VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

Final Project
Service Oriented Architecture

Informatics Center Management

Instructors: **MSc Duong Huu Phuc**

Students: **To Vinh Khang - 51800408**

Bui Quang Khai - 51800785

Truong Van Long - 51800897

Course: **22**

Ho Chi Minh City, April 1st 2021

VIETNAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

Final Project

Service Oriented Architecture

Informatics Center Management

Instructors: **MSc Duong Huu Phuc**

Students: **To Vinh Khang - 51800408**

Bui Quang Khai - 51800785

Truong Van Long - 51800897

Course: **22**

Ho Chi Minh City, April 1st 2021

THANK YOU

Sincerely, thank Mr.Duong Huu Phuc for your support and assistance during the process of completing this project. This project help us understand and apply in it.

During this process, our group avoid mistakes, which are out of our control. Hopefully, you continue to assist and give feedback for us to improve project.

THE PROJECT HAS BEEN DONE AT TON DUC THANG UNIVERSITY

We hereby declare that this thesis was carried out by ourselves under the guidance and supervision of Mr.Duong Huu Phuc; and that the work contained and the results in this project are true and have not been either submitted anywhere for any previous purpose or published in any other literature. The data and figures presented in this project are for analysis, comments, and evaluations from various resources by my own work and have been fully acknowledged in the reference part.

In addition, other comments, reviews and data from other authors, and organizations used in this project have been acknowledged, and explicitly cited.

I will take full responsibility for any fraud detected in my project. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any).

Ho Chi Minh City, 04/2021

Students Signature

To Vinh Khang

Bui Quang Khai

Truong Van Long

SUMMARY

In a growing society, education becomes more and more important. The computer center is established to satisfy the demand for learning on-line. According to demand, the KKL Software Team has implemented the project, which is Informatics Center Management. The software is conveniently deployed to store information about teachers, students, and courses in the center. It helps the centers collect and store customer data, searching potential customers and manage customer relationship.

With this topic, our group wants to apply the knowledge learned in the school along with research, system analysis, knowledge outside the society to build a management system

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction topic

In a growing society, education becomes more and more important. The computer center is established to satisfy the demand for learning online. According to demand, the KKL Software Team has implemented the project, which is Informatics Center Management.The software is conveniently deployed to store information about teachers, students, and courses in the center. It helps the centers collect and store customer data, searching potential customers and manage customer relationship.

1.2 Method use

We use NodeJS and ExpressJS for back-end; HTML, CSS and JS for font-end. Data is stored on MongoDB. Besides, JQuery, Bootstrap4, and PugEngine are applied. The interface is easy to use and suitable for customers. The Information is kept secret and updated in real-time, which ensures data integrity.

1.3 Requirement system

1.3.1 Functional requirement

- User:
  - Admin: Open courses, make announce, class-exam schedule, manage student and teacher information.
  - Visitor: View information about courses, tuition fees, teacher.

  - Students: View the announcement, view class-exam schedules, check tuition fees that need to be paid and pay tuition fees.
  - Teacher: View the announcement, view teaching schedules.

1.3.2 Non-functional requirement

Performance

- Data is updated quickly, which ensures integrity.
- The Information is received quickly and effectively.
- Friendly interface, easy to use, fast response speed (less than 1s for 1000 requests).
- Satisfying a large number of accesses at the same time.

Security

- Store and back-up data when the system meets problem.
- Manage user by the function and data. Functions(View, Add, Delete, Edit) are designed independently , which help the system be flexible in organizing many users and control data

The quality of software

- Built on the web platform and be suitable for many types of browsers (Chrome, Edge, Firefox,...) to use anytime, anywhere
- Update, maintain and develop easily after implement.

Chapter 2

REQUIREMENT SPECIFICATION

2.1 Define Actors and Use Cases

2.1.1 Define Actors

ID	Actor	Description
1	Visitor	Access to system to view information about courses, teacher.
2	Student	Enroll course, view the announcement or schedule, pay tuition fees
3	Teacher	View the information class of them and time to teach
4	Admin	Manage all activities of system

Table 2.1: List actors

2.1.2 Define Use Cases

ID	**Use Case**	**Description**	**Actor**
SU	Sign Up	User signs up system by student or teacher	ST
LILO	Log-In,Log-Out	User log in system ,User log out system	ST TE AD
ENR	Enroll	User enroll courses which are opening in list	ST
VAN	View Announcement	User select to view the announcement	ALL
VINF	View information	User select to view information about course	ALL
VSC	View schedule	User select to view schedule	ALL
VSE	View Exam Schedule	User select to view exam schedule	ALL
VTT	View Tuition Fees	User check tuition fees, which need to be paid, and status	ST
PTT	Pay tuition fees	User pay tuition fees	ST
MNAN	Manage Announcement	User view/add/delete/edit the announcements of center to student and teacher	AD
MC	Manage course	User view/add/delete/edit the information of course	AD
MSC	Manage Schedule	User view/add/delete/edit the class and exam schedule	AD
MCI	Manage Information Student	User view/add/delete/edit students	AD
MTI	Manage Information Teacher	User view/add/delete/edit teachers	AD

Table 2.2: List Use Case

2.2 System specification

The center management system support to 4 actors: visitor, student, teacher, admin to manage.

About visitor, the system provides some functions as view the information about course, teacher. Besides, when they want to become student, they can register with admin or they can sign up online.

About student, the system provides some functions as view the announcement, the information about courses, tuition fees, class schedule. The information of student includes ID, Name, Address, Phone, Email and Password.

About teacher, the system provides functions as view the announcement, the class schedule. The information of teacher includes ID, Name, Address, Phone, Description, Email and Password.

Users, like admin, teacher and student, signed up an account, so they can access to system. These actors have the different access. Admin, is the high level in the center ,has responsibility to manage the system in the center. Account of admin is stored on the system. The information of admin includes ID, Name, Email and Password.

Chapter 3

DESIGN SYSTEM

3.1 Use case diagram of system

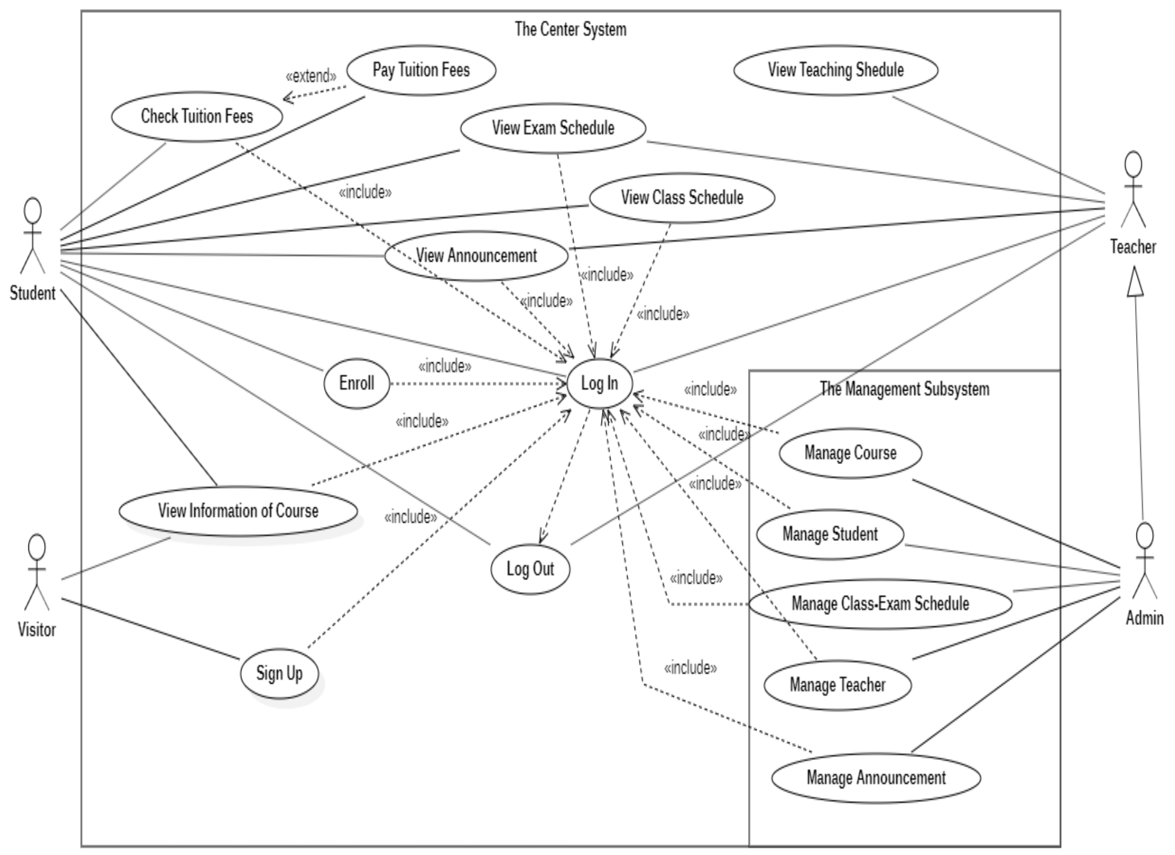

Figure 3.1: Use case diagram

3.2 Use case

3.2.1 Use case sign up

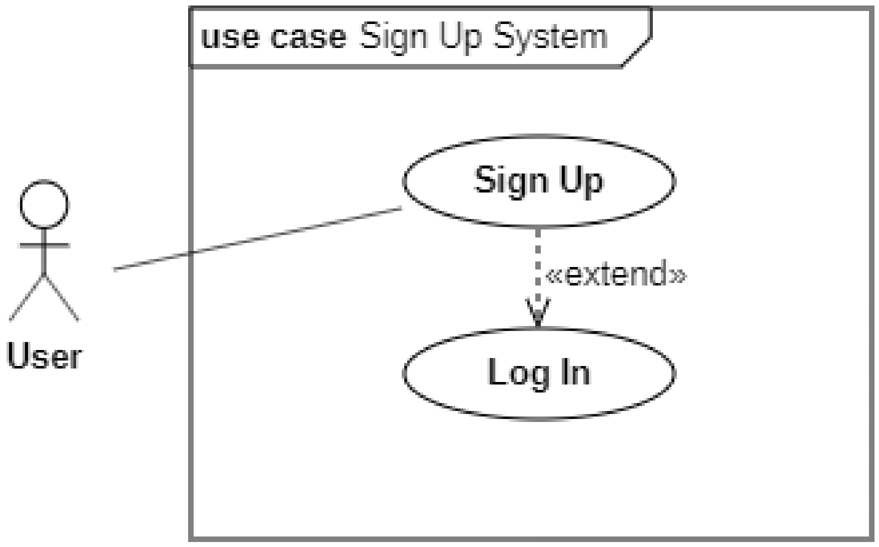

Figure 3.2: Use case sign up

User Case Name	Sign Up	
Scenario	Sign up page	
Trigger Event	Go to Log in page or press button "Sign up" on navbar.	
Brief description	Customer use function to sign up an account.	
Actors	Customer	
Related Use Case	None	
Stakeholder	Admin	
Pre-condition	Go to Log in page	
Post-condition	Sign up successfully.	
Flow of activities	Actor 1. Customer fills in the account registration form. 2. Customer press button to sign up.	System The system displays on the login page relying on the information in the form. 2.1 System checks the information in the form. 2.2 It will send request that add an account in database to DAO, if information is valid. Otherwise, it will notify customer of re-filling form.

		2.3 DAO checks the unique of information (as username, email, id card, phone) on database. If the information is invalid, it will notify customer of re-filling form. 2.4 System sends authentication code to phone/email, which is used to register, and requires customer to enter.
	3. Customer enter the authentication code.	3.1 If it is successful, the system will display message "Sign up successfully" and send request to DAO about the transformed status of account from "wait" to "active". 3.2 DAO requests database to add a new account and authorize as a member. 3.3 Notifying the successful authentication.
Exception condition	None	

Table 3.1: Use case sign up

3.2.2 Use case log in

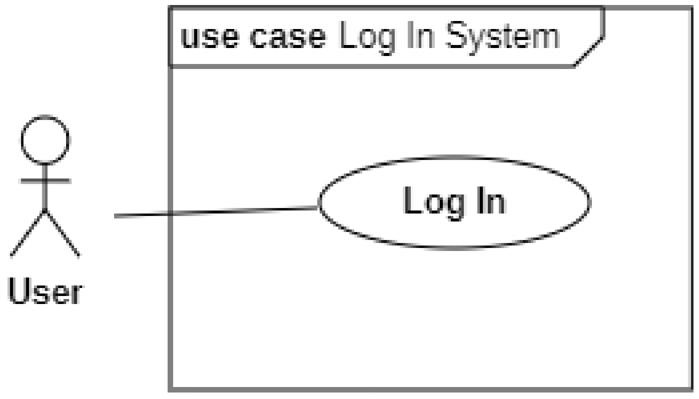

Figure 3.3: Use case log in

User Case Name	Log In	
Scenario	Log in page	
Trigger Event	Go to Log in page or press button ”Log in” on navbar.	
Brief description	Customer accesses to the system and uses functions.	
Actors	Student (had account), Teacher, Admin	
Related Use Case	Sign up	
Stakeholder	Admin	
Pre-condition	Go to Log in page	
Post-condition	Log in successfully.	
Flow of activities	Actor 1. Enter username and password. 2. User presses button to log in.	System The system displays the log in form, which filled in. 2.1 The system send request to DAO with username and password, which is hashed. 2.2 DAO sends authentic request to database. 2.3 Database checks username and password, which is existed in it or not. 2.4 If existed, the system will provide authority. Otherwise, it will display message ”Log in unsuccessfully . 2.5 The system returns final result. If be successful, the system goes to home with corresponding authority. Otherwise, display message ”username or password incorrect”.
Exception condition	Visitor doesn’t have account.	

Table 3.2: Use case log in

3.2.3 Use case log out

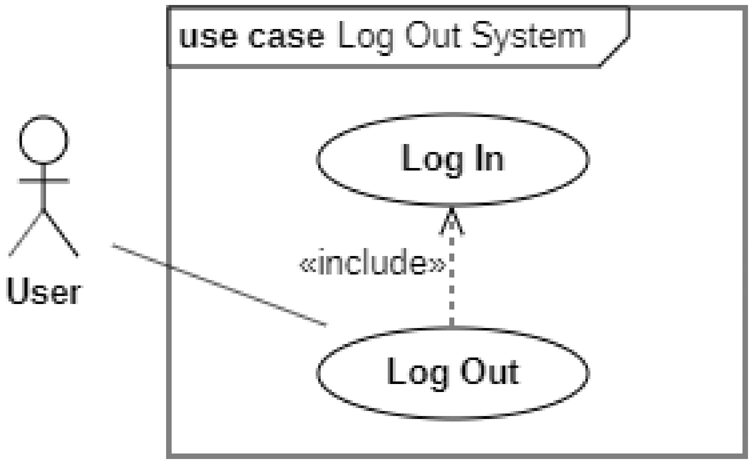

Figure 3.4: Use case log out

User Case Name	Log Out	
Scenario	Home page	
Trigger Event	Press button "Log out" on navbar.	
Brief description	Users log out when they don't use or a session is overtime.	
Actors	User (had account), Teacher, Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Log in successfully.	
Post-condition	Back to Log in page	
Flow of activities	Actor	System
	1. Press button "Log out".	1.1 The system displays the notify about accepting to log out. If user choose "YES", log out account and end session. The system deletes necessary cookies. Otherwise, it does nothing.
	2. When session is overtime.	2.1 The system logs out automatically and deletes necessary cookies. 2.2 Back to log in page and require to enter again system.
Exception condition	None	

Table 3.3: Use case log out

3.2.4 Use case enroll

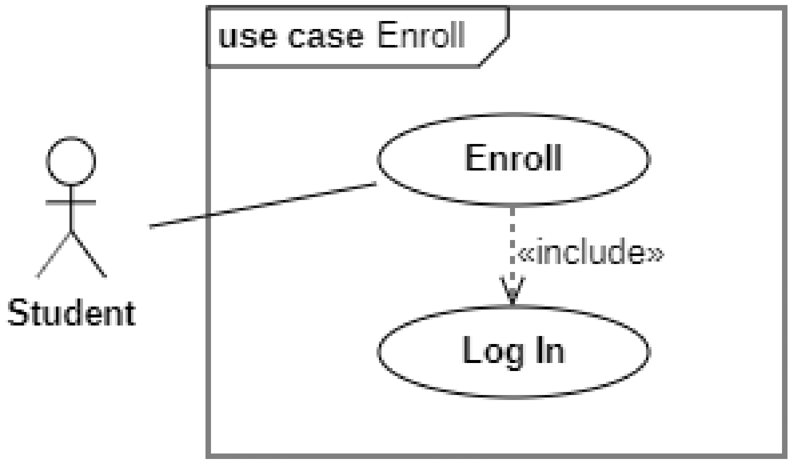

Figure 3.5: Use case enroll

User Case Name	Enroll	
Scenario	At Home page when users log in system.	
Trigger Event	Users select course and enroll that.	
Brief description	Users use this function to enroll online course.	
Actors	Student	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Log in successfully.	
Post-condition	Enroll course successfully/	
Flow of activities	Actor	System
	1. Press button ”Enroll” on navbar.	1.1 The system displays course list, which can enroll now.
	2. Customer enroll courses they want.	2.1 The system require customer to fill in the enroll course form and choose date, that goes to the center to pay tuition fees.
	3. Customer fills in the enroll course form and choose date	3.1 The system check the valid information of customer.
	4. Customer press button ”Confirm” to enroll.	4.1 The system saves the information of customer filled and requires DAO to stored them on database.

		4.2 DAO stored the information of customer enrolled on database and return message "Send successfully". The status of this course is "wait to verify". 4.3 The system displays message "Enroll successfully" and shows the information of customer and course, which he/she enrolled.
Exception condition	None	

Table 3.4: Use case enroll

3.2.5 Use case view announcement

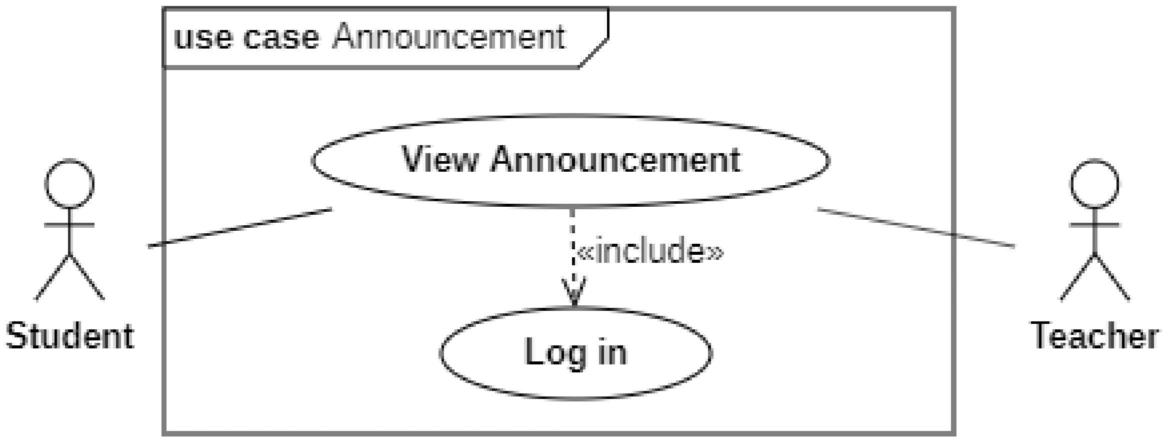

Figure 3.6: Use case view announcement

User Case Name	View announcement
Scenario	When users access to the system, it will notifies them of the class and exam schedule in a week
Trigger Event	Users log in the system
Brief description	The system notifies the class and exam schedule in a week to student and the teaching schedule to teacher
Actors	Student, Teacher
Related Use Case	Log In
Stakeholder	Admin
Pre-condition	Enter to system and at least one course now.

Post-condition	None	
Flow of activities	Actor	System
	1. After user logs in system	1.1 DAO connect to database about the class and exam schedule of user, and then DAO will return for system and it displays the announcement about schedule in a week.
	2. User clicks "Confirm" or press "X" to close.	2.1 Users can see the old announcement when they click icon, which next to icon(user).
	3. When user clicks the old announcement about class or exam schedule	3.1 Go to Use case ¡¡View Class Schedule¿¿ or ¡¡View Exam Schedule¿¿
Exception condition	None	

Table 3.5: Use case view announcement

3.2.6 Use case view information

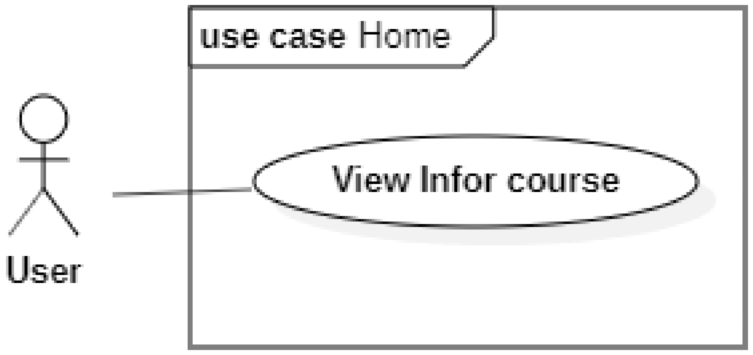

Figure 3.7: Use case view information

User Case Name	View information
Trigger Event	User logs in the system
Brief description	User views the information of course
Actors	All
Related Use Case	None
Stakeholder	None
Pre-condition	User clicks random course on the system

Post-condition	None	
Flow of activities	Actor 1. User clicks the course 2. User clicks "Enroll".	System 1.1 The information of course (like schedule, start day, end day, exam schedule) will be displayed with selection "Enroll"and button "Back" to return home page to view another course. 2.1 Go to Use case "Enroll"
Exception condition	None	

Table 3.6: Use case view information

3.2.7 Use case view class schedule

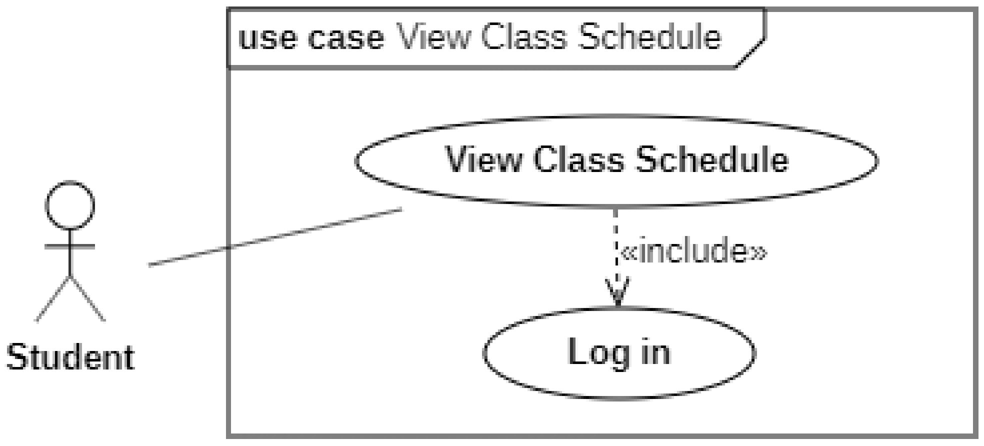

Figure 3.8: Use case view class schedule

User Case Name	View class schedule
Trigger Event	User logs in the system
Brief description	User can view user's class schedule
Actors	Student
Related Use Case	Log In
Stakeholder	Admin

Pre-condition	Users log in the system	
Post-condition	None	
Flow of activities	Actor	System
	1. User clicks ”View class schedule”	1.1 The system send request to DAO about taking data of user’s class schedule. 1.2 Database returns result to DAO and displays schedule, the information of course, start date, end date, classroom. 1.3 If database returns blank, GUI view class schedule will notify user of not having courses with selection ”View information” and ”Back” to back home page.
	2. User clicks ”View information”.	2.1 Go to Use case ”View information”
Exception condition	None	

Table 3.7: Use case view class schedule

3.2.8 Use case view exam schedule

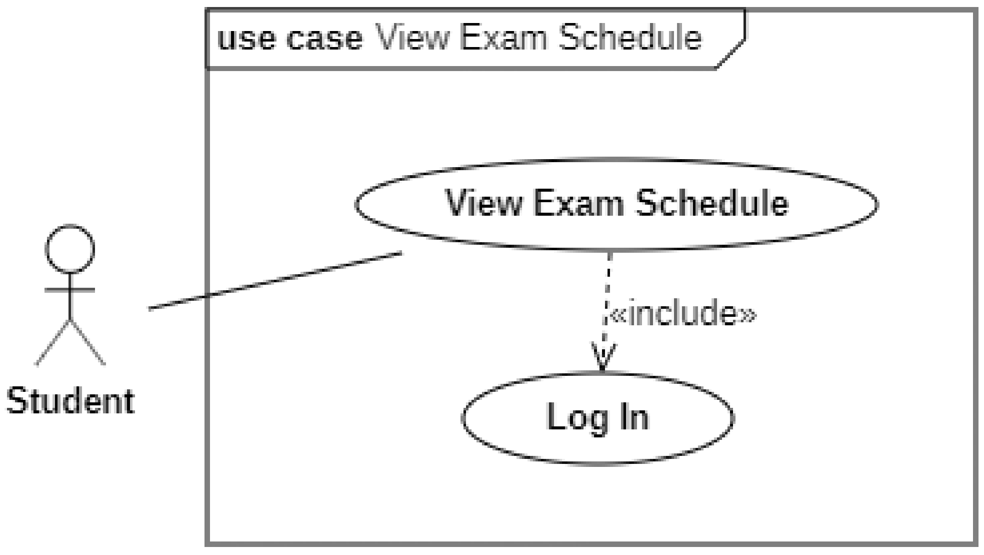

Figure 3.9: Use case view exam schedule

User Case Name	View exam schedule	
Trigger Event	User logs in the system	
Brief description	User can view user's exam schedule	
Actors	Student	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system	
Post-condition	None	
Flow of activities	Actor	System
	1. User clicks "View exam schedule"	1.1 The system send request to DAO about taking data of user's exam schedule. 1.2 Database returns result to DAO and displays schedule, the information of course, start date, end date, classroom. 1.3 If database returns blank, GUI view exam schedule will notify user of not having courses with selection "View information" and "Back" to back home page.
	2. User clicks "View information".	2.1 Go to Use case "View information"
Exception condition	None	

Table 3.8: Use case view exam schedule

3.2.9 Use case view teaching schedule

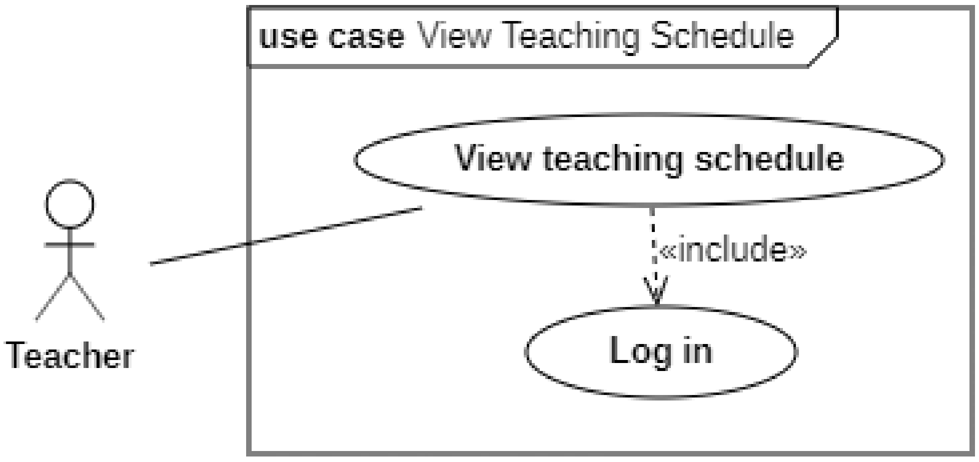

Figure 3.10: Use case view teaching schedule

User Case Name	View teaching schedule	
Trigger Event	User logs in the system	
Brief description	User can view user's teaching schedule	
Actors	Teacher	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system	
Post-condition	None	
Flow of activities	Actor 1. User clicks "View teaching schedule"	System 1.1 The system send request to DAO about taking data of user's teaching schedule. 1.2 Database returns result to DAO and displays schedule, the information of course, start date, end date, classroom. 1.3 If database returns blank, GUI view teaching schedule will notify user of not having courses with selection "View information" and "Back" to back home page.

	2. User clicks ”View information”.	2.1 Go to Use case ”View information”
Exception condition	None	

Table 3.9: Use case view teaching schedule

3.2.10 Use case manage course

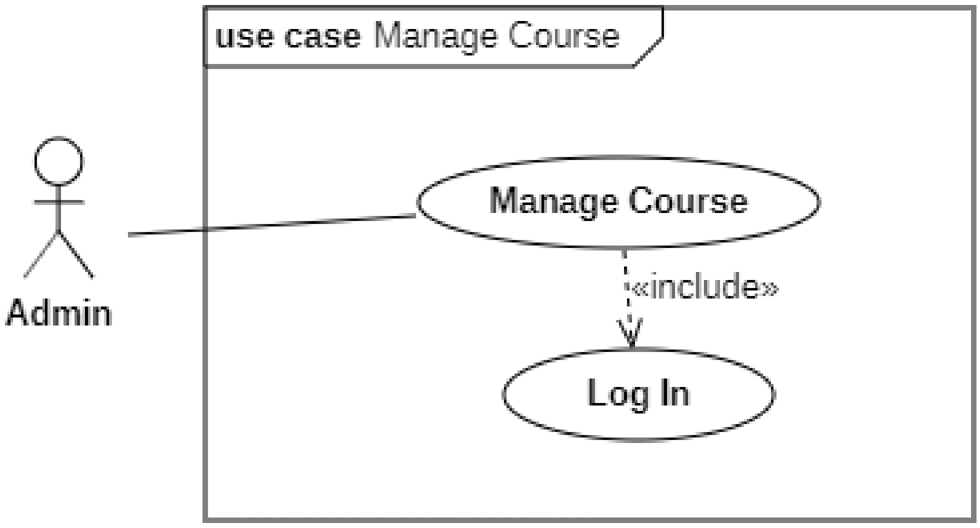

Figure 3.11: Use case manage course

User Case Name	Manage course	
Trigger Event	User choose function to manage course	
Brief description	User can manage course	
Actors	Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system by Admin	
Post-condition	None	
Flow of activities	Actor	System
	1. User clicks ”Mange course”.	1.1 The system requires DAO to get the data of course and display course list.
	2. User chooses a course.	2.1 The system returns the information of course.
	3. User have functions (as add/edit/delete).	3.1 IF:

		-Add: The system creates a form to enter the information of new course.The systems requires user to fill in. -Edit: The system displays the information of course. User re-enter things what user want to edit. -Delete: The system displays the verify delete notification.
	4. IF: -ADD: User input the information of new course and click ”verify” to add.	4.1 IF: -ADD: The system requests DAO to have a new addition. After adding successfully, the system will notify and end use case.
	-EDIT: User edits the information, which is displayed.	-EDIT: The system request to change. After editing successfully, the system will notify and end use case.
	-DELETE: User confirm to delete.	-DELETE: The system request to delete the information of course. After deleting successfully, the system will notify and end use case.
Exception condition	None	

Table 3.10: Use case Use case manage course

3.2.11 Use case manage exam-class schedule

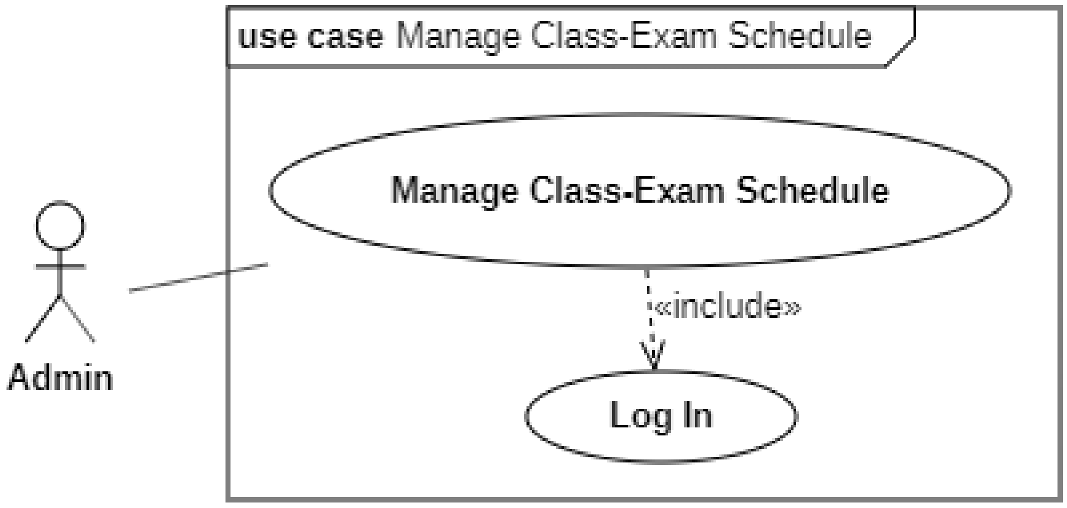
use case Manage Class-Exam Schedule
Manage Class-Exam Schedule
«include»
Log In
Admin

Figure 3.12: Use case manage exam-class schedule

User Case Name	Manage exam-class schedule	
Trigger Event	User choose function to update exam-class schedule	
Brief description	User can manage manage exam-class schedule	
Actors	Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system by Admin	
Post-condition	Update exam-class schedule	
Flow of activities	Actor	System
	1. User clicks "Update exam-class schedule".	1.1 The system requires DAO to get the data of exam-class schedule and display these.
	2. User chooses a course.	2.1 The system returns the information of course: exam-class schedule.
	3. User have functions (as add/edit/delete).	3.1 IF: -Add: The system creates a form to enter the information of new exam-class schedule.The systems requires user to fill in.

		-Edit: The system displays the information of exam-class schedule. User re-enter things what user want to edit. -Delete: The system displays the verify delete notification.
	4. IF: -ADD: User input the information of new exam-class schedule and click ”verify” to add.	4.1 IF: -ADD: The system requests DAO to have a new addition. After adding successfully, the system will notify and end use case.
	-EDIT: User edits the information, which is displayed.	-EDIT: The system request to change. After editing successfully, the system will notify and end use case.
	-DELETE: User confirm to delete.	-DELETE: The system request to delete the information of course. After deleting successfully, the system will notify and end use case.
Exception condition	None	

Table 3.11: Use case manage exam-class schedule

3.2.12 Use case manage announcement

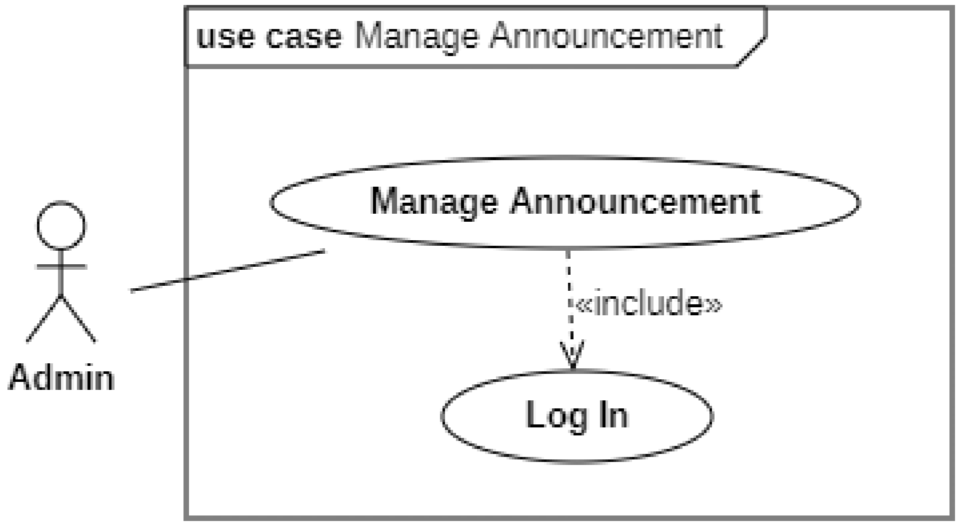

Figure 3.13: Use case manage announcement

User Case Name	Manage announcement	
Trigger Event	User choose function to manage announcement	
Brief description	User can manage announcement	
Actors	Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system by Admin	
Post-condition	None	
Flow of activities	Actor 1. User clicks "Mange announcement". 2. User chooses an announcement. 3. User have functions (as add/edit/delete).	System 1.1 The system requires DAO to get the data of course and display announcement. 2.1 The system returns the information of announcement. 3.1 IF: -Add: The system creates a form to enter the information of new announcement.The systems requires user to fill in.

		-Edit: The system displays the information of announcement. User re-enter things what user want to edit. -Delete: The system displays the verify delete notification.
	4. IF: -ADD: User input the information of new announcement and click "verify" to add.	4.1 IF: -ADD: The system requests DAO to have a new addition. After adding successfully, the system will notify and end use case.
	-EDIT: User edits the information, which is displayed.	-EDIT: The system request to change. After editing successfully, the system will notify and end use case.
	-DELETE: User confirm to delete.	-DELETE: The system request to delete the information of course. After deleting successfully, the system will notify and end use case.
Exception condition	None	

Table 3.12: Use case Use case manage announcement

3.2.13 Use case check tuition fees

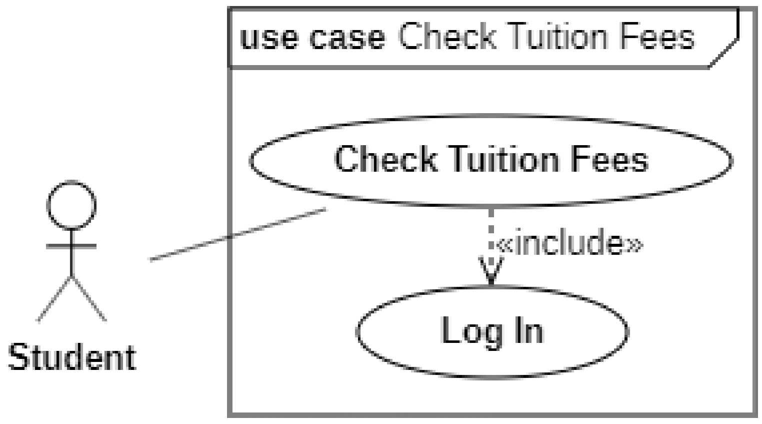

Figure 3.14: Use case check tuition fees

User Case Name	Check tuition fees	
Trigger Event	use function ”Pay tuition fees”	
Brief description	User checks tuition fees, which paid or not	
Actors	Student	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Enter to system and at least one course now.	
Post-condition	None	
Flow of activities	Actor 1. User clicks ”View tuition fees” 2. User clicks ”Confirm”	System 1.1 The system request to select by school-year, course. 2.1 The system requires DAO to get the data of course and the status of tuition fees. 2.2 The system displays user’s recent course list and tuition fees of every courses, the sum of tuition fees, the status of tuition fees.
Exception condition	None	

Table 3.13: Use case view teaching schedule

3.2.14 Use case pay tuition fees

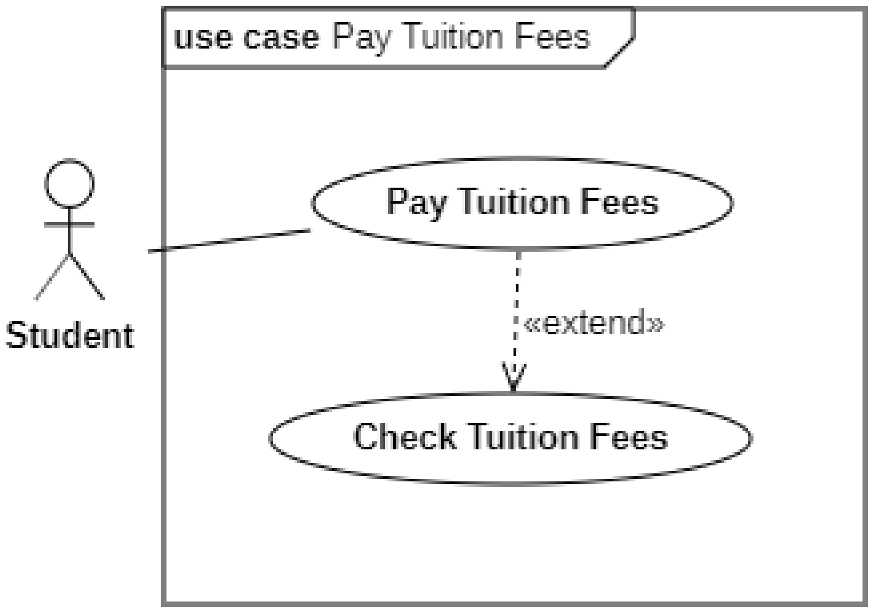

Figure 3.15: Use case pay tuition fees

3.2.15 Use case manage info student

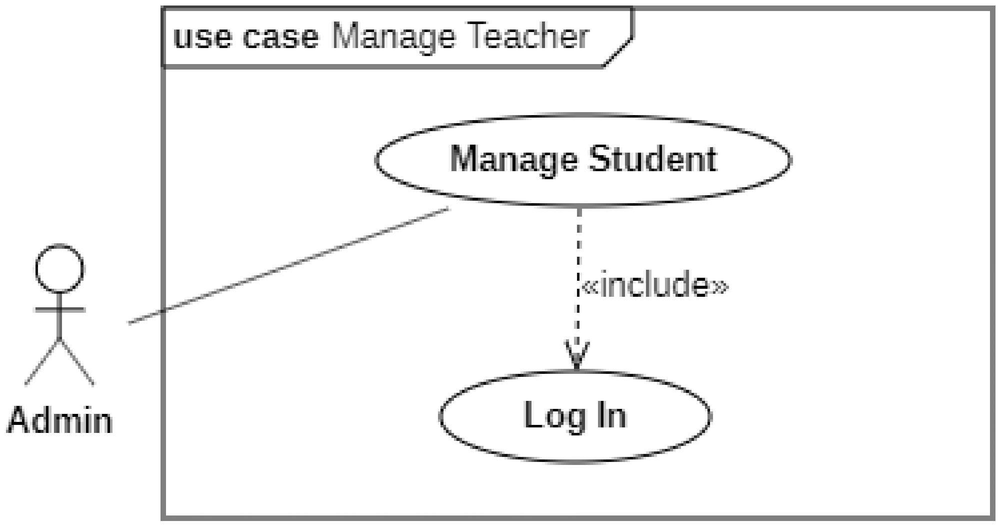

Figure 3.16: Use case manage info student

User Case Name	Manage info student	
Trigger Event	User choose function to manage info student	
Brief description	User can manage info student	
Actors	Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system by Admin	
Post-condition	None	
Flow of activities	Actor	System
	1. User clicks "Mange student".	1.1 The system requires DAO to get the data of student and sort by ID and Name.
	2. User input school-year, course and click "Confirm".(not require)	2.1 The system request DAO to get data of student by above attributes. If user don't choose anything, the system will displayed all data of student, which ordered by school-year and ID student.
	3. User chooses a student.	3.1 The system returns information, status, class schedule and time to work.
	4. User have functions (as add/edit/delete).	4.1 IF: -Add: The system creates a form to enter the information of new student (have a least ID number, phone, email). The other attributes can be blank. The systems requires student update when he/she logs in. -Edit: The system displays the information of student. -Delete: The system displays the verify delete notify.
	5. IF:	5.1 IF:

	-ADD: User input the information of new student and click ”verify” to add.	-ADD: The system requests a new add to DAO. DAO check the validity and unique of ID number, Phone, email. If they are valid, DAO will add the information in database. Otherwise, The invalid notification will be displayed and request to re-enter.
	-EDIT: User edits the information, which is displayed.	-EDIT: The system request to change, DAO will check the unique of ID number, Phone, email. If valid, the system will update database. If not, it will notify user of the invalid information and request to re-enter.
	-DELETE: User confirm to delete.	-DELETE: The system request to delete the information of student. DAO sends this request to database. The system delete these information. All of these information store in document, which will be cleaned up in 15 days, the status of account transforms inactive until deleting forever.
Exception condition	None	

Table 3.14: Use case manage info student

3.2.16 Use case manage info teacher

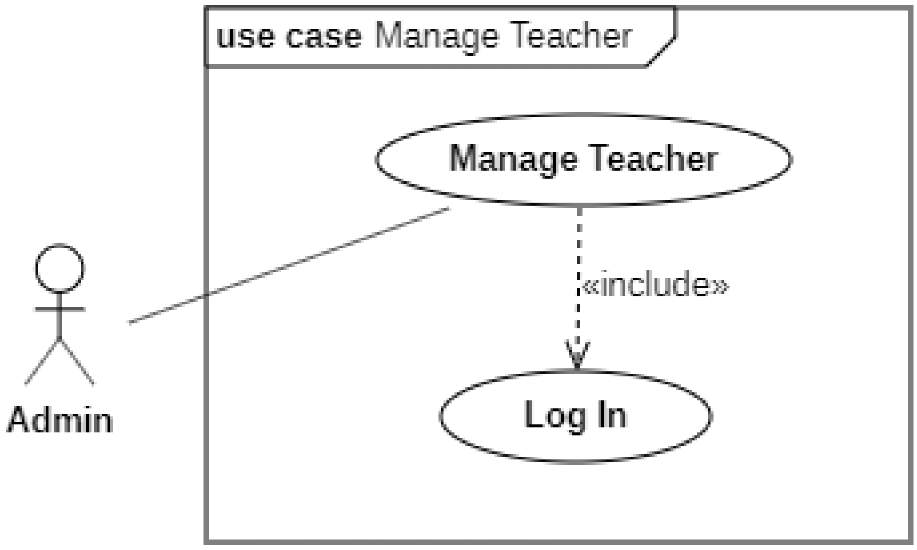

Figure 3.17: Use case manage info teacher

User Case Name	Manage info teacher	
Trigger Event	User choose function to manage info teacher	
Brief description	User can manage info teacher	
Actors	Admin	
Related Use Case	Log In	
Stakeholder	Admin	
Pre-condition	Users log in the system by Admin	
Post-condition	None	
Flow of activities	Actor	System
	1. User clicks "Mange teacher".	1.1 The system requires DAO to get the data of teacher and sort by ID and Name.
	2. User chooses a teacher.	2.1 The system returns information, status, teaching schedule and time to work.
	3. User have functions (as add/edit/delete).	3.1 IF:

		-Add: The system creates a form to enter the information of new teacher (have a least ID number, phone, email). The other attributes can be blank. The systems requires teacher update when he/she logs in. -Edit: The system displays the information of teacher. -Delete: The system displays the verify delete notify.
	4. IF: -ADD: User input the information of new teacher and click "verify" to add.	4.1 IF: -ADD: The system requests a new add to DAO. DAO check the validity and unique of ID number, Phone, email. If they are valid, DAO will add the information in database. Otherwise, The invalid notification will be displayed and request to re-enter.
	-EDIT: User edits the information, which is displayed.	-EDIT: The system request to change, DAO will check the unique of ID number, Phone, email. If valid, the system will update database. If not, it will notify user of the invalid information and request to re-enter.
	-DELETE: User confirm to delete.	-DELETE: The system request to delete the information of teacher. DAO sends this request to database. The system delete these information. All of these information store in document, which will be cleaned up in 15 days, the status of account transforms inactive until deleting forever.
Exception condition	None	

Table 3.15: Use case manage info teacher

3.3 ER diagram

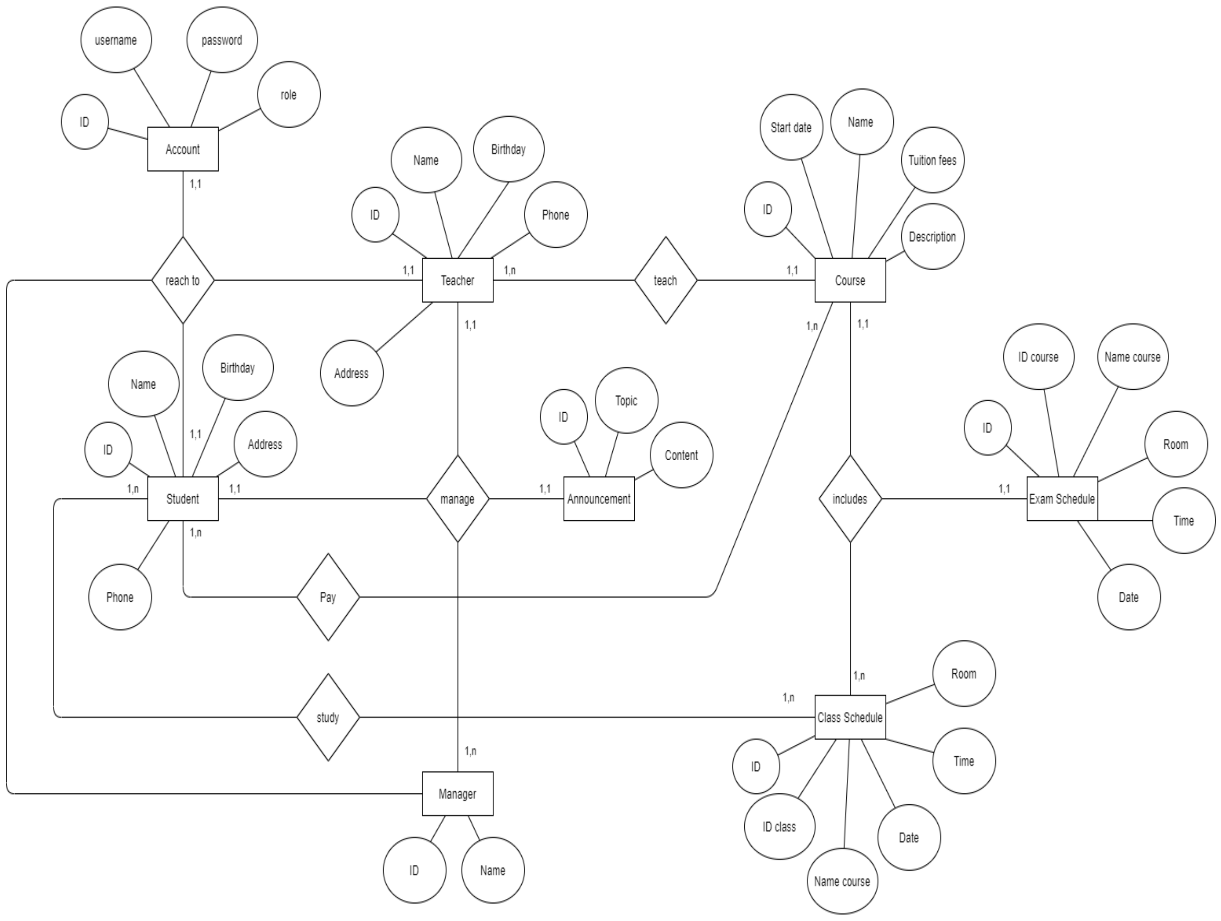

Figure 3.18: Entity Relationship Diagram

Chapter 4

CONCLUSION

This report has analyzed the business process of Informatics center management.The implementation of this system will save a lot of money, time and avoid errors for management because we have the use case specification. The storage is easy and varied. The entry and retrieval of information is simplified. All of them will bring the interesting experience for customer.

Thereby we knew thoroughly about the business process of Informatics center management. However, the system interface has not been designed to be eye-catching, the color schemes have not really matched with each other. Relative complexity is due to the fact that some command lines are still not optimized.

In the future, we will fix the optimization of the more succinct command lines. At the same time we will increase the usability and experience (Not only interface but also experiences across different device platforms).

Chapter 5

REFERENCES

1. J.W. Satzinger, R.B. Jackson, S.D. Burd, [2010], Object-Oriented Analysis and Design with the Unified Process, Course Technology, Boston.
2. L.Richardson, M.Amundsen, S.Ruby, [2013], RESTful Web APIs: Services for a Changing World, USA
3. Valentin Bojinov, [2018], RESTful Web API Design with Node.js 10, Technical University, Bulgaria
4. Howard Podeswa, [2010], UML for the IT Business Analyst, Course Technology, Boston.
5. J.W. Satzinger, R.B. Jackson, S.D. Burd, [2011], Systems Analysis and Design in a Changing World, 6th edition, Course Technology, Australia.
6. Ian Sommerville , [2011], Software Engineering 9th, University of St Andrews, Scotland.
7. Basarat Ali Syed, [2014], Beginning Node.js, Melbourne, Australia
8. https://pugjs.org/api/getting-started.html
9. https://docs.mongodb.com/manual/
10. https://expressjs.com/en/guide/routing.html
11. https://www.mongodb.com/cloud/atlas