VIETNAM – KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Faculty of Computer Science

GRADUATION PROJECT

BUILDING AN ONLINE LEARNING SYSTEM WITH INTEGRATED AI TO SUGGEST LEARNING CONTENT

Students: **Thua-Huy Khuc**

Van-Huy Pham

Class: **20SE6**

Supervisor: Dr. Dai-Tho Dang

Da Nang, December - 2024

VIETNAM – KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Faculty of Computer Science

GRADUATION PROJECT

BUILDING AN ONLINE LEARNING SYSTEM WITH INTEGRATED AI TO SUGGEST LEARNING CONTENT

Students: **Thua-Huy Khuc**

Van-Huy Pham

Class: **20SE6**

Supervisor: Dr. Dai-Tho Dang

Da Nang, December - 2024

SUPERVISOR'S COMMENTS

ACKNOWLEDGEMENTS

We would like to profoundly acknowledge the people who have helped us during my studies:

First of all, we would like to sincerely thank .....

We would like to thank .....

Many thanks to .....

Finally, we want to thank .....

TABLE OF CONTENTS

ABBREVIATIONS

ABBREVIATIONS	MEANING
AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
UI	User Interface
API	Application Programming Interface
SQL	Structured Query Language
XSS	Cross-Site Scripting
JSON	JavaScript Object Notation
NoSQL	Not Only SQL
MVC	Model-View-Controller
CXR	Chest X-Ray
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
IoT	Internet of Things
UX	User Experience
HTML	HyperText Markup Language

CSS	Cascading Style Sheets
JS	JavaScript
DB	Database
IDE	Integrated Development Environment

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Introduction to the topic

1.1.1 Reason for choosing the topic

In the context of the Fourth Industrial Revolution and the global trend of digital transformation, the demand for online learning has been growing strongly. Platforms like Udemy, Coursera, and edX have clearly demonstrated the convenience and accessibility of knowledge without limitations in geography, time, and cost. However, to increase personalization and effectiveness in learning, the application of Artificial Intelligence (AI) in these platforms has become an essential trend.

1.1.2 Scope of the topic

The scope of this project involves developing a comprehensive online learning platform that integrates AI to enhance user experience, learning efficiency, and course management. Key components of the project include:

- AI-Driven Course Recommendation: Building a system that uses AI to suggest personalized courses based on users' preferences, learning history, and performance.
- User and Content Management: Creating robust systems for managing user accounts, roles (admin, instructor, student), and course content efficiently.
- Multi-Device Accessibility: Designing the platform to be accessible from various devices, including desktops, tablets, and smartphones, to ensure flexibility for users.
- By addressing these components, the project aims to deliver an innovative and practical solution to enhance the online learning experience while leveraging the power of AI.

1.1.3 Topic objectives

The objective of this project is to develop an AI-integrated online learning platform that enhances the personalization and efficiency of the learning experience. Specifically, the platform aims to suggest personalized courses based on learners' preferences and progress, track and analyze learning outcomes for optimization, streamline user and content management, foster user engagement through interactive and gamified content, and provide actionable analytics for instructors and learners. Additionally, the platform will prioritize scalability, security, and multi-device accessibility to ensure a robust, user-friendly, and inclusive learning environment.

1.2 Support tools

To develop the AI-integrated online learning platform, a combination of modern technologies and frameworks has been carefully selected to ensure scalability, efficiency, and a seamless user experience. The chosen tools span across frontend development, backend development, database management, version control, and artificial intelligence integration. Below is a detailed description of the technologies utilized:

Frontend Development

ReactJS

ReactJS, a powerful JavaScript library, is employed for building the platform's user interface. Its component-based architecture allows developers to create reusable UI elements, ensuring consistency and maintainability. ReactJS provides the flexibility to create a highly dynamic and responsive user experience, catering to the diverse needs of learners and administrators. Its virtual DOM capability enhances performance by minimizing updates to the actual DOM.

Tailwind CSS

Tailwind CSS is used as the CSS framework to design a clean, modern, and highly customizable user interface. Its utility-first approach allows developers to style elements directly within the HTML, significantly reducing the time required for UI development. Tailwind CSS ensures that the platform is visually appealing, responsive, and accessible on multiple devices, including desktops, tablets, and smartphones.

Backend Development

Node.js

Node.js serves as the runtime environment for building the backend of the platform. Known for its event-driven, non-blocking architecture, Node.js ensures high performance and scalability, making it ideal for handling the real-time interactions and high traffic demands of an online learning platform.

Express.js

Built on top of Node.js, Express.js simplifies the development of robust APIs and server-side logic. This lightweight framework provides essential tools for managing routes, middleware, and HTTP requests, allowing for efficient communication between the frontend and the backend. Its flexibility and

modularity make it an excellent choice for building a server that handles user authentication, course management, and AI model integration.

Database Management

MongoDB

MongoDB, a NoSQL database, is chosen for its ability to handle unstructured and semi-structured data. Its document-oriented data model offers flexibility in storing user profiles, course content, and AI model parameters, adapting seamlessly to the evolving needs of the platform. MongoDB's scalability ensures that the system can manage large datasets efficiently as the user base and course library expand.

Version Control and Collaboration

Git

Git is utilized as the version control system to track and manage changes to the codebase. By enabling developers to work on different features simultaneously and merge their work seamlessly, Git ensures a smooth development workflow and prevents code conflicts.

GitHub

GitHub provides a collaborative platform for hosting the repository and facilitating teamwork. With features like issue tracking, pull requests, and automated workflows, GitHub streamlines project management and promotes efficient collaboration among team members.

Artificial Intelligence Integration

The integration of AI capabilities is a cornerstone of the platform, enhancing features like personalized learning and automated assessment. The following tools and libraries are used:

Joblib

Used for saving and loading trained AI models, joblib ensures that machine learning models are efficiently stored and retrieved, reducing the computational cost of retraining models.

Scikit-learn

A comprehensive machine learning library, scikit-learn provides a wide range of algorithms for data preprocessing, classification, regression, and clustering. It

plays a critical role in implementing AI features, such as recommending courses and analyzing learning patterns.

Pandas

pandas is employed for data manipulation and analysis. Its powerful DataFrame structure simplifies the handling of large datasets, allowing for efficient cleaning, transformation, and preparation of data for AI model training.

tqdm

tqdm is used to display progress bars during lengthy data processing or training tasks, improving the development experience by providing real-time feedback on task completion.

re (Regular Expressions)

The re module is utilized for text processing, including cleaning and normalizing input data. This ensures that user-generated content and course materials are free from unnecessary characters and formatted consistently before being used in AI algorithms.

1.2.2 Language used

The development of this project utilizes the following programming languages:

- **JavaScript**: The primary language for both frontend and backend development, enabling dynamic and scalable application development.
- **CSS**: For styling the user interface, implemented using the Tailwind CSS framework.
- **HTML**: To structure the content and layout of the web application.
- **Python**: Used in AI model development for its rich ecosystem of libraries, including scikit-learn, pandas, and joblib.

1.2.3 Support tools

- **Visual Studio Code**: Used for writing and managing the codebase.
- **Figma**: Employed to design UX/UI interfaces for a visually appealing and user-friendly platform.
- **MongoDB Atlas**: Utilized for database management and data handling in a secure and scalable environment.
- **Postman**: A tool for testing API endpoints and ensuring robust backend performance.

1.3 Works related to the topic

1.3.1 Overviews of online learning trends

In recent years, online learning (E-learning) has emerged as a transformative approach to education, allowing learners to access high-quality courses from anywhere at any time. Popular platforms like Coursera, Udemy, and Khan Academy have set benchmarks for delivering diverse learning experiences. However, the sheer number of available courses often overwhelms users, highlighting the need for personalized learning pathways. Artificial Intelligence (AI) has played a crucial role in addressing this challenge by enabling personalized content delivery, intelligent tutoring systems, and course recommendation systems.

1.3.2 Related Research and Applications

- **Academic Research:** Numerous studies have explored the integration of AI into E-learning systems. Key contributions include:
  + *"Using Collaborative Filtering to Recommend Courses in Online Learning Platforms"*: This study highlights the use of collaborative filtering techniques to analyze user behavior and recommend courses based on similar users' preferences.
  + *"AI-based Personalized Learning in E-Learning Systems"*: This research investigates the application of machine learning algorithms to adapt learning materials to individual users' needs.
  + *"Deep Learning for Adaptive Learning Systems"*: The paper discusses how deep learning models can predict student success and provide tailored recommendations.
- **Online Learning Platforms:** Several leading platforms have integrated AI to enhance user experiences:
  + **Coursera**: Utilizes AI algorithms to suggest courses based on a user’s prior learning history and career goals.
  + **Udemy**: Employs data-driven insights to recommend popular courses relevant to the user’s interests.
  + **Duolingo**: Leverages AI to personalize language lessons by analyzing users’ strengths and weaknesses.
  + **LinkedIn Learning**: Offers AI-powered suggestions based on users’ professional profiles and trending skills.
- **Technological Innovations:** AI techniques like Collaborative Filtering, Content-Based Filtering, and Hybrid Recommendation Systems are widely used in course recommendation.

1.3.3 Comparative Analysis and Gaps

While the above platforms and studies have significantly advanced the E-learning landscape, certain limitations persist:

- Lack of deep personalization for niche or local markets.
- Limited consideration of non-academic factors such as user motivation or time constraints.
- Challenges in integrating real-time feedback for continuous improvement.

These gaps provide opportunities for innovation. For instance, combining AI-driven insights with real-time analytics can enable dynamic course recommendations tailored to individual learners' evolving needs.

1.3.4 Relevance to the Current Project

This project aims to leverage AI to develop a course recommendation system that addresses the limitations of existing platforms. By integrating Collaborative Filtering, Content-Based Filtering, and user behavior analysis, the system will:

- Provide highly personalized course recommendations.
- Enhance accessibility for regional or underserved user groups.
- Offer a seamless and intuitive user experience through advanced AI techniques.

In summary, this project builds on existing works by addressing current gaps and offering innovative solutions to improve online learning outcomes.

1.4 Project structure

The report includes the following main sections:

Chapter 1: Introduction

This chapter introduces the purpose, significance, and reasons for choosing the topic. It also provides an overview of the current state of research on the topic.

Chapter 2: System Analysis and Design

This chapter outlines the system requirements, including user requirements, functional and non-functional requirements, and system requirements. Additionally, the system design is described in detail.

Chapter 3: Implementation and Experimental Results

This chapter focuses on the implementation of the system and presents the experimental results.

Chapter 4: System Testing

This chapter focuses on system testing and executing test cases.

CHAPTER 2. SYSTEM ANALYSIS AND DESIGN

In the process of developing a software system, the analysis and design phase plays an important role in ensuring the accuracy and efficiency of the final product. This phase helps to clearly define the system requirements, including functional and non-functional requirements, and then build a suitable design model.

Analysis and design is not simply a description of functions, but also a process of converting user requirements into detailed technical components. This helps to shape the system infrastructure and clarify how the components interact with each other throughout the process.

In this chapter, we will present the process of analyzing requirements and designing the system, including identifying the main functions, building Use Case, Class, Activity diagrams and designing basic interfaces. The goal is to build a solid foundation for proceeding to the installation and deployment phase in the next steps.

2.1 Requirements analysis

2.1.1 User requirements

Role: Student

As a student, I want to search and view a list of courses so I can register for the appropriate course, including details such as course name, description, price, reviews, etc.

I want to view details of courses I have registered for so I can track my progress, including a list of completed lectures and remaining lectures.

I want to watch the videos of lectures I have registered for so I can learn from them.

I want to leave a review for the course I have registered for so that I can share my experience with the course to people who would have the intention to register the course.

I want to discuss the contents, lectures of the course and communicate with other students, author so that I can connect with the community to exchange and clarify inquiries.

I want to change my password to enhance the security of my account.

I want to update my personal information in my profile, including my avatar and account name.

Role: Instructor

As an instructor, I want to create, manage, update and remove the courses, categories I offer, including creating course info, lecture lists, attaching learning videos, and categorizing content.

I want to see the revenue chart from the published courses to track income performance and student learning demand.

Role: Administrator

As a manager, I want to have particular control over the E-Learning system, including assigning roles, management users (students, instructors), courses, categories, and all revenue from the courses on the system as a classification method to maintain consistency within the system.

I want to see the list of student and instructor accounts on the system to facilitate role assignment.

I want to create accounts for instructors to grant them management permission for their own courses and categories.

2.1.2 Functional requirements

Based on the user requirements analysis, the system includes the following main functions:

Course Search:
- Allows users to search for courses by title, author, or price.
- Supports search suggestions based on using filters by prices, categories.

Course Enrollment:
- Manages course registration, payment, and access to course content.
- Allows users to view a list of enrolled courses.

Course Creation and Management (Administrator, Instructor):
- Allows to create courses and add detailed content such as videos, lectures, and supporting documents.
- Updates the lecture list or deletes outdated courses.

User Management (Administrator):

- Allows administrators to view the list of users on the system.
- Assigns roles to users (Student, Instructor).
- Creates instructor accounts.
- Deletes invalid or rule-violating accounts.

Course Categories Management (Administrator, Instructor):

- Allows to create, edit, or delete course categories.
- Associates courses with categories to facilitate better management and searching.

Learning Progress and Interaction (Student & Instructor):

Student:

- Tracks personal learning progress through the list of completed lectures and remaining content.
- Provides rating and reviewing courses after experience.
- Provides commenting/discussion within courses.

Instructor:

- Monitors student progress in their courses.
- Provides feedback or assistance to students to help them complete the course.

Payment and Transaction Management:

- Supports payment transactions for course purchases.
- Sends transaction success notifications to users.
- Manages transaction history, including transaction statuses (pending, completed, failed).

Business revenue management (Administrator, Instructor):

Instructor:

- Allows to view and track income from owned published courses including total sales and total income.

Administrator:

- Allows to view and track income from published courses on the system including total sales and total income from other instructors and owned courses.
- Provides statistical charts to facilitate viewing business progress.

2.1.3 Non-functional requirements

The system must meet the following non-functional requirements:

Performance:
- The system must respond within 3 seconds for operations such as course search, enrollment, and content access.
- It must handle at least 500 concurrent users without performance degradation.

Scalability:
- The system must be scalable to accommodate a significant increase in user traffic.
- The system should allow easy integration of new features in the future, such as assignments, quizzes, and certificates.

Security:
- User passwords must be securely encrypted and stored.
- The system must protect user data against attacks such as SQL Injection and XSS.
- Role-based access control must be enforced to ensure users can only access functionalities appropriate to their roles.

Stability:
- The system must operate continuously.
- It must provide quick recovery in case of system failures.

User-Friendliness:
- The system interface must be intuitive and easy to use for all types of users.
- It must support access on popular devices, including computers, tablets, and smartphones.

Compatibility:
- The system must be compatible with popular browsers such as Google Chrome, Firefox, Safari, and Microsoft Edge.
- It should function seamlessly on major operating systems like Windows, macOS, Android, and iOS.

Maintainability:
- The system codebase must be clean, readable, and well documented to facilitate maintenance and future upgrades.
- Logs must be maintained to monitor critical system activities.

2.1.4 Use case diagram

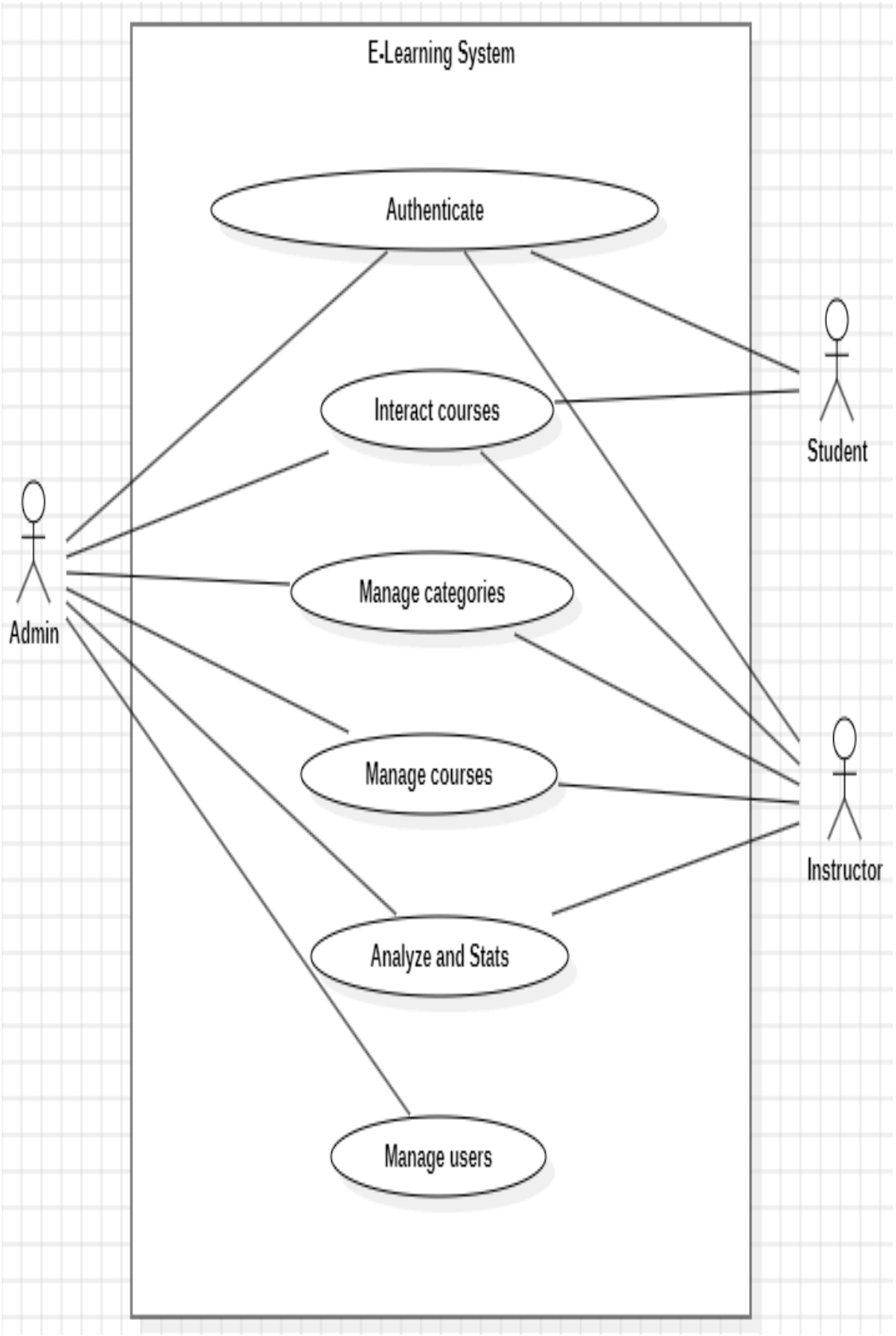


Figure 1. E-Learning system overview use-case diagram

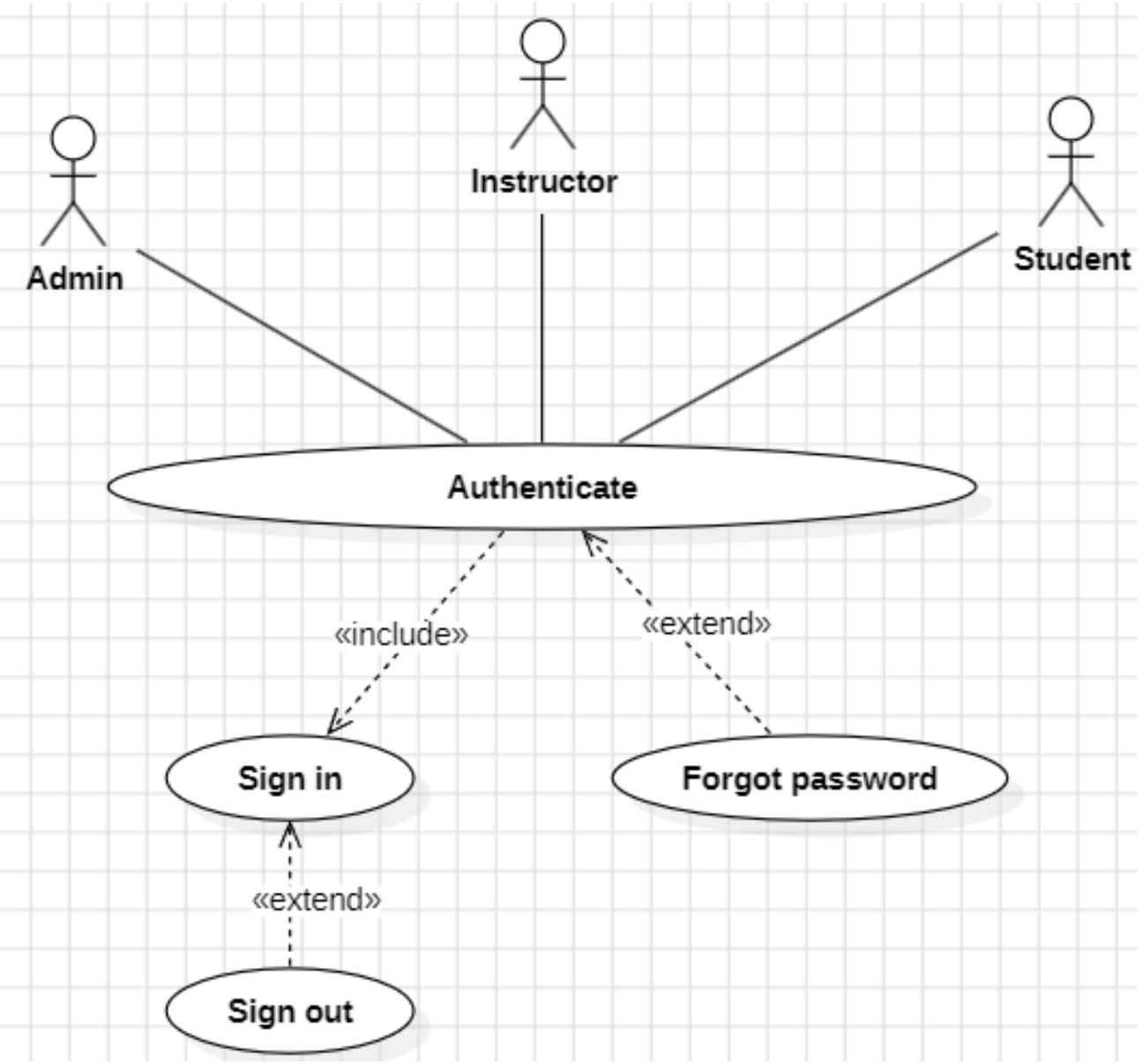


Figure 2. Authentication use-case diagram

Use-case name	Authentication use-case
Description	Authenticate user credentials to access the system.
Actor	Student, Instructor, Administrator
Trigger	The user opens the login page and enters login information.
Pre-Condition	User already has a valid account.

Post-Condition	**Success**: The user successfully logs in and is redirected to the main page. **Failure**: The user cannot log in due to invalid information or a system problem.
Basic flow	1. The user opens the login screen. 2. The system displays the interface asking for login information (username and password). 3. The user enters the login information (username and password). 4. The user presses the "Login" button. 5. The system checks the login information: If the information is valid: a. The system authenticates successfully. b. The system redirects the user to the main page. If the information is invalid: a. The system displays an error message: "Incorrect login name or password". 6. End of use case.
Alternative flow	Flow 1: User forgets password 1. At the login screen, the user

	clicks on the "Forgot password" link. 2.The system displays an interface asking for an email address. 3.The user enters the email and clicks the "Send request" button. 4.The system sends a password reset link to the provided email. 5.The user performs the password reset steps. 6.Ends the sub-flow and returns to the login screen. 7.End of use-case. Flow 2: User register an account 1.At the login screen, the user clicks on the "Forgot password" link. 2.The system displays the registration form 3.The user enters the information and clicks Register. 4.The system checks the validity of the information: If valid: the account is successfully created and redirected to the login page. If invalid: the system displays an

	error message. 5.End of use-case.

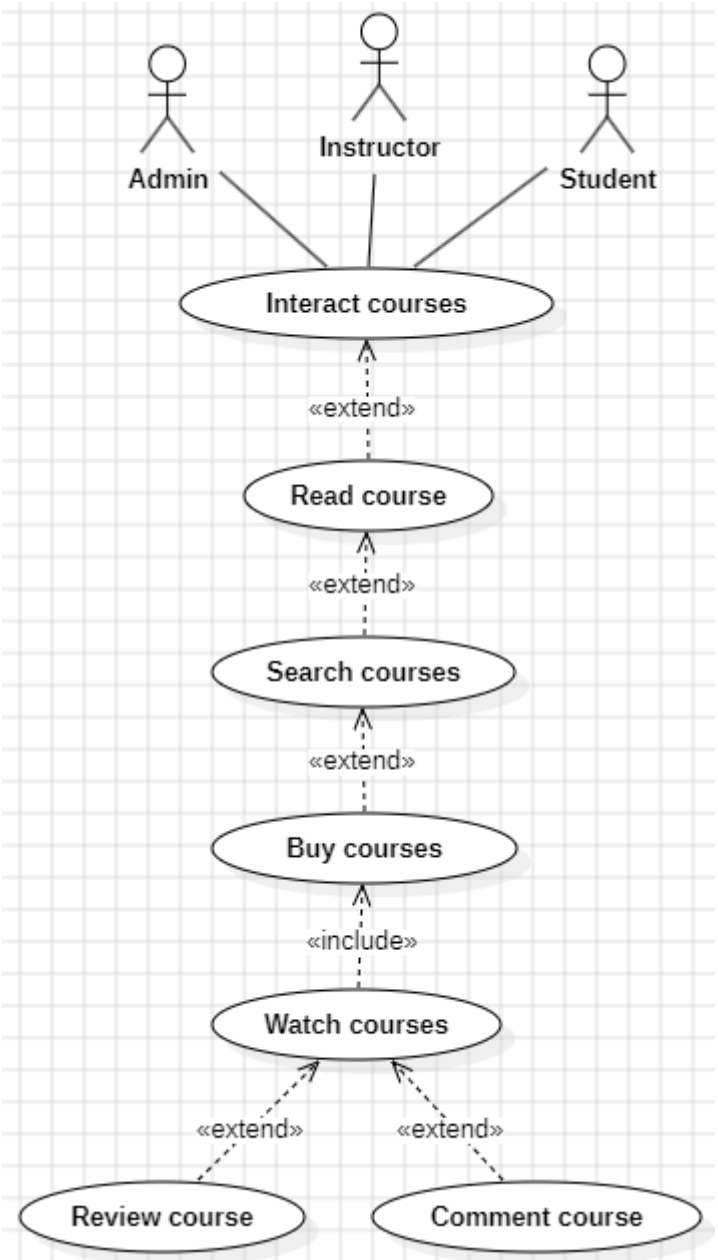


Figure 3. Interact courses use case diagram

Use-case name	Interact courses use-case
Description	Users can interact with the course including: viewing content, rating, commenting, and completing activities in the course.
Actor	Student, Instructor, Administrator
Trigger	The person selects a specific course to interact with.
Pre-Condition	The user is logged in and has purchased or been granted access to the course.
Post-Condition	**Success:** User completes an interaction such as viewing a lesson, posting a comment, or rating a course. **Failure:** The user cannot interact because they have not purchased the course or the course is unavailable.
Basic flow	1. The user accesses the Course List screen and selects a specific course. 2. The system checks the user's course access rights: If the user has rights: the system switches to the Course Content screen. If no rights: the system displays the message "You do not have access to this course" or redirect to the course detail page. 3. The user can perform the following operations: View course content: The system

	displays the list of lessons and allows the user to open each lesson. Comments in the course: a. The user enters the comment content and clicks "Submit". b. The system authenticates the content and displays the comment on the interface. Rate the course: a. The user selects the number of stars (1-5) and writes a review (optional). b. The system records and saves the review. Complete course activities: The user finishes watching a video lecture or test. 4.The system updates the user's learning progress and displays the completion status for each lesson. 5.The user exits the course. 6. End of use-case.
Alternative flow	Flow 1: User does not have access to the course 1.When the user selects a course, the system detects that the user has not purchased the course or has not been granted permission. 2.The system displays the message: "You need to purchase the course to continue" and redirects to the payment screen.

	3.End of use-case. Flow 2: User submits invalid comment 1.The user enters a comment that contains inappropriate keywords or exceeds the allowed length. 2.The system displays the error message: "Invalid comment content". 3.End of use-case. Flow 3: User reviews the course 1.The user only selects the number of stars and does not write a review. 2.The system records the star rating and updates the overall rating of the course. 3.End of use-case. Flow 4: Disconnection while interacting with the course 1.When the user views the course content or submits a comment, the system detects a disconnection. 2.The system displays the message: "Connection is unstable, please try again". 3.End of use-case.

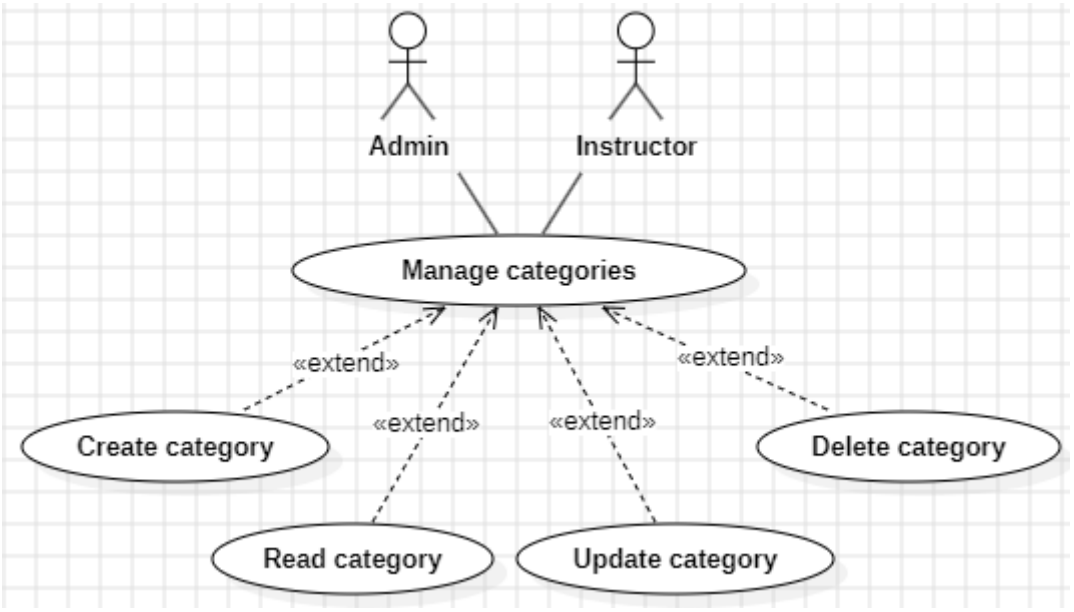


Figure 4. Manage categories use-case diagram

Use-case name	Manage categories use-case
Description	Allows Administrators to perform category management functions, including adding, editing, hiding, and showing categories.
Actor	Administrators, Instructor
Trigger	Administrator access to category management feature.
Pre-Condition	The administrator has successfully logged into the system.
Post-Condition	**Success:** Category added, edited, hidden or shown successfully. **Failure:** Category management operation failed due to system error or invalid information.

Basic flow	1. The administrator selects the Category Management feature from the system menu. 2. The system displays a list of current categories. 3. The administrator performs one of the following functions: **Add a new category:** 3.1. The administrator selects Add category. 3.2. The system displays a form to enter category information (category name, status). 3.3. The administrator enters information and confirms saving. 3.4. The system checks the information and saves the new category. **Edit category:** 3.5. The administrator selects the category to edit from the list. 3.6. The system displays a form to edit category information. 3.7. The administrator updates information and confirms saving. 3.8. The system saves the edited information. **Hide category:** 3.9. The administrator selects the category to delete from the list. 3.10. The system displays a hidden confirmation message. 3.11. The administrator confirms hiding.

	3.12. The system hides the category from the list. **Show categories:** 3.13. The administrator selects the category to display from the list. 3.14. The system displays a display confirmation message. 3.15. The administrator confirms displaying. 3.16. The system displays the category in the list. 4. The system updates the list of categories and notifies the administrator of the result. 5.End of use-case.
Alternative flow	Flow 1: Add category - Invalid information 1.The administrator enters invalid category information (e.g., leaving the category name blank). 2.The system displays an error message: "Category name cannot be blank." 3.The administrator re-enters the information. 4.End of use-case.

	Flow 2: Hide/show category - Cancel operation 1.The administrator chooses to hide/show the category but cancels at the confirmation message. 2.The system does not operate hiding/showing and returns to the category list. 3.End of use-case. Flow 3: System error 1.The administrator performs any function (add, edit, hide/show), but the system encounters an error. 2.The system displays a message: "An error occurred, please try again later." The administrator returns to the category list page. 3.End of use-case.

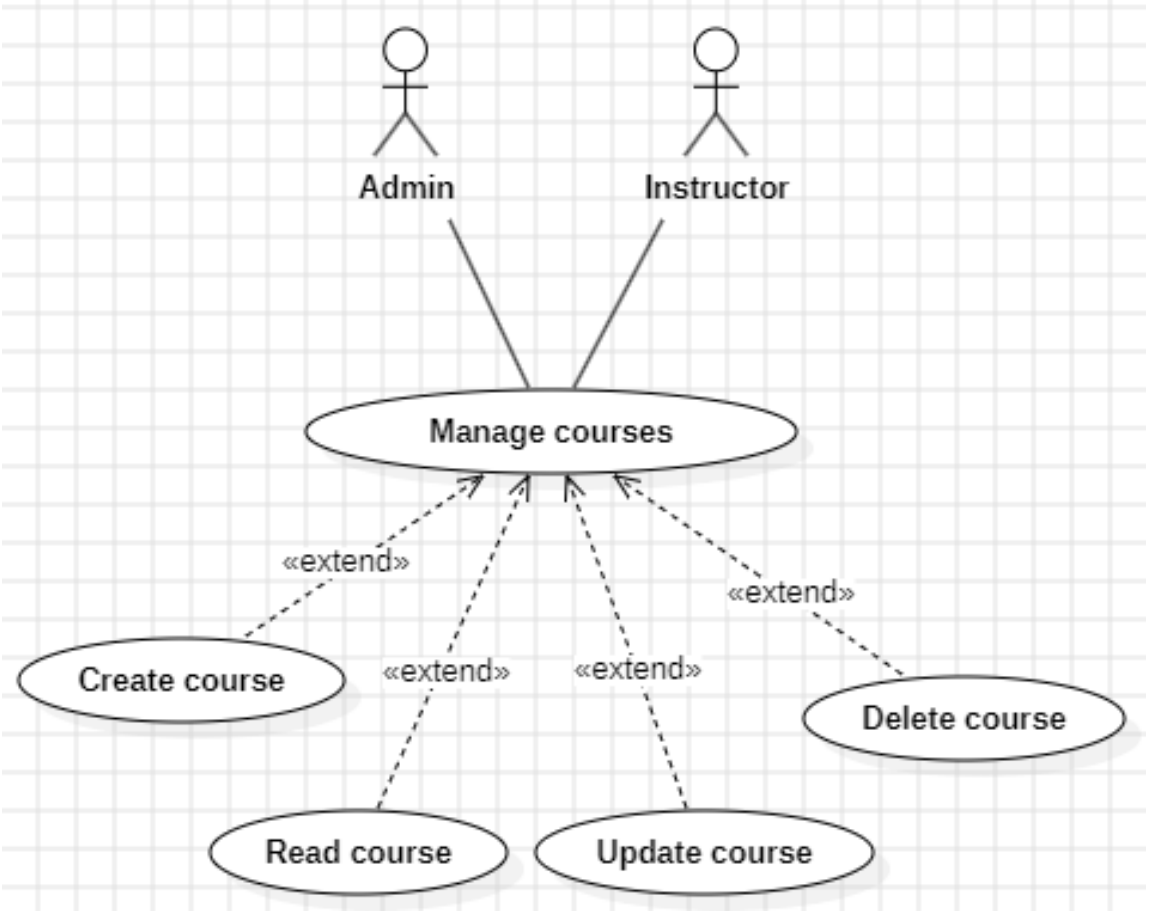


Figure 5. Manage courses use-case diagram

Use-case name	Manage courses use-case
Description	Allows administrators to perform course management functions, including adding, editing, deleting, and viewing course information.
Actor	Administrator, Instructor
Trigger	Administrator access to course management features.
Pre-Condition	The administrator has successfully logged into the system.
Post-Condition	**Success:** The course was successfully added, edited, deleted, or viewed.

	Failure: The course management operation failed due to a system error or invalid information.
Basic flow	1. The administrator selects the Course Management feature from the system menu. 2. The system displays a list of current courses. 3. The administrator performs one of the following functions: Add a new course: 3.1. The administrator selects Create Course. 3.2. The system displays a course information entry form with the following fields: - Course name 3.3. The administrator enters information and confirms saving. 3.4. The system checks the validity and saves the new course information. Edit course: 3.5. The administrator selects a course to edit from the list. 3.6. The system displays a course information editing form.

	3.7. The administrator updates information and confirms saving. 3.8. The system checks and saves the edited information. Publish course: 3.9. The administrator selects a course to publish 3.10. The administrator confirms publishing the course. 3.11. The system checks and publishes the course to the user. Unpublish the course: 3.9. The administrator selects a course to be unpublished 3.10. The administrator confirms unpublishing the course. 3.11. The system checks and unpublished the course from the system. Delete the course: 3.12. The administrator selects a course to be deleted from the list. 3.13. The system displays a confirmation message to delete. 3.14. The administrator confirms deleting the course. 3.15. The system deletes the course and updates the list.

	View course details: 3.16. The administrator selects a course from the list. 3.17. The system displays detailed course information, including: - Course name - Description - Course type - Difficulty - Price - List of lectures - Number of students 4.The system updates the course list and notifies the successful operation result. 5.End of use-case.
Alternative flow	Flow 1: Add course - Invalid information 1. The administrator enters missing or incorrect information (eg: leave the course name blank). 2. The system displays an error message: "Please enter complete and correct course information." 3. The administrator re-enters the information. 4.End of use-case. Flow 2: Delete course - Cancel operation

	1. The administrator chooses to delete the course but cancels the operation at the confirmation message. 2. The system does not delete the course and returns to the course list. 3.End of use-case. Flow 3: Delete course - The course has registered students 1. The administrator chooses to delete the course but the course has registered students. 2. The system does not delete the course and displays the message: "Cannot delete a course that has registered students". 3.End of use-case. Flow 4: Publish course - Course has no lecture 1.The administrator chooses to publish the course but the course has no lecture video. 2.The system does not publish the course and displays the message: "Cannot publish a course without a lecture video". 3.End of use-case. Flow 5: System error

	1.The administrator performs an operation (add, edit, delete, publish or unpublish a course), but the system encounters an error. 2.The system displays the message: "An error occurred, please try again later." The administrator returns to the course list page. 3.End of use-case.

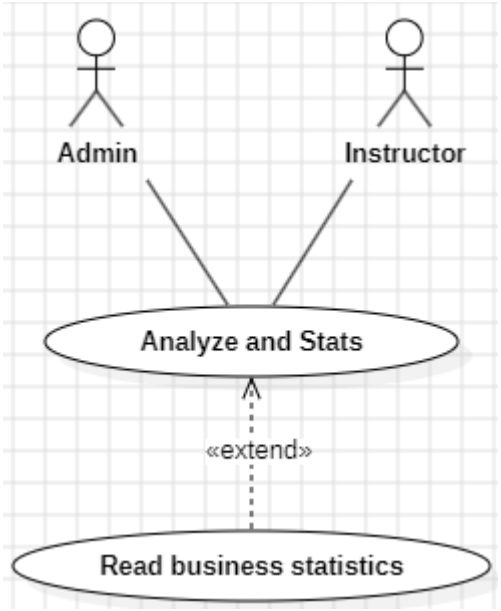


Figure 6. Analyze and Stats use-case diagram

Use-case name	Analyze and Stats use-case
Description	Allows Admin and Instructor to view statistics and analyze data.
Actor	Administrator, Instructor

Trigger	Admin or Instructor access Analyze and Stats function.
Pre-Condition	The user is logged into the system.
Post-Condition	**Success**: Statistics and reports are displayed successfully. **Failure**: System error or data unavailable.
Basic flow	1.The admin selects the Analyze and Stats function from the system interface. 2.The system displays statistics and reports. 4.The system retrieves and analyzes data. 5.The system displays statistics and reports on the interface. 6.The admin views the report results. 7. End of use-case.
Alternative flow	1.The admin selects to view business statistics. 2.The system retrieves business statistics data and displays the report chart. 3.The admin views the analysis results and charts. 4.End of use-case.

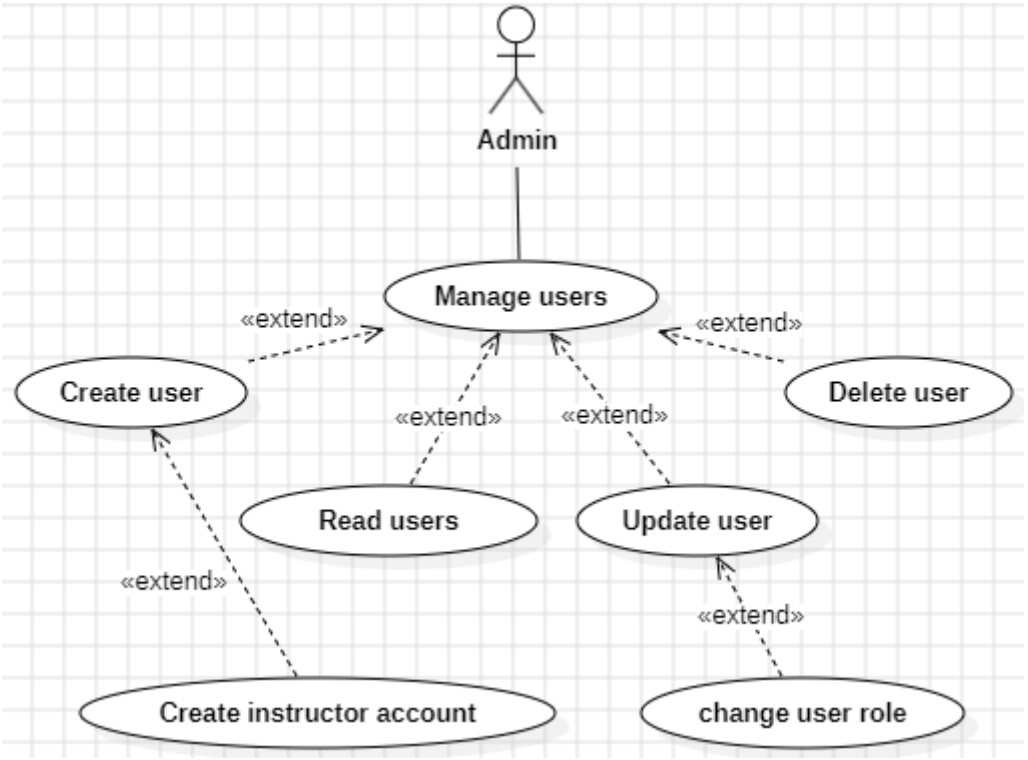


Figure 7. Manage users use case diagram

Use-case name	Manage user use-case
Description	Allows Admin to manage user information and permissions.
Actor	Admin
Trigger	Admin accesses the Manage Users function.
Pre-Condition	Admin has logged into the system.
Post-Condition	**Success:** User information was successfully created or updated. **Failure**: System error or user data unavailable.
Basic flow	1.Admin selects the Manage Users function from the interface.

	2.The system displays the current user list. 3.Admin performs the following operations: - View user information. - Change user role. - Create an instructor account. 4.The system confirms and updates the corresponding data. 5.Admin completes user management. 6. End of use-case.
Alternative flow	Flow 1: Change user role 1.Admin selects a user from the list. 2.Admin changes the user's role (eg from User to Instructor). 3.The system updates the user's role information. 4.End of use-case. Flow 2: Watch all users 1.Admin selects the function to view all users. 2.The system displays the full list of current users. 3.End of use-case.

	Flow 3: Create instructor 1.Admin selects to create an instructor account. 2.The system displays the account creation form with the following fields: Name Email Password 3.Admin enters the information and clicks confirm. 4.The system successfully creates a new instructor account. 5. End of use-case.

2.1.5 Class diagram

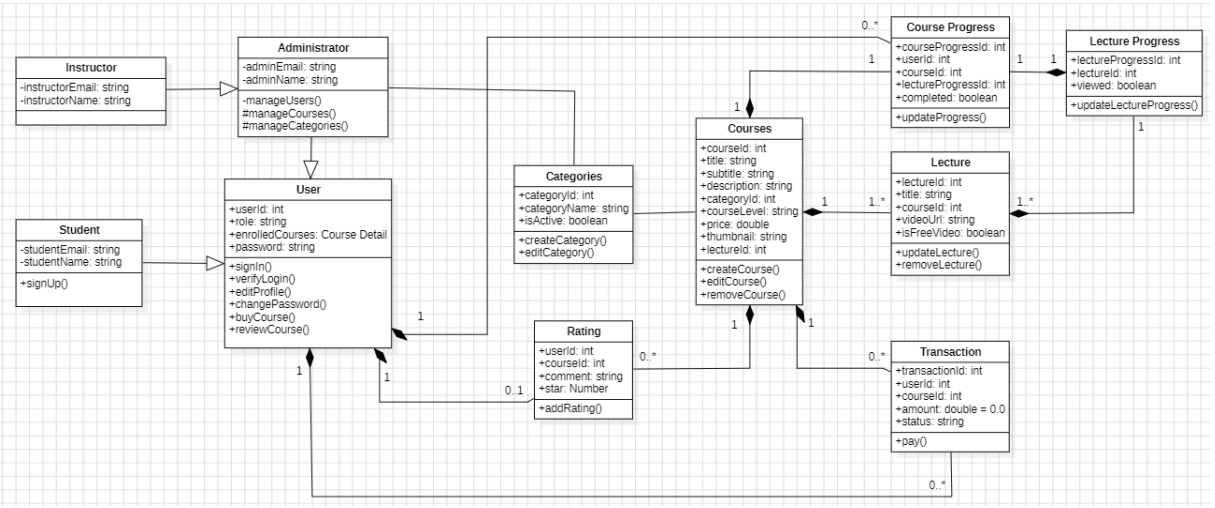


Figure 8. E-Learning system class diagram

Figure 8 presents the class diagram of the E-Learning system, illustrating the structure and relationships between various entities within the system.
User: This is the base class from which two subclasses, **Student** and **Instructor**, inherit. It contains general information about users, such as username and password.

Course: Represents the courses available in the system, including details like course name, description, and related lectures. Each course can contain multiple **Lectures** and receive various **Ratings** from users.
Categories: Classifies courses to help users easily search for and select appropriate courses.
Transaction: Handles transactions related to course registration, including information about the user and the associated course.
Course Progress: Tracks the learning progress of students in courses, indicating the number of completed and remaining lectures.
Rating: Allows users to leave feedback on courses, providing valuable insights for other users and instructors.
Comment: Allows users to comment on courses, providing questions, communications for other users and instructors.

2.1.6 Activity diagram

Figure 9 presents activity diagram includes the following sections:

User: Represents user actions, starting with "Click on login".
System: Represents system responses and actions, including "Check information", "authorize" and "Back to home page".

Flows:
- **Main flow:** User enters account information, system verifies valid information and allows access.
- **Branch flow**: If account information is invalid, system asks user to re-enter.

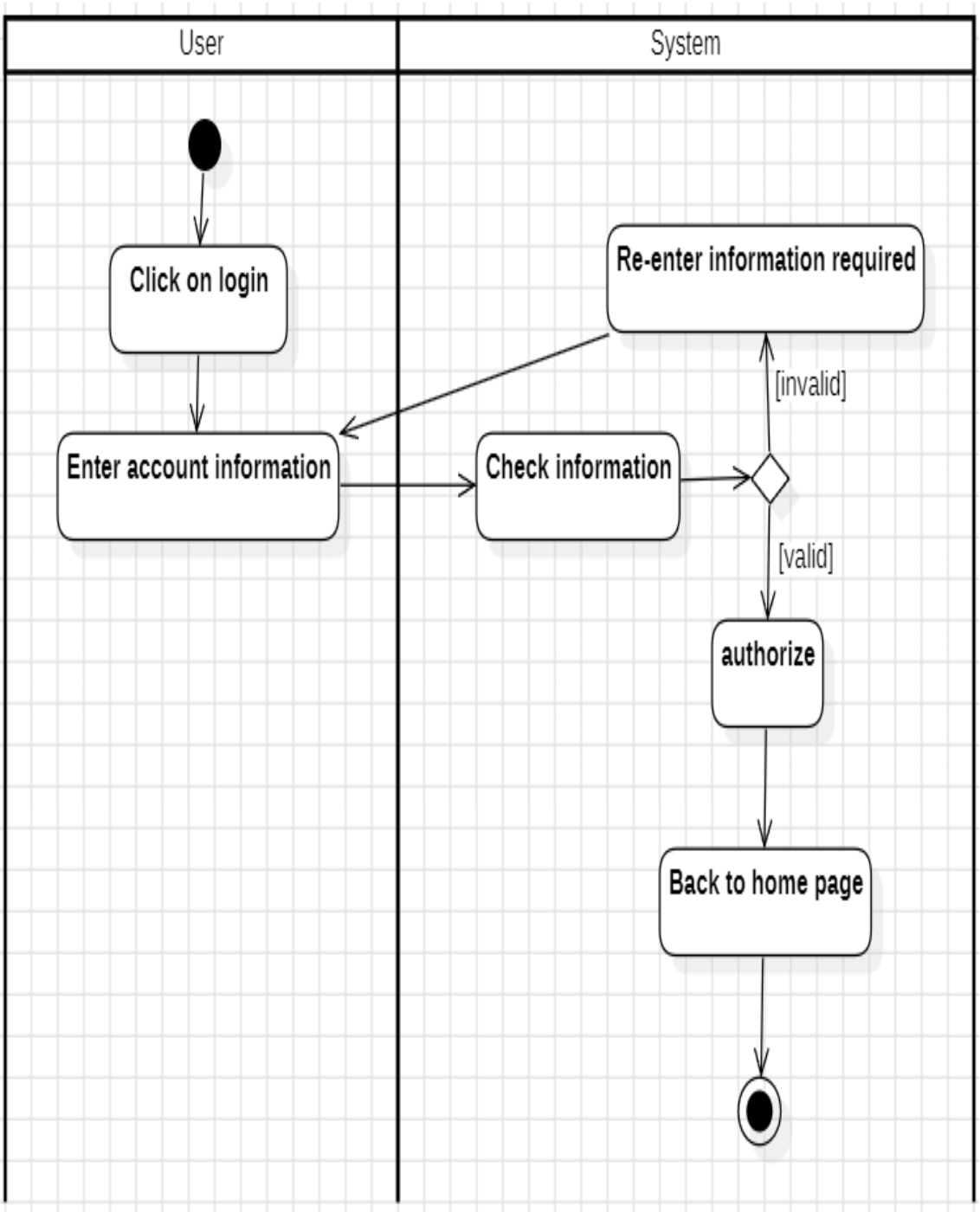


Figure 9. sign in activity diagram

Figure 10 presents activity diagram describes the process of registering a user account in the e-learning system. The main steps include:

User:

- Click on "Click on signup" button to start the registration process.
- Enter the required account information.

System:

- Check the user's registration information.

- If the information does not exist in the system, the system will create a new account for the user with the role of "student".
- After successfully creating an account, the system will redirect the user to the login page.

This diagram clearly illustrates the steps of interaction between the user and the system during the account registration process, including the main flows and branch flows when errors occur.

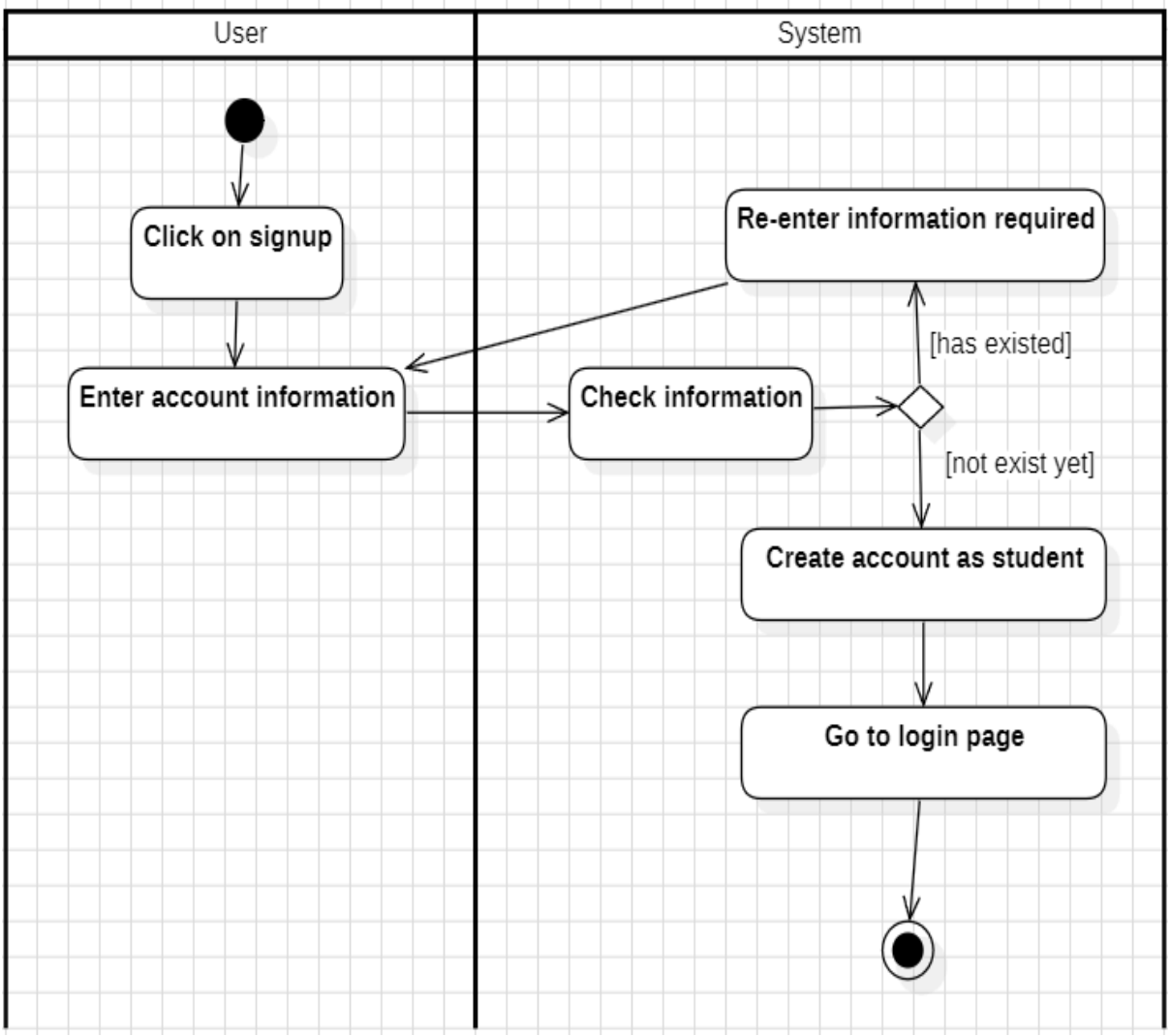


Figure 10. Signup activity diagram

Figure 11 presents activity diagram shows the process for an admin to create a new instructor account in an e-learning system. The key steps are:

Admin:
- Logs in to the system.
- Goes to the "Users Management" page.
- Goes to the "Create Instructor" page.

- Enters the instructor's information.

System:
- Checks if the entered information already exists in the system.
- If the information does not exist, it creates a new instructor account.

The diagram clearly illustrates the flow of actions between the admin and the system, including the decision point where the system checks if the instructor information already exists. This allows the system to either create a new account or prompt the admin to re-enter the information if it already exists.

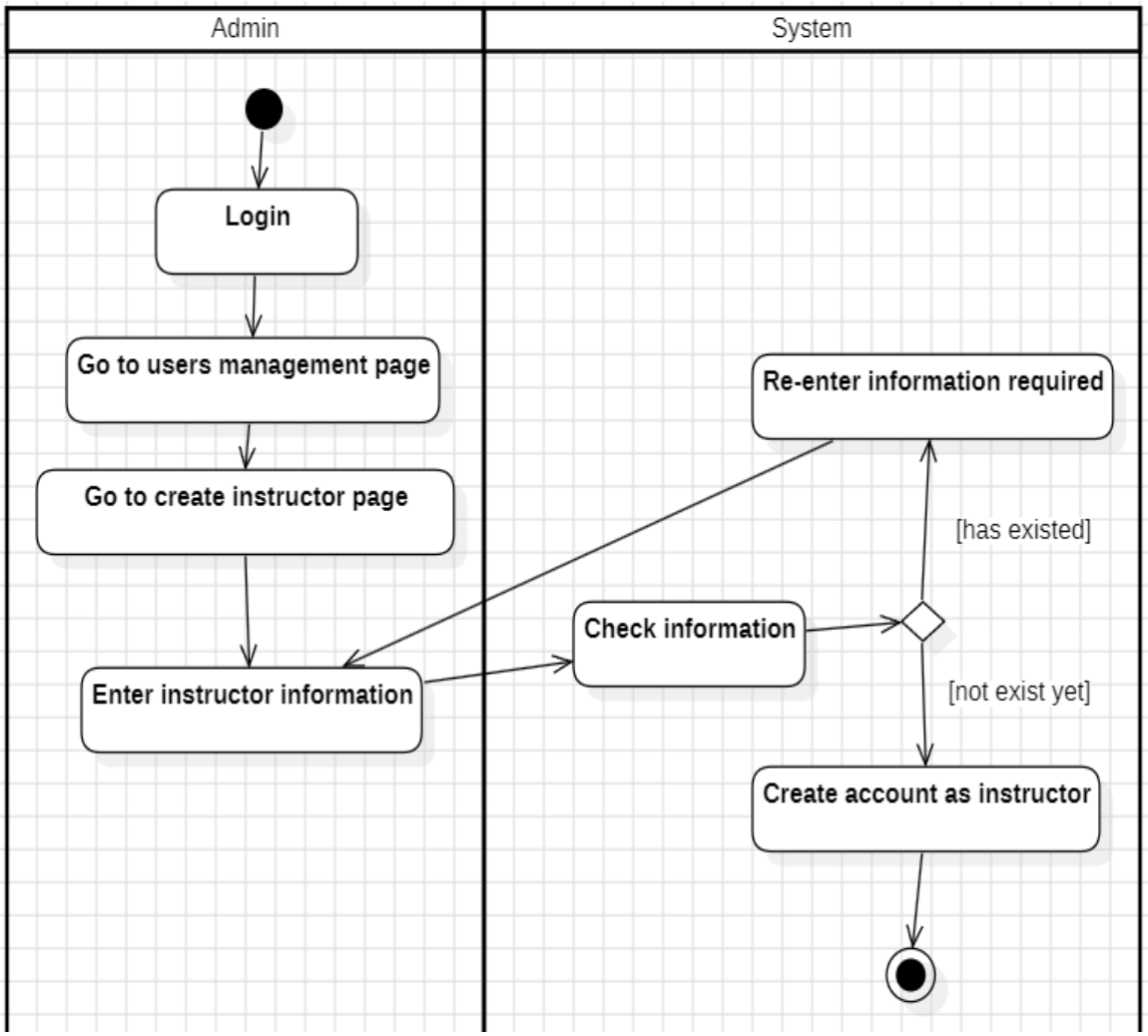


Figure 11. create new instructor activity diagram

Figure 12 presents diagram shows the process for an admin to update a user's role in an e-learning system. The key steps are:

Admin:
- Logs in to the system.

- Goes to the "Users Management" page.
- Selects the user to update.
- Selects the change role option.
- Selects a new role for the user.

System:
- Shows the list of users.
- Displays the user's information.
- Shows the list of roles.
- Updates the user's role with the new selection.

The diagram illustrates the flow of actions between the admin and the system, including the steps for selecting the user, changing their role, and the system updating the user's information accordingly.

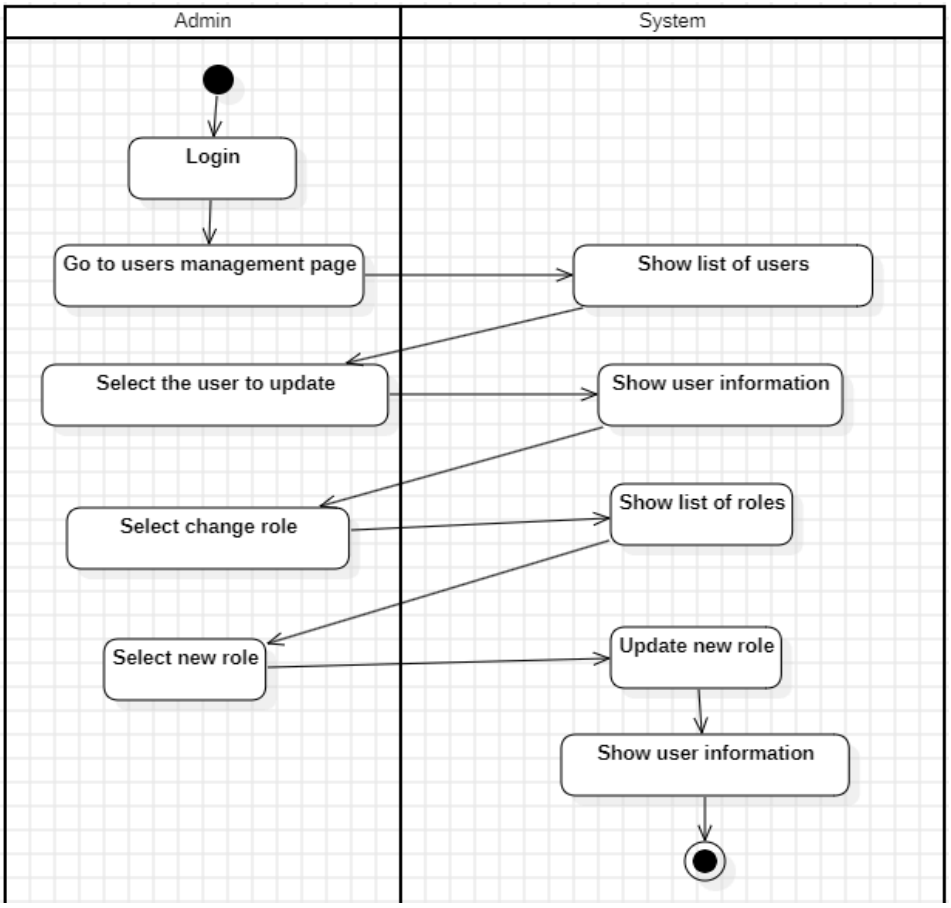


Figure 12. Update user role activity diagram

Figure 13 presents diagram shows the process for an admin to manage categories in an e-learning system. The key steps are:

Admin:

- Logs in to the system.
- Goes to the "Categories Management" page.

System:

- Shows the list of categories.

Admin:

Can perform the following actions on categories:

- Add a new category.
- Update an existing category.
- Delete a category.

System:

- Shows the data entry form when adding or updating a category.
- Checks the data entered by the admin.
- If the data is valid, updates the category information.
- Shows a confirmation form before deleting a category.
- Returns the admin to the categories management page after any changes.

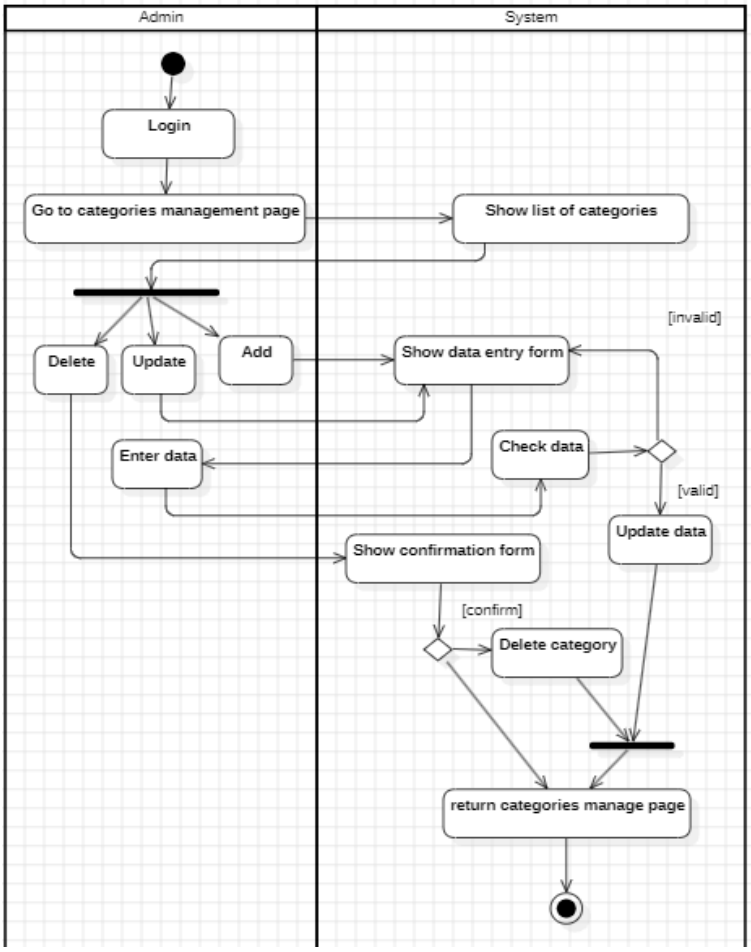


Figure 13. Manage categories activity diagram

Figure 14 presents diagram shows the process for an instructor to create a new course in an e-learning system. The key steps are:

Instructor:

- Logs in to the system.
- Goes to the "Courses Management" page.
- Selects the option to create a new course page.
- Enters the course data.

System:

- Shows the list of courses.
- Displays the data entry form.
- Checks the data entered by the instructor.
- If the data is valid, saves the new course.
- Returns the instructor to the courses management page.

The diagram illustrates the flow of actions between the instructor and the system, including the steps for creating a new course, entering the course data, and the system validating and saving the new course.

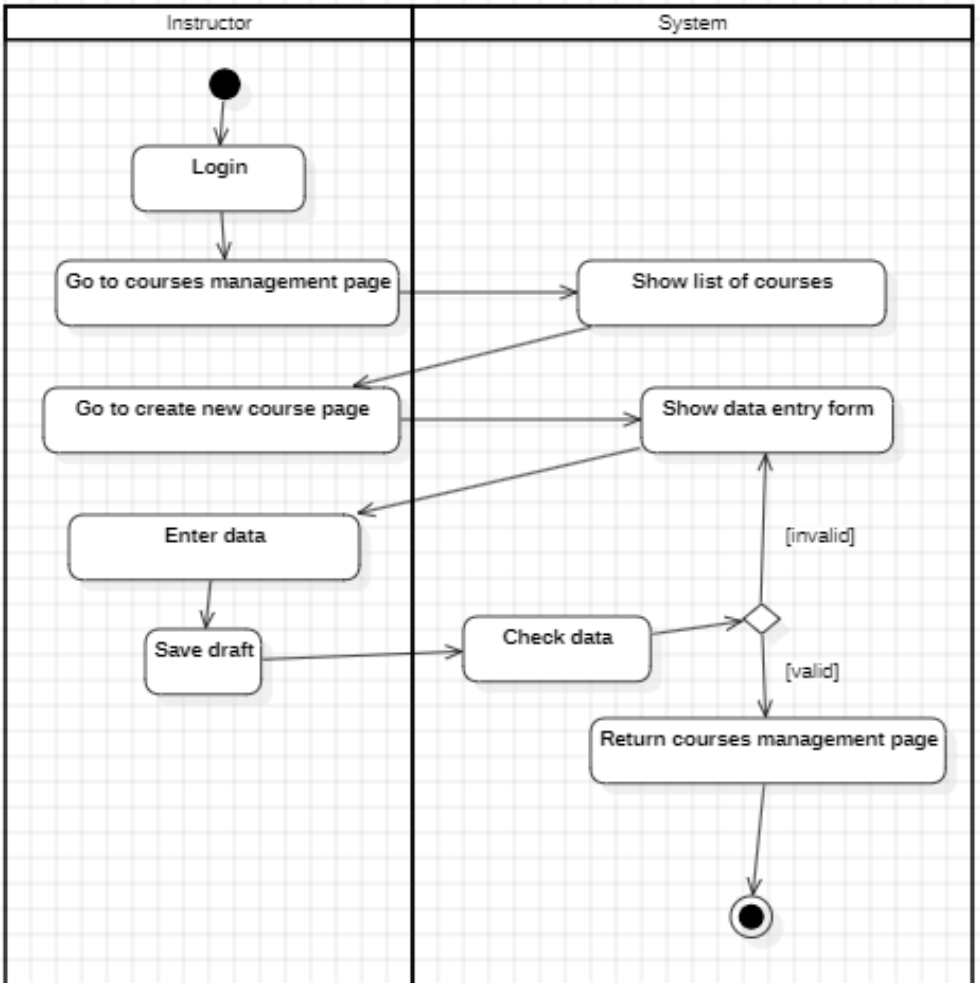


Figure 14. Create course activity diagram

Figure 15 presents diagram shows the process for an instructor to update a course in an e-learning system. The key steps are:

Instructor:
- Logs in to the system.
- Goes to the "Courses Management" page.

System:
- Shows the list of courses.

Instructor:
- Selects a course to update.

System:
- Shows the data entry form for the selected course.

Instructor:
- Enters the updated course data.

System:
- Checks the data entered by the instructor.
- If the data is valid, updates the course information.
- Shows the updated course information.

The diagram illustrates the flow of actions between the instructor and the system, including the steps for selecting a course, entering/updating course data, and the system validating and updating the course information.

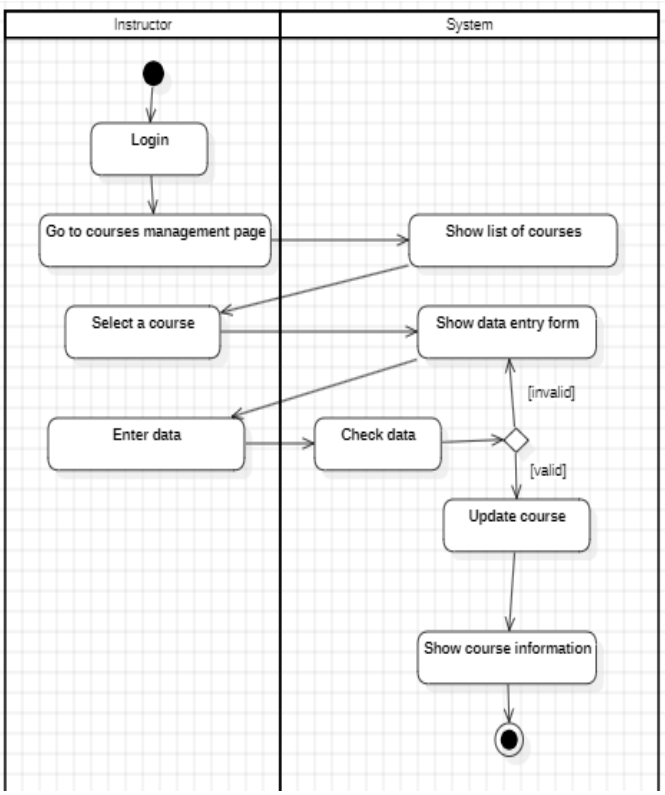


Figure 15. Update course activity diagram

Figure 16 presents diagram shows the process for an instructor to publish a course in an e-learning system. The key steps are:

Instructor:
- Logs in to the system.
- Goes to the "Courses Management" page.
- Selects a course to publish.

System:
- Checks the lectures for the selected course.
- If the lectures are valid, publishes the course.
- If the lectures are invalid, returns an error message.

Instructor:
- Receives the result of the publish operation.
- If the course was published successfully, the process is complete.
- If there was an error, the instructor can make corrections and try publishing the course again.

The diagram illustrates the flow of actions between the instructor and the system, including the steps for selecting a course, the system checking the course lectures, and publishing the course or returning an error message.

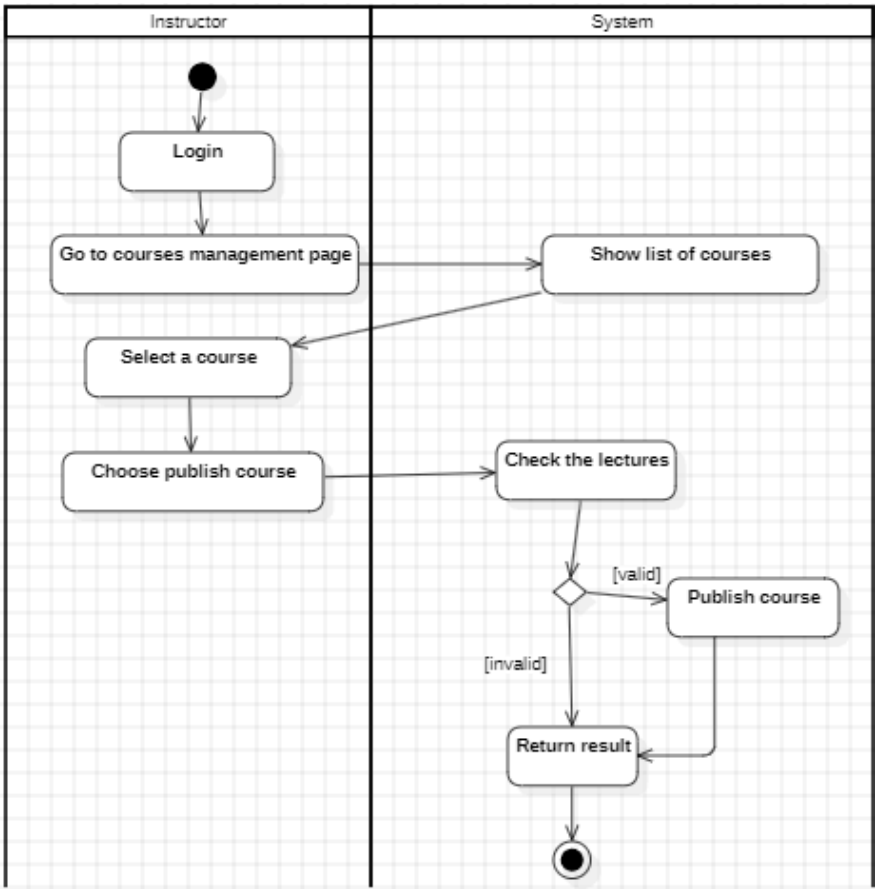


Figure 16. Publish course activity diagram

Figure 17 presents diagram shows the process for a student to interact with a course in an e-learning system, specifically focusing on the steps involved in deleting a course.The key steps are:

Student:

- Logs in to the system.
- Selects a course to view.

System:

- Shows the course details to the student.

Student:

- Watches the course content.

System:

- Checks the student's data.
- If the student has not purchased the course, requires the student to purchase it.
- If the student has purchased the course, shows the course lectures.

Student:

- Purchases the course if required.
- Enters data (e.g. personal information, payment details).
- Reviews the entered data.

System:

- Checks the entered data.
- If the data is valid, updates the student's data.
- If the data is invalid, shows an error message.

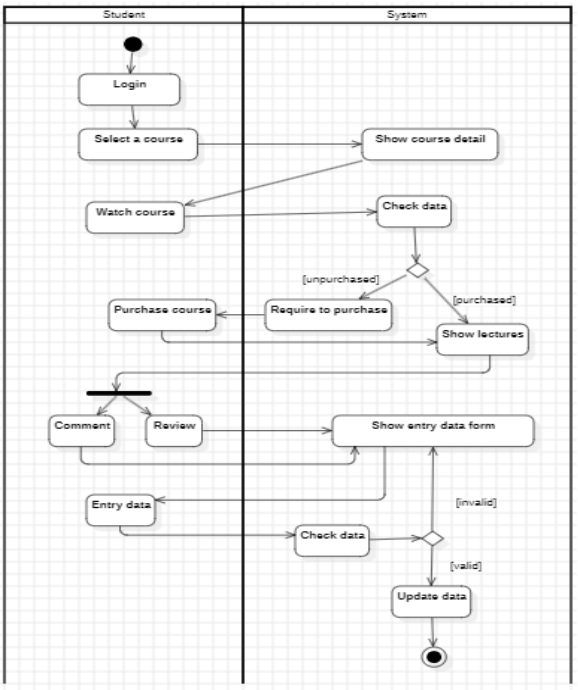


Figure 17. Delete course activity diagram

Figure 18 presents diagram shows the process for an instructor to manage lectures in an e-learning system. The key steps are:

Instructor:
- Logs in to the system.

System:
- Shows the list of users.

Instructor:
- Selects a course to manage.

System:
- Shows the course information.

Instructor:
Can perform the following actions on the lectures:
- Delete.
- Add.
- Update.

Instructor:
- Enters data (e.g. lecture details).

System:
- Checks the entered data.
- If the data is valid, updates the lecture information.
- If the data is invalid, shows an error message.
- Shows the updated lecture information.

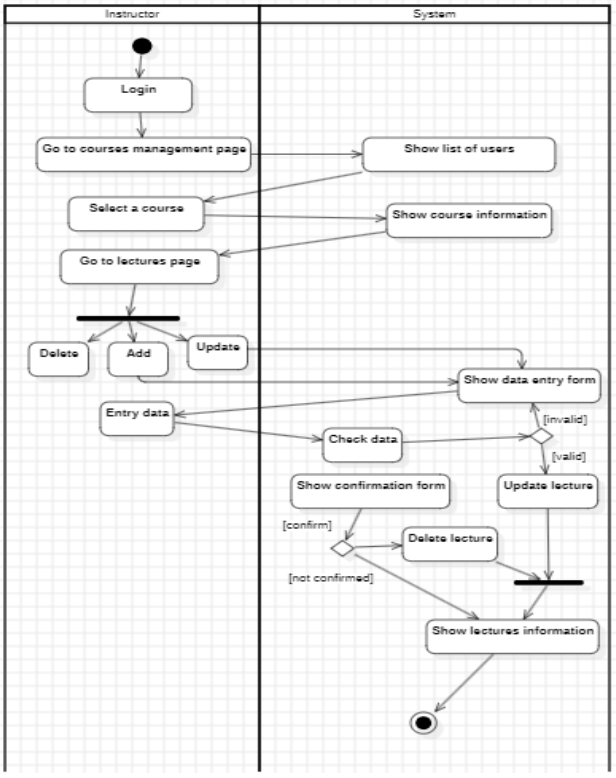


Figure 18. Manage lecture activity diagram

Figure 19 presents diagram shows the process for a student to interact with an e-learning system. The key steps are:

Student:
- Logs in to the system.

System:
- Shows the list of available courses.

Student:
- Selects a course to view.

System:
- Shows a confirmation form for the selected course.

Student:
- Can confirm or not confirm the selection.

System:
- If confirmed, checks the course information.
- If not confirmed, returns the student to the courses management page.
- If the course can be deleted, allows the student to delete the course.
- If the course cannot be deleted, unpublishes the course.
- Returns the student to the courses management page.

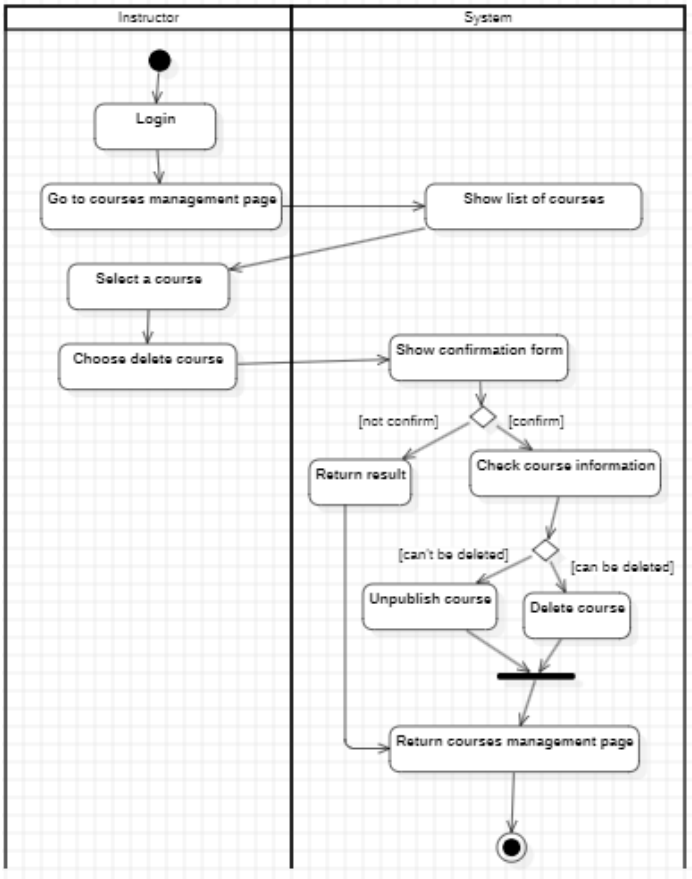


Figure 19. student activity diagram

2.2 Design

2.2.1 User interface design

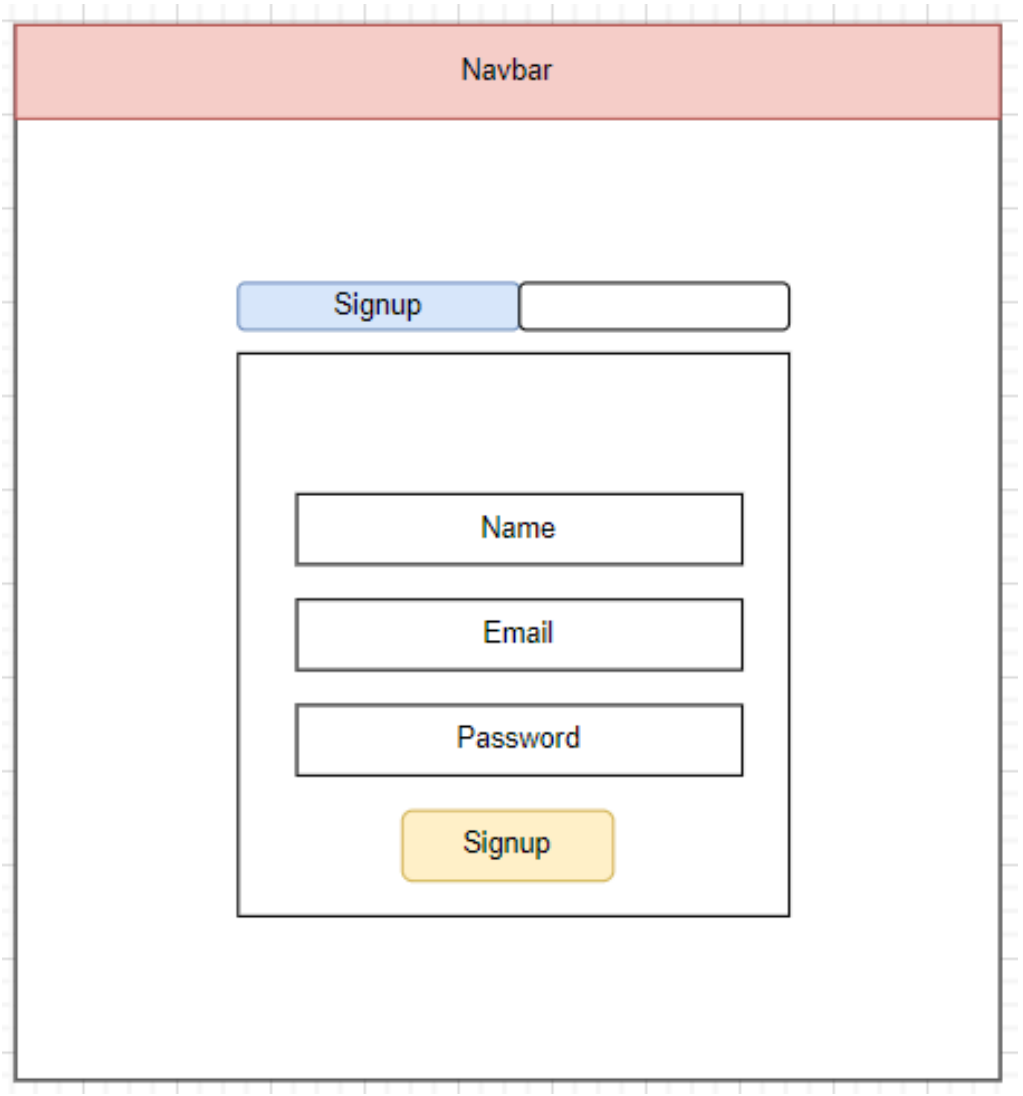


Figure 20. Sign up page UI

Figure 21 displays the user interface (UI) design for the sign-up page of the application.

- **Navbar**: Positioned at the top, it provides navigation options for users to access different sections of the application.
- **Form Fields**: The sign-up form includes fields for entering the user's name, email, and password, ensuring that essential information is collected for account creation.
- **Signup Button**: A prominent button labeled "Signup" allows users to submit their information and complete the registration process.

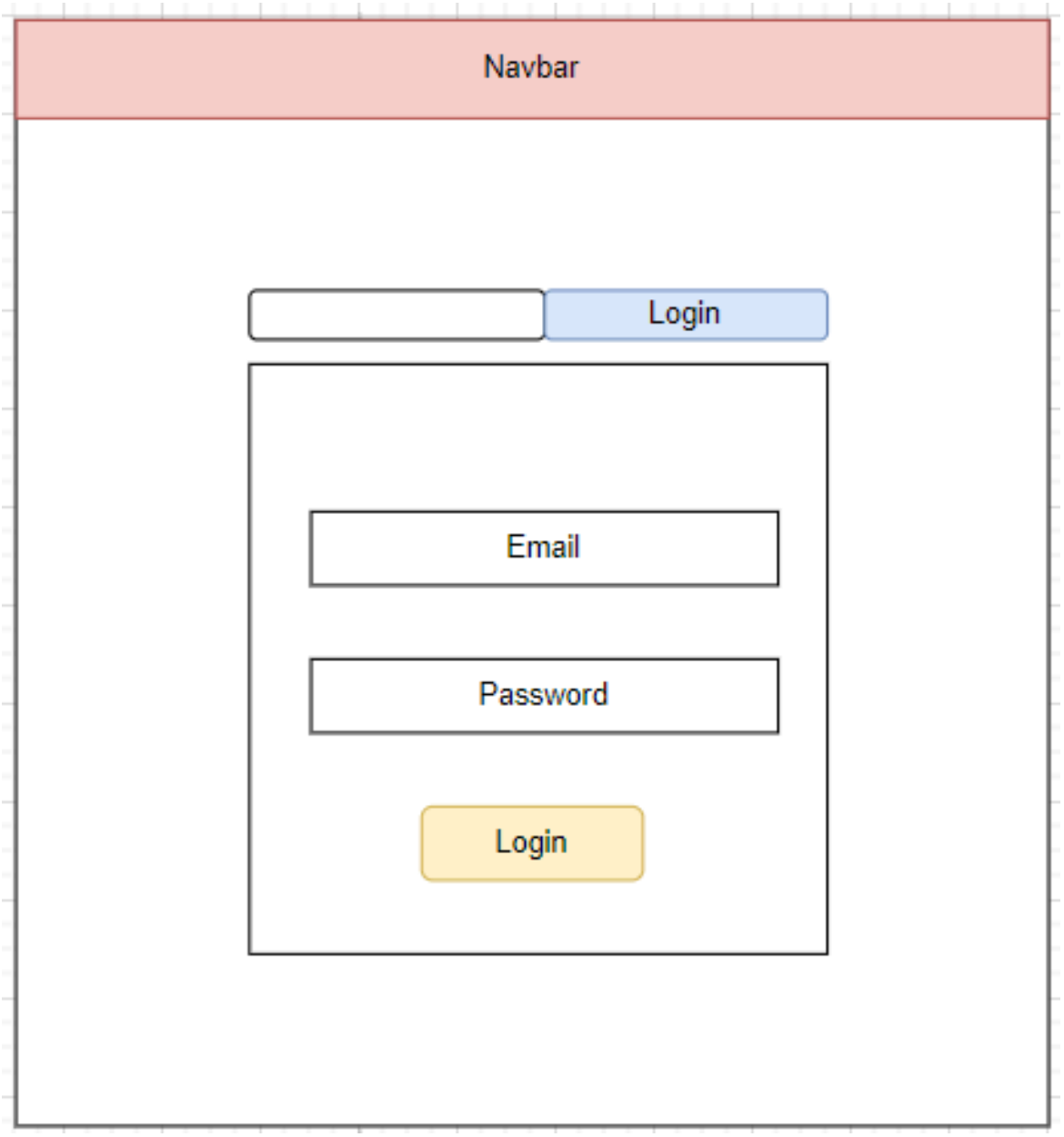


Figure 21. Sign in page UI

Figure 22 illustrates the user interface design for the sign-in page.

- **Navbar**: Located at the top, providing easy navigation to various sections of the application.
- **Form Fields**: Includes fields for entering email and password, essential for user authentication.
- **Login Button**: A prominently displayed button labeled "Login" for users to submit their credentials.

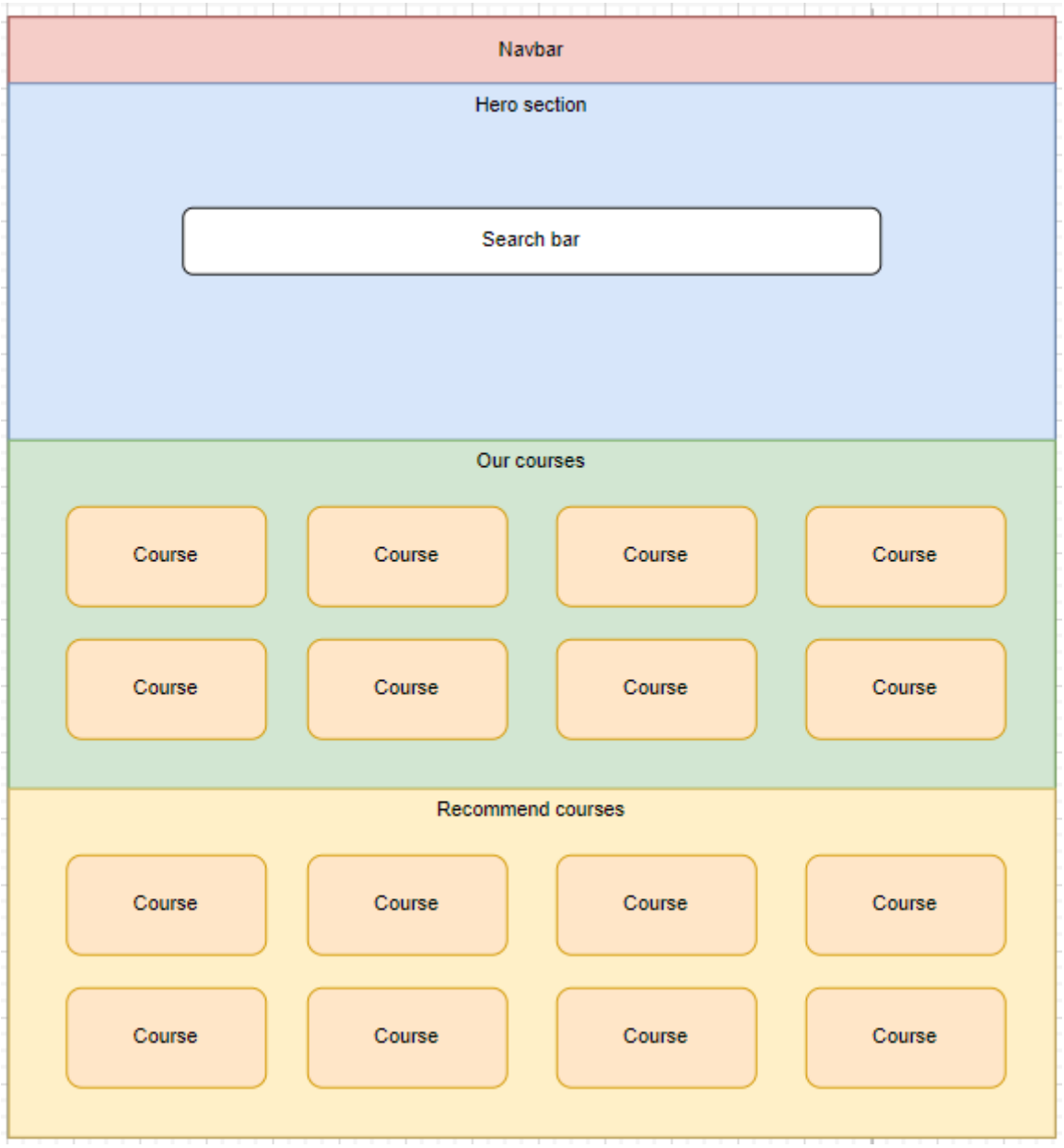


Figure 23. Home page UI

Figure 23 displays the user interface design for the home page.

- **Navbar**: Positioned at the top for easy navigation throughout the application.
- **Hero Section**: A prominent area featuring a search bar, allowing users to quickly find courses.
- **Our Courses**: This section showcases a selection of available courses, displayed in a grid format for easy browsing.
- **Recommended Courses**: A separate area highlighting courses tailored to the user's interests or previous activity.

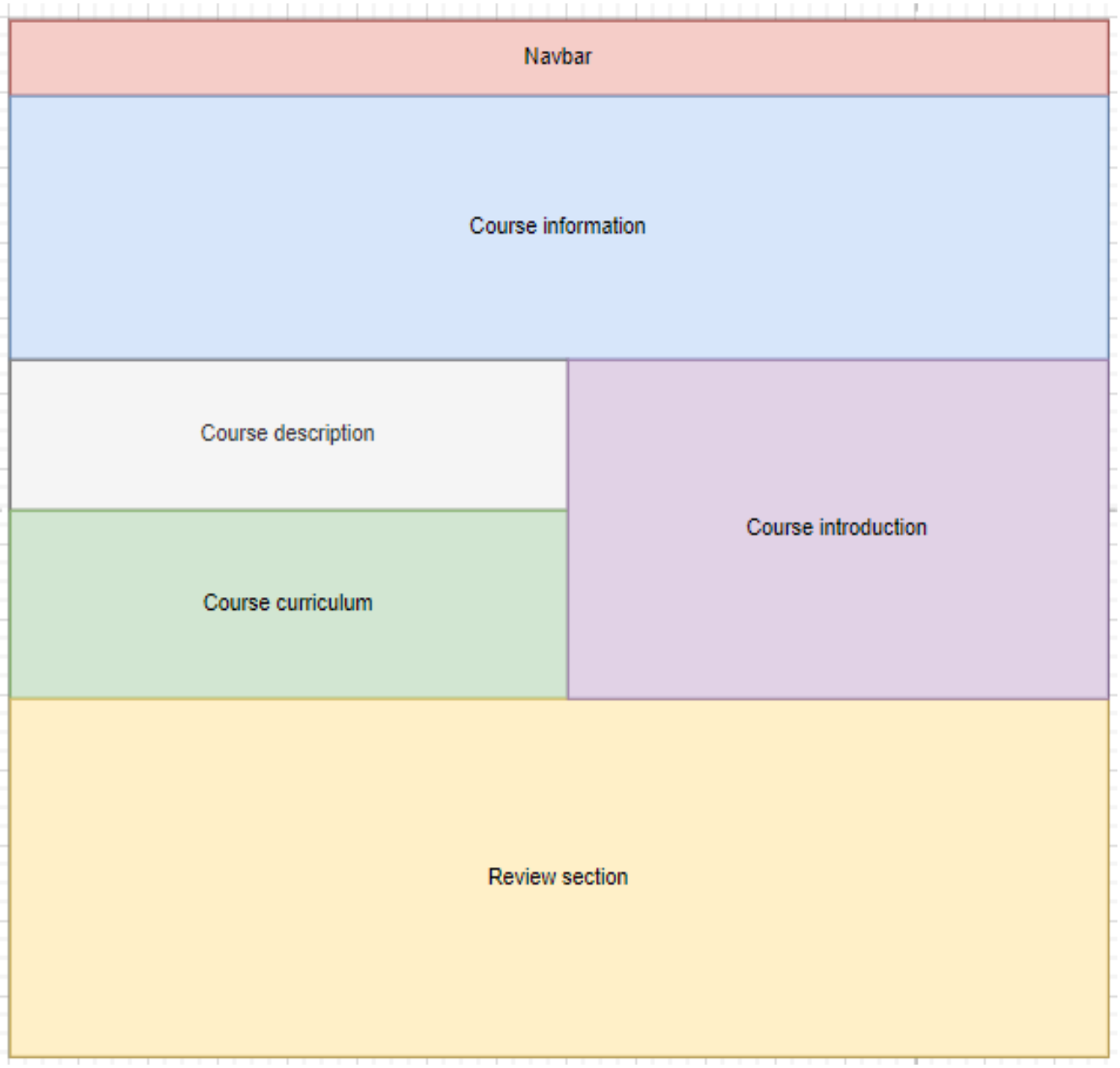


Figure 24. Course detail page UI

Figure 24 presents the user interface design for the course detail page.

- **Navbar**: Located at the top for easy access to other sections of the application.
- **Course Information**: A section that provides essential details about the course.
- **Course Description**: Offers an overview of what the course entails.
- **Course Curriculum**: A detailed outline of the topics and materials covered in the course.
- **Review Section**: Allows users to read and submit reviews about the course.

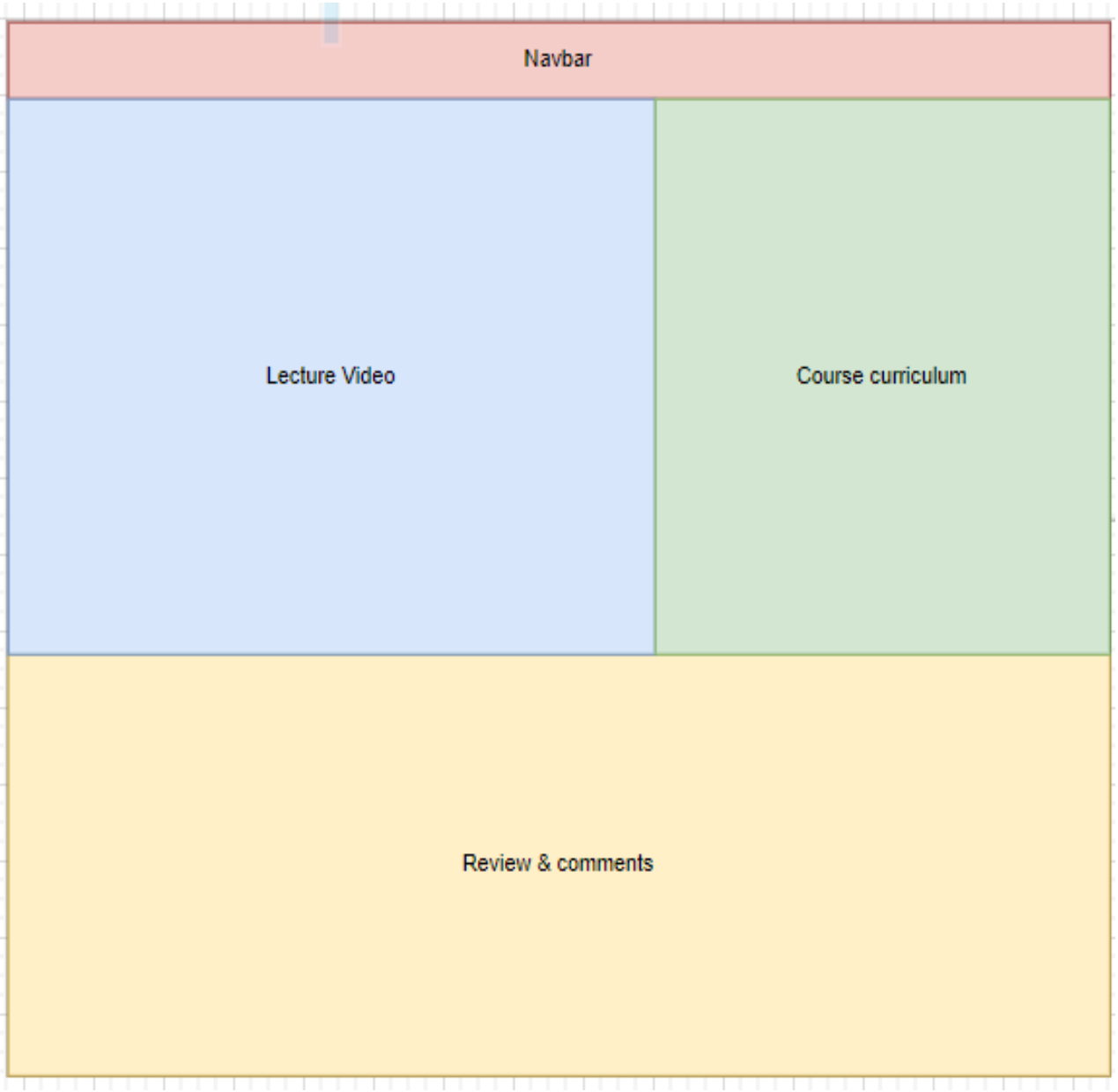


Figure 25. Course progress page UI

Figure 25 illustrates the user interface design for the course progress page.

- **Navbar**: Positioned at the top for streamlined navigation across the application.
- **Lecture Video**: A dedicated section for streaming the current lecture, ensuring easy access to course content.
- **Course Curriculum**: Displays the structure of the course, allowing users to track their progress through the topics.
- **Review & Comments**: A space for users to leave feedback and engage with other learners regarding the course material.

Figure 26. Dashboard page UI

Figure 26 showcases the user interface design for the dashboard page.

- **Navbar**: Located at the top for easy access to different sections of the application.
- **Sidebar**: Positioned on the left for navigation through various dashboard options.
- **Dashboard**: The main area where key metrics and data are displayed.
- **Revenue Statistics Section**: Provides a summary of revenue-related data, helping users to quickly understand financial performance.
- **Revenue Chart**: A visual representation of revenue trends, enhancing data comprehension and analysis.

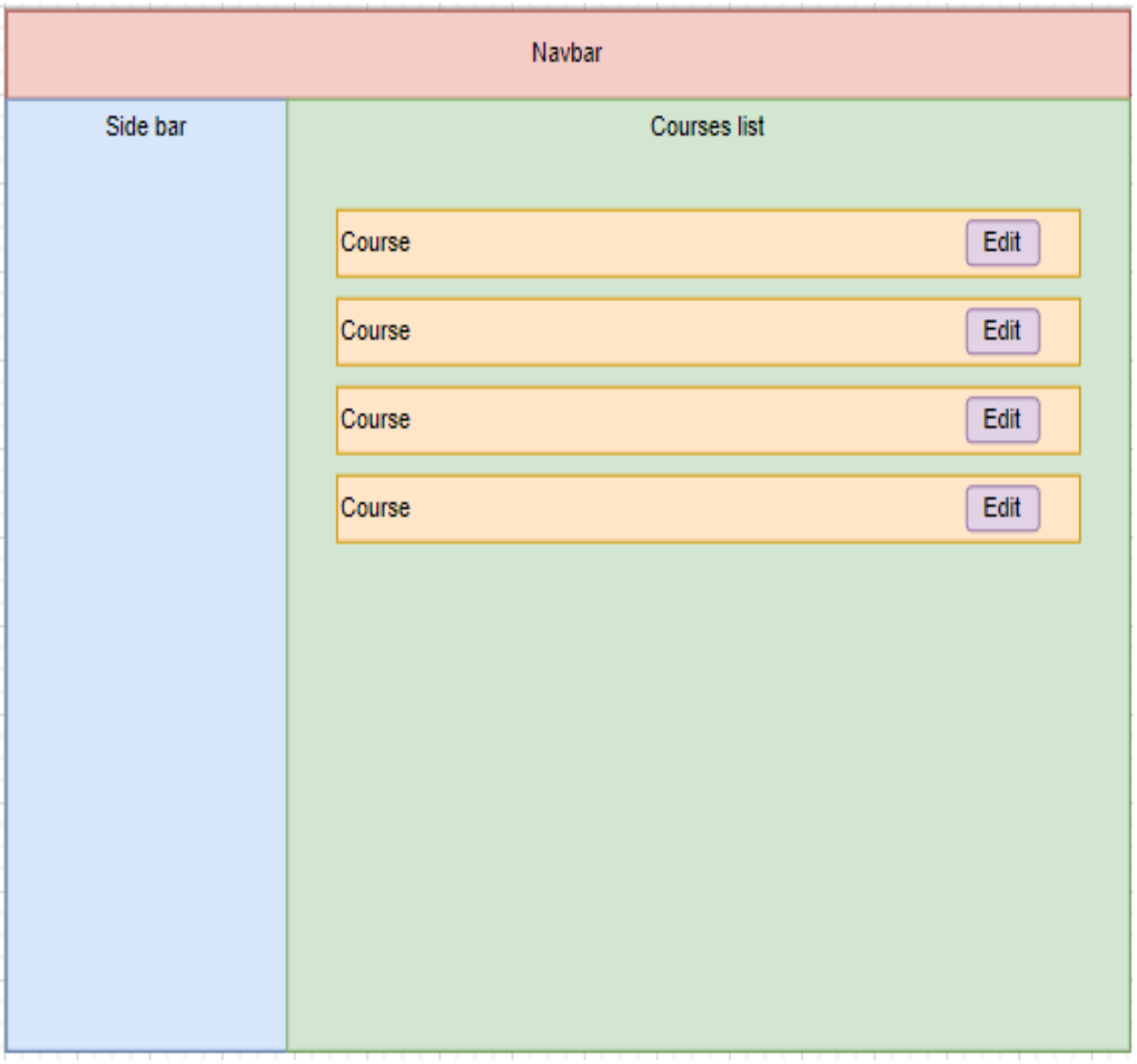


Figure 27. Course management page UI

Figure 27 presents the user interface design for the course management page.

- **Navbar**: Positioned at the top, providing access to various sections of the application.
- **Sidebar**: Located on the left for easy navigation through different management options.
- **Courses List**: Displays a list of courses currently managed within the system. Each course entry includes an "Edit" button for quick modifications.

CHAPTER 3. INSTALLATION AND RESULT

3.1 Installing

The system is built in the integrated development environment (IDE) Visual Studio Code. To ensure objectivity in comparison, all algorithms were tested on the same environment, Windows 10 operating system, with an Intel Core i7-9700K processor and 16GB DDR4 RAM.

3.1.1. Development tools

The project leverages a robust technology stack designed for optimal performance and scalability. On the frontend, we utilize ReactJS, complemented by HTML, CSS, and JavaScript, to create a dynamic and responsive user interface. For the backend, we employ Node.js in conjunction with the Express framework, ensuring a streamlined and efficient server-side experience. Data management is facilitated through MongoDB, allowing for flexible and scalable data storage solutions. Communication between the frontend and backend is achieved via a RESTful API, which enhances data exchange and integration capabilities. Additionally, we utilize Postman for comprehensive API testing and Git for effective source code management and version control.

3.1.2. Algorithms and technical solutions

The system implements user authentication using the JWT (JSON Web Token) mechanism to ensure secure user login. For course and catalog management, database queries are optimized through data filters and pagination. The user interface is enhanced using the ReactJS library, providing a friendly and smooth experience, while responsive design ensures optimal display on both desktop and mobile devices. Data management is facilitated by integrating MongoDB for quick information storage and retrieval. Additionally, APIs are built to perform CRUD operations (Create, Read, Update, Delete) for courses and catalogs.

3.2 Experimental results

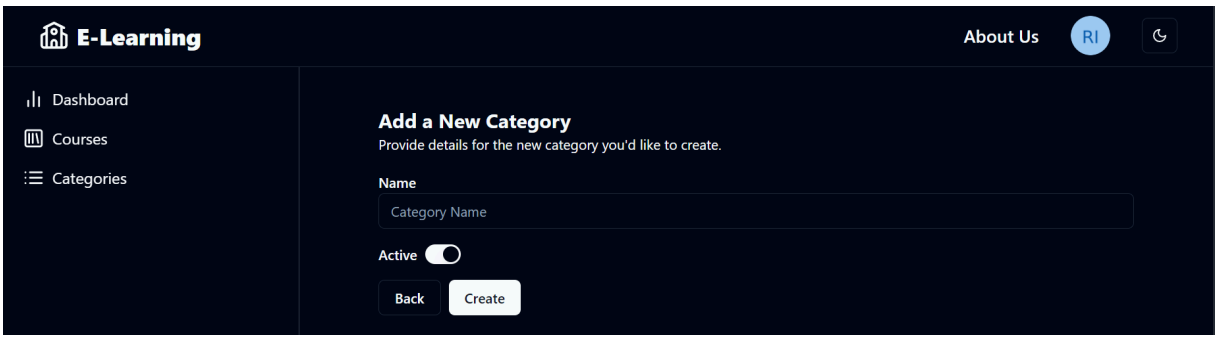


Figure 28. Add new category screen

Category creation screen includes category name and display status

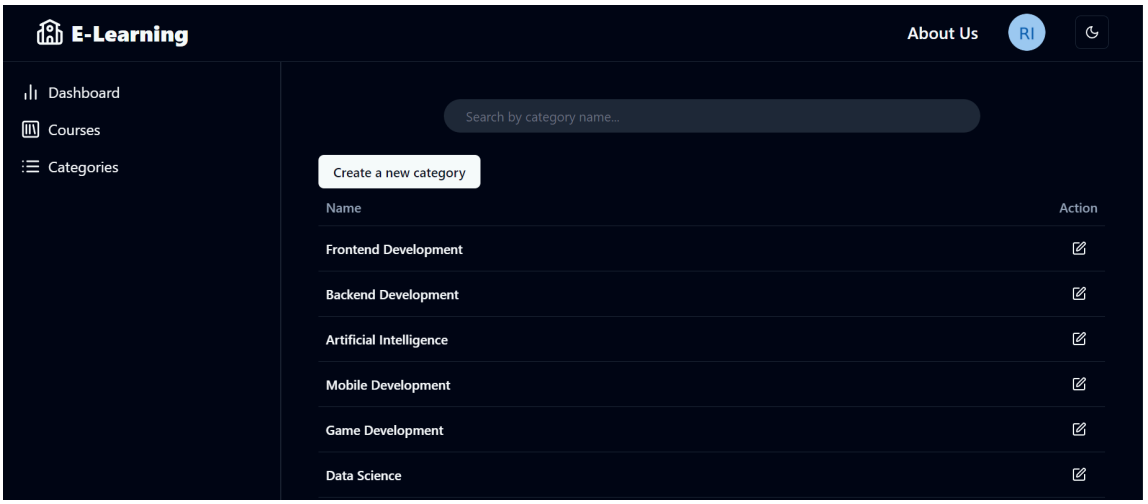


Figure 29. Category management screen

The category management screen includes a list of categories and function buttons to create and edit categories

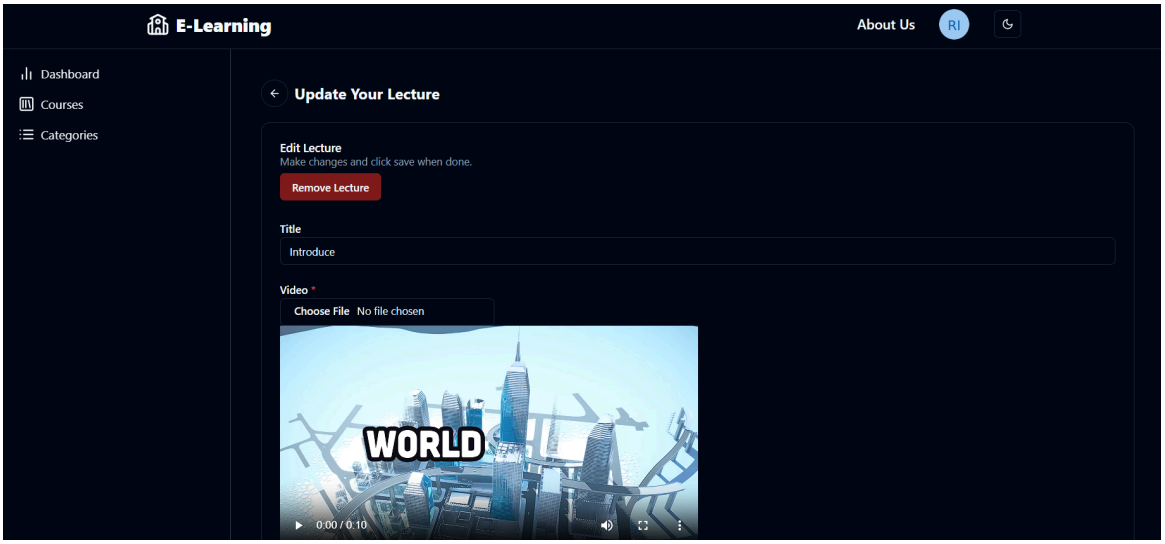


Figure 30. Lecture editing screen

The lecture editing screen includes the lecture name, lecture video and the lecture delete function button.

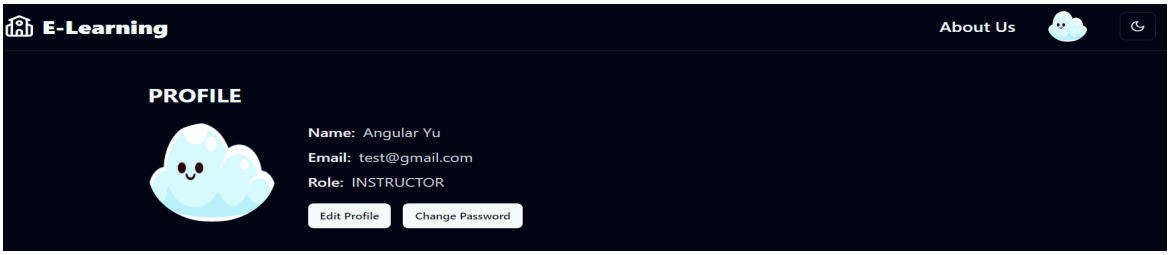


Figure 31. Account management screen

The account management screen includes 2 option buttons: "Edit profile" and "Change password".

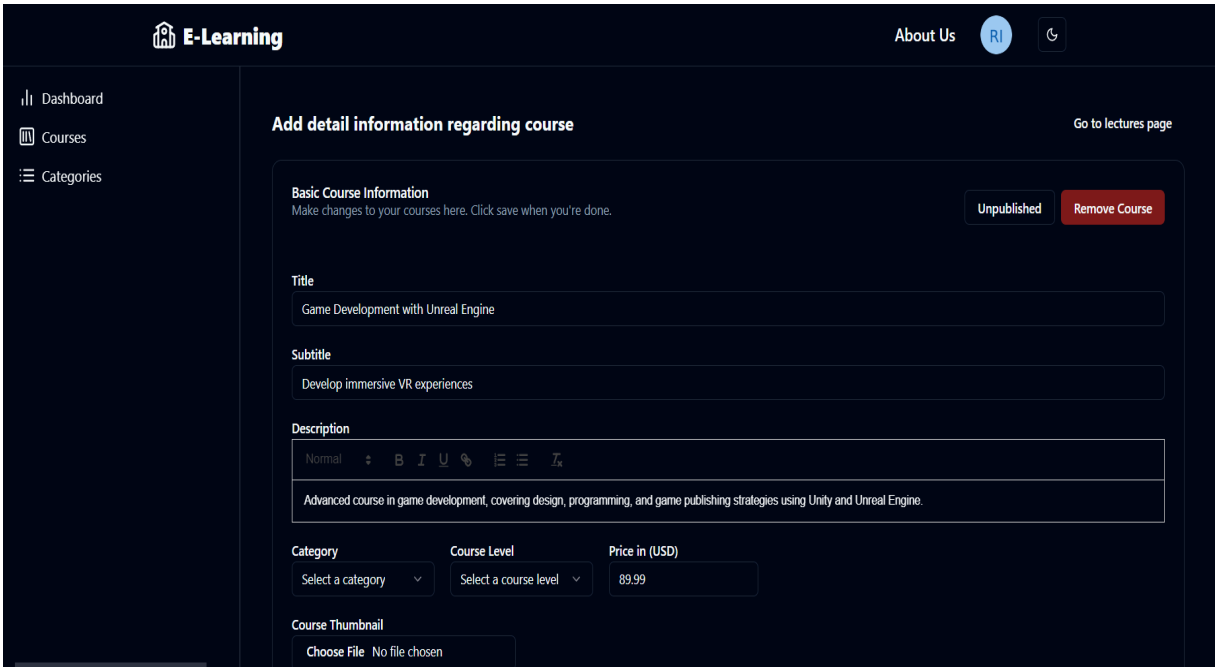


Figure 32. Course information editing screen

Course information editing screen enables instructors to input essential details such as the course title, subtitle, and a comprehensive description. Additionally, users can specify the course level and upload a thumbnail image, enhancing the course's presentation. This screen is designed for efficient course management, ensuring that critical information is easily accessible and editable.

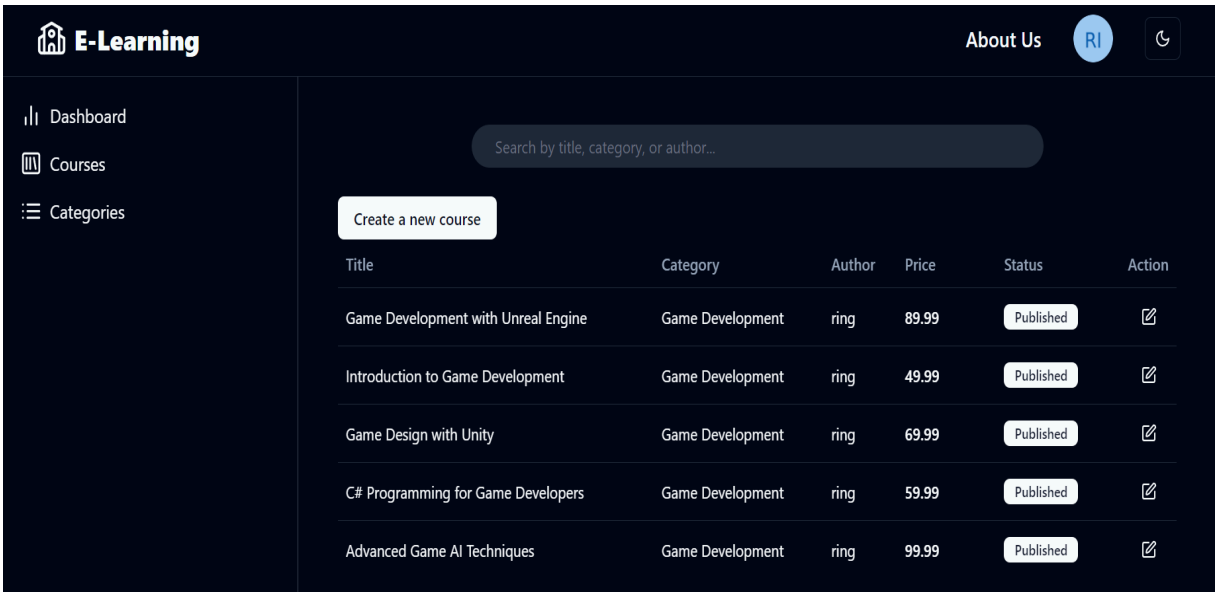


Figure 33. Course information editing screen

Course information editing screen includes course name, description, category type, difficulty, price and thumbnail image.

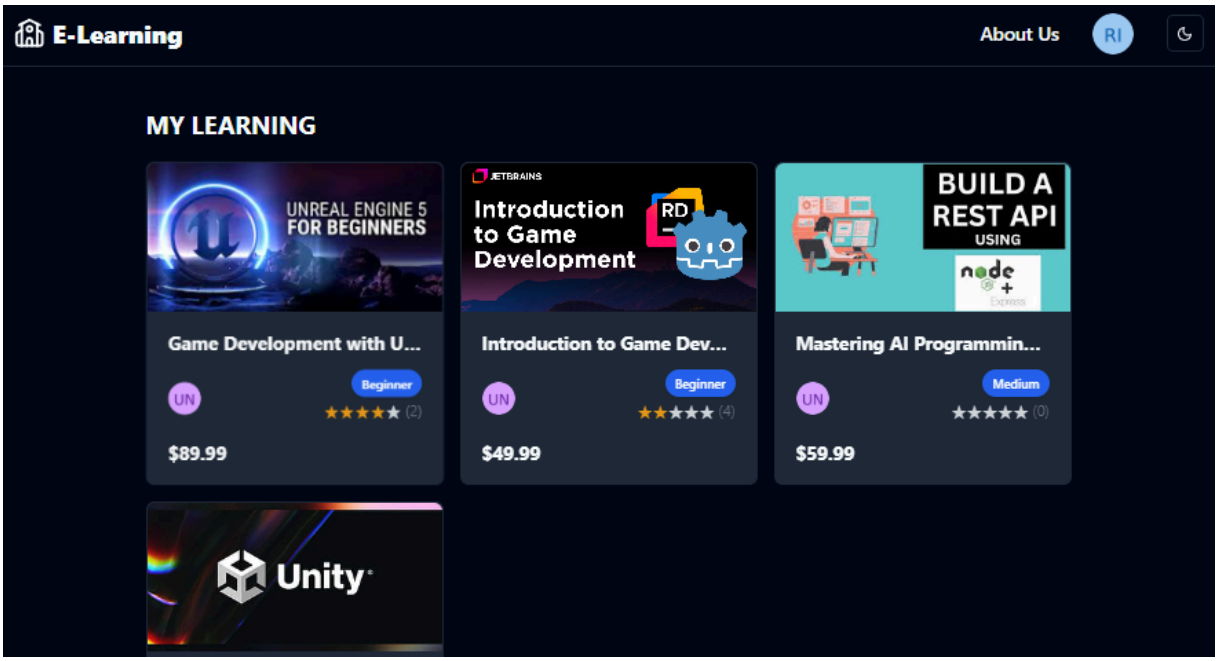


Figure 34. My Learning screen

The screen list of the courses purchased. After the user completes payment, the course will be displayed in My Learning. Upon completing payment for a course, it is automatically displayed in this section, allowing users to easily access and manage their enrolled courses.

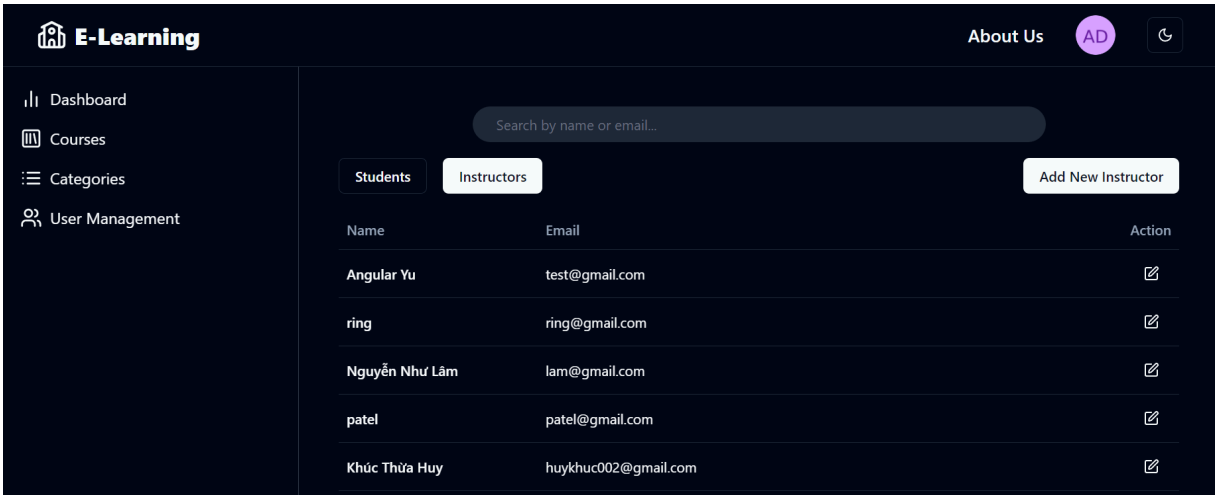


Figure 35. Manage all user accounts screen

Screen to manage all user accounts on the system including students and lecturers from the Administrator role and the add lecturer account button. This

interface is designed for administrators to effectively oversee and manage all user accounts, including both students and instructors.

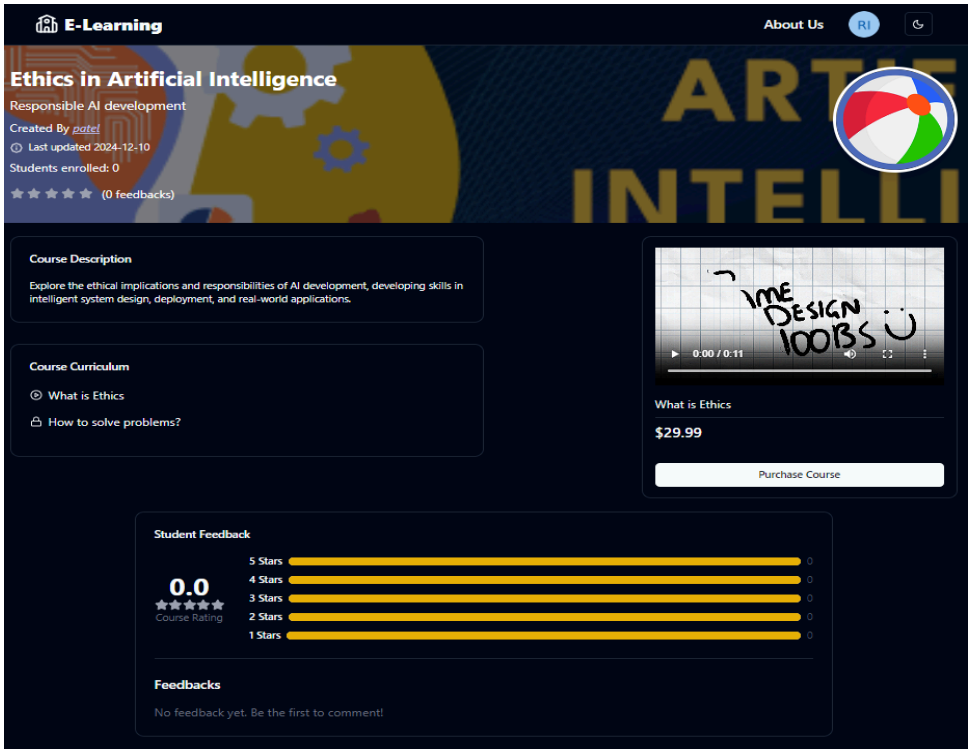


Figure 36. Course details screen

The course details screen includes the course name, description, list of video lectures, reviews and the option to purchase the course.

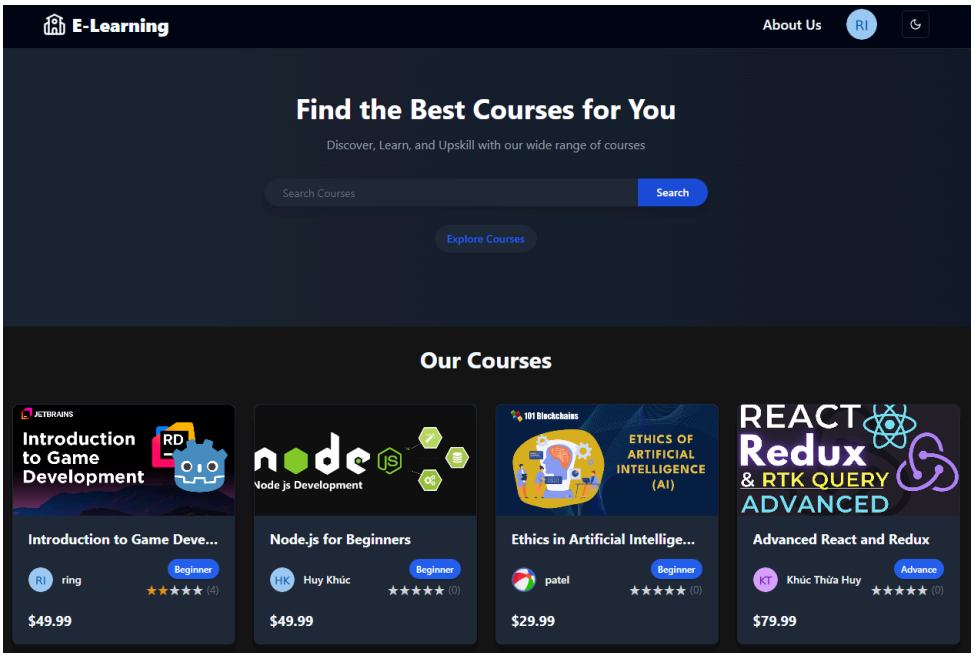


Figure 37. Home screen

The home screen displays available courses on the system and recommended courses to users based on learning needs and course quality.

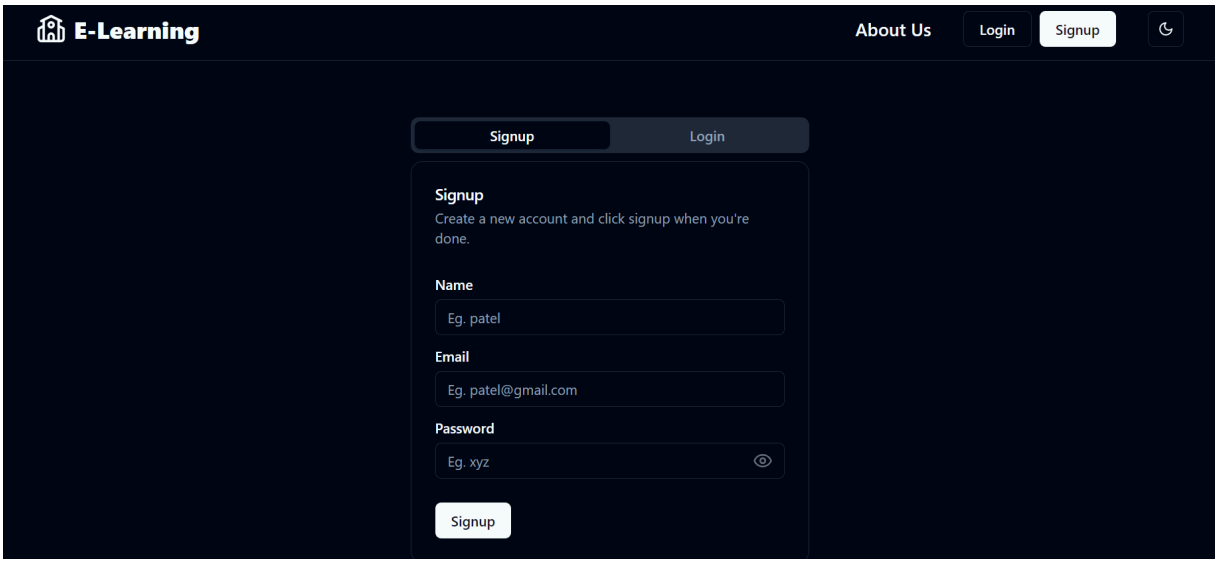


Figure 38. Registration screen

The account registration screen includes account name, email and password. Signup button to submit signup form. Each field is clearly labeled to guide users through the registration process. Once all required information is entered, users can click the "Signup" button to submit their registration form.

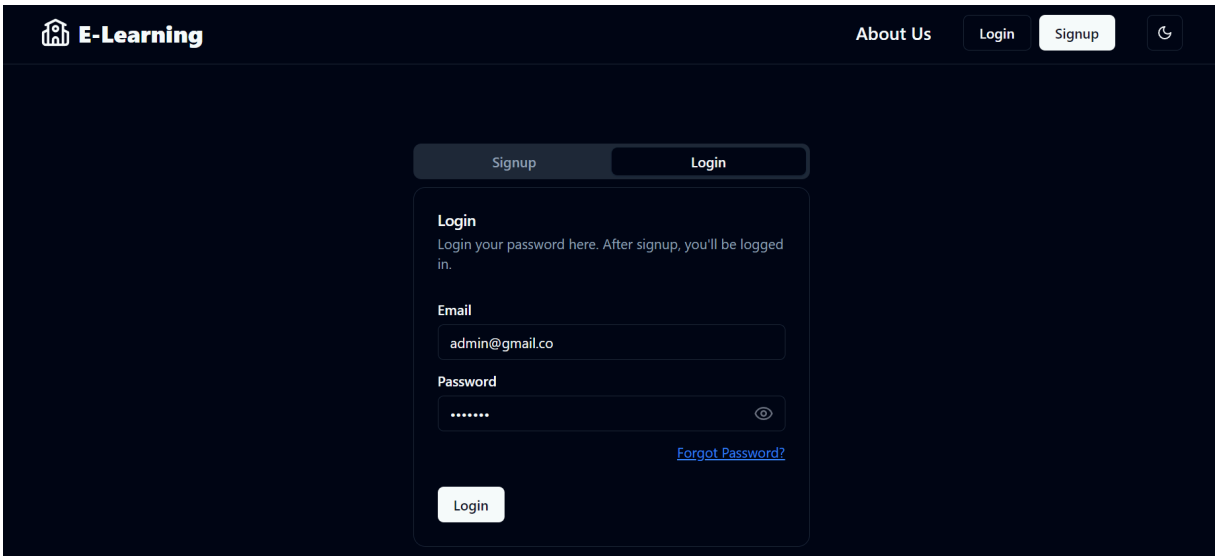


Figure 39. Login screen

Account login screen includes an email and password input field. Login button to submit login form. Once the information is filled in, users can click the "Login" button to submit their credentials and gain access to their accounts. This

simple and intuitive layout ensures a smooth login experience, allowing users to quickly connect to their learning resources and continue their educational activities without unnecessary complications.

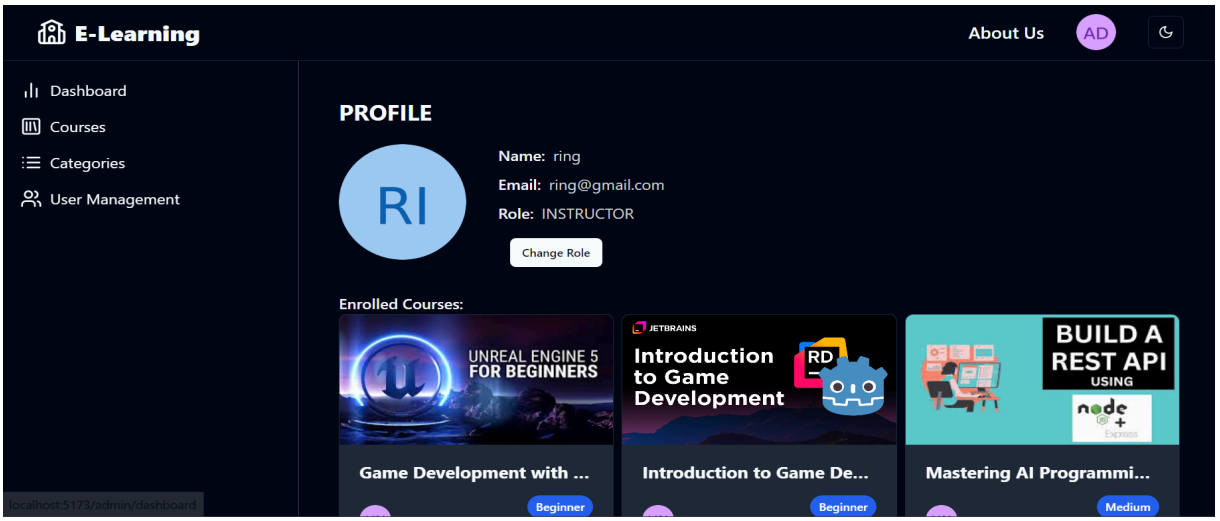


Figure 40. View user information screen

Screen to view user information on the system from the Administrator role including name, email, role and button to change user role. The layout is designed for clarity and ease of use, enabling administrators to quickly view and update user information. This functionality is crucial for maintaining an organized and effective learning environment, ensuring that users have appropriate access and capabilities based on their roles.

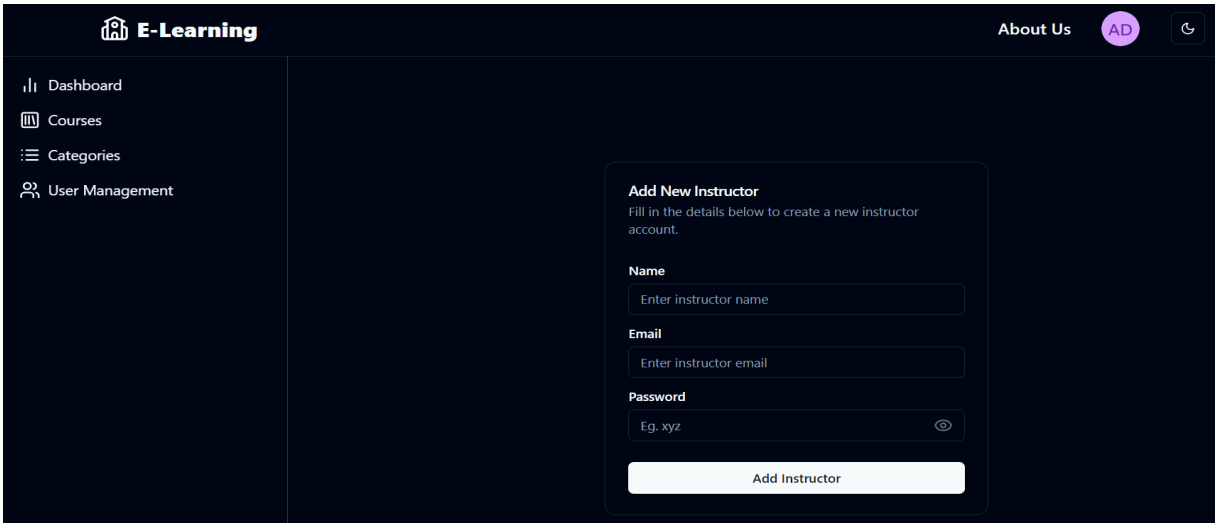


Figure 41. Create instructors account screen

Screen to create an instructor account from the Administrator role including account name, email and password. This interface is designed for administrators to set up new instructor accounts easily.

CHAPTER 4. SYSTEM TESTING

4.1. Test Planning

4.1.1. Define Testing Strategy

The testing types used in this project include:

- Functional Testing
- Non-functional Testing, including:
    + Performance Testing
    + Usability Testing
    + Compatibility Testing

4.1.2. Set Up Test Environment

The test environment includes the following hardware and software:

- Device: Laptop
- Browser: Google Chrome, version 124.0.6367.62
- System Access: Project Website

4.1.3. Test Data

Accounts for testing include:

- Admin: admin@gmail.com / *****
- Student: thuahuy@gmail.com / *****
- Instructor: ring@gmail.com / *****

4.2. Test Case Design

4.2.1. Test case for Student

Test	What is being tested	How	Test data	Expected Results
1	Register new account	Enter sample data and evaluate results	Name: Email: test@gmail.com Pass:	Missing field error message
2	Register new account	Enter sample data and evaluate results	Name: Huy Email: pvhuy@gmail.com Pass: ********	Error message account already exists
3	Register new account	Enter sample data and evaluate results	Name: Huy Email: kthuy@gmail.com Pass: ********	Success

4	Login	Enter sample data and evaluate results	Email: kthuy@gmail.com Pass: ********	Missing field error message
5	Login	Enter sample data and evaluate results	Email: test@gmail.com Pass: ********	Wrong account information error message
6	Login	Enter sample data and evaluate results	Email: kthuy@gmail.com Pass: ********	Success
7	Search courses by name	Enter sample data and evaluate results	"Java"	Success
8	Search courses by description	Enter sample data and evaluate results	"Angular Yu"	Success
9	Search for a non-existent course	Enter sample data and evaluate results	"Key does not exist"	Shows no available courses
10	View registered course information	Click on My Learning	User ID	Success
11	View purchased video lectures	Click to view course lectures	User ID, Course ID	Success
12	View unpurchased video lectures	Click to view course lectures	User ID, Course ID	Error message not purchased course

4.2.2. Test case for Instructor

Test	What is being tested	How	Test data	Expected Results
1	Create new course	Enter sample data and evaluate results	Missing field	Missing field error message
2	Create new course	Enter sample data and evaluate results	Course already exists	Error message course already exists
3	Create new course	Enter sample data and evaluate results	Correct information	Success
4	Update course	Enter sample data	Missing field	Missing field

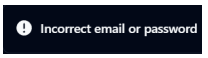
		and evaluate results		error message
5	Update course	Enter sample data and evaluate results	Correct information	Success
6	Delete course with no students enrolled	Select the course and click delete	Course ID	Success
7	Delete course with students enrolled	Select the course and click delete	Course ID	Course cannot be deleted message

4.2.3. Test case for Admin

Test	What is being tested	How	Test data	Expected Results
1	User authorization	Select user and select role	User ID Role: Student	Success
2	User authorization	Select user and select role	User ID Role: Instructor	Success
3	Create instructor account	Enter sample data and evaluate results	Name: John Doe Email: jdoe@gmail.com Pass: ********	Success
4	Create instructor account	Enter sample data and evaluate results	Name: John Doe Email: Pass:	Missing field error message
5	Create instructor account	Enter sample data and evaluate results	Name: Huy Email: pvhuy@gmail.com Pass: ********	Error message account already exists
6	Create new category	Enter sample data and evaluate results	“Block Chain”	Success
7	Create new category	Enter sample data and evaluate results	“ ”	Missing field error message
8	Update category	Enter sample data and evaluate results	Category ID	Success

4.3. Test Results

4.3.1. Test log for Student

Test	What is being tested	Test data	Expected Results	Acture result	Action taken
1	Register new account	Name: Email: test@gmail.com Pass:	Missing field error message		None
2	Register new account	Name: Huy Email: pvhuy@gmail.com Pass: ********	Error message account already exists		None
3	Register new account	Name: Huy Email: kthuy@gmail.com Pass: ********	Success	Pass	None
4	Login	Email: kthuy@gmail.com Pass: ********	Missing field error message		None
5	Login	Email: test@gmail.com Pass: ********	Wrong account information error message		None
6	Login	Email: kthuy@gmail.com Pass: ********	Success	Pass	None
7	Search courses by name	"Java"	Success	Pass	None
8	Search courses by description	"Angular Yu"	Success	No results	Recode Test again
9	Search for a non-existent course	"Key does not exist"	Shows no available courses		None
10	View registered course information	User ID	Success	Pass	None
11	View purchased video lectures	User ID, Course ID	Success	Pass	None

12	View unpurchased video lectures	User ID, Course ID	Error message not purchased course	No error message displayed	Recode Test again

4.3.2. Test log for Instructor

Test	What is being tested	Test data	Expected Results	Acture result	Action taken
1	Create new course	Missing field	Missing field error message		None
2	Create new course	Course already exists	Error message course already exists	No error message displayed	Recode Test again
3	Create new course	Correct information	Success	Pass	None
4	Update course	Missing field	Missing field error message		None
5	Update course	Correct information	Success	Pass	None
6	Delete course with no students enrolled	Course ID	Success	Pass	None
7	Delete course with students enrolled	Course ID	Course cannot be deleted message		None

4.3.3. Test log for Admin

Test	What is being tested	Test data	Expected Results	Acture result	Action taken
1	User authorization	User ID Role: Student	Success	Pass	None
2	User authorization	User ID Role: Instructor	Success	Pass	None
3	Create instructor account	Name: John Doe Email: jdoe@gmail.com Pass: ********	Success	Pass	None
4	Create instructor account	Name: John Doe Email: Pass:	Missing field error message		None
5	Create instructor account	Name: Huy Email: pvhuy@gmail.com Pass: ********	Error message account already exists		None
6	Create new category	“Block Chain”	Success	Pass	None
7	Create new category	“ ”	Missing field error message		None
8	Update category	Category ID	Success	No results	Recode Test again

CONCLUSIONS AND DEVELOPMENT DIRECTION

1. Conclusions

After the research and development process, the project has achieved the following results:

- Building a course management system with the main functions**:**
  + Safe and effective user authentication.
  + Managing the course catalog in a clear and easy-to-use manner.
  + Allowing users to interact with the course (view, search, study).
- Applying modern technology**:**
  + ReactJS and Node.js help the system operate quickly, with a smooth interface.
  + MongoDB database ensures stable information querying and storage.
- The system operates as required, ensuring to meet the needs of users.
  + The project has completed the initial goals and has the potential to be applied in practice in learning management systems.

2. Development direction

- Online payment integration**:**
  + Allow users to pay when registering for paid courses.
- Develop course review and comment functions**:**
  + Users can share reviews and comments directly.
- Machine Learning application**:**
  + Suggest suitable courses based on users' learning history.
  + Analyze user behavior to improve learning experience.
- Optimize system performance**:**
  + Improve data access and processing speed.
  + Deploy the system on cloud computing platforms such as AWS, Azure or Google Cloud.

- REFERENCES

[1] Bateman, David, et al. Full-Stack React Projects: Modern Web Development Using React, Node, Express, and MongoDB. Packt Publishing, 2020.

[2] Sharma, P., & Sharma, S. Developing e-Learning Systems with Modern Web Technologies. Springer, 2021.

[3] Wasson, Mike. Designing and Building Scalable Web Applications with Node.js and React. Packt Publishing, 2020.

[4] Vyas, Vishal. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Apress, 2019.