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

FINAL REPORT

ENTERPRISE SYSTEMS DEVELOPMENT CONCEPTS

DEVELOPING AN INTEGRATED MANAGEMENT SYSTEM FOR SKILLS DEVELOPMENT CENTER

Advised by: **MSc. DUONG HUU PHUC**
Student: **BACH PHUONG BINH - 522H0001**
PHAM XUAN TRUONG - 522H0069
TRAN MY VAN - 522H0083
Course: **26**

HO CHI MINH CITY, YEAR 2025

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

FINAL REPORT

ENTERPRISE SYSTEMS DEVELOPMENT CONCEPTS

DEVELOPING AN INTEGRATED MANAGEMENT SYSTEM FOR SKILLS DEVELOPMENT CENTER

Advised by: **MSc. DUONG HUU PHUC**

Student: **BACH PHUONG BINH - 522H0001**

PHAM XUAN TRUONG - 522H0069

TRAN MY VAN - 522H0083

Course: **26**

HO CHI MINH CITY, YEAR 2025

ACKNOWLEDGMENTS

We would like to extend our deepest gratitude to Mr Duong Huu Phuc for his teaching and imparting invaluable knowledge to us. Throughout our learning journey, Mr. Phuc has been a constant companion, addressing our queries, offering encouragement, and providing guidance that has helped us progress each day.

Furthermore, we truly appreciate Mr. Phuc for creating opportunities to motivate, assist, and always being ready to guide and clarify any issues we encountered during our research and completion of the essay.

As we complete this essay, we acknowledge that we may have made mistakes and respectfully seek sincere advice and suggestions from Mr. Phuc.

Finally, we wish Mr. Phuc good health, happiness, and success in his teaching career and life.

PROJECT COMPLETED
AT TON DUC THANG UNIVERSITY

We hereby declare that this project is solely our own work and has been guided by lecturer Duong Huu Phuc. The research content and results in this topic are truthful and have not been previously published in any form. The data in the tables used for analysis, comments, and evaluations were collected by the authors from various sources clearly stated in the reference section.

Additionally, the project also incorporates some comments, evaluations, and data from other authors and organizations, all of which are properly cited and referenced.

If any form of academic dishonesty is detected, we take full responsibility for the content of our project. Ton Duc Thang University is not responsible for any copyright violations or infringements caused by us during the implementation process (if any).

Ho Chi Minh City - April 10, 2025

Author

Bach Phuong Binh

Pham Xuan Truong

Tran My Van

INSTRUCTOR’S CONFIRMATION AND EVALUATION SECTION

Supervisor’s Confirmation Section

... , date ... month ...year…

Supervisor’s Evaluation Section

... , date ... month ...year…

TABLE OF CONTENTS

SECTION 1 – OVERVIEW TOPIC

1.1. Reason for choosing the topic

In today’s fast-paced world, the demand for structured education and skill development is on the rise. As individuals and organizations increasingly invest in professional training and personal growth, educational institutions face the challenge of managing courses, students, instructors, and administrative tasks effectively. This is particularly evident in environments offering diverse programs, such as public speaking, teamwork, and leadership courses, where tracking registrations, schedules, payments, and performance evaluations becomes essential.

However, traditional methods of managing educational programs—such as paper-based records or basic software—often lead to inefficiencies, including data inconsistencies, difficulty in retrieving student information, and challenges in coordinating schedules for instructors and learners. These limitations hinder the ability of institutions to scale operations and deliver high-quality educational experiences, ultimately affecting student satisfaction and institutional growth.

To overcome these obstacles, applying technology to create a modern course management system is a practical solution. Such a system automates processes like storing course details, managing student registrations, tracking attendance, and evaluating outcomes, ensuring accuracy and efficiency. This is especially critical in education, where historical data—like student grades and instructor feedback—plays a key role in tailoring future learning experiences.

Additionally, the global trend toward digital transformation in education highlights the need for integrated systems that can connect with broader platforms, enabling institutions to share data, monitor progress, and adapt to evolving needs. Developing a course management system not only enhances operational efficiency but also contributes to the sustainable advancement of the education sector. For these reasons, the topic "Course Management System Development" is both timely and impactful, aligning with the push for digitization while addressing real-world educational challenges.

1.2. The goal of choosing a topic

The primary aim of this project is to design an automated system that streamlines the management of educational processes, from storing course and student information to tracking attendance, payments, and performance evaluations. By reducing manual workloads, the system seeks to improve operational efficiency, enhance the quality of educational services, and provide a seamless experience for students, instructors, and administrative staff. It also prioritizes data security and process digitization, fostering trust among users and supporting the long-term growth of educational institutions.

The overarching goal is to apply knowledge from database design and system development to explore the intricacies of managing course-related processes. This includes researching the educational service industry, understanding student enrollment workflows, and optimizing the scheduling and evaluation of courses. By doing so, the project aims to create a scalable solution that meets the needs of diverse stakeholders while laying the foundation for future enhancements, such as performance analytics or integration with external learning platforms.

1.3. Topic scope

Topic: “Developing an Integrated Management System for Skills Development Center”

The scope of the proposed system encompasses the following core processes:

- Course Management: Creating and organizing course details, content, and schedules.
- Student Management: Handling registrations, attendance tracking, and performance evaluations.
- Administrative Support: Managing payments, instructor assignments, and operational workflows.

The system is designed to fully address the needs of managing educational programs, optimizing data storage for courses, students, and instructors, and supporting administrative tasks. During development, the project team will establish the following acceptance criteria:

- Students expect a system that simplifies course registration, provides clear schedule information, and stores their academic history for future reference.
- Instructors require tools to manage schedules, record attendance, and evaluate student progress efficiently.
- Administrators need a platform to oversee course offerings, track payments, and assign resources effectively.

Upon deployment and testing, the system will address these key functionalities:

- For each course, it will store detailed information about schedules, instructors, and maximum enrollment capacity.
- Student registrations can be updated (e.g., from "registered" to "completed" or "cancelled").
- Upon course completion, the system will record grades, instructor comments, and evaluation ratings for analysis and reporting.

The scope is focused on a single institution’s needs but can be adapted for broader use. It does not currently include advanced features like online course delivery or mobile access, which could be explored in future iterations.

1.4. Meaning of topic

The development of a course management system carries significant practical and theoretical value, depending on the perspective and approach to managing educational processes. Researching and building this system provides a deep understanding of course administration workflows, identifies critical activities, and offers solutions to optimize traditional methods. An automated system enables students to access educational opportunities more easily while helping institutions manage data efficiently.

Transitioning from manual record-keeping to a digital platform reduces errors, saves time for both learners and staff, and enhances the ability to deliver high-quality education. The system supports the management of large datasets—such as student registrations, attendance records, and payment histories—making it easier to analyze trends and make informed decisions. For instance, tracking instructor attendance and student evaluations can improve accountability and course quality.

Beyond operational benefits, adopting technology in course management drives the modernization of educational institutions, aligning with global digitization trends. It enhances service accessibility for students, boosts institutional efficiency, and contributes to the broader goal of advancing education through innovation. This project not only serves as a practical tool but also enriches the study of system design, offering insights into relational databases, user-centered development, and process automation.

SECTION 2 – SYSTEM ANALYSIS AND DESIGN

2.1. Business process analysis

2.1.1. Studying the Actual Process

The course management system aims to enhance how educational institutions operate by streamlining their key academic and administrative workflows. The actual process includes:

- Course Registration: Students choose the courses they wish to attend, fill out registration forms, and await confirmation from the administrative staff.
- Payment Processing: Students are required to pay the course fee after registration, either online or at the center, before the course starts.
- Course Content Preparation: The center prepares and distributes course materials (e.g., topics, handouts, or digital resources) to students before or during the course.
- Teaching Schedule Management: Instructors are assigned courses and provided with teaching schedules. They are also responsible for recording attendance during sessions.
- Student Evaluation: After the course ends, instructors input students' grades and give feedback based on their performance.
- Feedback Collection: Students complete feedback forms to evaluate the course and instructor, contributing to course improvement.
- Reporting: Administrators generate reports on course registrations, revenue, and student feedback to evaluate the center’s performance and make informed decisions.

2.1.2. Bussiness Process in the System

From the real-world procedures above, we identify how the system supports business processes:

- ➢ Students:
    - Register or cancel course enrollments online.

- View registered courses, schedules, and materials.
- Pay fees and view payment history.
- Submit feedback on courses and instructors.

➢ Instructors:

- Access teaching schedules.
- Record attendance and evaluate students.
- View payslips with earnings and deductions.

➢ Administrative Staff:

- Confirm or reject registrations and manage student records.
- Verify payments and review feedback.
- Handle refunds and check schedule conflicts.
- Manage instructor salaries, including payroll calculation and finalization.

➢ Administrator:

- Manage courses, users, and content.
- Generate reports on registrations, revenue, and performance.
- Oversee system security and access control.

➢ Guests:

- Browse courses and register accounts without logging in.

2.2. System analysis

2.2.1. System specification

The course management system is a robust, web-based platform meticulously designed to streamline the organization, registration, and evaluation of educational programs at training centers. Built with a scalable and secure architecture, the system caters to multiple user roles—Guests, Students, Instructors, Administrative Staff, and Administrators—each with tailored interfaces and functionalities to enhance operational efficiency and user experience. By automating manual processes and providing real-time data access, the system addresses inefficiencies identified in traditional workflows, such as delays in registration confirmation, manual payment tracking, and error-prone payroll calculations, aligning with the overarching goal of optimizing educational administration.

The system features a public-facing interface accessible to unregistered users, enabling them to explore the training center's offerings without requiring authentication. This interface showcases detailed course information, including titles, descriptions, schedules, fees, and instructor profiles, presented in an intuitive layout with search and filter options. Guests can also initiate account creation through a streamlined registration process, requiring essential details like full name, email address, and phone number. For example, a prospective student visiting the website can browse a Python Programming course, view its syllabus and schedule, and register an account to enroll—all within a few clicks. The public interface is optimized for search engine visibility to attract potential learners and supports multilingual options to accommodate diverse audiences.

Registered students access a personalized dashboard upon secure login, where they can manage their academic journey. The system allows students to browse and search for available courses using advanced filters, such as course name, instructor, or schedule compatibility, and register for desired courses through an interactive form. Students can also cancel registrations before staff confirmation, providing flexibility for changing plans. For instance, a student might register for a Data Science course but cancel it if their schedule changes, receiving a confirmation email.

Post-registration, students access detailed course information, including session dates, times, room assignments, and downloadable materials like lecture slides or assignments. The system supports online fee payments via integrated gateways and tracks payment history, ensuring transparency. After course completion, students view their grades and instructor feedback and submit course evaluations with ratings and comments. Data is stored securely in a relational database, with access restricted to authorized users to protect privacy.

Instructors, upon logging in, access a dedicated dashboard displaying their teaching schedules, including course names, session dates, times, and room assignments. For example, an instructor assigned to a Web Development course can view all sessions for the month and prepare accordingly. The system enables instructors to record detailed attendance for each session, marking students as present, absent, or late, with optional notes. These records are updated in real-time, allowing administrators to monitor participation.

Instructors also evaluate student performance by entering grades and qualitative feedback post-course, which students can later access. Additionally, instructors can view their payslips, detailing earnings, deductions, and net pay for specific pay periods. This feature ensures transparency in compensation, addressing issues like delayed or erroneous manual payrolls. Course materials uploaded by administrators are also accessible to instructors, streamlining preparation.

Administrative Staff manage critical backend operations through a secure management interface. They review and confirm student registrations, updating statuses and contacting students if verification is needed. Staff also verify payments, ensuring fees are recorded accurately with details like amount, method, and date. For example, a staff member might confirm a $200 payment for a Machine Learning course via bank transfer, updating the system to reflect the transaction.

The system supports refund processing for canceled registrations, with staff logging refund amounts and reasons to maintain financial accuracy. Staff also manage schedules, checking for conflicts and creating or updating session details. For instance, they might reschedule a session to avoid overlapping with another course. Additionally, staff handle instructor payroll by calculating salaries based on teaching hours and compensation rules, reviewing calculations for accuracy, and finalizing payments. Student feedback is accessible for quality assurance, enabling staff to identify trends, such as high ratings for a specific instructor.

Administrators have comprehensive control over the system, managing courses, users, and content through an advanced dashboard. They create and update course records, including titles, descriptions, fees, and schedules, and upload materials like lecture notes or videos. For example, an administrator might add a new Cybersecurity course with a detailed syllabus and assign it to an instructor. User management includes creating, editing, or deactivating accounts for students, instructors, and staff, with role-based access control.

Administrators generate detailed reports on operational metrics, such as course registrations, revenue, average ratings, and top-performing courses. For instance, a report might show that Python Programming generated $10,000 in revenue with a 4.5/5 rating. They also oversee payroll processes, ensuring payments are authorized correctly, and monitor system security, including authentication logs and access permissions.

The system is built on a scalable, modular architecture using a web framework and a relational database for efficient data management. User authentication employs secure protocols to protect against unauthorized access. Password recovery is supported via email-based resets, and profile updates ensure data accuracy.

Real-time updates are facilitated through asynchronous processing for tasks like attendance recording and payment confirmation, ensuring minimal latency. The system integrates with external payment gateways for secure transactions and supports email notifications for key actions. Data security is prioritized with encrypted storage, CSRF protection, and prepared statements to prevent SQL injection, aligning with non-functional requirements.

The platform is optimized for performance, with page load times under 2 seconds and database queries indexed for efficiency, even with large datasets. It is fully responsive, supporting desktops, tablets, and smartphones, and includes multilingual support for accessibility. Scalability is ensured through cloud-based deployment, allowing the system to handle increasing users and courses.

Future enhancements, such as mobile apps or integration with learning management systems, are supported by the modular design.

The system directly addresses inefficiencies, such as manual registration delays, error-prone payment tracking, and inconsistent feedback collection. By automating course registration, payment processing, and payroll management, it reduces administrative workload and errors. Real-time scheduling prevents conflicts, while digital attendance and evaluation streamline instructor tasks. Comprehensive reporting empowers administrators to make data-driven decisions, enhancing the center’s performance.

Example Scenario:

Consider a student, Minh, who wants to enroll in a Data Analytics course. Minh visits the public interface, browses courses, and creates an account. After logging in, he searches for the course, registers, and pays the $150 fee online. Staff confirm his registration, and Minh accesses the course schedule and materials. During the course, the instructor records Minh’s attendance and later submits his grade. Minh views his results and submits feedback rating the course 4/5. Meanwhile, staff calculate the instructor’s salary, finalize it, and the instructor views their payslip. Administrators generate a report showing the course’s $5,000 revenue, using the data to plan future offerings.

2.2.2. Requirement Specification

2.2.2.1. Requirements Elicitation Techniques

¤ Interviews:

- ➢ Objective:

  The interviews aim to collect detailed information from key stakeholders—including administrators, instructors, staff, and students—to understand current challenges, needs, and expectations for the new course management system.

- ➢ Interview Participants:

  - Administrators: 1–2 people

- Instructors: 2–3 people
- Administrative Staff: 2–3 people
- Students: 2–3 people

➢ Interview structure: Each interview lasts 30–45 minutes and includes:

- Introduction: Explain the purpose and ensure confidentiality.
- Core Questions: Focus on current challenges, desired features, and expectations.
- Conclusion: Ask about the participant's willingness to try the new system.

➢ Questions:

Respondent	Question
Admin	What are the biggest challenges you face in managing the current course management system?
Admin	What features do you think the new system should have to manage courses more effectively?
Admin	What would your ideal course registration and management process look like?
Admin	How do you currently handle payments, and are there any difficulties?
Admin	What types of reports do you frequently need to generate (e.g., registrations, revenue)?
Admin	Are you willing to participate in testing the new system or provide additional feedback?
Instructor	How do you currently manage your teaching schedule and evaluate students?

Instructor	What tools or features would make your work easier?
Instructor	Do you have any concerns about switching to a new system?
Instructor	How do you communicate with students outside of class, and how could the system support this?
Instructor	What support do you need for managing course content (materials, assignments)?
Instructor	Are you willing to participate in testing the new system or provide additional feedback?
Staff	What are the main challenges in confirming registrations and handling payments?
Staff	What features do you think the new system should have to better support your work?
Staff	How would you like the process of managing student records to be improved?
Staff	What tools do you need to better manage feedback from students?
Staff	Are you willing to participate in testing the new system or provide additional feedback?
Student	What do you like and dislike about the current course registration process?
Student	What features would you like to see in the new system?
Student	How often do you access course materials online, and how is your current experience?
Student	Would you like to receive notifications about schedule changes? Through which channels?

Student	Are you willing to participate in testing the new system or provide additional feedback?

- Results:

No.	Questioner's Name	Responder's Name	Question	Answer Content
1	Bạch Phương Bình	Anh Tuấn (Admin)	What are the biggest challenges you face in managing the current course management system?	The current system is not automated, requires manual data entry, is time-consuming, and prone to errors.
2	Bạch Phương Bình	Anh Tuấn (Admin)	What features do you think the new system should have to manage courses more effectively?	Automation of registration, integration of payments, and quick report generation.

3	Bạch Phương Bình	Anh Tuấn (Admin)	What would your ideal course registration and management process look like?	Students self-register online, the system automatically confirms and sends notifications.
4	Bạch Phương Bình	Anh Tuấn (Admin)	How do you currently handle payments, and are there any difficulties?	Manual bank transfers, difficult to track and reconcile.
5	Bạch Phương Bình	Anh Tuấn (Admin)	What types of reports do you frequently need to generate (e.g., registrations, revenue)?	Revenue reports, registration numbers, and course progress.

6	Bạch Phương Bình	Anh Tuấn (Admin)	Are you willing to participate in testing the new system or provide additional feedback?	Yes, very willing.
7	Bạch Phương Bình	Thầy Minh (Instructor)	How do you currently manage your teaching schedule and evaluate students?	Using Excel, but it's hard to update when there are sudden changes.
8	Bạch Phương Bình	Thầy Minh (Instructor)	What tools or features would make your work easier?	Online calendar with automatic updates and online grading tools.
9	Bạch Phương Bình	Thầy Minh (Instructor)	Do you have any concerns about switching to a new system?	Concerned about a complex and hard-to-use interface.

10	Bạch Phương Bình	Thầy Minh (Instructor)	How do you communicate with students outside of class, and how could the system support this?	Using email, but would like integrated chat in the system.
11	Bạch Phương Bình	Thầy Minh (Instructor)	What support do you need for managing course content (materials, assignments)?	Online repository for materials and online assignment submissions.
12	Bạch Phương Bình	Thầy Minh (Instructor)	Are you willing to participate in testing the new system or provide additional feedback?	Yes, if given guidance on how to use it.
13	Bạch Phương Bình	Anh Đức (Staff)	What are the main challenges in confirming registrations and handling payments?	Manual checks, time-consuming and prone to mistakes.

14	Bạch Phương Bình	Anh Đức (Staff)	What features do you think the new system should have to better support your work?	Automatic confirmation and real-time payment tracking.
15	Bạch Phương Bình	Anh Đức (Staff)	How would you like the process of managing student records to be improved?	A centralized, easily searchable database.
16	Bạch Phương Bình	Anh Đức (Staff)	What tools do you need to better manage feedback from students?	A system that automatically collects feedback from students.
17	Bạch Phương Bình	Anh Đức (Staff)	Are you willing to participate in testing the new system or provide additional feedback?	Yes, if it saves time.

18	Bạch Phương Bình	Học viên Lan (Student)	What do you like and dislike about the current course registration process?	Like: simplicity; Dislike: long wait for confirmation.
19	Bạch Phương Bình	Học viên Lan (Student)	What features would you like to see in the new system?	Online registration, notifications via email or app.
20	Bạch Phương Bình	Học viên Lan (Student)	How often do you access course materials online, and how is your current experience?	Rarely, because materials are scattered and hard to find.
21	Bạch Phương Bình	Học viên Lan (Student)	Would you like to receive notifications about schedule changes? Through which channels?	Yes, via email or messages.

22	Bạch Phương Bình	Học viên Lan (Student)	Are you willing to participate in testing the new system or provide additional feedback?	Yes, if it is easy to use.

¤ Questionnaires:

➢ Objective:

The questionnaire targets students to gather both quantitative and qualitative data on their experiences and expectations with the system.

➢ Target Group:

- Primary Audience: Students

➢ Questionnaire structure: Divided into 4 sections:

- Current Experience
- Desired Features
- Usability
- Payment & Feedback

➢ Questionnaire:

Section	Question	Type	Options / Format
Section 1: Current Experience	How satisfied are you with the current course registration process?	Multiple Choice	1. Very dissatisfied 2. Dissatisfied 3. Neutral 4. Satisfied 5. Very satisfied
	What challenges do you	Open-ended	Text response

	face during the course registration process?		
	Overall, how satisfied are you with the current course management process?	Multiple Choice	1. Very dissatisfied 2. Dissatisfied 3. Neutral 4. Satisfied 5. Very satisfied
Section 2: Desired Features	What features would you like to see in the new system?	Checkbox (Multiple)	Online registration, Payment integration, View schedule, Provide feedback, Receive notifications
	Do you have additional feature suggestions?	Open-ended	Text response
Section 3: Usability	How important is mobile-friendliness of the system to you?	Multiple Choice	1. Not important 2. Slightly important 3. Neutral 4. Important 5. Very important
	What type of interface do you prefer for the system?	Multiple Choice w/ Other	Simple, Modern, Colorful, Other (Please specify: ________________)
Section 4: Payments and Feedback	How satisfied are you with the current payment process?	Multiple Choice	1. Very dissatisfied 2. Dissatisfied 3. Neutral 4. Satisfied

			5. Very satisfied
	How often do you provide feedback about courses?	Multiple Choice	Always, Occasionally, Rarely, Never
	What suggestions do you have to improve the feedback process?	Open-ended	Text response

- Results:

Question		Results
1	How satisfied are you with the current course registration process?	Average: 3 - Neutral
2	What challenges do you face during course registration?	Long waiting time for confirmation, unclear information.
3	Overall, how satisfied are you with the current course management process?	Average: 2 - Dissatisfied
4	What features would you like to see in the new system?	80% chose "Online registration," 70% chose "Receive notifications," 60% chose "Payment integration."
5	Do you have additional feature suggestions?	Mobile app notifications, online grade viewing.
6	How important is mobile-friendliness of the system to you?	Average: 4 - Important
7	What type of interface do you prefer for the system?	60% chose "Simple," 30% chose "Modern."
8	How satisfied are you with the current	Average: 2 - Dissatisfied

	payment process?	
9	How often do you provide feedback about courses?	50% chose "Rarely," 30% chose "Occasionally."
10	What suggestions do you have to improve the feedback process?	Create short, accessible surveys through the system.

2.2.2.2. Business Requirement

Challenges in the Course Management Process:

- Students face inconvenience due to the need for physical presence at training centers, causing delays for those with limited time or living far away.
- Manual registration and scheduling processes are error-prone, leading to inaccurate records, scheduling conflicts, or misplaced information.
- Scheduling conflicts between courses and instructors are common, and manual distribution of materials is time-consuming.
- Manual payment tracking causes errors in financial records and delays in revenue reporting.
- Manual payroll calculations for instructors are inefficient, leading to errors and delays in salary processing.
- Centers struggle to systematically collect and analyze student feedback, hindering course quality improvements.

2.2.2.3. Functional Requirements of the System

- ➢ Guests (Unregistered Users):
  - Browse available courses without authentication, viewing details such as titles, descriptions, schedules, fees, and instructor profiles.
  - Register for a new account by providing essential information (e.g., full name, email, phone number) to access student functionalities.
- ➢ Students:

- Log in and out securely using registered credentials to access personalized features.
- Manage personal account details, including updating full name, email address, and phone number, with secure password changes and recovery options.
- Register for courses through an interactive online form and cancel registrations before staff confirmation, with real-time status updates (e.g., registered, completed, cancelled).
- Access registered courses, including detailed schedules (session dates, times, room assignments), downloadable materials (e.g., slides, assignments), evaluation results (grades, instructor comments), and registration status.
- Pay course fees online via integrated payment gateways (e.g., credit card, bank transfer) and view payment history with details like amount, method, and date.
- Submit feedback on courses and instructors, including numerical ratings and textual comments to contribute to quality improvement.

➢ Instructors:

- View teaching schedules, including course names, session dates, times, and assigned rooms, to prepare for classes effectively.
- Record detailed attendance for each session, marking students as present, absent, or late, with optional notes for clarification (e.g., reason for absence).
- Evaluate student performance by submitting grades and qualitative comments after course completion, stored securely for student access.
- Access payslips detailing earnings, deductions (e.g., taxes), and net pay for specific pay periods, ensuring transparency in compensation.

- ➢ Administrative Staff:
    - Confirm or decline course registrations, updating statuses (e.g., “Confirmed,” “Rejected”) and contacting students for verification if needed.
    - Manage student records, including viewing and updating personal details (e.g., enrollment dates, notes) and tracking completed courses for academic history.
    - Verify and record student payments, logging details such as amount, payment method, and date to ensure financial accuracy.
    - Review student feedback, including ratings and comments, to assess course quality and learner satisfaction for continuous improvement.
    - Process refunds for canceled registrations, recording refund amounts and reasons to maintain accurate financial records.
    - Check for schedule conflicts between courses and instructors, ensuring no overlapping sessions to optimize resource allocation.
    - Create, update, or delete course schedules, specifying session details like dates, times, and rooms to maintain an accurate timetable.
    - Manage instructor salaries by calculating payroll based on teaching hours and compensation rules, reviewing calculations, and finalizing payments for processing.
- ➢ Administrators (Management Team):
    - Create and modify profiles for instructors and staff, including details like areas of expertise, biographical information, hire dates, and employment status (e.g., full-time, part-time).
    - Assign user roles (e.g., student, instructor, staff, admin) to enforce appropriate access levels and permissions within the system.

- Manage user accounts by editing, deleting, or locking accounts, with status updates (e.g., active, inactive) to maintain system integrity.
- View, edit, and delete student and course records, ensuring accurate details like course titles, descriptions, fees, and schedules.
- Upload and manage course content, such as syllabi, lecture notes, or videos, to support teaching and learning activities.
- Review attendance data for instructors and students, analyzing participation trends to monitor engagement and compliance.
- Oversee financial operations by tracking payment records (amounts, dates) and authorizing refunds or payroll payments where necessary.
- Generate statistical reports on operational metrics, including course registrations, revenue, average course ratings, and top-performing courses, to support data-driven decisions.
- Monitor system activity logs to track user actions (e.g., login attempts, data changes) for security and auditing purposes.
- Perform system maintenance tasks, such as database backups and software updates, to ensure reliability and performance.

2.2.2.4. Non-Functional Requirements

- User-Friendly Interface: The system should feature a simple, intuitive, and well-organized interface, allowing users to easily navigate and access features such as course registration, schedules, and feedback submission.
- Optimized Page Loading Speed: The system must ensure rapid page loading times to deliver a seamless user experience, particularly when displaying lists of courses or schedules.

- Database Performance: Ensure database queries are optimized with proper indexing to handle large datasets efficiently, especially for reports and user management.
- Mobile Device Support: The system should be fully compatible with mobile devices, such as smartphones and tablets, to ensure accessibility for users on the go.
- Data Security and Privacy Protection: The system must implement robust security measures, including secure authentication, data protection techniques, CSRF protection for all form submissions, and the use of prepared statements for all database queries to prevent SQL injection, safeguarding sensitive user information such as personal details and payment records.
- Reliability: The system should handle errors gracefully (e.g., database connection failures) and ensure high availability with minimal downtime.
- Multilingual Support: The system should offer language options to accommodate users from diverse linguistic backgrounds, thereby enhancing accessibility.
- Search Engine Optimization (SEO): The website should be designed and structured to achieve high visibility on search engines, increasing the training center's reach to potential students.
- Scalability: The system should be designed to handle growth, supporting an increasing number of users, courses, and registrations without compromising performance.
- Maintainability: The system should be structured to support easy maintenance, enabling straightforward bug fixes, upgrades, and the addition of new features as needed.

2.2.3. Identify actors and use cases

2.2.3.1 Identify actors

STT	Actor	Description
1	Admin	The person responsible for overall system management, including managing user accounts, courses, and generating reports on system activities.
2	Student	The person who registers for and participates in courses, manages their course registrations, views schedules, and provides feedback on courses.
3	Staff	The person who supports administrative tasks such as confirming course registrations, managing student records, and reviewing student feedback.
4	Instructor	The person responsible for teaching courses, managing teaching schedules, recording attendance, and evaluating student performance.
5	Guest	The person who has not logged into the system but can browse courses and register for an account.

2.2.3.2 Identify use cases

ID	Use Case	Description	Actor
UC01	Log in	Function that allows actors to log into the system using their credentials. Depending on the role, actors are directed to their respective dashboards.	Admin, Student, Staff, Instructor
UC02	Sign out	Allow logged-in users to log out of their accounts.	Admin, Student, Staff, Instructor
UC03	Change Password	Allow users to change their account password.	Admin, Student,

			Staff, Instructor
UC04	Reset Password	Allow users to reset their account password.	Admin, Student, Staff, Instructor
UC05	Manage personal profile	Allow users to view and update their personal account information, such as full name, email, and contact number.	Admin, Student, Staff, Instructor
UC06	Register for new course	Allow students to register for a new course by selecting a course and submitting registration details.	Student
UC07	Cancel registration	Allow students to cancel their course registration before it is confirmed by staff.	Student
UC08	View registered courses	Allow students to view the list of courses they have registered for, including registration status (e.g., registered, completed, cancelled).	Student
UC09	View personal schedule	Allow students to view the schedule of their registered courses, including session dates, times, and room assignments.	Student
UC10	Make Payment	Allow students to pay course fees online for their registered courses.	Student
UC11	View payment history	Allow students to view their payment history, including details such as amount, method, and date.	Student
UC12	Submit course	Allow students to submit feedback and ratings	Student

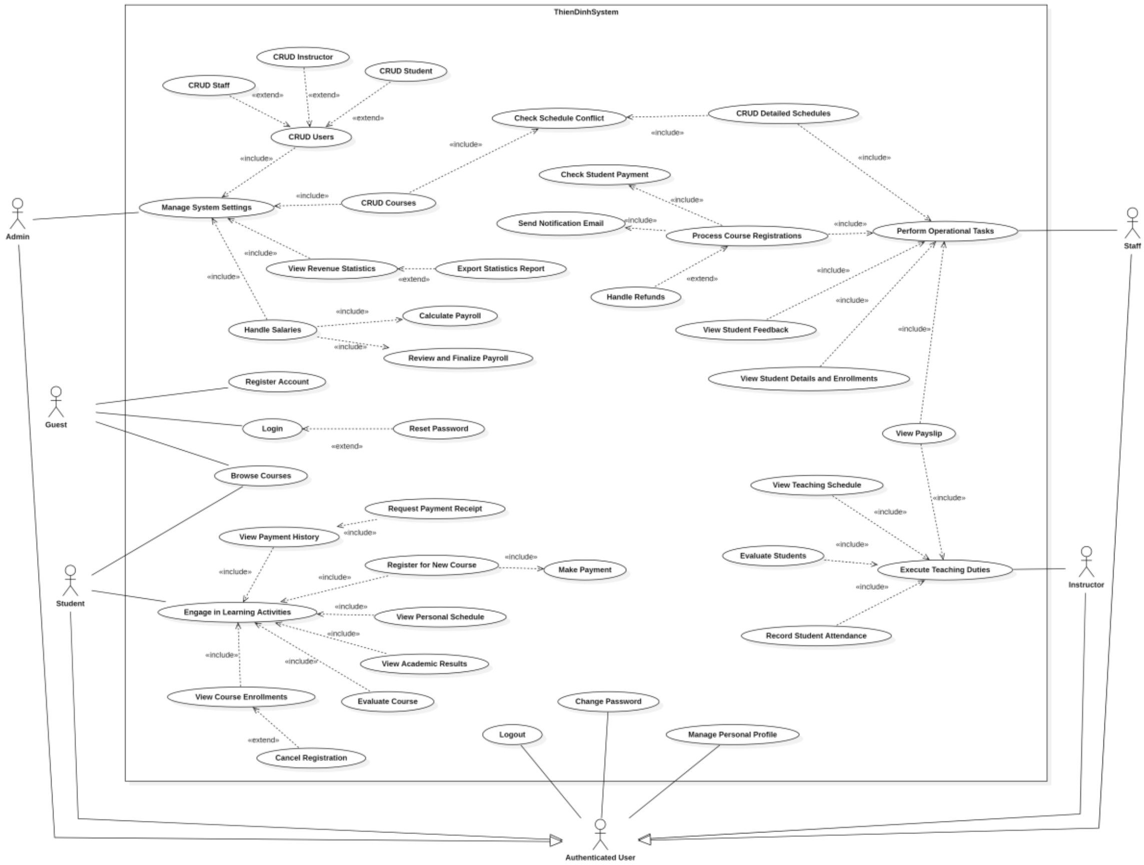
	feedback	fro a course and its instructor after completion	
UC13	Request Payment Receipt	Allow students to request an official payment receipt for completed transactions	Student
UC14	View teaching schedule	Allow instructors to view their teaching schedules, including session dates, times, and room assignments.	Instructor
UC15	Record student attendance	Allow instructors to record detailed attendance (e.g., present, absent, late) for each student in a session.	Instructor
UC16	Evaluate students	Allow instructors to enter grades and performance comments for students after course completion.	Instructor
UC17	Process course registration	Allow staff to confirm or decline student course registrations, updating enrollment statuses accordingly.	Staff
UC18	View student details and enrollments	Allow staff to view, update, and manage student records, including viewing the list of completed courses for each student.	Staff
UC19	Check student payment	Allow staff to verify and record student payments for course fees, including details such as amount, method, and date.	Staff
UC20	View student feedback	Allow staff to review feedback from students.	Staff
UC21	CRUD Users	Admin manages user accounts for students, instructors, and staff.	Admin
UC22	CRUD courses	Admin creates, updates, or deletes courses.	Admin
UC23	View revenue	Allow admins to generate statistical reports on	Admin

	statistics	registrations, revenue, average course ratings, and top-performing courses.	
UC24	Handle refunds	Admin handles refunds and cancellation returns.	Staff
UC25	Check Schedule Conflict	Allow staff to verify if any schedule conflicts exist when assigning instructors or courses.	Staff
UC26	CRUD Detailed Schedules	Allow staff to create, read, update, and delete detailed teaching schedules for instructors.	Staff
UC27	Send Notification Email	Allow staff to send email notifications to students about registration approval or changes.	Staff
UC28	View Academic Results	Allow students to view their academic results, such as grades or evaluations from instructors.	Student
UC29	Browse Courses	Allow students and guests to browse the list of available courses without needing to log in.	Student, Guest
UC30	Register Account	Allow guests to register for an account to become a student.	Guest
UC31	Export Statistics Report	Allow admins to export statistical reports (e.g., revenue, attendance) as PDF or Excel files for further analysis or sharing.	Admin
UC32	Handle Salaries	Allow Staff to manage instructor salaries, including calculating payroll and reviewing/finalizing payments.	Staff
UC33	Calculate Payroll	Allow Staff to calculate payroll for instructors based on teaching hours or other compensation rules.	Staff
UC34	Review and Finalize Payroll	Allow Staff to review calculated payroll and authorize payments to instructors.	Staff
UC35	View Payslip	Allow Instructors to view their payslips, including earnings, deductions, and payment	Instructor

		history.	

2.3. System design

2.3.1. Use case diagram



2.3.2. Use case specification

2.3.2.1 Log in

Use case name: Log in	ID: UC01	Priority: High
Actor: Admin, Student, Staff, Instructor		
Brief Description: The user logs into the system using their registered account to access role-specific functions.		
Trigger: The user attempts to access the system or use functions that require authentication but		

has not yet logged in. Type: ☑ External ☐ Temporal	
Precondition: • The user must have a registered account in the system (stored in the users table). • The system must be accessible (online and operational).	
Normal Course: 1. The user navigates to the login page of the system. 2. The system displays the login page with fields for username and password. 3. The user enters their username and password. 4. The user clicks the "Login" button. 5. The system verifies the credentials against the users table in the database. 6. If the credentials are correct, the system sets a session for the user. 7. The system redirects the user to their role-specific dashboard. 8. The system notifies the user of a successful login.	Information for Steps: • Step 1: The login page is accessible via a URL. • Step 2: The login page includes input fields for username and password, with a "Login" button. • Step 3: Username and password must match the data in the users table. • Step 4: The login request is sent to the server for processing. • Step 5: The system compares the entered credentials with stored data; if incorrect, an error is thrown (see Postconditions). • Step 6: Session data includes user ID and role. • Step 7: Dashboards are role-specific. • Step 8: A success message is displayed.
Postconditions:	

<table>
<tr><td colspan="4">• The user is successfully logged into the system.
• A session is created and stored in the system.
• The user is redirected to their role-specific dashboard.
• If the credentials are incorrect, the system displays an error message ("Invalid credentials, please try again") and prompts the user to re-enter their credentials.</td></tr>
<tr><th>Summary Inputs</th><th>Source</th><th>Outputs</th><th>Destination</th></tr>
<tr><td>Username
Password</td><td>User</td><td>User session
Role-specific dashboard
Success message</td><td>System
User</td></tr>
</table>

2.3.2.2 Sign out

<table>
<tr><td colspan="2">Use case name: Sign out</td><td>ID: UC02</td><td>Priority: Medium</td></tr>
<tr><td colspan="4">Actor: Admin, Student, Staff, Instructor</td></tr>
<tr><td colspan="4">Brief Description: The user logs out of the system, terminating their current session and returning to the login page.</td></tr>
<tr><td colspan="4">Trigger: The user wants to end their session after using the system.
Type: ☑ External ☐ Temporal</td></tr>
<tr><td colspan="4">Precondition:
• The user must be logged into the system with an active session (session data exists in the system).
• The system must be accessible (online and operational).</td></tr>
<tr><td colspan="2">Normal Course:</td><td colspan="2">Information for Steps:</td></tr>
</table>

1 The user clicks the "Sign out" button or link on their dashboard. 2 The system receives the sign-out request. 3 The system terminates the user's session by clearing session data. 4 The system redirects the user to the login page. 5 The system displays a confirmation message indicating successful logout.	• Step 1: The "Sign out" button/link is available on the user's dashboard. • Step 2: The request is sent to a logout script. • Step 3: The system uses a session management function to clear the user's session data. • Step 4: The login page URL is loaded. • Step 5: A message like "You have successfully logged out" is displayed on the login page.

Postconditions:

- The user's session is terminated, and they are no longer logged into the system.
- The user is redirected to the login page.
- The user must log in again to access any role-specific functions.

Summary Inputs	Source	Outputs	Destination
Sign-out request	User	Terminated session Login page Confirmation message	System User

2.3.2.3 Change Password

Use case name: Change Password	ID: UC03	Priority: Medium
Actor: Admin, Student, Staff, Instructor		

Brief Description: The user changes their account password by providing their current password and a new password.	
Trigger: The user wants to update their password for security reasons. Type: ☑ External ☐ Temporal	
Precondition: - The user must be logged into the system with an active session. - The user must know their current password.	
Normal Course: 1. The user navigates to the "Account Settings" section on their dashboard. 2. The user selects "Change Password." 3. The system displays a form with fields for current password, new password, and confirmation of the new password. 4. The user enters the required information and clicks "Submit." 5. The system verifies the current password against the users table in the database. 6. The system checks if the new password meets security requirements (e.g., minimum 8 characters, includes special characters).	Information for Steps: - Step 1: "Account Settings" is a menu option on the dashboard. - Step 2: "Change Password" is a sub-option under "Account Settings." - Step 3: The form includes three fields: current password, new password, and confirm new password. - Step 4: The user must ensure the new password and confirmation match. - Step 5: The system compares the current password with the stored data. - Step 6: Security requirements are predefined in the system (e.g., regex pattern). - Step 7: The new password is hashed and stored in the database. - Step 8: The success message confirms the

7. The system updates the password in the users table. 8. The system notifies the user: "Password changed successfully."	action.

Postconditions:

- The user’s password is updated in the users table.
- The user is logged out and must log in again using the new password.
- If the current password is incorrect, the system displays an error message ("Current password is incorrect, please try again") and prompts the user to re-enter.
- If the new password does not meet requirements or does not match the confirmation, the system displays an error message ("New password does not meet requirements" or "Passwords do not match") and prompts the user to correct the input.

Summary Inputs	Source	Outputs	Destination
Current password	User	Updated password	System
New password		Success message	User
Confirm new password		Login page	

2.3.2.4 Reset Password

Use case name: Reset Password	**ID: UC04**	**Priority: Medium**
Actor: Admin, Student, Staff, Instructor		
Brief Description: The user resets their account password by verifying their identity via email and setting a new password.		
Trigger: The user forgets their password and cannot log into the system.		

Type: ☑ External □ Temporal	
Precondition: • **The user must have a registered account with a valid email address.** • **The system must be accessible and capable of sending emails .**	
Normal Course: 1. The user clicks "Forgot Password" on the login page. 2. The system displays a form asking for the user’s email address. 3. The user enters their email address and clicks "Submit." 4. The system verifies the email address against the users table in the database. 5. The system generates a password reset link and sends it to the user’s email. 6. The user clicks the reset link in the email. 7. The system displays a form to enter a new password and confirm it. 8. The user submits the new password. 9. The system updates the password in the users table and notifies the user: "Password reset successfully."	Information for Steps: • Step 1: "Forgot Password" is a link on the login page (/login). • Step 2: The form includes a single field for email address. • Step 3: The email must match a record in the users table. • Step 4: The system checks if the email exists in the database. • Step 5: The reset link is a unique URL (e.g., /reset-password?token=xyz). • Step 6: The link is time-limited (e.g., expires in 1 hour). • Step 7: The form includes fields for new password and confirmation. • Step 8: The new password must meet security requirements (e.g., minimum 8 characters). • Step 9: The success message confirms the action.

Postconditions: • The user’s password is updated in the system. • The user can log in with the new password. • If the email is not found, the system displays an error: "Email not registered." • If the reset link expires, the system displays an error: "Link expired, please request a new one."			
Summary Inputs	Source	Outputs	Destination
Email address New password Confirm new password	User	Reset link Success message	System User

2.3.2.5 Manage personal profile

Use case name: Manage personal profile	ID: UC05	Priority: Medium
Actor: Admin, Student, Staff, Instructor		
Brief Description: The student views and updates their personal account information, such as full name, email, and contact number.		
Trigger: The user wants to update their personal information in the system. Type: ☑ External ☐ Temporal		
Precondition: • The student must be logged into the system with an active session. • The student’s personal information must exist in the student and users tables.		

<table>
<tr><td colspan="2">Normal Course:
1. The user navigates to the "Profile" section on their dashboard.
2. The system displays the user’s current profile information (e.g., full name, email, contact number).
3. The user updates their information as needed and clicks "Save."
4. The system validates the updated information (e.g., email format, contact number format).
5. The system updates the user’s profile in the users table.
6. The system notifies the user: "Profile updated successfully."</td><td colspan="2">Information for Steps:
• Step 1: "Profile" is a menu option on the dashboard.
• Step 2: Profile information is retrieved from the users table.
• Step 3: The user can edit fields like full name, email, and contact number.
• Step 4: Validation includes checking email format (e.g., regex) and contact number (e.g., 10 digits).
• Step 5: Updated data is saved to the database.
• Step 6: The success message confirms the action.</td></tr>
<tr><td colspan="4">Postconditions:
• The user’s profile information is updated in the system.
• If the email format is invalid, the system displays an error: "Invalid email format."
• If the contact number is invalid, the system displays an error: "Invalid contact number."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Updated profile information (full name, email, contact number)</td><td>User</td><td>Updated profile data
Success message</td><td>System
User</td></tr>
</table>

2.3.2.6 Register for a new course

Use case name: Register for a new course	ID: UC06	Priority: High
Actor: Student		
Brief Description: The student registers for a course by selecting an available course and submitting their registration details.		
Trigger: The student wants to enroll in a new course offered by the center. Type: ☑ External ☐ Temporal		
Precondition: • The student must be logged into the system with an active session. • The course must be available for registration (stored in the courses table with available slots).		
Normal Course: 1. The student navigates to the "Course Catalog" section on their dashboard. 2. The system displays a list of available courses with details (e.g., name, schedule, fees). 3. The student selects a course and clicks "Register." 4. The system displays a registration form with the course details and required fields (e.g., confirmation of intent). 5. The student submits the registration form. 6. The system checks if the course has	Information for Steps: • Step 1: "Course Catalog" is a menu option on the student dashboard. • Step 2: Course details are retrieved from the courses table. • Step 3: The "Register" button is available for courses with open slots. • Step 4: The registration form includes course details and a checkbox for confirmation. • Step 5: The form submission sends the registration request to the server. • Step 6: The system compares the current number of registrations with	

available slots. 7. The system records the registration in the registrations table with a "pending" status. 8. The system notifies the student: "Registration submitted, awaiting confirmation."	the course capacity. • Step 7: The registration record includes student ID, course ID, and status. • Step 8: The notification confirms the action.
Postconditions: • The student’s registration is recorded in the system with a "pending" status. • The student must wait for Staff confirmation (see UC17). • If the course is full, the system displays an error: "Course is full, please choose another course."	

Summary Inputs	Source	Outputs	Destination
Course selection	Student	Registration record Success message	System Student

2.3.2.7 Cancel registration

Use case name: Cancel course registration	ID: UC07	Priority: Medium
Actor: Student		
Brief Description: The student cancels their registration for a course before it is confirmed by staff.		
Trigger: The student decides they no longer want to take a registered course.		

Type: ☑ External ☐ Temporal	
Precondition: • The student must be logged into the system with an active session. • The student must have a pending registration (status = "pending" in the registrations table).	
Normal Course: 1. The student navigates to the "My Courses" section on their dashboard. 2. The system displays a list of the student’s registered courses with their statuses. 3. The student selects a pending registration and clicks "Cancel." 4. The system prompts for confirmation: "Are you sure you want to cancel this registration?" 5. The student confirms the cancellation. 6. The system updates the registration status to "cancelled" in the registrations table. 7. The system notifies the student: "Registration canceled successfully."	Information for Steps: • Step 1: "My Courses" is a menu option on the student dashboard. • Step 2: The list includes course name, status, and action buttons (e.g., "Cancel"). • Step 3: The "Cancel" button is only available for pending registrations. • Step 4: The confirmation prompt prevents accidental cancellations. • Step 5: The student clicks "Yes" to confirm. • Step 6: The status is updated in the database. • Step 7: The success message confirms the action.
Postconditions: • The student’s registration is marked as "cancelled" in the system. • The course slot becomes available for other students.	

- If the registration is already confirmed, the system displays an error: "Cannot cancel, registration already confirmed."

Summary Inputs	Source	Outputs	Destination
Cancellation request Confirmation	Student	Updated registration status Success message	System Student

2.3.2.8 View registered courses

Use case name: View registered courses	ID: UC08	Priority: High

Actor: Student

Brief Description: The student views the list of courses they have registered for, including registration status (e.g., registered, completed, cancelled).

Trigger: The student wants to check their registered courses and their statuses.

Type: ☑ External □ Temporal

Precondition:

- The student must be logged into the system with an active session.

Normal Course:	Information for Steps:
1. The student navigates to the "My Courses" section on their dashboard. 2. The system retrieves the student’s registrations from the registrations table. 3. The system displays a list of registered courses with their statuses (e.g., pending,	• Step 1: "My Courses" is a menu option on the student dashboard. • Step 2: The system queries the registrations table using the student’s ID. • Step 3: The list includes course

confirmed, completed, cancelled).	name, registration date, and status.

Postconditions:

- The student views their registered courses and statuses.
- If no registrations exist, the system displays a message: "No registered courses."

Summary Inputs	Source	Outputs	Destination
Request to view courses	Student	List of registered courses	System Student

2.3.2.9 View personal schedule

Use case name: View **personal** schedule	ID: UC09	Priority: Medium

Actor: Student

Brief Description: The student views the schedule of their registered courses, including session dates, times, and room assignments.

Trigger: The student wants to check their course schedule to plan their attendance.

Type: ☑ External ☐ Temporal

Precondition:

- The student must be logged into the system with an active session (role: Student).
- The student must have at least one registered course with a schedule in the schedule table.

Normal Course: 1. The student navigates to the "My Schedule" section on their	Information for Steps: - Step 1: "My Schedule" is a menu option on the student dashboard.

dashboard. 2. The system retrieves the student's confirmed registrations and associated schedules from the registrations and schedules tables. 3. The system displays the schedule, including session dates, times, and room assignments.	• Step 2: The system joins the registrations and schedules tables to fetch the schedule for confirmed courses. • Step 3: The schedule includes details like "Session 1: 08/07/2025, 18:00-21:00, Room A102."

Postconditions: • The student views their course schedule. • If no schedule is available, the system displays a message: "No schedule available."

Summary Inputs	Source	Outputs	Destination
Request to view schedule	Student	Course schedule	System Student

2.3.2.10 Make payment

Use case name: Pay course fees	ID: UC10	Priority: High
Actor: Student		
Brief Description: The student pays course fees online for their registered courses.		
Trigger: The student needs to pay fees for a confirmed course registration. Type: ☑ External ☐ Temporal		
Precondition: • The student must be logged into the system with an active session.		

• The student must have a confirmed course registration with unpaid fees (status = "confirmed", payment_status = "unpaid" in the registrations table).	
Normal Course: 1. The student navigates to the "Payments" section on their dashboard. 2. The system retrieves the student's unpaid course fees from the registrations table. 3. The system displays a list of unpaid fees with course details. 4. The student selects a course and clicks "Pay Now." 5. The system redirects the student to a payment gateway (e.g., VNPay, PayPal). 6. The student completes the payment on the gateway. 7. The system receives the payment confirmation from the gateway. 8. The system updates the payment_status to "paid" in the registrations table. 9. The system records the payment details (amount, method, date) in the payments table. 10. The system notifies the student: "Payment successful."	Information for Steps: • Step 1: "Payments" is a menu option on the student dashboard. • Step 2: The system queries the registrations table for unpaid fees. • Step 3: The list includes course name, fee amount, and a "Pay Now" button. • Step 4: The "Pay Now" button initiates the payment process. • Step 5: The payment gateway URL is provided by the system (e.g., /pay?course_id=123). • Step 6: The student enters payment details (e.g., card number). • Step 7: The gateway sends a success or failure response to the system. • Step 8: The payment_status field is updated in the database. • Step 9: Payment details are stored for future reference (e.g., in UC11). • Step 10: The success message confirms the action.
Postconditions: • The student's course registration is marked as paid. • The payment details are recorded in the system.	

• If the payment fails, the system displays an error: "Payment failed, please try again."			
Summary Inputs	Source	Outputs	Destination
Registered request Payment details	Student	Updated payment status Payment record Success message	System Student

2.3.2.11 View payment history

Use case name: View payment history	ID: UC11	Priority: Medium
Actor: Student		
Brief Description: The student views their payment history, including details such as amount, method, and date.		
Trigger: The student wants to review their past payments for courses. Type: ☑ External ☐ Temporal		
Precondition: • The student must be logged into the system with an active session.		
Normal Course: 1. The student navigates to the "Payment History" section on their dashboard. 2. The system retrieves the student's payment records from the payments table. 3. The system displays a list of	Information for Steps: • Step 1: "Payment History" is a menu option on the student dashboard. • Step 2: The system queries the payments table using the student's ID. • Step 3: The list includes details like "Course: Digital Marketing, Amount: 3,500,000 VND,	

payments with details (e.g., amount, method, date, course name).	Method: VNPay, Date: 01/07/2025."
Postconditions: • The student views their payment history. • If no payments exist, the system displays a message: "No payment history available."	

Summary Inputs	Source	Outputs	Destination
Request to view payment history	Student	Payment history list	System Student

2.3.2.12 Submit course feedback

Use case name: Submit course feedback	ID: UC12	Priority: Medium
Actor: Student		
Brief Description: The student submits feedback and ratings for a course and its instructor after completion.		
Trigger: The student wants to provide feedback on a completed course. Type: ☑ External □ Temporal		
Precondition: • The student must be logged into the system with an active session. • The student must have completed a course.		
Normal Course: 1. The student navigates to the "My Courses" section on their dashboard. 2. The system displays a list of the	Information for Steps: • Step 1: "My Courses" is a menu option on the student dashboard. • Step 2: The list includes courses with	

student’s completed courses. 3. The student selects a completed course and clicks "Submit Feedback." 4. The system displays a feedback form with rating fields (e.g., 1-5 stars for course, instructor) and a comment section. 5. The student fills out the form and clicks "Submit." 6. The system records the feedback in the feedback table. 7. The system notifies the student: "Feedback submitted successfully."	status = "completed." • Step 3: The "Submit Feedback" button is available for completed courses. • Step 4: The form includes fields for ratings and comments. • Step 5: The student provides ratings (e.g., 4 stars) and comments (e.g., "Great course!"). • Step 6: The feedback record includes student ID, course ID, ratings, and comments. • Step 7: The success message confirms the action.

Postconditions:

- The student’s feedback is recorded in the system.
- The feedback is available for Staff to review.
- If the course is not completed, the system displays an error: "Cannot submit feedback until the course is completed."

Summary Inputs	Source	Outputs	Destination
Feedback (ratings, comments)	Student	Feedback record, Success message	System Student

2.3.2.13 Request payment receipt

Use case name: Request payment receipt	ID: UC13	Priority: Medium
Actor: Student		
Brief Description: The student requests a payment receipt for a completed payment		

transaction for a course, which can be used for financial records or reimbursement purposes.	
Trigger: The student wants to obtain a receipt for a payment they made for a course. Type: ☑ External □ Temporal	
Precondition: • The student must be logged into the system with an active session. • The student must have at least one completed payment transaction for a course (stored in the payments table).	
Normal Course: 1. The student navigates to the "Payment History" section on their dashboard. 2. The system retrieves the student's payment records from the payments table. 3. The system displays a list of completed payments with details (e.g., amount, method, date, course name). 4. The student selects a payment and clicks "Request Receipt." 5. The system generates a payment receipt in PDF format, including details like student name, course name, amount, payment method, and date. 6. The system provides a download link for the receipt. 7. The student downloads the receipt. 8. The system notifies the student:	Information for Steps: • Step 1: "Payment History" is a menu option on the student dashboard. • Step 2: The system queries the payments table using the student's ID. • Step 3: The list includes details like "Course: Digital Marketing, Amount: 3,500,000 VND, Method: VNPay, Date: 01/07/2025." • Step 4: The "Request Receipt" button is available for each completed payment. • Step 5: The receipt is generated using a PDF library (e.g., iTextPDF) and includes a unique receipt number. • Step 6: The download link is a temporary URL. • Step 7: The student clicks the link to download the receipt to their device.

<table>
<tr><td colspan="2">"Payment receipt generated successfully."</td><td colspan="2">• Step 8: The success message confirms the action.</td></tr>
<tr><td colspan="4">Postconditions:
• The payment receipt is generated and downloaded by the student.
• If no payment history exists, the system displays a message: "No payment history available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Payment selection</td><td>Student</td><td>Payment receipt (PDF)
Download link
Success message</td><td>System
Student</td></tr>
</table>

2.3.2.14 View teaching schedule

<table>
<tr><td>Use case name: View teaching schedule</td><td>ID: UC14</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actor: Instructor</td></tr>
<tr><td colspan="3">Brief Description: The instructor views their teaching schedules, including session dates, times, and room assignments.</td></tr>
<tr><td colspan="3">Trigger: The instructor wants to check their teaching schedule to prepare for sessions.
Type: ☑ External ☐ Temporal</td></tr>
<tr><td colspan="3">Precondition:
• The instructor must be logged into the system with an active session (role: Instructor).
• The instructor must have at least one assigned course with a schedule in the schedule table.</td></tr>
</table>

<table>
<tr><td colspan="2">Normal Course:
1. The instructor navigates to the "My Schedule" section on their dashboard.
2. The system retrieves the instructor’s teaching schedules from the schedules table.
3. The system displays the schedule, including session dates, times, and room assignments.</td><td colspan="2">Information for Steps:
- Step 1: "My Schedule" is a menu option on the instructor dashboard.
- Step 2: The system queries the schedules table using the instructor’s ID.
- Step 3: The schedule includes details like "Course: Digital Marketing, Session: 01/07/2025, 19:00-21:00, Room D404."</td></tr>
<tr><td colspan="4">Postconditions:
- The instructor views their teaching schedule.
- If no schedule is available, the system displays a message: "No schedule available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Request to view schedule</td><td>Instuctor</td><td>Teaching schedule</td><td>System
Instuctor</td></tr>
</table>

2.3.2.15 Record student attendance

<table>
<tr><td>Use case name: Record student attendance</td><td>ID: UC15</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actor: Instructor</td></tr>
<tr><td colspan="3">Brief Description: The instructor records detailed attendance (e.g., present, absent, late) for each student in a session.</td></tr>
<tr><td colspan="3">Trigger: The instructor needs to record attendance for a session they are teaching.
Type: ☑ External ☐ Temporal</td></tr>
</table>

<table>
<tr><td colspan="2">Precondition:
• The instructor must be logged into the system with an active session (role: Instructor).
• The instructor must have at least one assigned course with a schedule in the schedule table.
• The session must have students enrolled (via the registration table).</td></tr>
<tr><td>Normal Course:
1. The instructor navigates to the "Attendance" section on their dashboard.
2. The system displays a list of scheduled sessions for the instructor’s courses.
3. The instructor selects a session.
4. The system displays a list of enrolled students for that session.
5. The instructor marks each student as present, absent, or late.
6. The instructor clicks "Save."
7. The system records the attendance in the attendance table.
8. The system notifies the instructor: "Attendance recorded successfully."</td><td>Information for Steps:
• Step 1: "Attendance" is a menu option on the instructor dashboard.
• Step 2: The list includes session details (e.g., date, time, course name).
• Step 3: The instructor selects a session to record attendance.
• Step 4: The list includes student names (retrieved from the registrations table).
• Step 5: Attendance options are dropdowns (e.g., Present, Absent, Late).
• Step 6: The "Save" button submits the attendance data.
• Step 7: The attendance record includes session ID, student ID, and status.
• Step 8: The success message confirms the action.</td></tr>
<tr><td colspan="2">Postconditions:
• The attendance data is recorded in the system.
• The data is available for Admin to view (see UC23).</td></tr>
</table>

• If no students are enrolled, the system displays a message: "No students enrolled in this session."			
Summary Inputs	Source	Outputs	Destination
Attendance data (present, absent, late)	Instructor	Attendance records Success message	System Intructor

2.3.2.16 Evaluate student

Use case name: Evaluate student performance	ID: UC16	Priority: High
Actor: Instructor		
Brief Description: The instructor enters grades and performance comments for students after course completion.		
Trigger: The instructor needs to evaluate students at the end of a course. Type: ☑ External ☐ Temporal		
Precondition: • The instructor must be logged into the system with an active session. • The course must be completed. • The course must have enrolled students.		
Normal Course: 1. The instructor navigates to the "Evaluate Performance" section on their dashboard. 2. The system retrieves the instructor’s assigned courses that are completed.	Information for Steps: • Step 1: The "Evaluate Performance" section is accessible via a link on the instructor dashboard. • Step 2: The system queries the database to find completed courses assigned to the instructor (e.g., via InstructorID and	

3. The system displays a list of completed courses with a "Evaluate" button for each course. 4. The instructor selects a course and clicks the "Evaluate" button. 5. The system retrieves the list of students enrolled in the course from the registration table. 6. The system displays the list of students with fields to enter grades and comments for each student. 7. The instructor enters grades and comments (e.g., "Excellent participation") for each student and clicks the "Submit" button. 8. The system validates the grades (e.g., ensures they are within the range 0-100). 9. The system saves the grades and comments in the evaluation table. 10. The system notifies the instructor that the evaluations have been recorded successfully.	registration status "completed"). • Step 3: The list shows columns like Course Name and a "Evaluate" button for each completed course. • Step 4: The "Evaluate" button loads the evaluation page for the selected course. • Step 5: The system queries the registration table to get the list of students enrolled in the course (via CourseID). • Step 6: The list shows each student's name (joined with student and users tables) with input fields for grade (numeric) and comments (text area). • Step 7: The instructor enters a grade (e.g., 85) and optional comments for each student. • Step 8: Validation ensures grades are between 0 and 100; comments are optional but may have a length limit (e.g., 500 characters). • Step 9: The system updates or inserts records into the evaluation table with fields like StudentID, CourseID, InstructorID, Grade, and Comments. • Step 10: A success message like "Student evaluations for [Course Name] recorded successfully" is displayed.
Postconditions:	

<table>
<tr><td colspan="4">• The student evaluations (grades and comments) are saved in the evaluation table.
• The instructor can no longer evaluate the same students for the same course (to prevent duplicates).
• If evaluations have already been submitted for the course, the system displays a message ("Evaluations for this course have already been submitted").
• If validation fails (e.g., invalid grade), the system displays an error message ("Grades must be between 0 and 100") and prompts the instructor to correct the input.</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Grades
Comments</td><td>Instructor</td><td>Evaluation records
Success message</td><td>System
Instructor</td></tr>
</table>

2.3.2.17 Process course registration

<table>
<tr><td>Use case name: Confirm course registration</td><td>ID: UC17</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actor: Staff</td></tr>
<tr><td colspan="3">Brief Description: The staff confirms or declines student course registrations, updating enrollment statuses accordingly.</td></tr>
<tr><td colspan="3">Trigger: The staff needs to process pending student registrations.
Type: ☑ External □ Temporal</td></tr>
<tr><td colspan="3">Precondition:
• The staff must be logged into the system with an active session (role: Staff).
• There must be at least one pending course registration.</td></tr>
<tr><td>Normal Course:
1. The staff navigates to the "Registrations" section on their</td><td colspan="2">Information for Steps:
• Step 1: "Registrations" is a menu option on the staff dashboard.</td></tr>
</table>

dashboard. 2. The system displays a list of pending registrations with student and course details. 3. The staff selects a registration to review. 4. The staff clicks "Confirm" or "Decline." 5. The system updates the registration status to "confirmed" or "declined" in the registrations table. 6. The system triggers a notification email to the student (see UC29). 7. The system notifies the staff: "Registration processed successfully."	• Step 2: The list includes student name, course name, and registration date. • Step 3: The staff can view detailed registration information (e.g., student profile). • Step 4: "Confirm" or "Decline" buttons are available for each registration. • Step 5: The status is updated in the database. • Step 6: The notification email is sent via UC29. • Step 7: The success message confirms the action.

Postconditions:

- The registration status is updated to "confirmed" or "declined" in the registration table.
- If declined, the decline reason is recorded in the registration table.
- The registration no longer appears in the pending list for the staff.
- If the registration has already been processed, the system displays a message ("This registration has already been processed").

Summary Inputs	Source	Outputs	Destination
Confirmation/Decline decision	Staff	Updated registration status	System Staff

		Notification trigger Success message	Student

2.3.2.18 View student details and enrollments

<table>
<tr><td>Use case name: View student details and enrollments</td><td>ID: UC18</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actor: Staff</td></tr>
<tr><td colspan="3">Brief Description: The staff views, updates, and manages student records, including their enrollments.</td></tr>
<tr><td colspan="3">Trigger: The staff needs to review or update a student’s information or enrollment history.
Type: ☑ External ☐ Temporal</td></tr>
<tr><td colspan="3">Precondition:
• The staff must be logged into the system with an active session.</td></tr>
<tr><td>Normal Course:
1. The staff navigates to the "Student Records" section on their dashboard.
2. The system displays a list of students with basic details (e.g., name, email).
3. The staff selects a student to view their details.
4. The system displays the student’s profile and enrollment history (e.g., completed courses).
5. The staff updates the student’s information if needed (e.g., email, contact number).</td><td colspan="2">Information for Steps:
• Step 1: "Student Records" is a menu option on the staff dashboard.
• Step 2: The list includes student ID, name, and email (retrieved from the users table).
• Step 3: The staff selects a student to view more details.
• Step 4: The enrollment history includes course names and statuses (retrieved from the registrations table).
• Step 5: The staff can edit fields like email or contact number.</td></tr>
</table>

6. The system saves the changes in the users table. 7. The system notifies the staff: "Student record updated successfully."	• Step 6: Updated data is saved to the database. • Step 7: The success message confirms the action.
Postconditions: • The staff views or updates the student’s records. • If no students exist, the system displays a message: "No student records available."	

Summary Inputs	Source	Outputs	Destination
Request to view/update Updated student info	Staff	Student details Enrollment history Success message	System Staff

2.3.2.19 Check student payments

Use case name: Check student payments	ID: UC19	Priority: High
Actor: Staff		
Brief Description: The staff verifies and records student payments for course fees, including details such as amount, method, and date.		
Trigger: The staff needs to confirm that a student has paid their course fees. Type: ☑ External ☐ Temporal		
Precondition: • The staff must be logged into the system with an active session. • There must be pending payments.		

<table>
<tr><td colspan="2">Normal Course:
1. The staff navigates to the "Payments" section on their dashboard.
2. The system displays a list of pending payments with student and course details.
3. The staff selects a payment to verify.
4. The staff confirms the payment details (e.g., amount, method, date).
5. The staff marks the payment as "Received."
6. The system updates the payment_status to "paid" in the registrations table.
7. The system records the payment details in the payments table.
8. The system notifies the staff: "Payment recorded successfully."</td><td colspan="2">Information for Steps:
• Step 1: "Payments" is a menu option on the staff dashboard.
• Step 2: The list includes student name, course name, and fee amount.
• Step 3: The staff selects a payment to review.
• Step 4: Payment details may be verified manually (e.g., via bank transfer receipt).
• Step 5: The "Received" button updates the payment status.
• Step 6: The payment_status field is updated in the database.
• Step 7: Payment details include amount, method, and date.
• Step 8: The success message confirms the action.</td></tr>
<tr><td colspan="4">Postconditions:
• The payment status is updated in the system.
• The payment details are recorded for future reference (e.g., in UC11).
• If no payments are pending, the system displays a message: "No pending payments."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Payment verification
Payment details</td><td>Staff</td><td>Updated payment status</td><td>System
Staff</td></tr>
</table>

		Payment record Success message	

2.3.2.20 View student feedback

Use case name: View student feedback	ID: UC20	Priority: Medium
Actor: Staff		
Brief Description: The staff reviews feedback and ratings submitted by students to assess course quality and learner satisfaction.		
Trigger: The staff wants to review student feedback to improve course quality. Type: ☑ External ☐ Temporal		
Precondition: • The staff must be logged into the system with an active session. • There must be submitted feedback		
Normal Course: 1. The staff navigates to the "Feedback" section on their dashboard. 2. The system retrieves feedback records from the feedback table. 3. The system displays a list of feedback entries with details (e.g., course, ratings, comments).	Information for Steps: • Step 1: "Feedback" is a menu option on the staff dashboard. • Step 2: The system queries the feedback table for all records. • Step 3: The list includes course name, student ID, ratings (e.g., 4 stars), and comments (e.g., "Great course!").	
Postconditions:		

<table>
<tr><td colspan="4">• The staff views the student feedback.
• If no feedback is available, the system displays a message: "No feedback available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Request to view feedback</td><td>Staff</td><td>Feedback list</td><td>System
Staff</td></tr>
</table>

2.3.2.21 CRUD Users

<table>
<tr><td colspan="2">Use case name: CRUD Users</td><td>ID: UC21</td><td>Priority: High</td></tr>
<tr><td colspan="4">Actor: Admin</td></tr>
<tr><td colspan="4">Brief Description: The admin manages user accounts for students, instructors, and staff (create, read, update, delete).</td></tr>
<tr><td colspan="4">Trigger: The admin needs to add, update, or remove user accounts in the system.
Type: ☑ External □ Temporal</td></tr>
<tr><td colspan="4">Precondition:
• The admin must be logged into the system with an active session.</td></tr>
<tr><td colspan="2">Normal Course:
1. The admin navigates to the "User Management" section on their dashboard.
2. The system displays a list of existing users with details (e.g., name, role, email).
3. The admin chooses an action: Create, Update, or Delete.
○ Create: The admin clicks "Add New User," enters user details (e.g., name, email, role,</td><td colspan="2">Information for Steps:
• Step 1: "User Management" is a menu option on the admin dashboard.
• Step 2: The list includes user ID, name, role, and email.
• Step 3: Actions are available as buttons (e.g.,</td></tr>
</table>

<table>
<tr><td colspan="2">password), and clicks "Save."
○ Update: The admin selects a user, edits their details, and clicks "Save."
○ Delete: The admin selects a user and clicks "Delete."
4. The system validates the input data (e.g., email uniqueness).
5. The system updates the users table (create, update, or delete the user record).
6. The system notifies the admin: "User account updated successfully."</td><td colspan="2">"Add New User", "Edit", "Delete").
• Step 4: Validation ensures email is unique and role is valid (e.g., Admin, Student).
• Step 5: The database is updated based on the action (insert, update, or delete).
• Step 6: The success message confirms the action.</td></tr>
<tr><td colspan="4">Postconditions:
• The user account is created, updated, or deleted in the system.
• If the email already exists (for create/update), the system displays an error: "Email already exists."
• If the user has active enrollments or assignments (for delete), the system displays an error: "Cannot delete user with active records."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>User details (name, email, role, password)</td><td>Admin</td><td>Updated user record
Success message</td><td>System
Admin</td></tr>
</table>

2.3.2.22 CRUD courses

Use case name: Manage courses	ID: UC22	Priority: High

Actor: Admin	
Brief Description: The admin creates, views, updates, and deletes course information, such as titles, descriptions, and fees.	
Trigger: The admin needs to manage the list of available courses Type: ☑ External □ Temporal	
Precondition: • The admin must be logged into the system with an active session.	
Normal Course: 1. The admin navigates to the "Course Management" section on their dashboard. 2. The system displays a list of existing courses with details (e.g., name, schedule, fees). 3. The admin chooses an action: Create, Update, or Delete. a. Create: The admin clicks "Add New Course," enters course details (e.g., name, schedule, fees, capacity), and clicks "Save." b. Update: The admin selects a course, edits its details, and clicks "Save." c. Delete: The admin selects a course and clicks "Delete." 4. The system validates the input data (e.g., schedule format, capacity limit). 5. The system updates the courses table (create, update, or delete the course record).	Information for Steps: • Step 1: "Course Management" is a menu option on the admin dashboard. • Step 2: The list includes course ID, name, schedule, and fees. • Step 3: Actions are available as buttons (e.g., "Add New Course", "Edit", "Delete"). • Step 4: Validation ensures schedule is valid and capacity is reasonable (e.g., positive integer). • Step 5: The database is updated based on the action (insert, update, or delete). • Step 6: The success message

6. The system notifies the admin: "Course updated successfully."	confirms the action.

Postconditions:

- The course is created, updated, or deleted in the system.
- If the course has active enrollments (for delete), the system displays an error: "Cannot delete course with active enrollments."

Summary Inputs	Source	Outputs	Destination
Course details (name, schedule, fees, capacity)	Admin	Updated course record Success message	System Admin

2.3.2.23 View revemue statistics

Use case name: View revenue statistics	ID: UC23	Priority: Medium

Actor: Admin

Brief Description: The admin generates statistical reports on registrations, revenue, average course ratings, and top-performing courses.

Trigger: The admin wants to analyze financial and performance metrics of the center.

Type: ☑ External ☐ Temporal

Precondition:

- The admin must be logged into the system with an active session.
- Payment and feedback data must be available (stored in the payments and feedback tables).

Normal Course:	Information for Steps:

<table>
<tr><td colspan="2">1. The admin navigates to the "Reports" section on their dashboard.
2. The admin selects "Revenue Statistics."
3. The system retrieves data from the payments and feedback tables.
4. The system displays the statistics report with details (e.g., total revenue, number of registrations, average course ratings, top-performing courses).</td><td colspan="2">- Step 1: "Reports" is a menu option on the admin dashboard.
- Step 2: "Revenue Statistics" is a sub-option under "Reports."
- Step 3: The system queries the payments table for revenue data and the feedback table for ratings.
- Step 4: The report includes metrics like "Total Revenue: 50,000,000 VND, Registrations: 100, Average Rating: 4.2/5."</td></tr>
<tr><td colspan="4">Postconditions:
- The admin views the revenue statistics report.
- If no data is available, the system displays a message: "No revenue data available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Request to view revenue statistics</td><td>Admin</td><td>Revenue statistics report</td><td>System
Admin</td></tr>
</table>

2.3.2.24 Handle refunds

<table>
<tr><td>Use case name: Handle refunds</td><td>ID: UC24</td><td>Priority: Medium</td></tr>
<tr><td colspan="3">Actor: Staff</td></tr>
<tr><td colspan="3">Brief Description: The staff processes refund requests for students who cancel their course registrations, ensuring proper financial adjustments.</td></tr>
<tr><td colspan="3">Trigger: A student cancels a registration, and a refund request is initiated
Type: ☑ External □ Temporal</td></tr>
</table>

Precondition: • The staff must be logged into the system with an active session. • The student must have a cancelled registration (status = "cancelled" in the registrations table). • The student must have paid for the course (payment_status = "paid" in the registrations table).	
Normal Course: 1. The staff navigates to the "Refunds" section on their dashboard. 2. The system displays a list of cancelled registrations with refund eligibility. 3. The staff selects a cancelled registration to process the refund. 4. The system displays the refund details (e.g., student name, course name, paid amount, refund amount). 5. The staff confirms the refund amount and clicks "Process Refund." 6. The system updates the payment_status to "refunded" in the registrations table. 7. The system records the refund transaction in the payments table (e.g., as a negative amount). 8. The system triggers a notification email to the student about the refund (using UC29).	Information for Steps: • Step 1: "Refunds" is a menu option on the staff dashboard. • Step 2: The list includes student name, course name, and refund eligibility (based on cancellation date and refund policy). • Step 3: The staff selects a registration to process. • Step 4: Refund details include the original paid amount and calculated refund (e.g., 80% of the fee). • Step 5: The "Process Refund" button initiates the refund process. • Step 6: The payment_status field is updated in the database. • Step 7: The refund transaction includes student ID, course ID, amount, and date. • Step 8: The notification email is sent

9. The system notifies the staff: "Refund processed successfully."	via UC29 (e.g., "Your refund of 2,800,000 VND has been processed"). • Step 9: The success message confirms the action.

Postconditions:

- The refund is processed, and the registration is marked as "refunded."
- The refund transaction is recorded in the system.
- The student is notified of the refund.
- If the registration is not eligible for a refund (e.g., past refund deadline), the system displays an error: "Refund not eligible for this registration."

Summary Inputs	Source	Outputs	Destination
Refund request, Refund confirmation	Staff	Updated payment status Refund transaction Notification trigger Success message	System Staff Student

2.3.2.25 Check Schedule Conflict

Use case name: Check schedule conflict	ID: UC25	Priority: High
Actor: Staff		
Brief Description: The staff verifies if any schedule conflicts exist when assigning instructors or courses.		
Trigger: The staff needs to ensure there are no overlapping schedules when creating or		

updating a course schedule.

Type: ☑ External □ Temporal

Precondition:

- The staff must be logged into the system with an active session.
- Schedules must be available.

Normal Course:	Information for Steps:
1. The staff navigates to the "Scheduling" section on their dashboard. 2. The staff selects an instructor or course to check for conflicts. 3. The system retrieves existing schedules for the selected instructor or course from the schedules table. 4. The system checks for overlapping schedules (e.g., same date and time). 5. The system displays the result: "No conflicts found" or "Conflict detected with [details]."	• Step 1: "Scheduling" is a menu option on the staff dashboard. • Step 2: The staff selects an instructor or course from a dropdown list. • Step 3: The system queries the schedules table for relevant records. • Step 4: The system compares the new schedule with existing ones (e.g., overlapping times). • Step 5: Conflict details include the conflicting session (e.g., "Conflict: 01/07/2025, 19:00-21:00").

Postconditions:

- The staff is informed of any schedule conflicts.
- If no schedules are available, the system displays a message: "No schedules to check."

Summary Inputs	Source	Outputs	Destination
Instructor/Course	Staff	Conflict check result	Systen

selection Schedule details			Staff

2.3.2.26 CRUD Detailed Schedules

Use case name: CRUD Detailed Schedules	ID: UC26	Priority: High
Actor: Stafff		
Brief Description: The staff creates, reads, updates, and deletes detailed teaching schedules for instructors.		
Trigger: The staff needs to manage teaching schedules for instructors. Type: ☑ External □ Temporal		
Precondition: • The staff must be logged into the system with an active session.		
Normal Course: 1. The staff navigates to the "Scheduling" section on their dashboard. 2. The system displays a list of existing schedules with details (e.g., date, time, instructor, course). 3. The staff chooses an action: Create, Update, or Delete. ○ Create: The staff clicks "Add New Schedule," enters schedule details (e.g., date, time, instructor, course), and	Information for Steps: • Step 1: "Scheduling" is a menu option on the staff dashboard. • Step 2: The list includes schedule ID, date, time, instructor, and course. • Step 3: Actions are available as buttons (e.g., "Add New Schedule", "Edit", "Delete"). • Step 4: The system triggers UC27 to check for conflicts. • Step 5: The database is updated based on the action (insert, update, or delete). • Step 6: The success message confirms	

clicks "Save." ○ Update: The staff selects a schedule, edits its details, and clicks "Save." ○ Delete: The staff selects a schedule and clicks "Delete." 4. The system checks for schedule conflicts (using UC27). 5. The system updates the schedules table (create, update, or delete the schedule record). 6. The system notifies the staff: "Schedule updated successfully."	the action.

Postconditions:

- The schedule is created, updated, or deleted in the system.
- If a conflict is detected, the system displays an error: "Schedule conflict detected."
- If the schedule has active sessions (for delete), the system displays an error: "Cannot delete active schedule."

Summary Inputs	Source	Outputs	Destination
Schedule details (date, time, instructor, course)	Staff	Updated schedule record Success message	System Staff

2.3.2.27 Send notification email

Use case name: Send notification	ID: UC27	Priority: Medium
Actor: Staff		

Brief Description: The staff sends email notifications to students about registration approval or changes.	
Trigger: The staff processes a registration or updates a student's status, requiring notification. Type: ☑ External ☐ Temporal	
Precondition: - The staff must be logged into the system with an active session. - The student must have a valid email address (stored in the users table). - The system must be able to send emails (SMTP service configured).	
Normal Course: 1 The staff navigates to the "Notifications" section on their dashboard. 2 The system displays a list of pending notifications (e.g., registration approvals, schedule changes). 3 The staff selects a notification to send. 4 The system generates an email with the notification details (e.g., "Your registration for Digital Marketing has been confirmed"). 5 The system sends the email to the student's email address. 6 The system marks the notification as "sent" in the notifications table. 7 The system notifies the staff: "Email sent successfully."	Information for Steps: - Step 1: "Notifications" is a menu option on the staff dashboard. - Step 2: The list includes notification type, student ID, and details. - Step 3: The staff selects a notification to process. - Step 4: The email content is auto-generated based on the notification type. - Step 5: The email is sent via the system's SMTP service. - Step 6: The notification record is updated in the database. - Step 7: The success message confirms the action.

Postconditions:

- The email notification is sent to the student.
- The notification status is updated in the system.
- If the student’s email is invalid, the system displays an error: "Invalid email address."

Summary Inputs	Source	Outputs	Destination
Notification selection	Staff	Email notification Updated notification status Success message	System Student Staff

2.3.2.28 View Academic Results

Use case name: View academic results	ID: UC28	Priority: Medium

Actor: Student

Brief Description: The student views their academic results, such as grades or evaluations from instructors.

Trigger: The student wants to check their performance in a completed course.

Type: ☑ External ☐ Temporal

Precondition:

- The student must be logged into the system with an active session.
- The student must have completed a course with recorded evaluations (stored in the evaluations table).

<table>
<tr><td colspan="2">Normal Course:
1. The student navigates to the "Academic Results" section on their dashboard.
2. The system retrieves the student’s evaluations from the evaluations table.
3. The system displays the academic results with details (e.g., course name, grade, instructor comments).</td><td colspan="2">Information for Steps:
• Step 1: "Academic Results" is a menu option on the student dashboard.
• Step 2: The system queries the evaluations table using the student’s ID.
• Step 3: The results include details like "Course: Digital Marketing, Grade: 85/100, Comments: Excellent work!"</td></tr>
<tr><td colspan="4">Postconditions:
• The student views their academic results.
• If no results are available, the system displays a message: "No results available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Request to view results</td><td>Student</td><td>Academic results</td><td>System
Student</td></tr>
</table>

2.3.2.29 Browse courses

<table>
<tr><td>Use case name: Browse courses</td><td>ID: UC29</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actor: Student, Guest</td></tr>
<tr><td colspan="3">Brief Description: The student or guest browses the list of available courses without needing</td></tr>
</table>

<table>
<tr><td colspan="4">to log in.</td></tr>
<tr><td colspan="4">Trigger: The student or guest wants to explore courses offered by the center.
Type: ☑ External □ Temporal</td></tr>
<tr><td colspan="4">Precondition:
• None (accessible to both logged-in users and guests).</td></tr>
<tr><td colspan="2">Normal Course:
1. The actor navigates to the "Course Catalog" section on the system’s homepage.
2. The system retrieves the list of available courses from the courses table.
3. The system displays the courses with details (e.g., name, schedule, fees, capacity).</td><td colspan="2">Information for Steps:
• Step 1: "Course Catalog" is a public section accessible via the homepage (/courses).
• Step 2: The system queries the courses table for active courses.
• Step 3: The list includes details like "Course: Advanced Public Speaking, Schedule: 08/07/2025, Fees: 3,500,000 VND."</td></tr>
<tr><td colspan="4">Postconditions:
• The actor views the list of available courses.
• If no courses are available, the system displays a message: "No courses available."</td></tr>
<tr><td>Summary Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Request to browse courses</td><td>Student/Guest</td><td>List of courses</td><td>System
Student/Guest</td></tr>
</table>

2.3.2.30 Register account

Use case name: Register account	ID: UC30	Priority: High
Actor: Guest		
Brief Description: The guest registers for an account to become a student and access the system.		
Trigger: The guest wants to enroll in courses and needs an account to log in. Type: ☑ External □ Temporal		
Precondition: • The guest must not have an existing account in the system.		
Normal Course: 1. The guest navigates to the "Register" page on the system's homepage. 2. The system displays a registration form with fields (e.g., name, email, password). 3. The guest enters their details and clicks "Register." 4. The system validates the input data (e.g., email uniqueness, password strength). 5. The system creates a new user account in the users table with the role "Student." 6. The system notifies the guest: "Account	Information for Steps: • Step 1: "Register" is a link on the homepage (/register). • Step 2: The form includes fields for name, email, password, and confirm password. • Step 3: The guest provides their details. • Step 4: Validation ensures email is unique and password meets requirements (e.g., 8 characters). • Step 5: The new user record includes name, email, hashed password, and role.	

created successfully, please log in."	• Step 6: The success message confirms the action.

Postconditions:

- A new student account is created in the system.
- The guest can log in as a student using their new credentials.
- If the email already exists, the system displays an error: "Email already exists."

Summary Inputs	Source	Outputs	Destination
Registration details (name, email, password)	Guest	New user account Success message	System Guest

2.3.2.31 Export statistics report

Use case name: Export statistics report	ID: UC31	Priority: Medium
Actor: Admin		
Brief Description: The admin exports statistical reports (e.g., revenue, attendance) as PDF or Excel files for further analysis or sharing.		
Trigger: The admin needs to save or share statistical reports for external use, such as presenting to stakeholders or archiving. Type: ☑ External ☐ Temporal		
Precondition: • The student must be logged into the system with an active session.		

• The student must have completed a course with recorded evaluations.	
Normal Course: 1. The admin navigates to the "Reports" section on their dashboard. 2. The admin selects a report type to export (e.g., "Revenue Statistics" or "Attendance Reports"). 3. The system retrieves the relevant data from the appropriate tables (e.g., payments table for revenue, attendance table for attendance). 4. The system displays the report preview with an "Export" option. 5. The admin chooses the file format (PDF or Excel) and clicks "Export." 6. The system generates the report file in the selected format. 7. The system provides a download link for the generated file. 8. The admin downloads the file. 9. The system notifies the admin: "Report exported successfully."	Information for Steps: • Step 1: "Reports" is a menu option on the admin dashboard. • Step 2: Report types include "Revenue Statistics" (from UC24) and "Attendance Reports" (from UC23). • Step 3: The system queries the relevant tables (e.g., payments for revenue, attendance for attendance data). • Step 4: The preview shows the report content (e.g., "Total Revenue: 50,000,000 VND"). • Step 5: The "Export" option includes a dropdown for file format (PDF or Excel). • Step 6: The system uses a library (e.g., iTextPDF for PDF, Apache POI for Excel) to generate the file. • Step 7: The download link is a temporary URL. • Step 8: The admin clicks the link to download the file to their device. • Step 9: The success message confirms the action.
Postconditions: • The statistical report is exported as a PDF or Excel file.	

- The admin can access the file for further use (e.g., sharing or archiving).
- If no data is available for the selected report type, the system displays an error: "No data available to export."

Summary Inputs	Source	Outputs	Destination
Report type selection, File format	Admin	Exported report file Download link Success message	System Admin

2.3.2.32 Handle Salaries

Use case name: Handle Salaries	ID: UC32	Priority: High
Actor: Staff		
Brief Description: The Staff manages instructor salaries, including calculating payroll and reviewing/finalizing payments.		
Trigger: The Staff needs to process instructor salaries, typically at the end of a pay period. Type: ☑ External □ Temporal		
Precondition: • The Staff must be logged into the system with appropriate permissions to manage salaries. • Instructor data, including teaching hours and compensation rules, must be available in the system.		
Normal Course: 1. The Staff navigates to the salary management section of the system.	Information for Steps: • Step 1: The salary management section is accessible via the Staff dashboard.	

2. The system displays options to calculate payroll and review/finalize payments. 3. The Staff selects the option to calculate payroll. 4. The system prompts the Staff to select the pay period and instructors. 5. The Staff selects the relevant pay period and instructors 6. The system calculates the payroll based on teaching hours and compensation rules. 7. The system displays the calculated payroll for review. 8. The Staff reviews the payroll and makes any necessary adjustments. 9. The Staff finalizes the payroll and authorizes payments. 10. The system records the finalized payroll and initiates payment processes.	• Step 2: Options are presented as buttons or menu items. • Step 3: Selecting "Calculate Payroll" initiates the payroll calculation process. • Step 4: The system provides a date range selector for the pay period and a list of instructors. • The Staff can select multiple instructors or all instructors for the pay period. • Step 6: Calculation is based on predefined rules stored in the system. • Step 7: The payroll is displayed in a tabular format with details for each instructor. • Step 8: Adjustments can be made manually if needed. • Step 9: Finalizing the payroll confirms the payments. • Step 10: The system updates the database and may integrate with payment gateways.
Postconditions: • - The payroll is calculated, reviewed, and finalized. • Payments are authorized and processed for instructors. • If there are errors in calculation or insufficient data, the system displays an error message and prompts the Staff to correct the issues.	

Summary Inputs	Source	Outputs	Destination
Pay period dates Instructor selection	Staff	Calculated payroll Finalized payroll records Payment authorizations	System Instructor

2.3.2.33 Calculate Payroll

Use case name: Calculate Payroll	ID: UC33	Priority: High
Actor: Staff		
Brief Description: The Staff calculates payroll for instructors based on teaching hours or other compensation rules.		
Trigger: The Staff initiates the payroll calculation process for a specific pay period. Type: ☑ External □ Temporal		
Precondition: • The Staff must be logged into the system with permissions to calculate payroll. • Instructor teaching hours and compensation rules must be recorded in the system.		
Normal Course: 1. The Staff selects the "Calculate Payroll" option from the salary management section. 2. The system prompts the Staff to enter the pay period dates.	Information for Steps: • Step 1: The "Calculate Payroll" option is available in the salary management menu. • Step 2: The system provides date input fields for the pay period. • Step 3: Dates must be in a valid format	

3. The Staff enters the start and end dates of the pay period. 4. The system retrieves the list of instructors who taught during the pay period. 5. The Staff selects the instructors for whom to calculate payroll. 6. The system calculates the payroll for each selected instructor based on their teaching hours and applicable compensation rules. 7. The system displays the calculated payroll details for each instructor. 8. The Staff reviews the calculated payroll. 9. The Staff confirms the calculation or makes adjustments if necessary. 10. The system saves the calculated payroll for further processing.	and within the system's date range. • Step 4: The system queries the database for instructors with teaching records in the specified period. • Step 5: The Staff can select all or specific instructors. • Step 6: Calculation formulas are predefined and may include hourly rates, bonuses, etc. • Step 7: Details include instructor name, hours taught, rate, and total pay. • Step 8: Review involves checking for accuracy and completeness. • Step 9: Adjustments can be made to hours or rates if errors are found. • Step 10: Saved payroll data is used for review and finalization.

Postconditions:

- The payroll is calculated and saved for the selected instructors and pay period.
- If teaching hours or compensation rules are missing, the system alerts the Staff to update the necessary data.

Summary Inputs	Source	Outputs	Destination
Pay period dates Instructor selection	Staff	Calculated payroll details	System Staff

2.3.2.34 Review and Finalize Payroll

Use case name: Review and Finalize Payroll	ID: UC34	Priority: High
Actor: Staff		
Brief Description: The Staff reviews the calculated payroll and authorizes payments to instructors.		
Trigger: The Staff needs to finalize the payroll after calculation. Type: ☑ External □ Temporal		
Precondition: • The payroll must have been calculated for the relevant pay period. • The Staff must have permissions to review and finalize payroll.		
Normal Course: 1. The Staff selects the "Review and Finalize Payroll" option. 2. The system displays the list of calculated payrolls pending review. 3. The Staff selects a specific payroll to review. 4. The system shows the detailed payroll information for the selected payroll. 5. The Staff reviews the payroll details, ensuring accuracy. 6. If adjustments are needed, the Staff makes the necessary changes. 7. The Staff finalizes the payroll by authorizing the payments.	Information for Steps: • Step 1: The option is available in the salary management section. • Step 2: Pending payrolls are listed with pay period and status. • Step 3: Selection can be by pay period or instructor. • Step 4: Details include instructor names, hours, rates, totals, and any adjustments. • Step 5: Review involves cross-verifying with teaching records and compensation rules. • Step 6: Adjustments might include correcting hours or applying	

8. The system updates the payroll status to "Finalized" and initiates the payment process. 9. The system notifies the Staff of successful finalization.	deductions. • Step 7: Authorization is done by clicking a "Finalize" button. • Step 8: The system may integrate with financial systems for payment processing. • Step 9: A confirmation message is displayed.
Postconditions: • The payroll is finalized and payments are authorized. • If there are discrepancies or missing data, the system prevents finalization and prompts the Staff to resolve issues.	

Summary Inputs	Source	Outputs	Destination
Payroll selection Adjustments (if any)	Staff	Finalized payroll Payment authorizations Confirmation message	System Instructor System

2.3.2.35 View Payslip

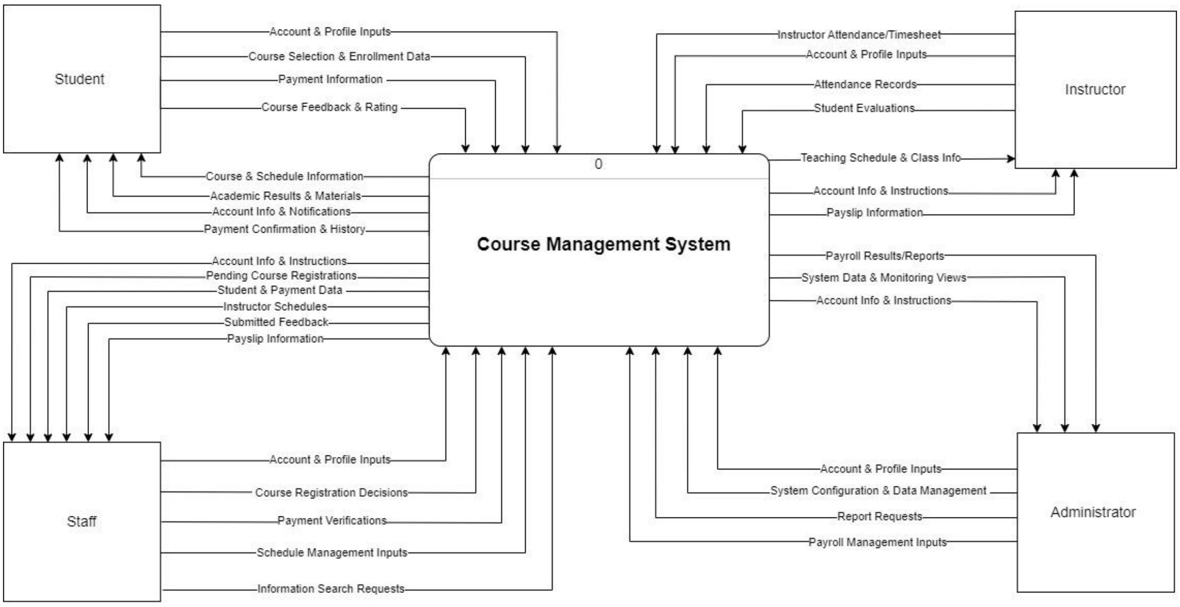
Use case name: View Payslip	ID: UC35	Priority: Medium
Actor: Instructor		
Brief Description: The Instructor views their payslips, including earnings, deductions, and payment history.		

Trigger: The Instructor wants to check their payment details for a specific pay period. Type: ☑ External □ Temporal	
Precondition: • The Instructor must be logged into the system. • Payslips must have been generated and stored in the system for the Instructor.	
Normal Course: 1. The Instructor navigates to the payslip section of their dashboard. 2. The system displays a list of available payslips, organized by pay period. 3. The Instructor selects a specific payslip to view. 4. The system retrieves and displays the detailed payslip, including earnings, deductions, and net pay. 5. The Instructor reviews the payslip details. 6. Optionally, the Instructor can download or print the payslip. 7. The system provides options to view previous payslips or return to the dashboard.	Information for Steps: • Step 1: The payslip section is accessible via the Instructor's dashboard menu. • Step 2: Payslips are listed in chronological order with pay period dates. • Step 3: Selection is done by clicking on the desired pay period. • Step 4: The payslip includes sections for earnings (e.g., base pay, bonuses), deductions (e.g., taxes), and net pay. • Step 5: The Instructor can verify the accuracy of the payslip. • Step 6: Download and print options are available as buttons or links. • Step 7: Navigation options allow easy access to other payslips or dashboard functions.
Postconditions: • The Instructor successfully views their payslip.	

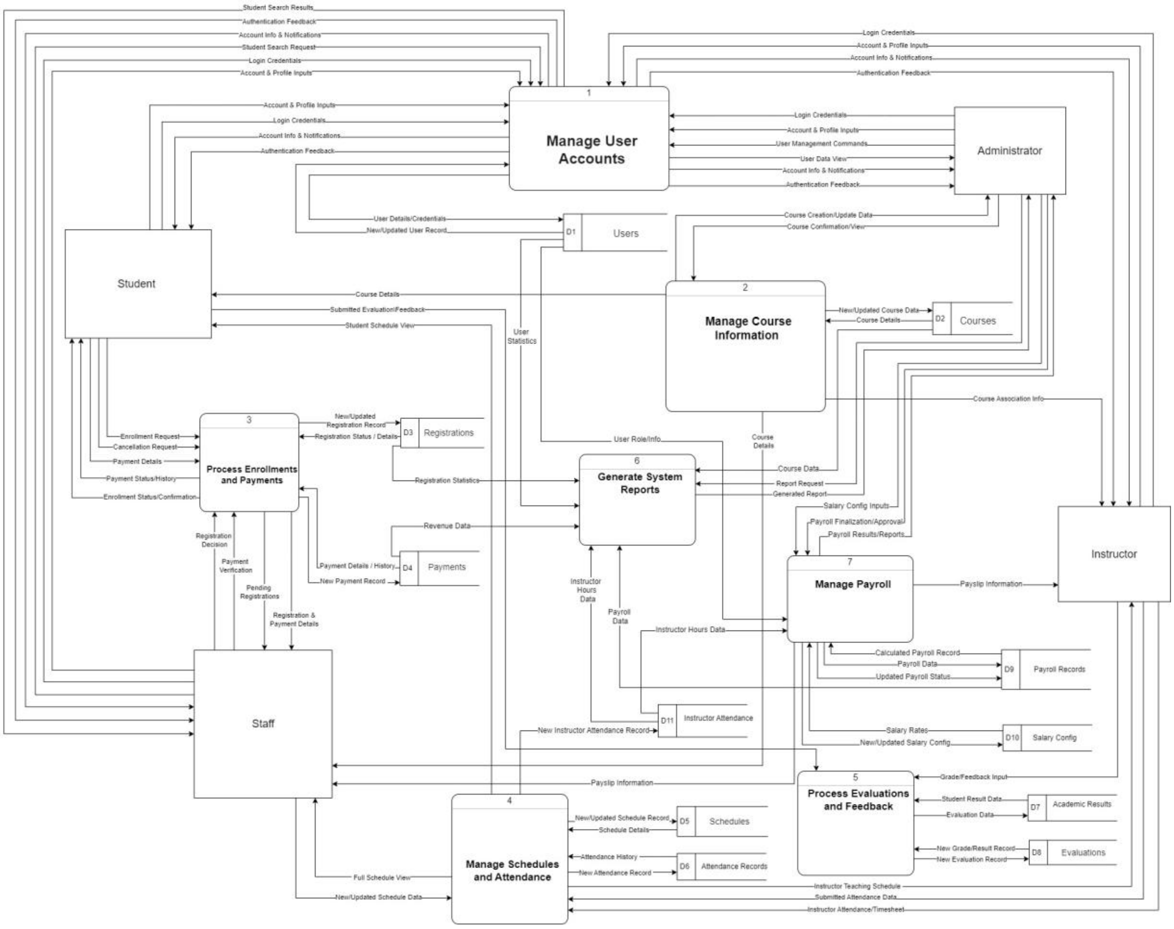
<table>
<tr><td colspan="4">• If no payslips are available for the selected period, the system displays a message indicating no data found.</td></tr>
<tr><th>Summary Inputs</th><th>Source</th><th>Outputs</th><th>Destination</th></tr>
<tr><td>Payslip selection</td><td>Instructor</td><td>Detailed payslip
Download/print options</td><td>Instructor</td></tr>
</table>

2.3.3. Data flow Diagram

2.3.3.1 DFD Context

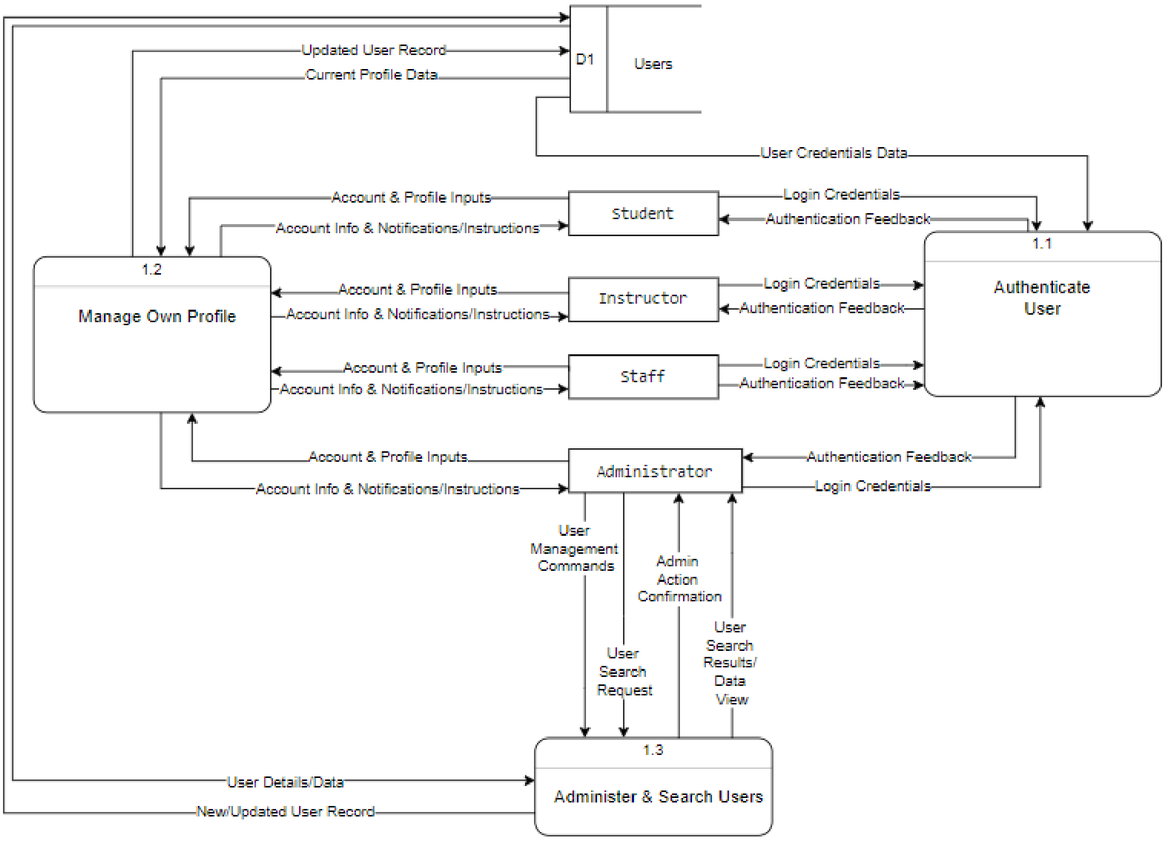

Student
Account & Profile Inputs
Course Selection & Enrollment Data
Payment Information
Course Feedback & Rating
Course & Schedule Information
Academic Results & Materials
Account Info & Notifications
Payment Confirmation & History
Instructor Attendance/Timesheet
Account & Profile Inputs
Attendance Records
Student Evaluations
Instructor
Teaching Schedule & Class Info
Account Info & Instructions
Payslip Information
0
Course Management System
Account Info & Instructions
Pending Course Registrations
Student & Payment Data
Instructor Schedules
Submitted Feedback
Payslip Information
Payroll Results/Reports
System Data & Monitoring Views
Account Info & Instructions
Staff
Account & Profile Inputs
Course Registration Decisions
Payment Verifications
Schedule Management Inputs
Information Search Requests
Administrator
Account & Profile Inputs
System Configuration & Data Management
Report Requests
Payroll Management Inputs

2.3.3.2 DFD Level 0

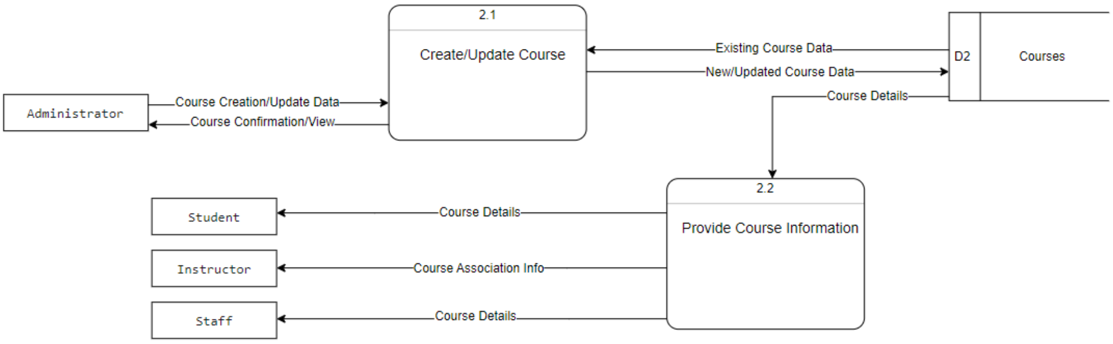


2.3.3.3 DFD Level 1

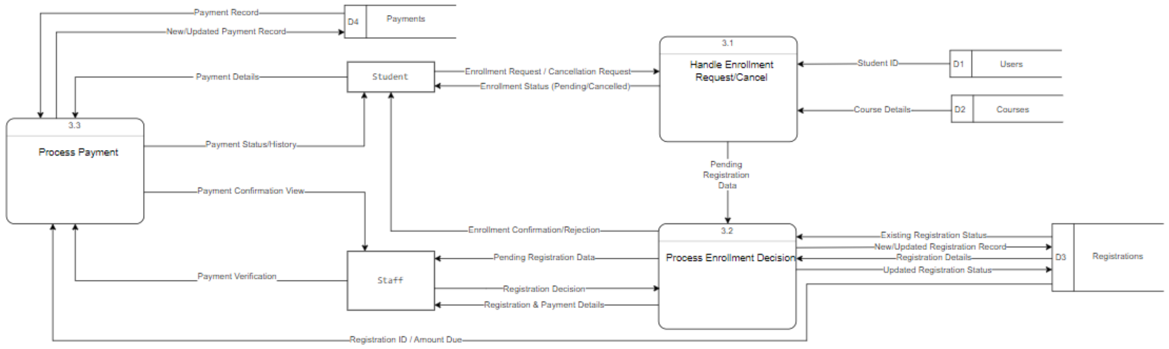
¤ *Manage User Account*



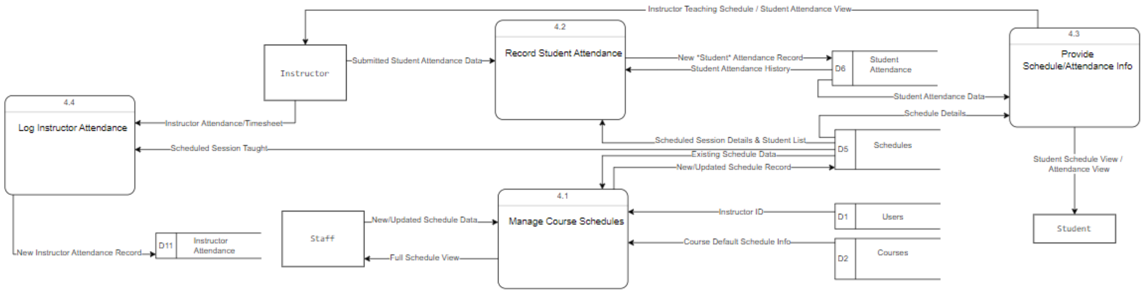
¤ *Manage Course Information*



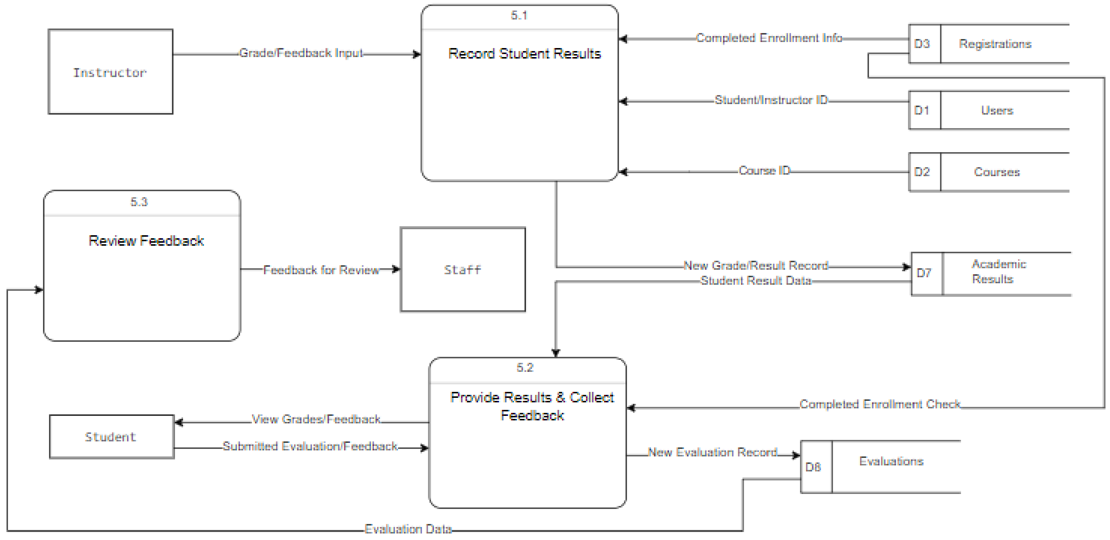
¤ *Process Enrollments and Payments*


Payment Record
D4
Payments
New/Updated Payment Record
Payment Details
Student
Enrollment Request / Cancellation Request
Enrollment Status (Pending/Cancelled)
3.1
Handle Enrollment Request/Cancel
Student ID
D1
Users
Course Details
D2
Courses
3.3
Process Payment
Payment Status/History
Pending Registration Data
Payment Confirmation View
Enrollment Confirmation/Rejection
3.2
Process Enrollment Decision
Existing Registration Status
New/Updated Registration Record
Registration Details
Updated Registration Status
D3
Registrations
Pending Registration Data
Payment Verification
Staff
Registration Decision
Registration & Payment Details
Registration ID / Amount Due

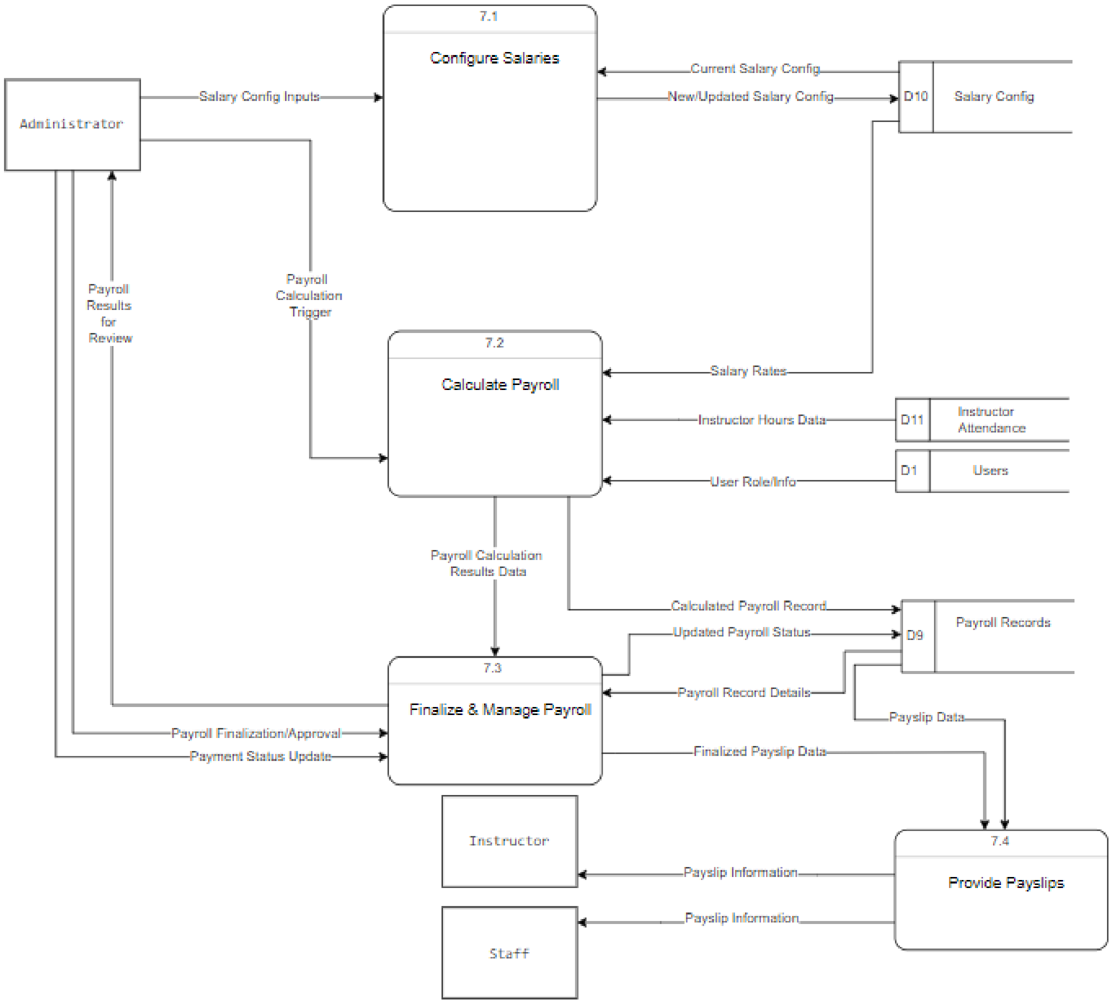
¤ *Manage Schedules and Attendance*


Instructor Teaching Schedule / Student Attendance View
4.2
Record Student Attendance
Instructor
Submitted Student Attendance Data
New *Student* Attendance Record
Student Attendance History
D6
Student Attendance
4.3
Provide Schedule/Attendance Info
Student Attendance Data
Schedule Details
4.4
Log Instructor Attendance
Instructor Attendance/Timesheet
Scheduled Session Taught
Scheduled Session Details & Student List
D5
Schedules
Existing Schedule Data
New/Updated Schedule Record
Student Schedule View / Attendance View
4.1
Manage Course Schedules
Instructor ID
D1
Users
New/Updated Schedule Data
Staff
Student
Full Schedule View
Course Default Schedule Info
D2
Courses
New Instructor Attendance Record
D11
Instructor Attendance

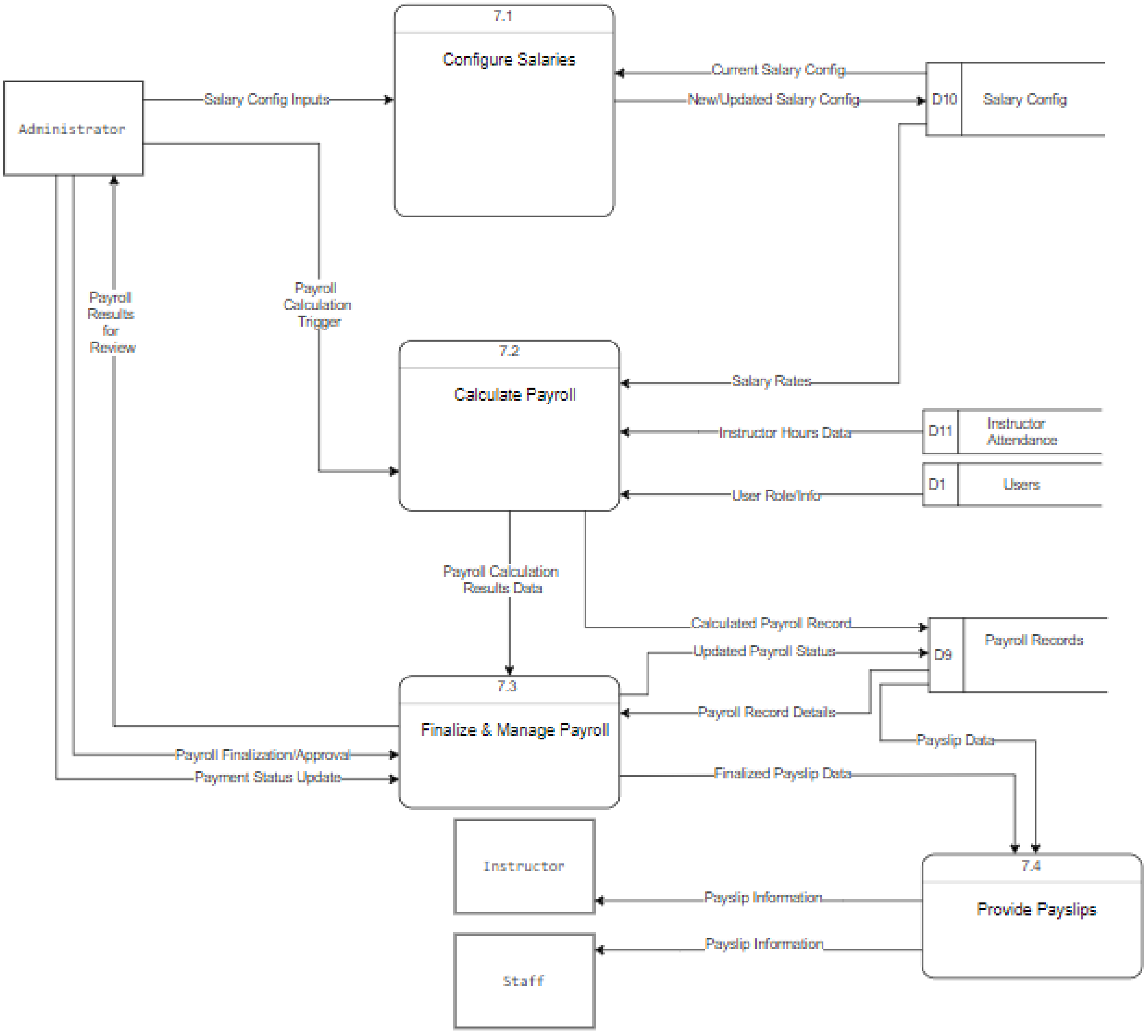
¤ *Process Evaluations and Feedback*



¤ *Generate System Reports*

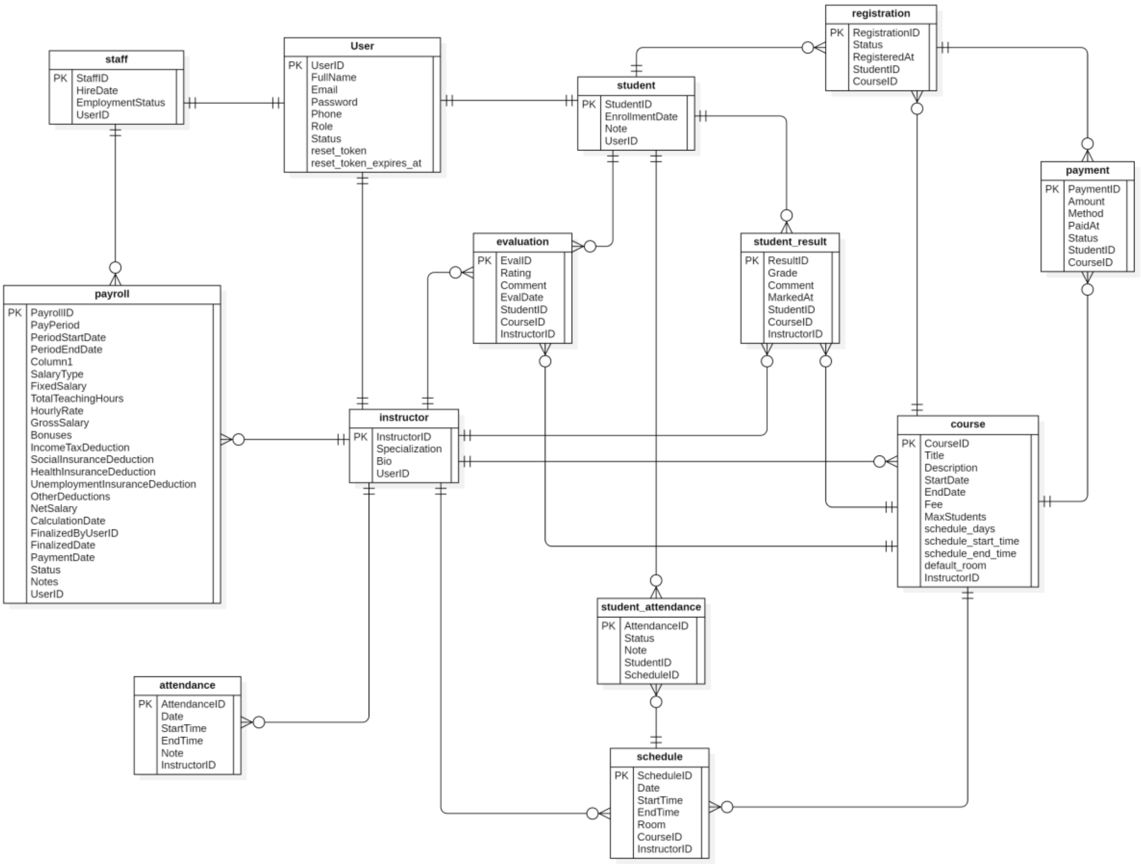


¤ *Manage Payroll*

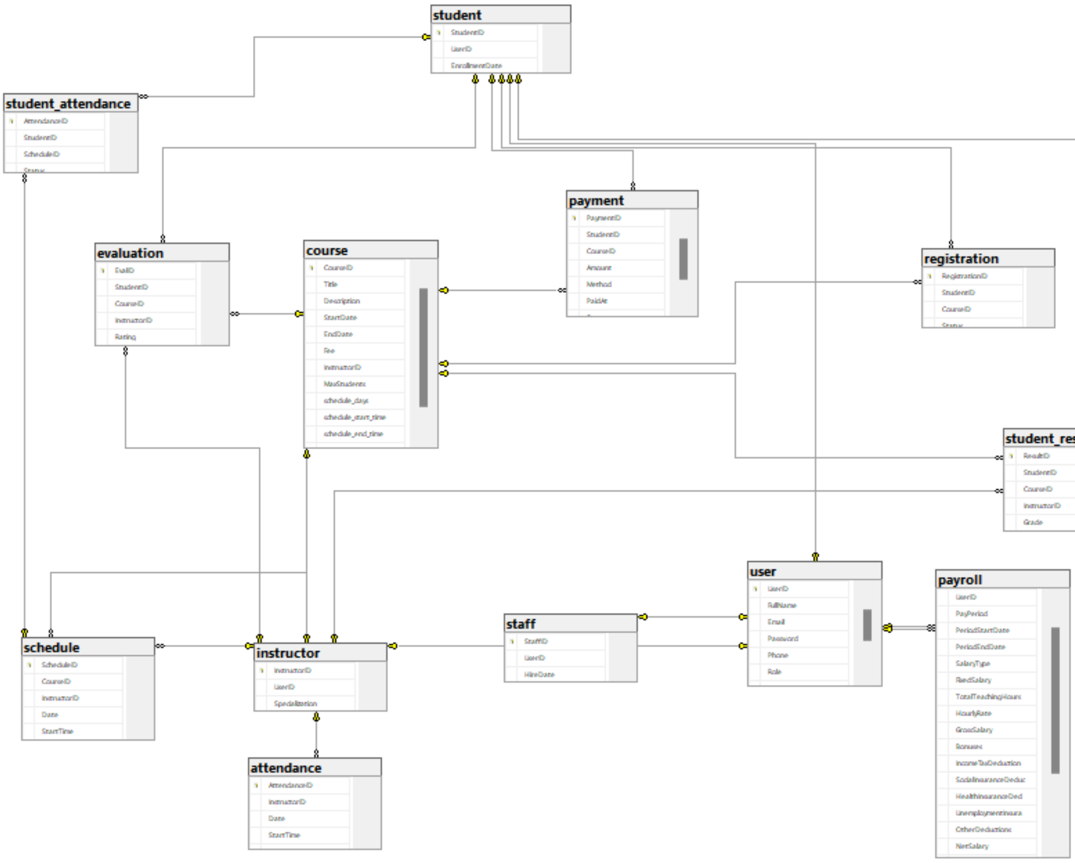


SECTION 3 – DATABASE DESIGN

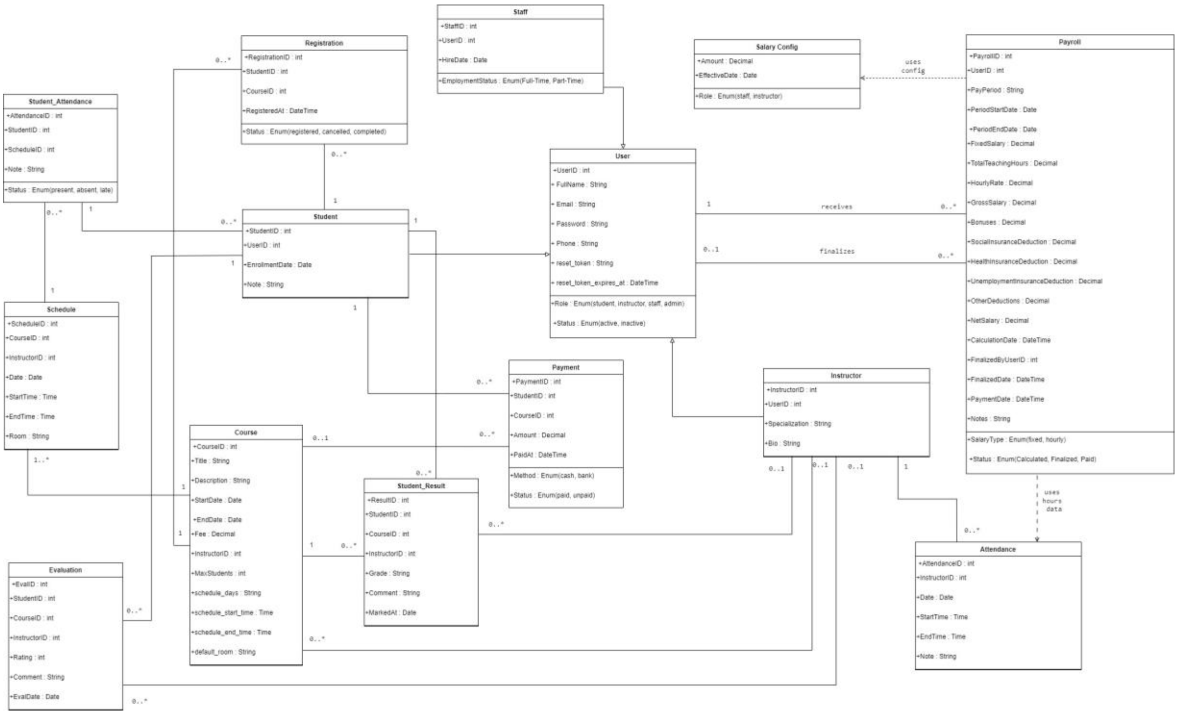
3.1. ERD model



3.2. Database schema



3.3. Class Diagram



SECTION 4 – SOFTWARE IMPLEMENTATION

4.1. Software development process

During the development of the course management system, the development team chose the Waterfall model as the software development process. This is one of the traditional models that follows a sequential flow including the stages: requirement analysis, design, implementation, testing, and maintenance.

The Waterfall model is suitable for this system because user requirements were clearly defined from the beginning, with no major changes during development. The separation of development phases helps in managing progress and controlling quality more easily.

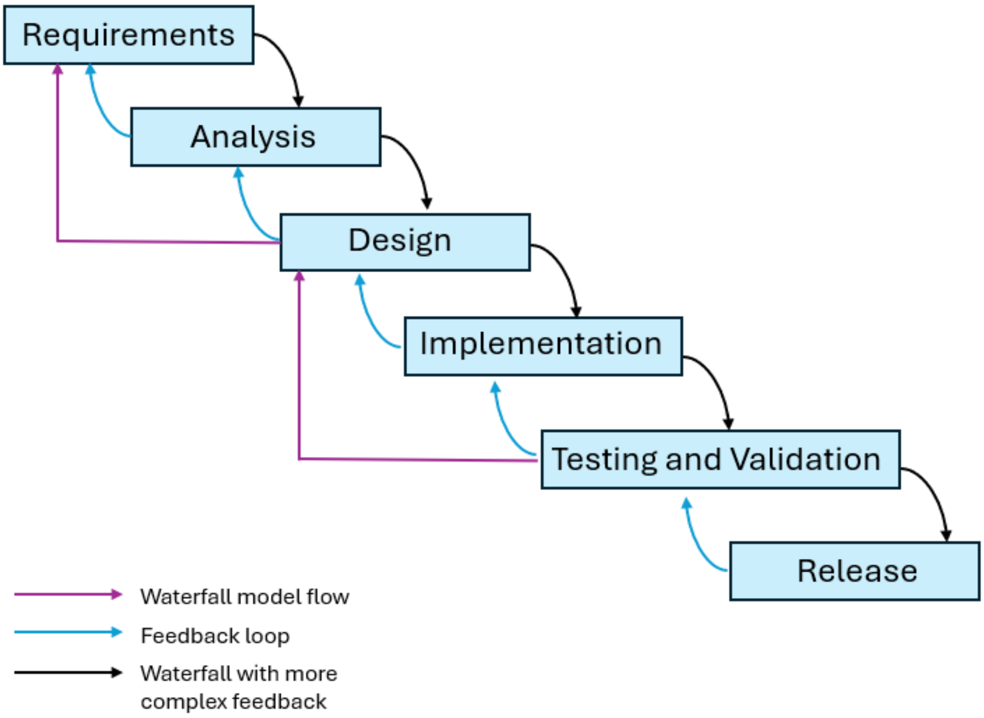


Figure 1: Waterfall Model

4.2. Development plan

The team divided the plan into the following specific stages:

Stage	Timeline	Description

Requirement Analysis	Week 1	Collect and analyze requirements from stakeholders, identify key Use Cases.
Database and Interface Design	Week 2	Design database schema, define tables, primary and foreign keys, and sketch the user interface.
System Implementation	Week 3-4	Implement frontend and backend functionalities for each user module.
Testing and Deployment	Week 5	Conduct functionality testing, fix bugs, refine UI, and deploy the system.

Due to limited team size (3 members) , the use of simple technologies like PHP and MySQL without a complex framework such as MVC was a suitable choice to optimize time and effort.

4.3. Technologies Used

The system was developed using the following main technologies:

Component	Technology Used
Backend Language	PHP
Database	MySQL
User Interface	HTML, CSS, Javascript
System Architecture	Simple Client – Server
Testing Method	Manual Testing
Server Platform	XAMPP (Apache + MySQL)

4.4. System Modules by User Roles

The system is organized into modules corresponding to four user roles:

a. Student Module

- Register and log in to an account.
- View enrolled courses.
- Register/unregister for courses.
- Submit assignments, view grades.
- Update personal information.

b. Instructor Module

- Log in and manage classes being taught.
- Upload lectures and study materials.
- Create assignments, grade submissions.
- Send announcements to students.

c. Staff Module

- Process course registration requests.
- Approve new students.
- Update course status.
- Generate statistics on students and instructors.

d. Admin Module

- Manage the entire system.
- Assign user permissions.
- Add/edit/delete user accounts.
- Backup and restore data.

Each module corresponds to a separate set of features, making it easier to develop and test in a modular fashion.

4.5. System Testing

The system was tested at multiple levels:

Test Type	Description
Unit Testing	Test individual functions like login or

	adding a course.
Integration Testing	Test continuous workflows such as register → view course info
System Testing	Test the whole system including UI and access permissions.
Acceptance Testing	Simulate real user behavior to evaluate operational readiness

4.6. System Deployment Architecture

The system is deployed following a simple Client – Server model:

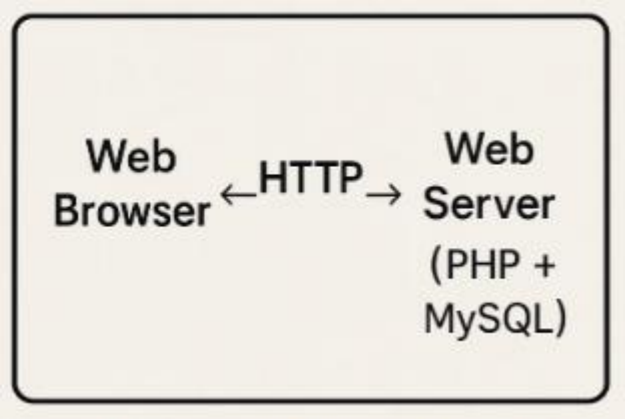


Figure 2: Client – Server Model

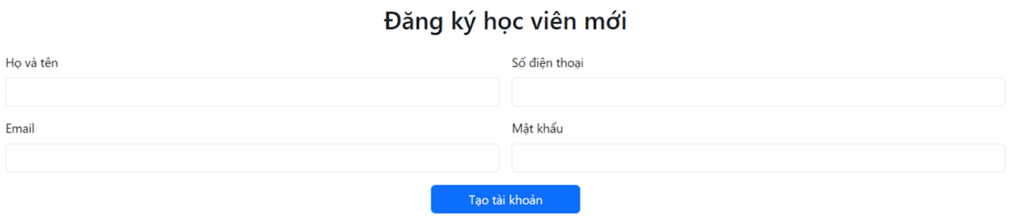
All interfaces are rendered in the browser using HTML/CSS/JS. Client-side requests are handled by PHP scripts on the server, which interact with MySQL to retrieve or store data.

4.7. Security and Access Control

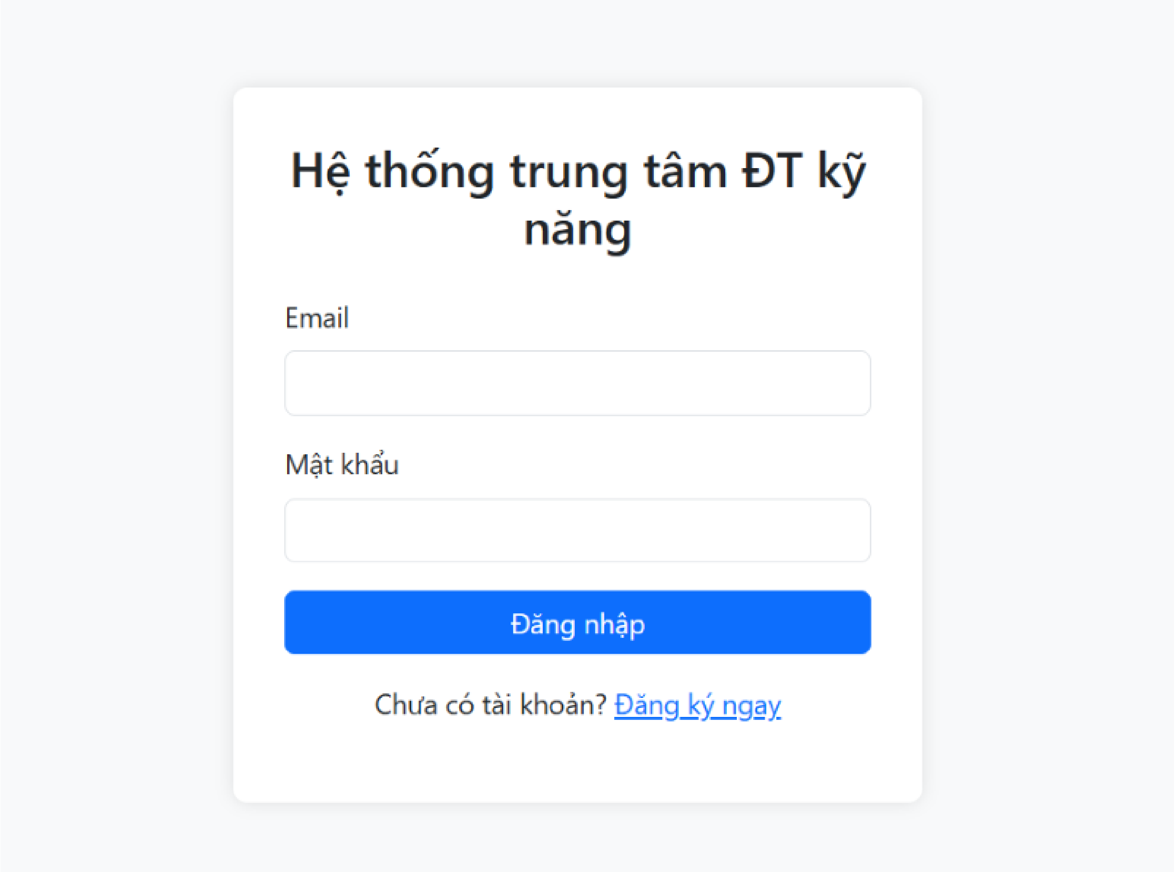
- The system uses sessions to authenticate users after login.
- Each functional page checks the user's role before granting access.
- Each account is assigned a single role (student/instructor/staff/admin), defined by the role field in the users table.
- All input fields are validated to prevent SQL Injection or XSS attacks.
- User passwords are encrypted using the md5() algorithm before being stored in the database.

SECTION 5 – PRODUCT RESULTS

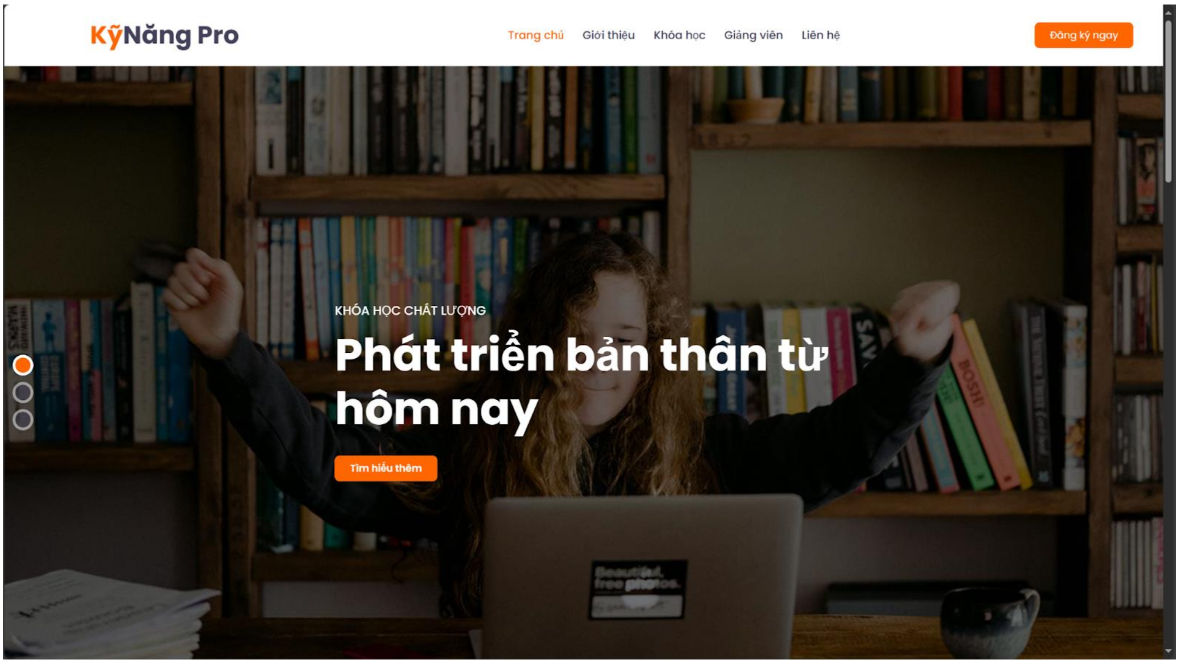
- **Register**



- **Login Screen**


Hệ thống trung tâm ĐT kỹ năng
Email
Mật khẩu
Đăng nhập
Chưa có tài khoản? Đăng ký ngay

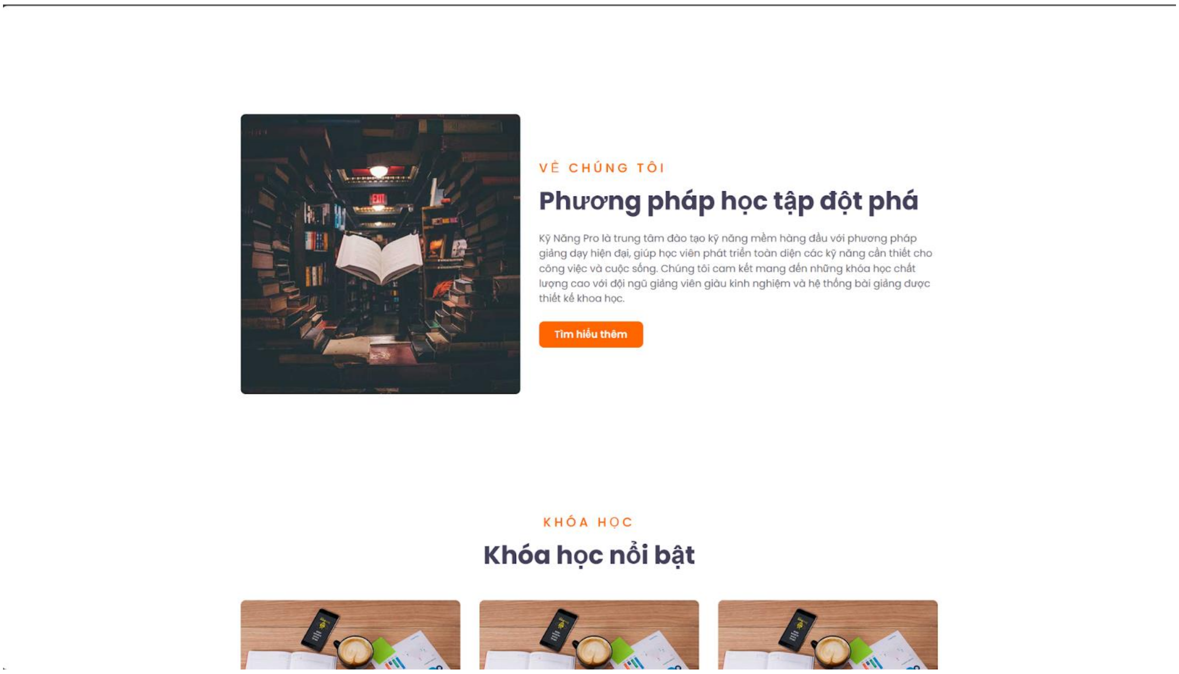
- **Home Screen 1**



KHÓA HỌC CHẤT LƯỢNG

Phát triển bản thân từ hôm nay

- **Home Screen 2**



VỀ CHÚNG TÔI

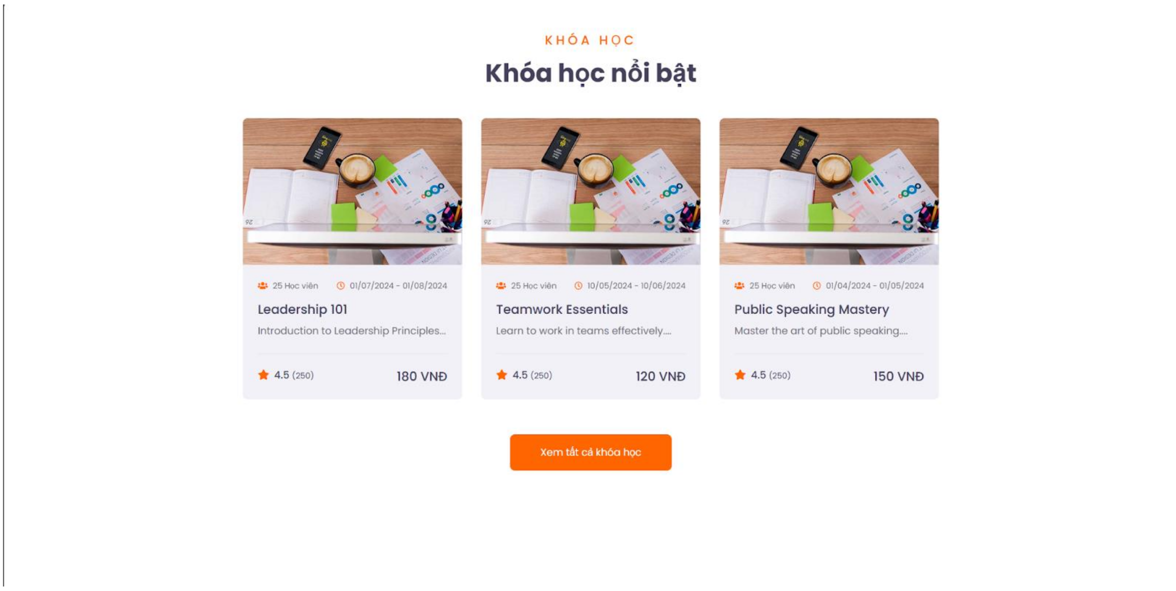
Phương pháp học tập đột phá

Kỹ Năng Pro là trung tâm đào tạo kỹ năng mềm hàng đầu với phương pháp giảng dạy hiện đại, giúp học viên phát triển toàn diện các kỹ năng cần thiết cho công việc và cuộc sống. Chúng tôi cam kết mang đến những khóa học chất lượng cao với đội ngũ giảng viên giàu kinh nghiệm và hệ thống bài giảng được thiết kế khoa học.

KHÓA HỌC

Khóa học nổi bật

- **Home Screen 3**



KHÓA HỌC

Khóa học nổi bật

25 Học viên 01/07/2024 - 01/08/2024

Leadership 101

Introduction to Leadership Principles...

★ 4.5 (250) 180 VNĐ

25 Học viên 10/05/2024 - 10/06/2024

Teamwork Essentials

Learn to work in teams effectively....

★ 4.5 (250) 120 VNĐ

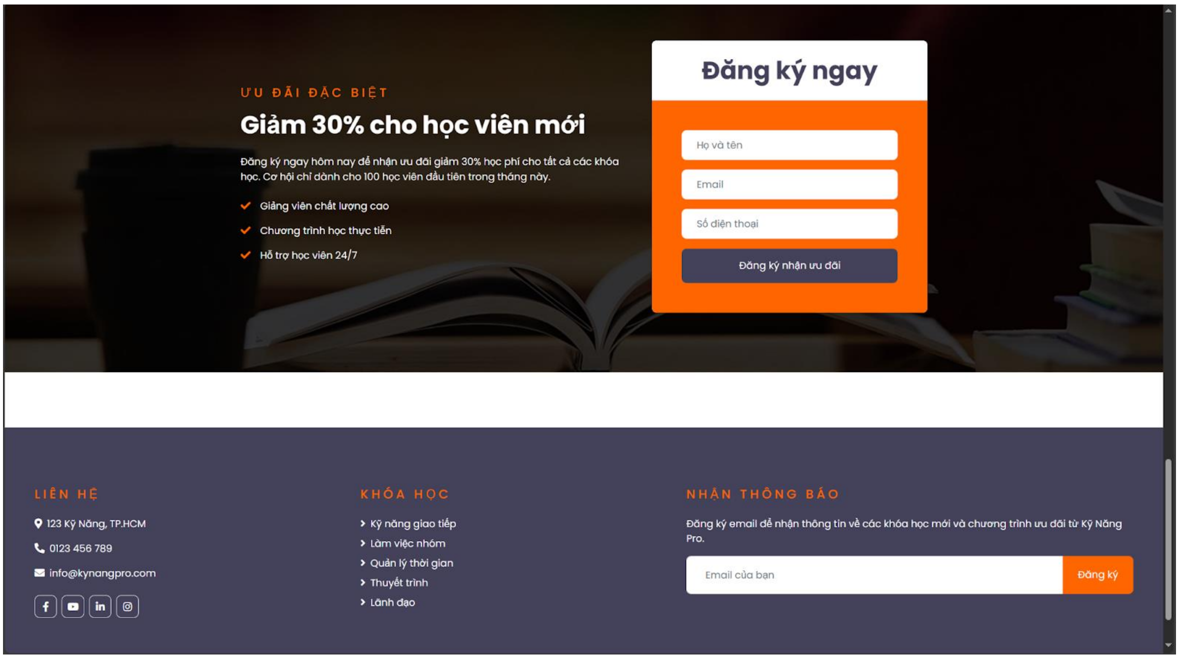
25 Học viên 01/04/2024 - 01/05/2024

Public Speaking Mastery

Master the art of public speaking....

★ 4.5 (250) 150 VNĐ

- **Home Screen 4**



ƯU ĐÃI ĐẶC BIỆT

Giảm 30% cho học viên mới

Đăng ký ngay hôm nay để nhận ưu đãi giảm 30% học phí cho tất cả các khóa học. Cơ hội chỉ dành cho 100 học viên đầu tiên trong tháng này.

✓ Giảng viên chất lượng cao

✓ Chương trình học thực tiễn

✓ Hỗ trợ học viên 24/7

Đăng ký ngay

LIÊN HỆ

123 Kỹ Năng, TP.HCM

0123 456 789

info@kynangpro.com

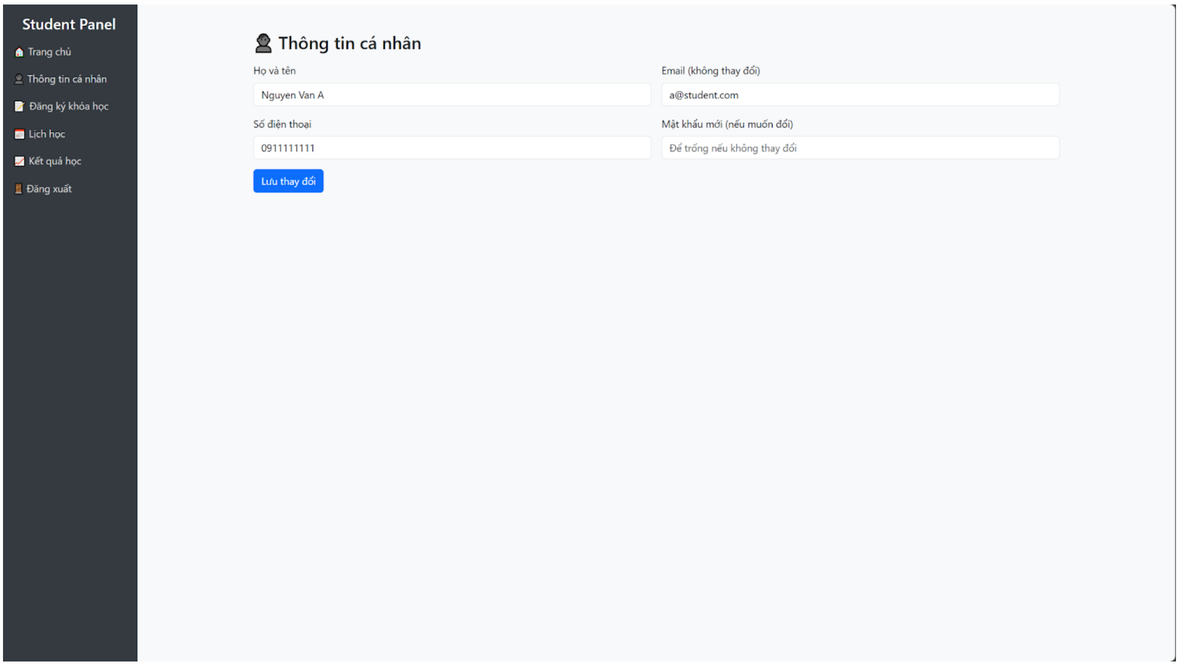
KHÓA HỌC

- Kỹ năng giao tiếp
- Làm việc nhóm
- Quản lý thời gian
- Thuyết trình
- Lãnh đạo

NHẬN THÔNG BÁO

Đăng ký email để nhận thông tin về các khóa học mới và chương trình ưu đãi từ Kỹ Năng Pro.

- **Student Dashboard - Profile**



Thông tin cá nhân

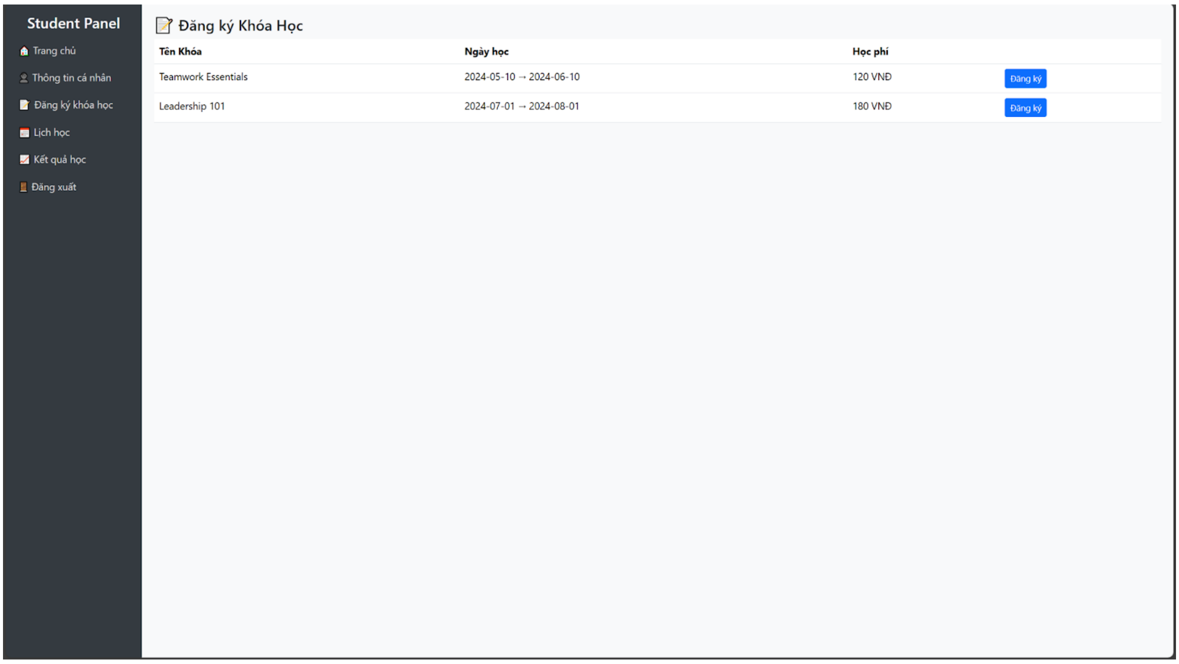
Họ và tên: Nguyen Van A

Email (không thay đổi): a@student.com

Số điện thoại: 0911111111

Mật khẩu mới (nếu muốn đổi): Để trống nếu không thay đổi

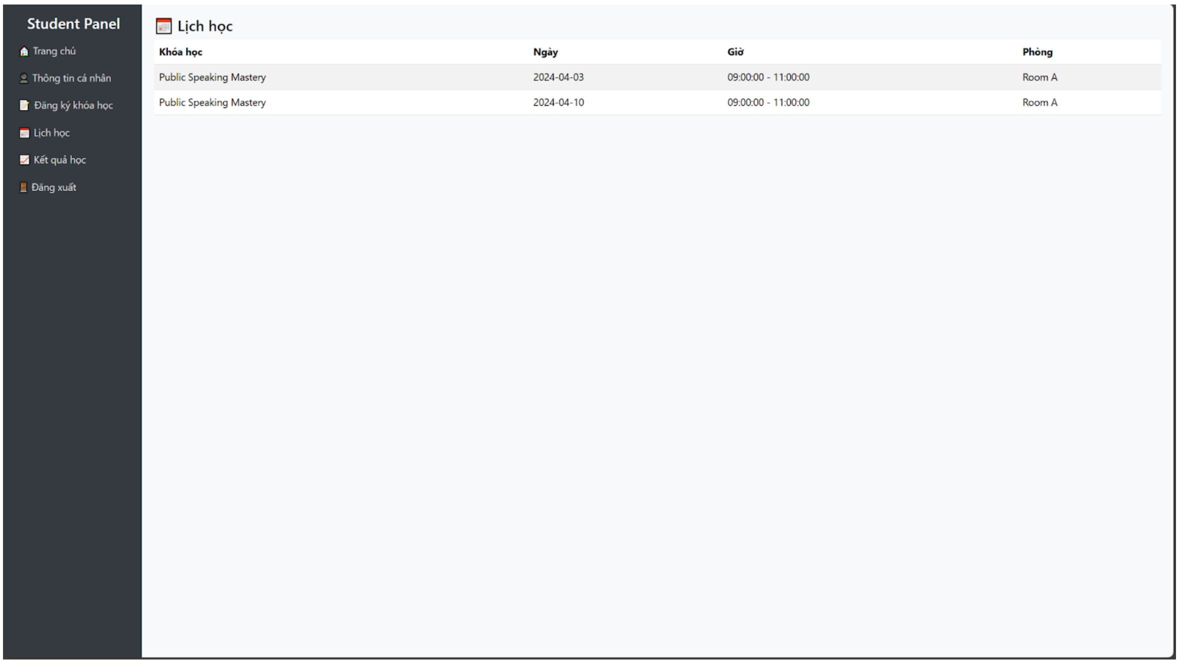
- **Student Dashboard - Course registration**



Đăng ký Khóa Học

Tên Khóa	Ngày học	Học phí	
Teamwork Essentials	2024-05-10 → 2024-06-10	120 VNĐ	Đăng ký
Leadership 101	2024-07-01 → 2024-08-01	180 VNĐ	Đăng ký

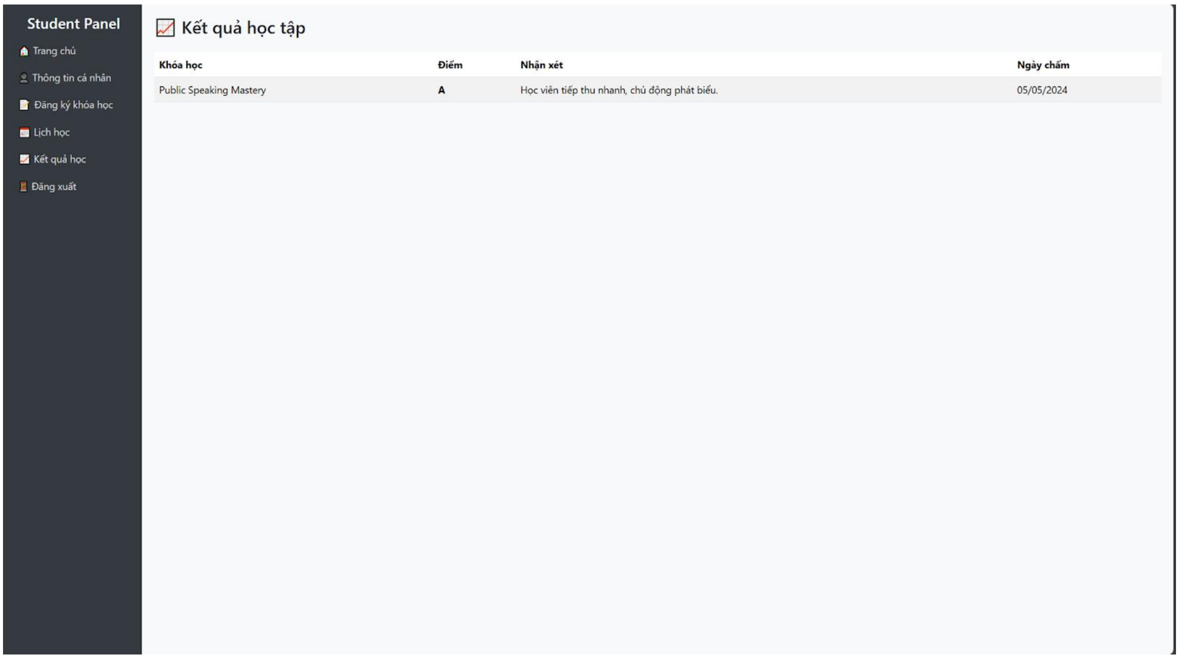
- **Student Dashboard - Schedule**



Lịch học

Khóa học	Ngày	Giờ	Phòng
Public Speaking Mastery	2024-04-03	09:00:00 - 11:00:00	Room A
Public Speaking Mastery	2024-04-10	09:00:00 - 11:00:00	Room A

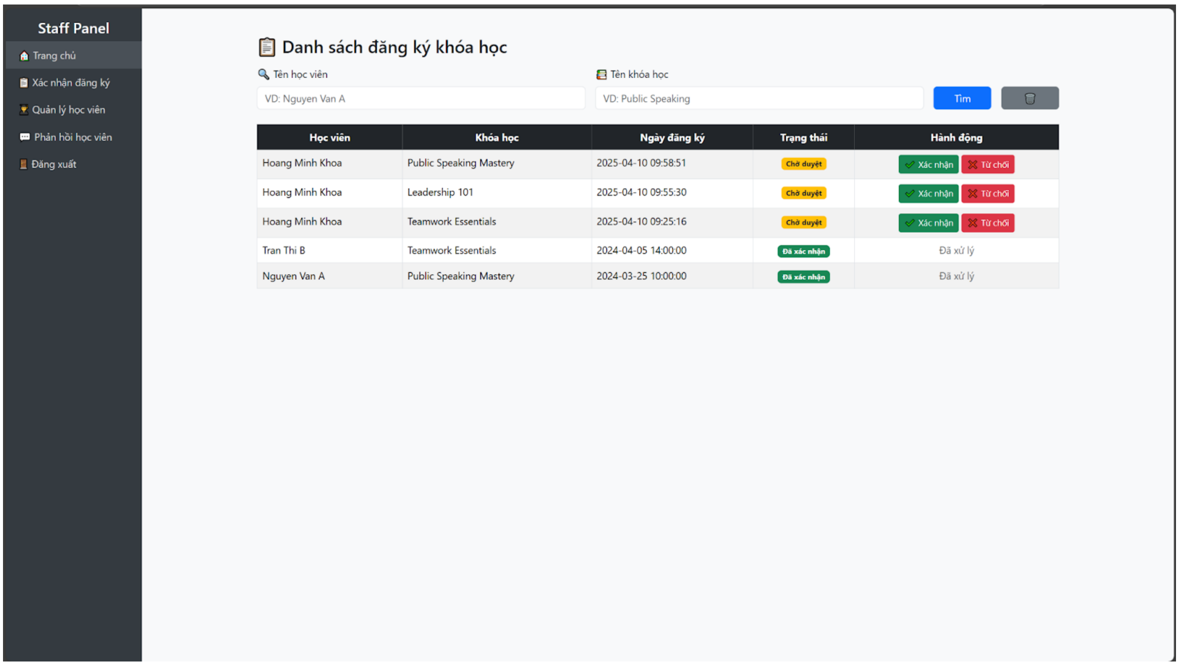
- **Student Dashboard - Academic Results**



Kết quả học tập

Khóa học	Điểm	Nhận xét	Ngày chấm
Public Speaking Mastery	**A**	Học viên tiếp thu nhanh, chủ động phát biểu.	05/05/2024

- **Staff Dashboard - Registration Confirmation Screen**



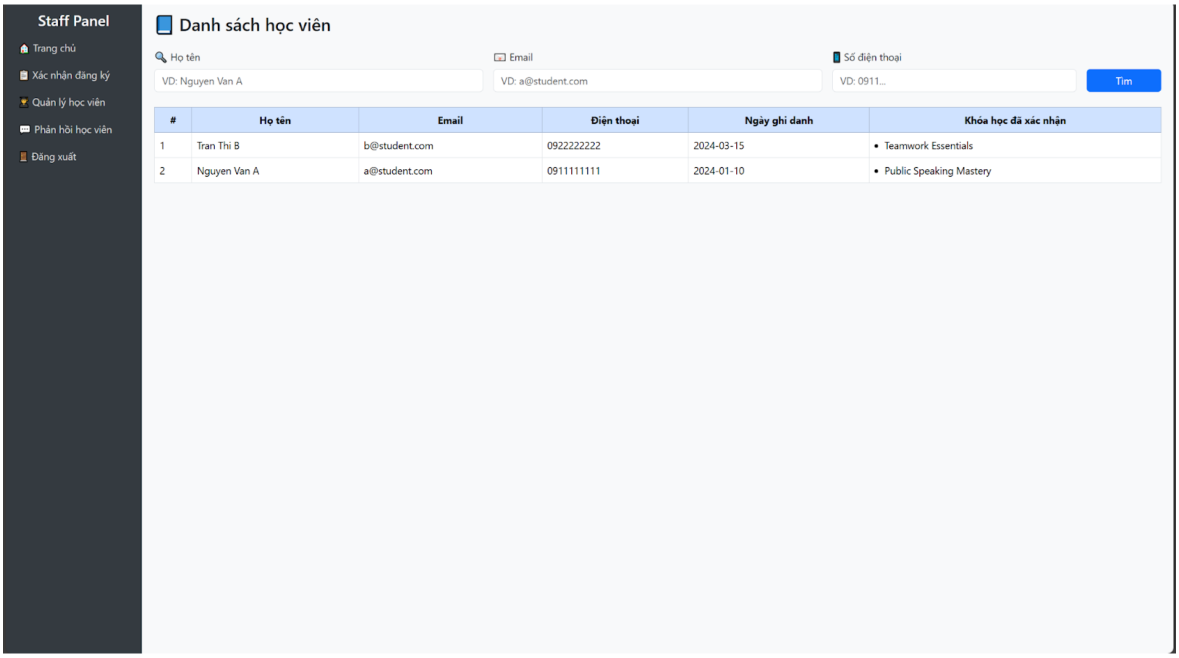
Danh sách đăng ký khóa học

Tên học viên: VD: Nguyen Van A

Tên khóa học: VD: Public Speaking

Học viên	Khóa học	Ngày đăng ký	Trạng thái	Hành động
Hoang Minh Khoa	Public Speaking Mastery	2025-04-10 09:58:51	Chờ duyệt	✓ Xác nhận ✗ Từ chối
Hoang Minh Khoa	Leadership 101	2025-04-10 09:55:30	Chờ duyệt	✓ Xác nhận ✗ Từ chối
Hoang Minh Khoa	Teamwork Essentials	2025-04-10 09:25:16	Chờ duyệt	✓ Xác nhận ✗ Từ chối
Tran Thi B	Teamwork Essentials	2024-04-05 14:00:00	Đã xác nhận	Đã xử lý
Nguyen Van A	Public Speaking Mastery	2024-03-25 10:00:00	Đã xác nhận	Đã xử lý

- **Staff Dashboard - Student List Screen**



Danh sách học viên

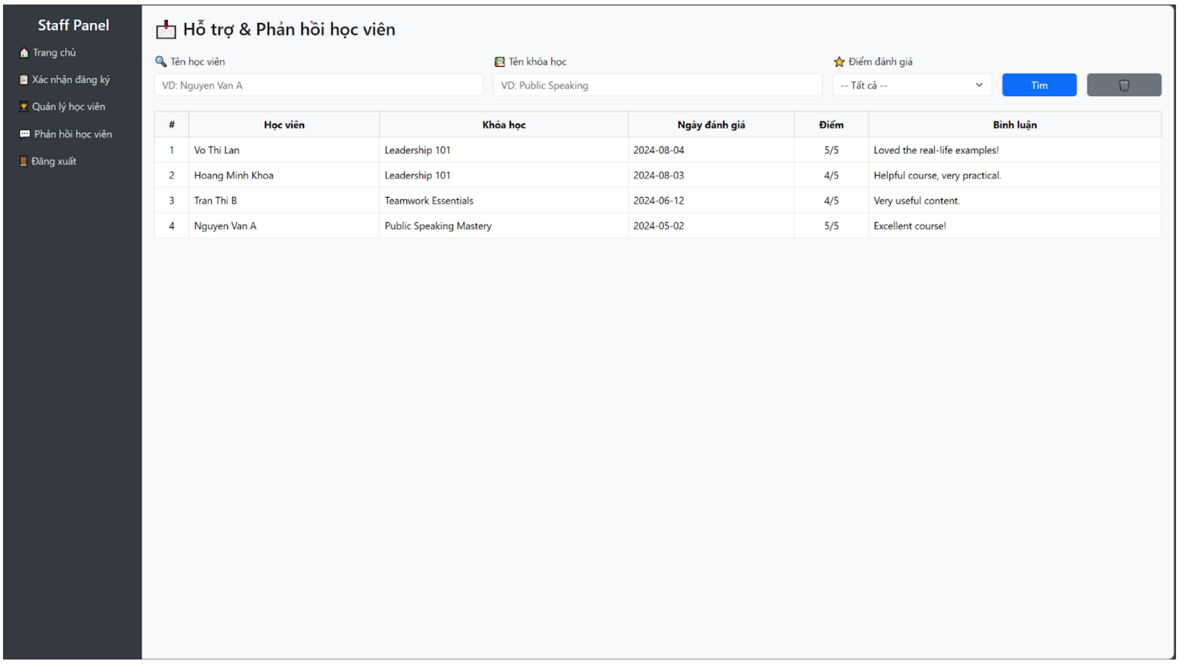
Họ tên: VD: Nguyen Van A

Email: VD: a@student.com

Số điện thoại: VD: 0911...

#	Họ tên	Email	Điện thoại	Ngày ghi danh	Khóa học đã xác nhận
1	Tran Thi B	b@student.com	0922222222	2024-03-15	• Teamwork Essentials
2	Nguyen Van A	a@student.com	0911111111	2024-01-10	• Public Speaking Mastery

- **Staff Dashboard - Student Support & Feedback Screen Screen**



Hỗ trợ & Phản hồi học viên

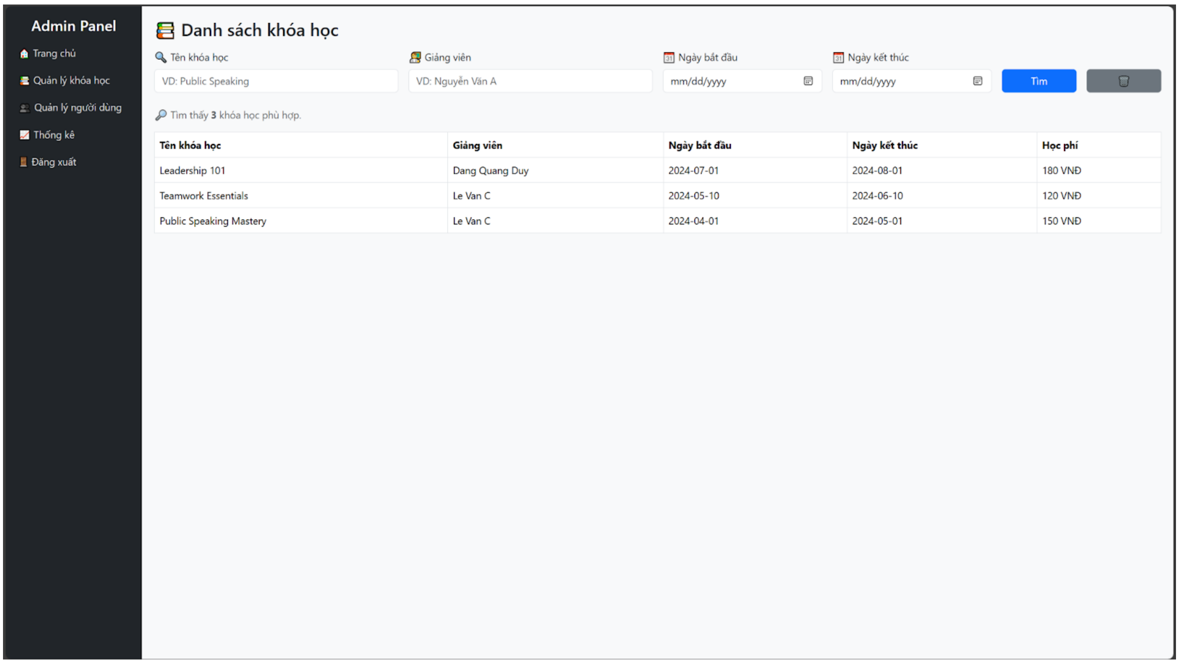
Tên học viên: VD: Nguyen Van A

Tên khóa học: VD: Public Speaking

Điểm đánh giá: -- Tất cả --

#	Học viên	Khóa học	Ngày đánh giá	Điểm	Bình luận
1	Vo Thi Lan	Leadership 101	2024-08-04	5/5	Loved the real-life examples!
2	Hoang Minh Khoa	Leadership 101	2024-08-03	4/5	Helpful course, very practical.
3	Tran Thi B	Teamwork Essentials	2024-06-12	4/5	Very useful content.
4	Nguyen Van A	Public Speaking Mastery	2024-05-02	5/5	Excellent course!

- **Admin Dashboard - Course List Screen Screen**



Danh sách khóa học

Tên khóa học: VD: Public Speaking

Giảng viên: VD: Nguyễn Văn A

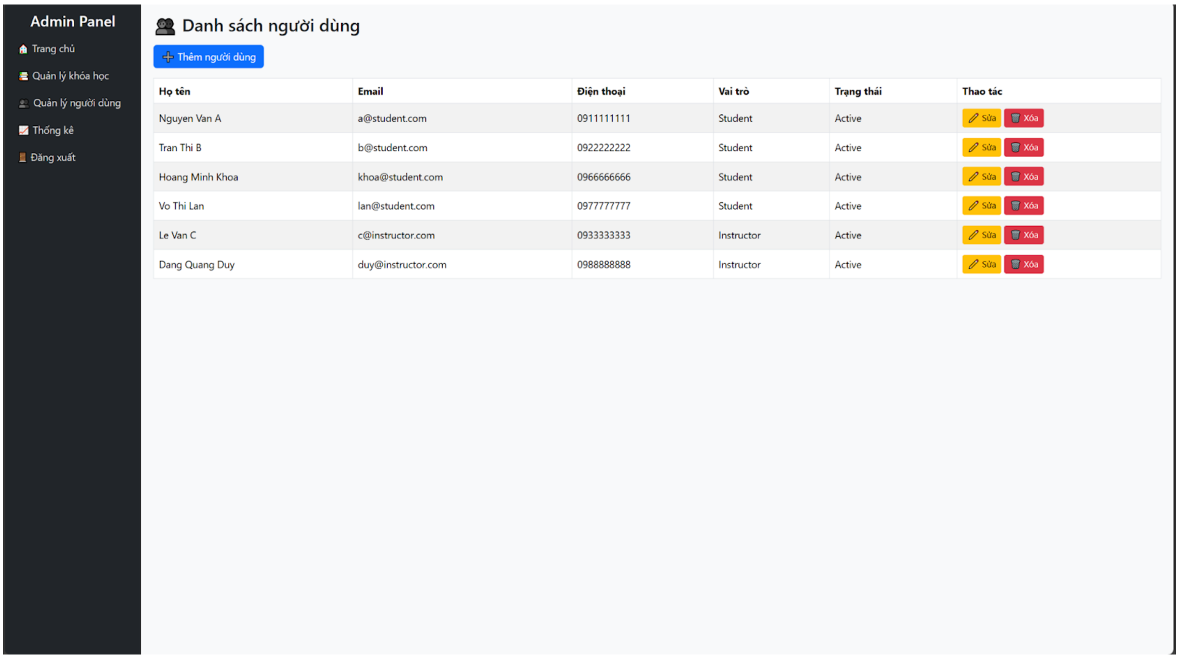
Ngày bắt đầu: mm/dd/yyyy

Ngày kết thúc: mm/dd/yyyy

Tìm thấy **3** khóa học phù hợp.

Tên khóa học	Giảng viên	Ngày bắt đầu	Ngày kết thúc	Học phí
Leadership 101	Dang Quang Duy	2024-07-01	2024-08-01	180 VNĐ
Teamwork Essentials	Le Van C	2024-05-10	2024-06-10	120 VNĐ
Public Speaking Mastery	Le Van C	2024-04-01	2024-05-01	150 VNĐ

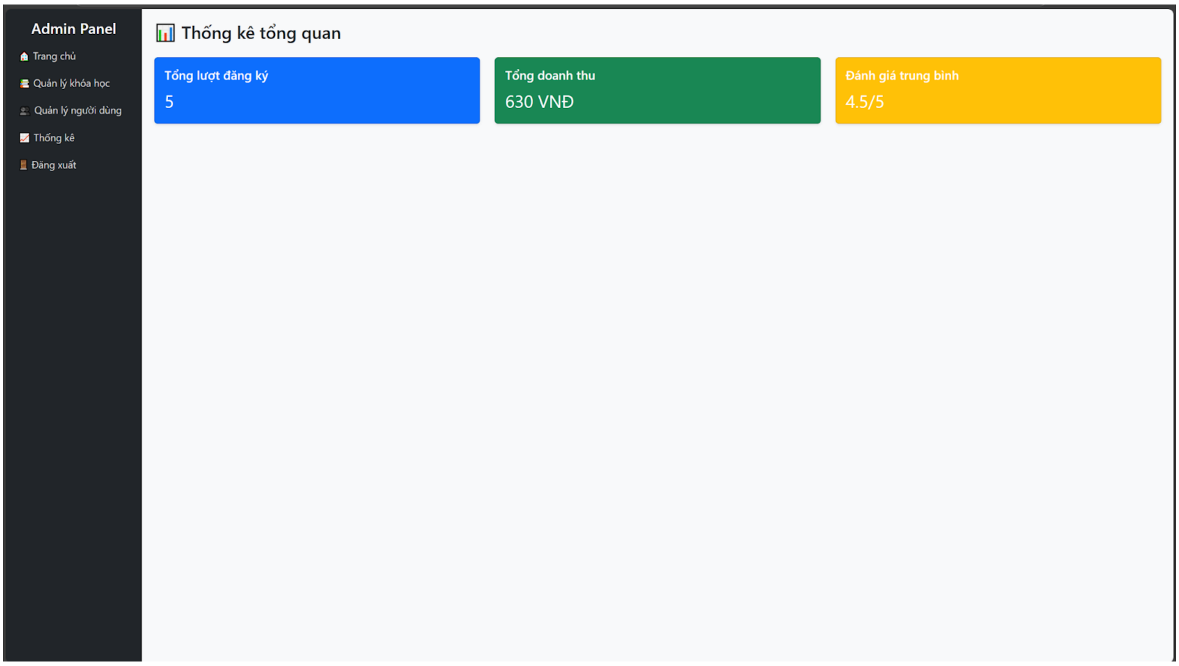
- ## Admin Dashboard - Users List Screen



Danh sách người dùng

Họ tên	Email	Điện thoại	Vai trò	Trạng thái	Thao tác
Nguyen Van A	a@student.com	0911111111	Student	Active	Sửa Xóa
Tran Thi B	b@student.com	0922222222	Student	Active	Sửa Xóa
Hoang Minh Khoa	khoa@student.com	0966666666	Student	Active	Sửa Xóa
Vo Thi Lan	lan@student.com	0977777777	Student	Active	Sửa Xóa
Le Van C	c@instructor.com	0933333333	Instructor	Active	Sửa Xóa
Dang Quang Duy	duy@instructor.com	0988888888	Instructor	Active	Sửa Xóa

- ## Admin Dashboard - Overview Statistics Screen



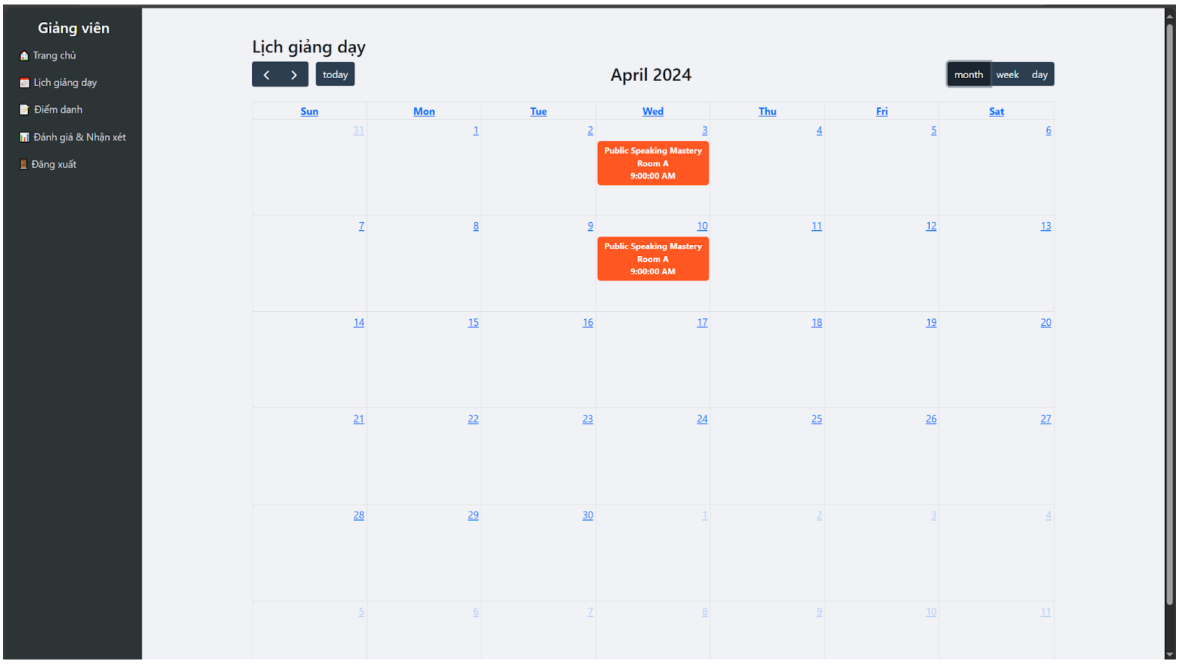
Thống kê tổng quan

Tổng lượt đăng ký
5

Tổng doanh thu
630 VNĐ

Đánh giá trung bình
4.5/5

- **Instructor Dashboard - Schedule Screen**

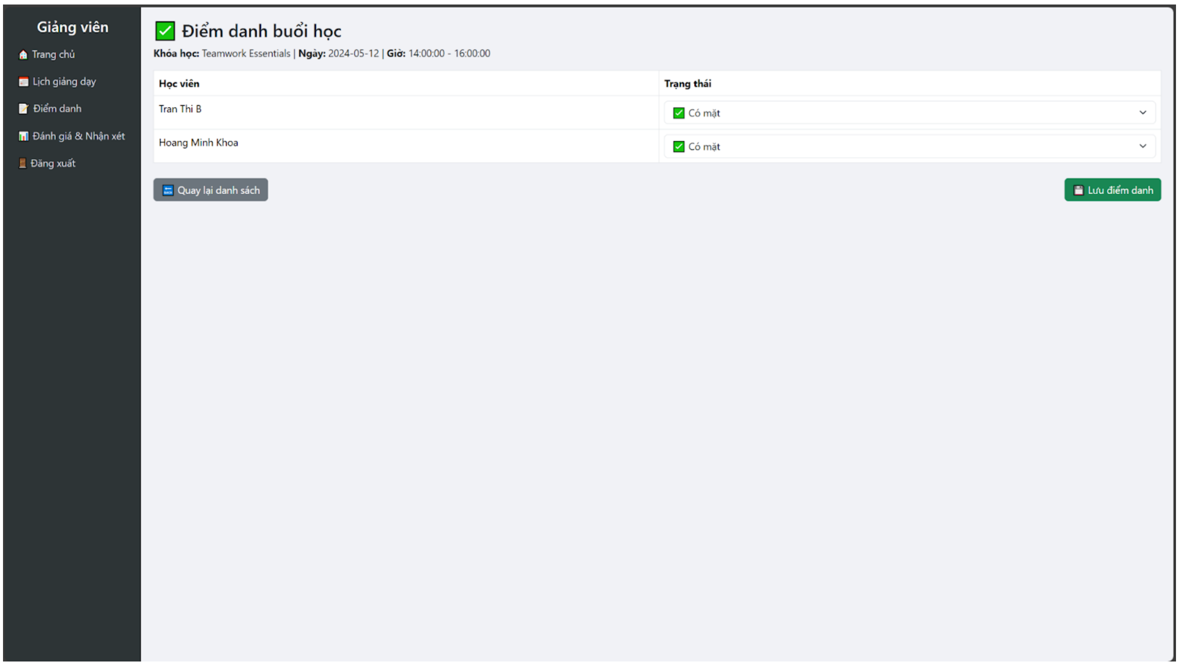


Lịch giảng dạy

April 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3 Public Speaking Mastery Room A 9:00:00 AM	4	5	6
7	8	9	10 Public Speaking Mastery Room A 9:00:00 AM	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

- **Instructor Dashboard - Attendance Tracking Screen**

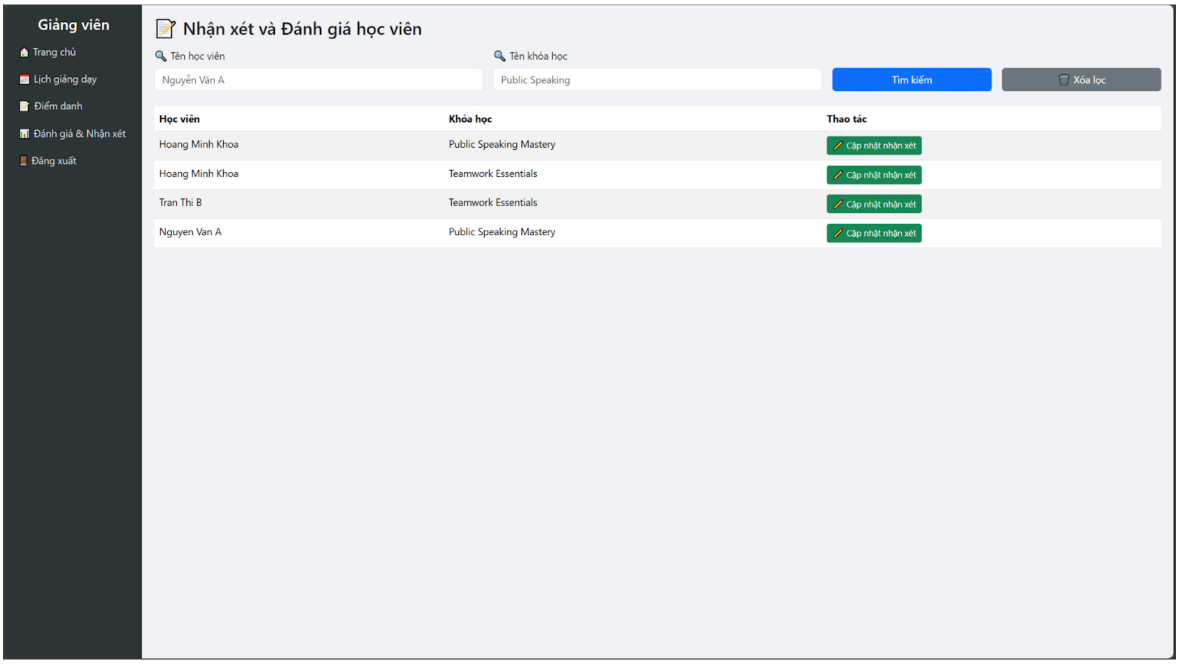


✅ Điểm danh buổi học

Khóa học: Teamwork Essentials | **Ngày:** 2024-05-12 | **Giờ:** 14:00:00 - 16:00:00

Học viên	Trạng thái
Tran Thi B	✅ Có mặt
Hoang Minh Khoa	✅ Có mặt

- **Instructor Dashboard - Student Evaluation Screen**



Nhận xét và Đánh giá học viên

Tên học viên: Nguyễn Văn A

Tên khóa học: Public Speaking

Học viên	Khóa học	Thao tác
Hoang Minh Khoa	Public Speaking Mastery	Cập nhật nhận xét
Hoang Minh Khoa	Teamwork Essentials	Cập nhật nhận xét
Tran Thi B	Teamwork Essentials	Cập nhật nhận xét
Nguyen Van A	Public Speaking Mastery	Cập nhật nhận xét

TEAM MEMBER RESPONSIBILITIES

Members	Main Task
Tran My Van – 522H0083	BA – Developer
Pham Xuan Truong – 522H0069	BA – Developer
Bach Phuong Binh – 522H0001	BA – Developer

Note: All members participate in learning all parts so the division of work is guaranteed to be even.

REFERENCES

- https://www.lucidchart.com/pages/uml-use-case-diagram
- https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/
- https://www.lucidchart.com/pages/er-diagrams
- https://www.geeksforgeeks.org/waterfall-model/