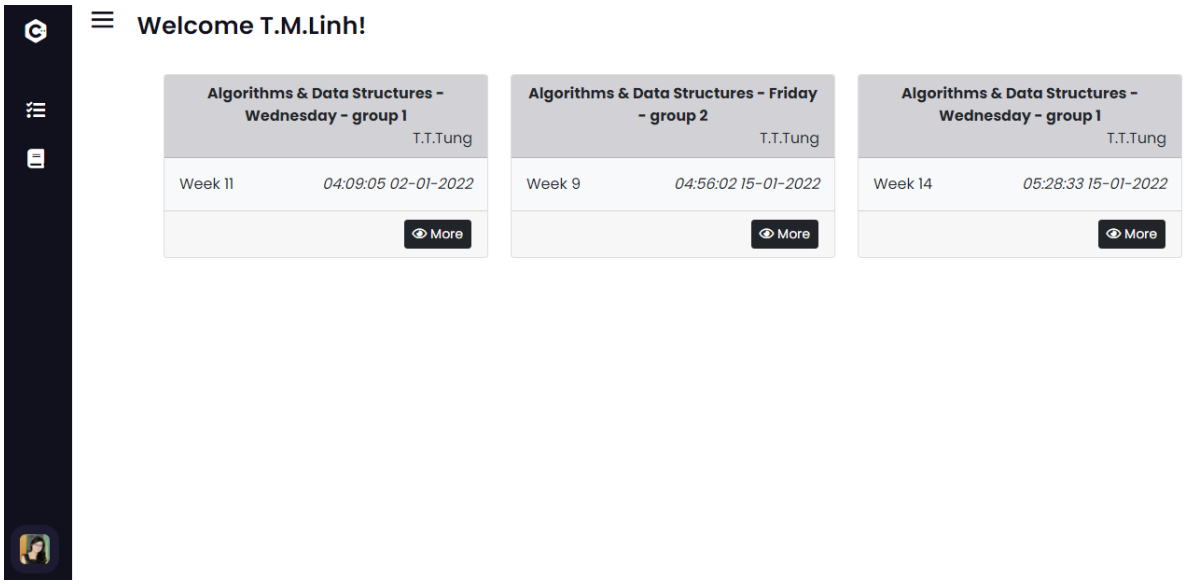

Figure 25: Check homepage

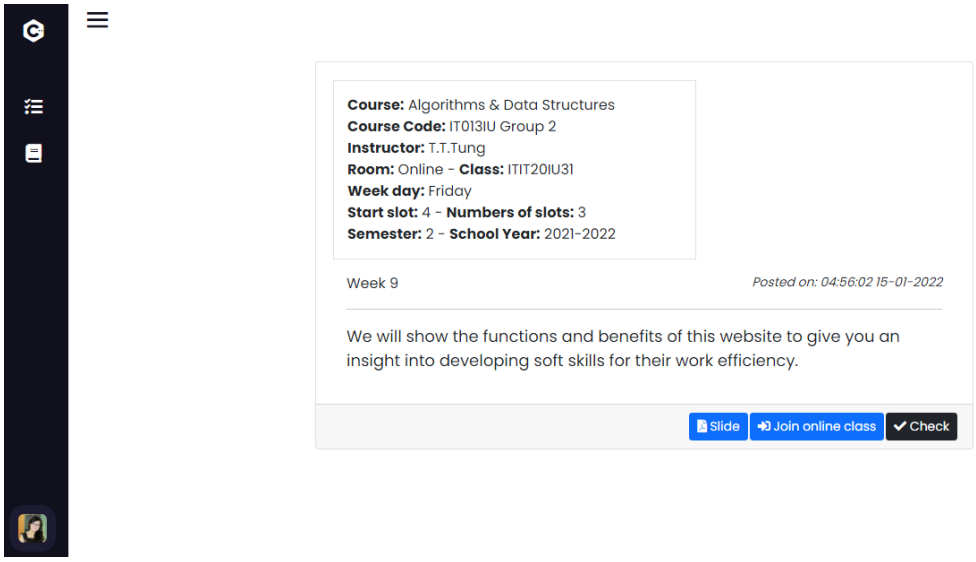

Figure 26: Post verification

In addition, in the faculty secretary's account, the user can view all the posts of each subject by each instructor for content management, classroom management, statistics, etc (Figure 27, 28).

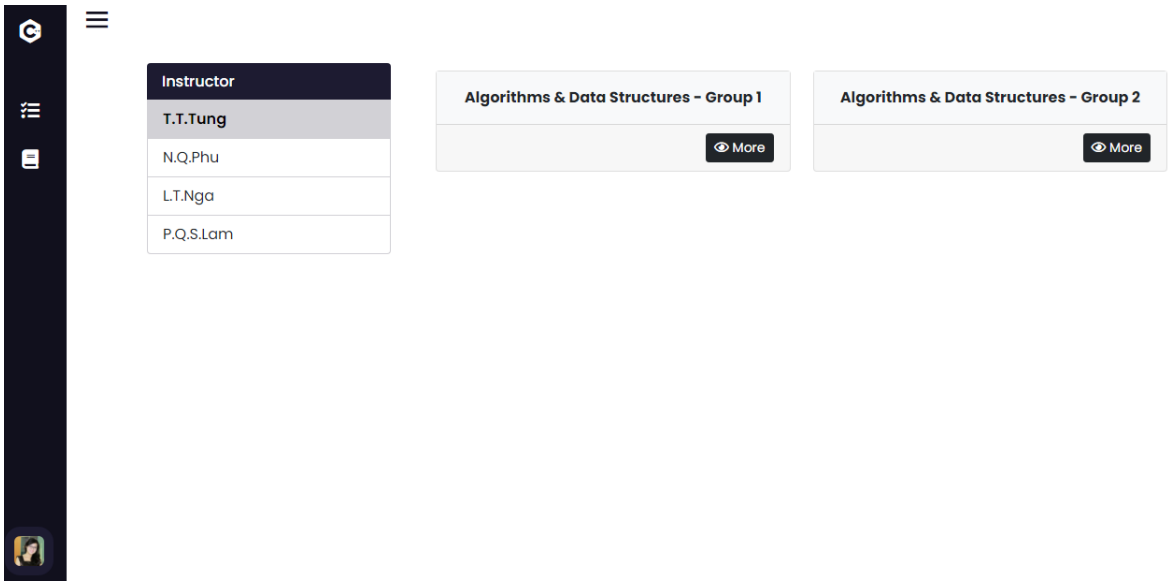

Figure 27: Choose instructor, choose subject

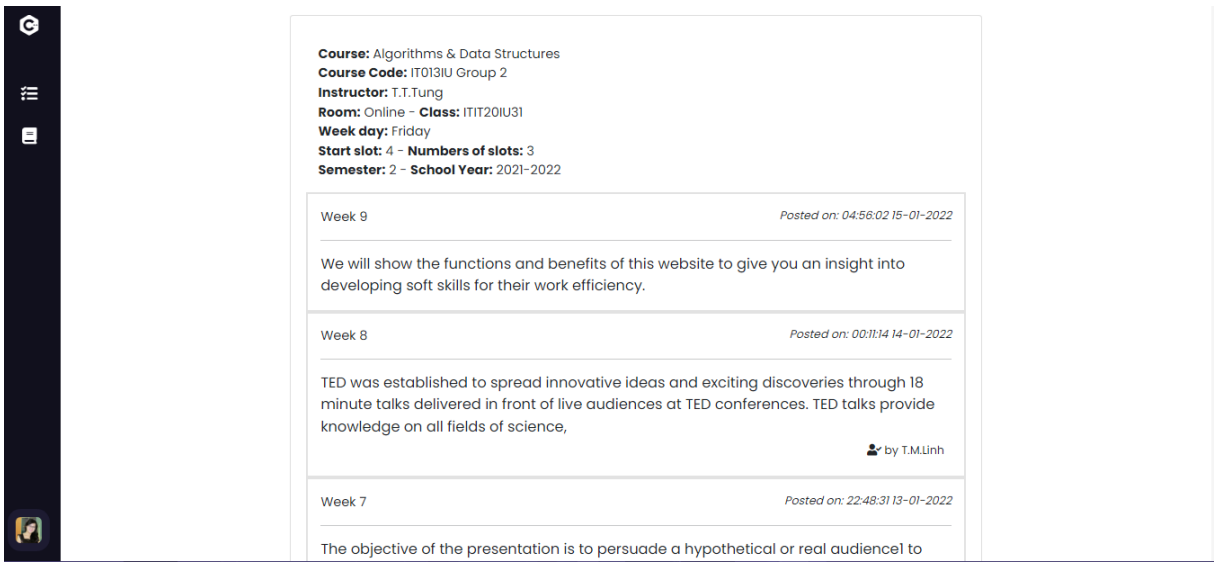

Figure 28: Show all posts of a subject

4.2.5. Student homepage

The student function on this website is quite simple, the student's account can only view the posts of each subject to get the content of that day's lesson as well as participate in the online class (Figure 29).

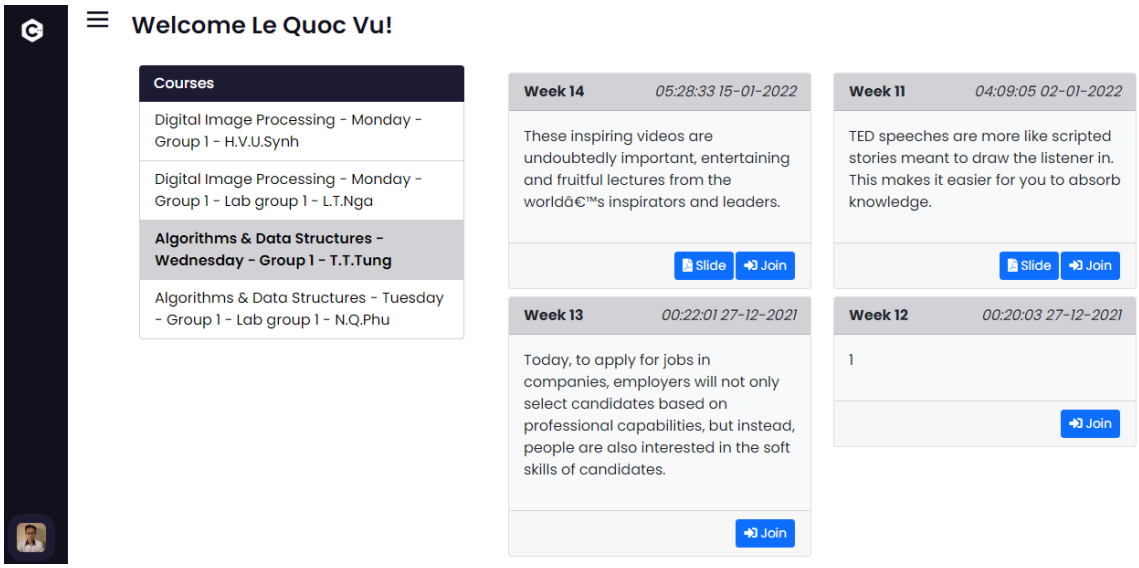

Figure 29: Student homepage

4.2.6. Add/delete course

With admin account, user can view all information of course, add/delele course (Figure 30, 31, 32). Note that with the delete course function, you can only delete all courses based on semester and school year, so be careful before deleting (Figure 33).

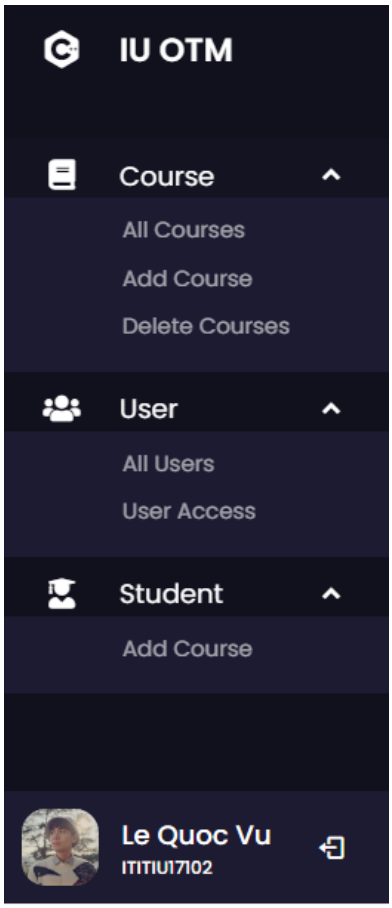

Figure 30: Admin function

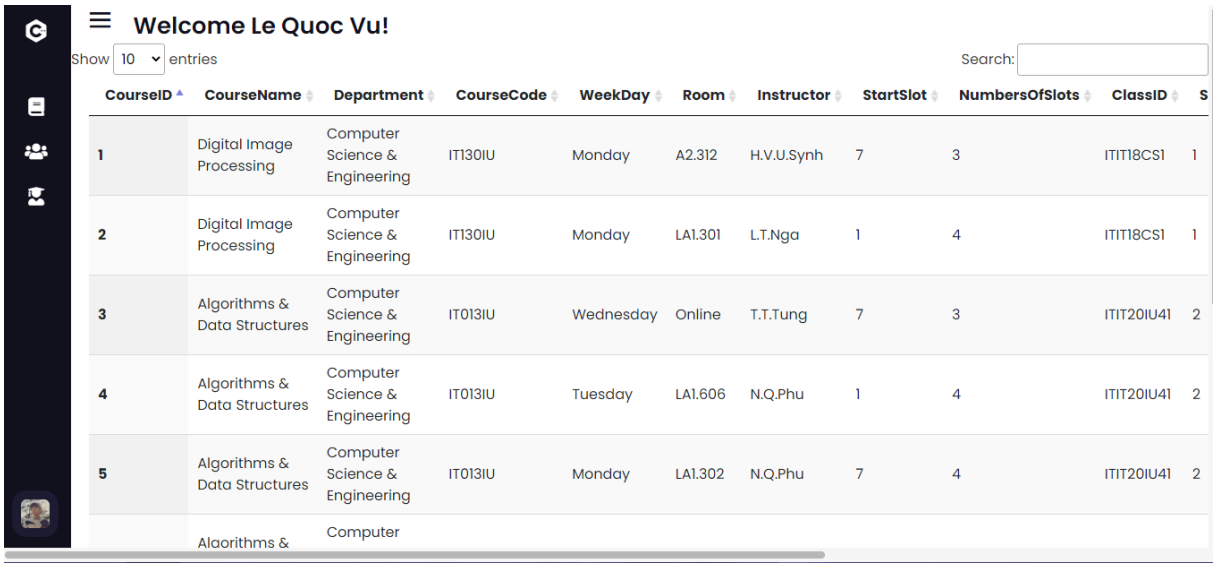

Figure 31: View all courses

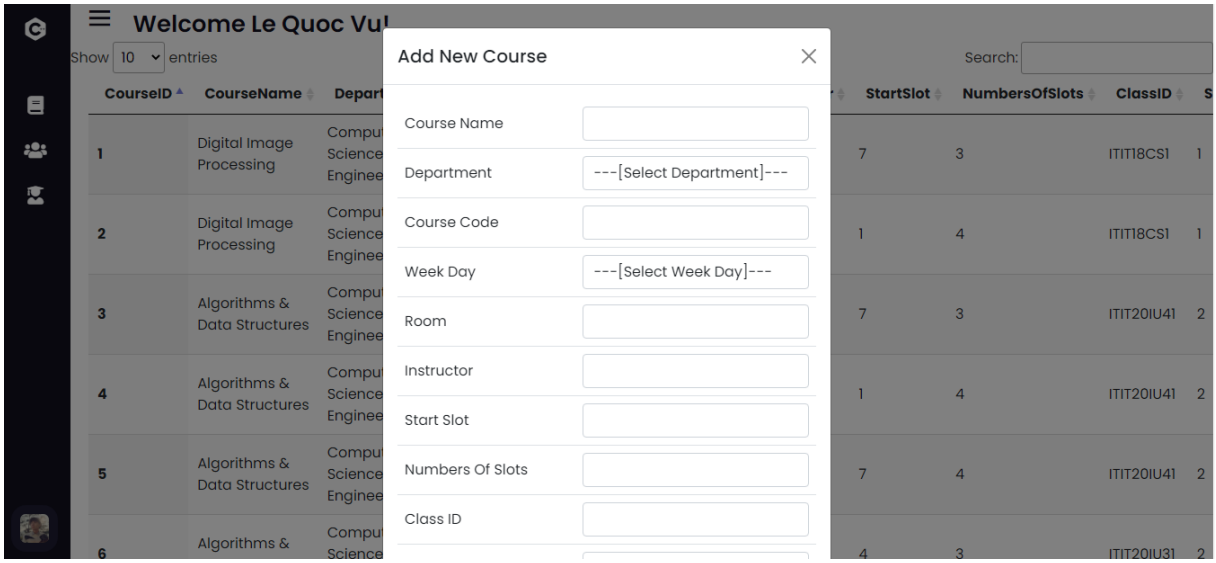

Figure 32: Add new course

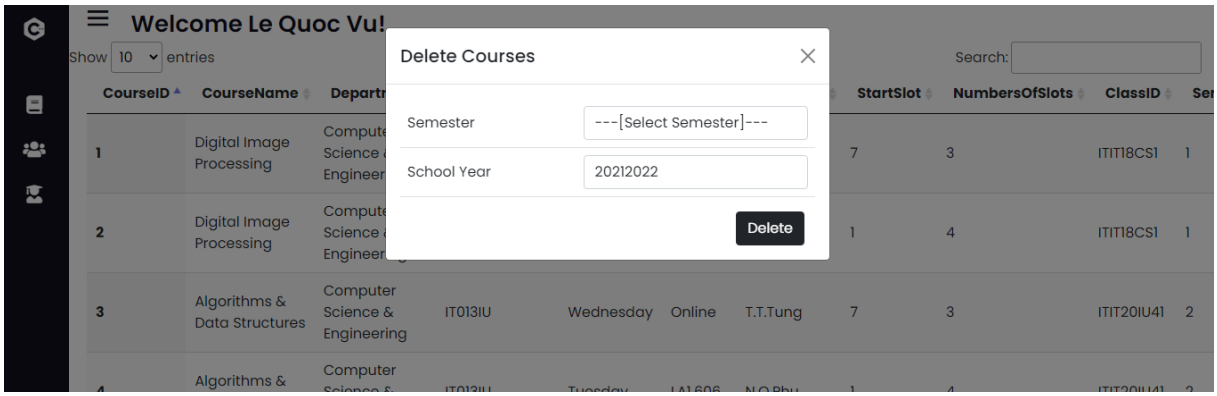

Figure 33: Delete courses

4.2.7. Grant user's permission

As mentioned above, there are all 4 roles in this system including admin, instructor, secretary and student. When creating a new account, the default account will be student account, so to grant instructor, secretary, (admin) permissions, you need to go through the admin page (Figure 35). In addition, the admin can also see all the users on the system (Figure 34).

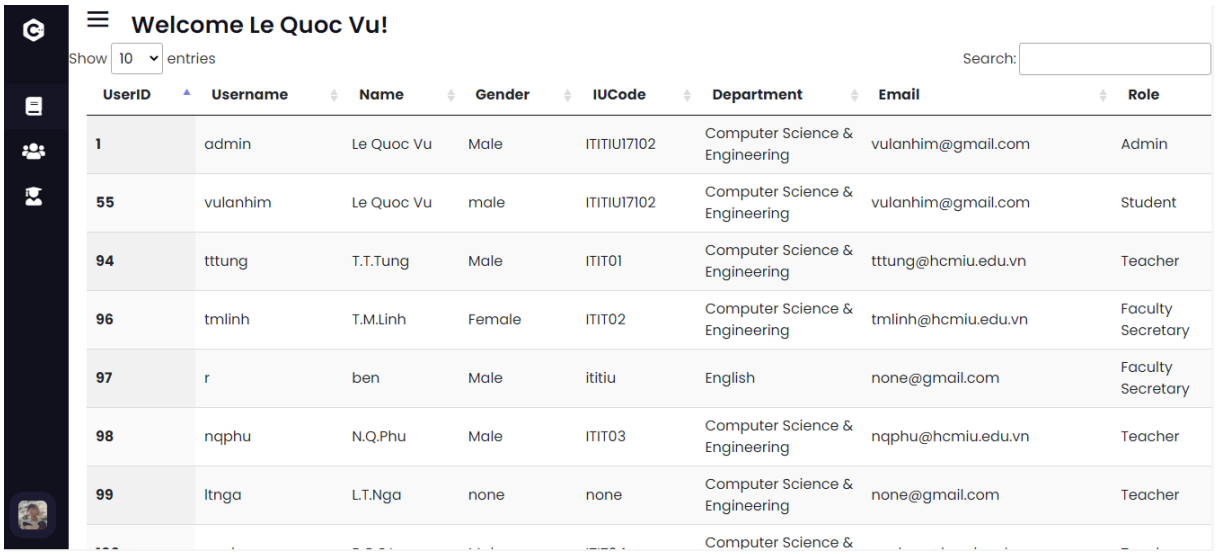

Figure 34: Show all users

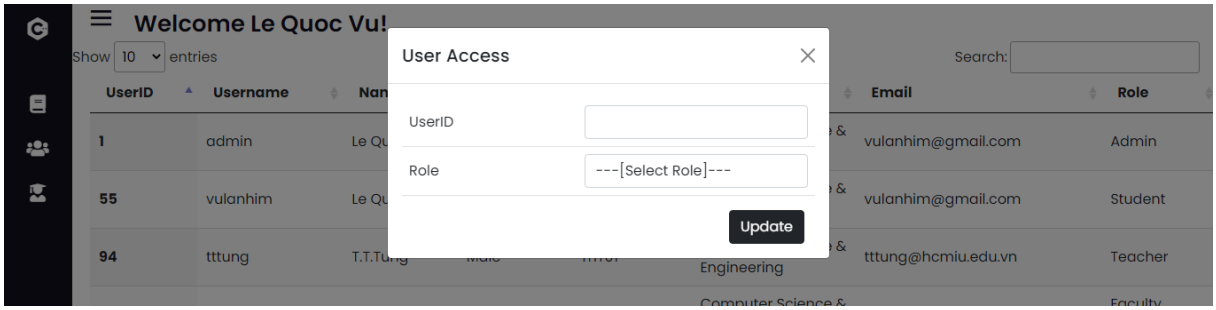

Figure 35: Grant user's permission

4.2.8. Add course for student

Admin account can add students to the course.

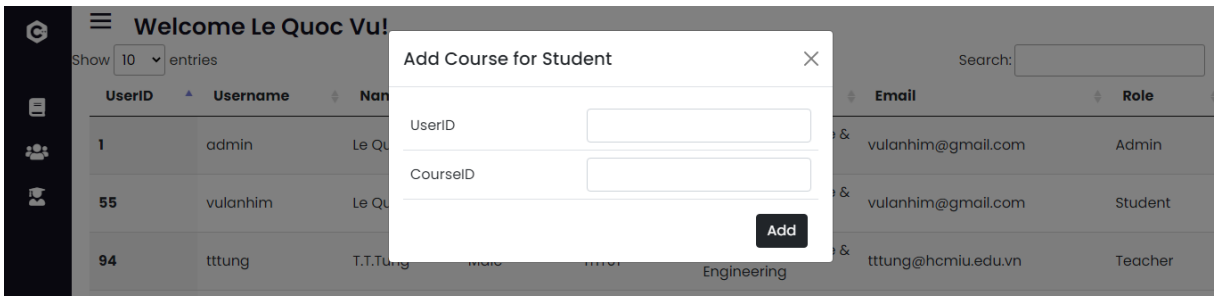

Figure 36: Add course for student

CHAPTER 5

DISCUSSION AND EVALUATION

5.1. Discussion

There is a problem while planning the concept to create the program. My first concept was an online-teaching management system, however, I am still unable to come up with many ideas for exploiting this class management section, resulting in the product lacking in functionality. It took time for another period of developing and improving additional functions for handling some problems of academic program and making it correctly.

This project was created with the JavaScript programming language. Bootstrap5 is used mostly to render user interfaces, besides I used Fontawesome [8], Sweetalert [9], DataTables, etc to make the interface more user-friendly. To generate and store connections, I used Microsoft SQL Server with UI Dbeaver and SQL Server Management Sudio. However, the cleaning code, clearing for future maintenance and scaling, is departing.

Furthermore, the program should employ automated testing, such as unit testing or client-side testing; nevertheless, this project is tested manually, and not all defects and errors have been identified.

5.2. Comparison

This online-teaching management application may be compared to several classroom management apps for use by an internal institution. By testing certain applications, I can learn how to utilize these apps and solve problems. This webpage is based on HCMIU's specifications. It is more readily and conveniently used by the university staffs. Access is provided to each function based on the features of the university. Convenience compared to the current BlackBoard IU is that the faculty is used to monitor its own, actively manage teaching records, lectures as well as quality assurance issues of online classes. Furthermore, the user interface is adaptable and simple to use.

5.3. Evaluation

Many common aspects should be considered while adjusting an application: The first standard should be the design and the user's experience with them. The strategy should be user-friendly, and the interface should have a clear vision for the user. It will then establish a connection between the client and the application. Second, it must meet all of the client's requirements. All work within the program should be as natural for the user to use as possible. If the user has several problems when operating the program, it is possible that the application

is a failure product. Finally, the structure of the site is critical. If the designer plans to use a genuine database for the framework, the system may be easily modified in the future, and security will be assured.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

A great academic program is a conglomeration of many aspects, the management of the courses becomes a matter of course to improve the quality of learning. However, in the current era of prioritizing online teaching, managing online classes is more necessary than ever, the essential goal is to assist managers in efficiently managing all courses and personnel. Keeping track of the academic program used to be a nightmare and a time-consuming task for both faculty under the old system. Understanding the limitations of traditional evaluation, a high-tech technique, the OTM system via the internet, has been developed to assist faculty to the greatest extent possible. And, with its large storage size and fast transfer speed, our application has effectively handled this problem. Aside from that, it provides several benefits for both improving the subject's quality and conserving time. This systems can monitor the classroom in real-time. The faculty secretary can log into the system, follow the links to check the class, and on the faculty secretary's interface can see the classes of the faculty in general. Furthermore, the faculty is used to monitoring its own, actively managing teaching records, lectures, and quality assurance concerns of online classrooms, as opposed to the present BlackBoard.

6.2. Future work

Many aspects of the online-teaching management system should be enhanced in the future to solve limitations. First and foremost, the user interface must be made more user-friendly and simple to use. The layout should be adaptable to a variety of platforms, including computers, tablets, and smartphones. Besides, adding more support functions for managing classes is something that is on the way such as export reports on all courses for management, add more interactive functions between teachers, faculty secretaries and students, or even store a video recording the entire online teaching process, etc, there is still a lot of work to do with this project in the future.

REFERENCES

[1] https://jquery.com/ [Accessed October 30, 2021]

[2] https://datatables.net/ [Accessed December 25, 2021]

[3] https://getbootstrap.com/ [Accessed October 30, 2021]

[4] https://www.microsoft.com/en-us/sql-server/ [Accessed October 30, 2021]

[5] https://netbeans.apache.org/ [Accessed October 30, 2021]

[6] http://edusoftweb.hcmiu.edu.vn/ [Accessed September 25, 2021]

[7] https://blackboard.hcmiu.edu.vn/ [Accessed September 25, 2021]

[8] https://fontawesome.com/ [Accessed October 30, 2021]

[9] https://sweetalert.js.org/guides/ [Accessed October 30, 2021]

[10] Learn Code with Durgesh https://www.youtube.com/channel/UC-Gn7EgShAINFthjuzxi9PQ [Accessed November 4, 2021]

[11] CodingLab https://www.youtube.com/channel/UCBlr2jG1onljL-gUy9bbhJw [Accessed October 30, 2021]

[12] Lucid Chart https://lucid.app/lucidchart/ [Accessed January 20, 2022]

[13] ERD https://erdplus.com/ [Accessed January 20, 2022]

[14] Use case diagram, Sequence diagram https://online.visual-paradigm.com/ [Accessed January 20, 2022]