

AMMAN ARAB UNIVERSITY

جامعة عمان العربية

College of Information Technology

كلية تكنولوجيا المعلومات



جامعة عمان العربية  
AMMAN ARAB UNIVERSITY

كلية تكنولوجيا المعلومات  
College of Information Technology

**Title of The project: UniVerse**

*Student*

**Name: Fayez Hasan Abu Hajar**

**ID: 202220628**

*Supervisor*

**Dr.Mohammad Shehab**

***A project report submitted in partial fulfillment of the requirements***

***for B.Sc. degree in Software Engineering***

***Amman - Jordan***

***2024/2025***

## Acknowledgements

First and foremost, I would like to express sincere gratitude to Dr. Mohammad Shehab for his continuous help, support, and insights throughout the length of this project. His encouragement and expertise have helped me a lot in completing this work successfully. My humble gratitude also goes to all faculty members and the administrative staff at the Amman Arab University for their support in making possible all resources that contributed to the running of this project. Special thanks go to students and instructors who gave interviews and provided essential feedback in the requirement-gathering phase. To my family and friends, I appreciate your support, patience, and encouragement throughout the entire process. Your belief in me has not ceased to motivate me. Finally, but indeed not the least, I would like to thank everyone who directly or indirectly played a role in the success of this project. This research is the final product of collaborative effort and shared commitment towards academic excellence.

## Abstract

UniVerse is a web-based learning platform developed to enhance student success in university exams by leveraging peer-to-peer instruction. Traditional learning support methods often rely on formal tutoring, which may not always address individual learning needs or provide relatable guidance. UniVerse addresses this gap by enabling outstanding students—those who have demonstrated excellence in specific subjects—to become peer instructors. The platform allows students to access personalized help from peers who have excelled in the same courses, promoting practical understanding, real-exam preparation, and academic collaboration. Core users include students, instructors, and administrators. The system offers features such as user registration, instructor application workflows, course creation, and session management. The results demonstrate that UniVerse can provide a scalable and effective alternative to traditional academic support, fostering a community-driven approach to educational success.

## Contents

Acknowledgements .....	2
Abstract .....	2
<b>Chapter 1: Introduction .....</b>	<b>5</b>
1.1 Overview .....	5
1.2 Problem Statement.....	5
1.3 Project Objectives.....	6
1.4 Research Strategy.....	7
1.5 Scope.....	8
Project Justification:.....	8
Project Characteristics and Requirements:.....	9
Project Management-Related Deliverables: .....	9
Product-Related Deliverables:.....	9
1.6 Gant chart .....	9
1.7 Project Outline.....	10
• Chapter One: Introduction .....	10
• Chapter Two: Literature Review.....	10
• Chapter Three: Methodology.....	11
• Chapter Four: System Design.....	11
• Chapter Five: Implementation and Testing.....	11
• Chapter Six: Conclusion and Future Work .....	11
<b>CHAPTER 2: Literature Review.....</b>	<b>12</b>
2.1 Overview .....	12
2.2 Related Work .....	12
2.4 Summary.....	15
<b>CHAPTER 3: Methodology.....</b>	<b>17</b>
3.1 Overview .....	17
3.2 Feasibility Study.....	17
3.3 Methodology Process.....	18
3.3.1 Requirements.....	19
<b>Chapter 4: System Design .....</b>	<b>22</b>
4.1 Overview .....	22
4.2 Context Diagram.....	22

4.3 Use Case Diagram .....	23
4.4 Entity-Relationship Diagram (ERD).....	26
4.4 Sequence Diagram .....	27
<b>Chapter 5: Implementation and Testing .....</b>	<b>31</b>
5.1 Overview .....	31
5.2 Testing methodologies .....	31
5.2.1 Unit Testing Results .....	31
5.2.2 Integration Testing Results .....	32
5.2.3 System Testing Results .....	34
5.2.4 Acceptance System Results.....	36
5.3 Discussion and Evaluation .....	37
<b>Chapter 6: Conclusion and Future Work .....</b>	<b>38</b>
6.1 Overview .....	38
6.2 Summary about the Project .....	38
6.3 Achieved Objectives .....	38
6.4 Main contributions of the work .....	39
6.5 Limitation .....	39
6.6 Future Work.....	39
<b>References: .....</b>	<b>40</b>

# Chapter 1: Introduction

## 1.1 Overview

**UniVerse** is a web-based educational platform aimed at empowering students in their academic journey by connecting them with qualified peer instructors. Designed primarily for university students, UniVerse facilitates structured, on-demand support for exam preparation and course-related topics. The system targets three main user roles: students, instructors, and administrators, each with distinct capabilities to streamline the teaching-learning process.

The platform allows students to register and request assistance in specific academic subjects. At the same time, academically distinguished students can apply to become instructors by submitting their credentials and undergoing an approval process managed by the administrator. Once approved, instructors can create and manage courses, upload educational content, and interact with students. The system supports scheduling sessions, sending notifications, and tracking student progress. Instructors can also view statistics related to their sessions, feedback, and performance.

Administrators oversee all system operations, including the approval of instructor applications, monitoring active courses, managing users, and ensuring content quality. The platform also features secure authentication, a responsive user interface, and integration with communication tools to ensure seamless interaction.

By combining peer-to-peer learning with administrative control and intuitive design, UniVerse aims to enhance academic success, encourage collaboration, and make quality educational support more accessible to all students.

## 1.2 Problem Statement

In many educational environments, students face significant financial challenges, making it difficult to cover essential expenses like transportation and study materials. At the same time, other students struggle with challenging subjects but cannot afford personalized academic assistance. Traditional tutoring options are often expensive and finding qualified

tutors can be time-consuming and inefficient. These financial pressures and educational gaps leave many students without the support they need to succeed academically.

## 1.3 Project Objectives

The UniVerse platform is designed with the primary objective of addressing key needs across three main user groups: Students (Learners), Student-Instructors (Tutors), and Admins.

- **For Students (Learners):**  
The objective is to provide an accessible, user-friendly platform where students can receive academic support from peer tutors in a cost-effective and timely manner. Students will be able to browse available courses or tutoring sessions, request help in specific subjects, and receive guidance through structured communication tools. The platform aims to minimize academic gaps by connecting students with knowledgeable peers who can provide personalized assistance at no cost or at affordable rates.
- **For Student-Instructors (Tutors):**  
The platform aims to empower academically strong students by giving them the opportunity to offer tutoring services, gain teaching experience, and potentially earn income or recognition. Tutors will have access to tools for managing their sessions, tracking student progress, uploading materials, and receiving ratings and feedback. This structure supports both the academic growth of tutors and their contribution to a collaborative learning environment.
- **For Admins:**  
The objective is to ensure seamless platform operation through effective user management, content moderation, and performance monitoring. Admins will oversee user registration, validate tutor applications, manage reported issues, and ensure that the platform maintains academic integrity. Additionally, they will be responsible for maintaining a fair and safe learning environment by implementing policies, managing notifications, and analyzing usage metrics.

## 1.4 Research Strategy

The research strategy for the UniVerse platform is structured to ensure the development of a reliable, user-centric educational system that responds effectively to the financial and academic needs of students. The process begins with a comprehensive literature review examining existing local and global e-learning platforms, peer tutoring systems, and educational support tools. The goal of this phase is to identify gaps in current solutions, such as affordability, accessibility, and efficiency in peer-based learning environments.

Following the literature review, a Feasibility Study is conducted to evaluate the technical and operational viability of the proposed system. This includes assessing the available development tools, hosting options, database requirements, and the ability of the system to scale with increased users and interactions.

The next step is Requirements Gathering, where input is collected from target users—students, potential peer tutors, and academic administrators—through online surveys, structured interviews, and observational research. These insights help define the system’s essential features and refine the user experience.

Once requirements are finalized, the System Design and Development phase begins. Given the need for responsiveness and adaptability, the team adopted the **Agile Scrum methodology** due to its iterative development, flexibility, and stakeholder collaboration [1]. This will ensure that the system dynamically evolves toward the needs expressed by the users during its construction. Once the development phase is done, testing will start. Testing is built into the development process to guarantee that the system will be executed in accordance with specified requirements and be free from bugs. It contains different levels of testing, some of which overlap with the development phase, allowing the identification and correction of possible problems on time.

The adopted strategy of research for developing the UniVerse is thus carefully structured to see that analysis and implementations are thorough and well applicable. Figure 1.1 below outlines the sequential steps followed by the researchers in developing this system, from the literature review up to system testing.

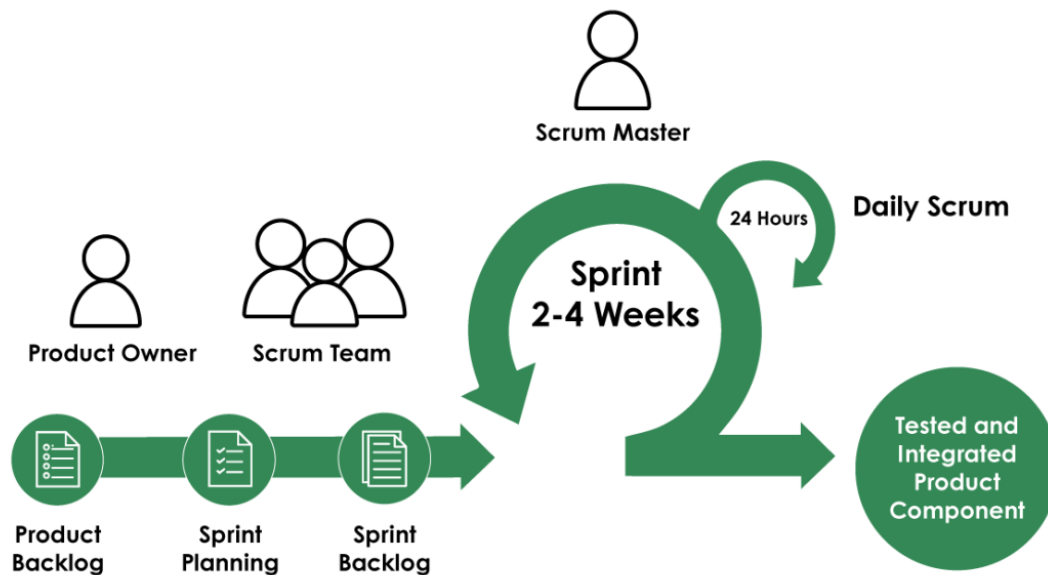


Figure 1.1: Research strategy

As depicted in Figure 1.1, the iterative nature of the Agile Scrum methodology allows for continuous feedback and refinement, ensuring that the system evolves in alignment with stakeholder needs.

## 1.5 Scope

The scope of this project is categorized according to the functionalities available to Students, Instructors, and Admins. It defines what the system will cover and explicitly outlines its boundaries.

### Project Justification:

The UniVerse platform is designed to connect university students who excel in certain subjects with those in need of academic support. It aims to provide a seamless, user-friendly environment for peer-to-peer tutoring, offering features like lesson scheduling, payment systems, and communication tools.



### Project Characteristics and Requirements:

- **User Registration and Profile Management:** Allows students and tutors to register, customize profiles, and specify academic details.
- **Tutor-Student Matching:** Implements search and filtering tools to help students find suitable tutors.
- **Lesson Scheduling and Payment Systems:** Includes flexible scheduling and secure payment methods.
- **Communication Tools:** Supports chat and voice calls for effective tutor-student interaction.
- **Security and Data Privacy:** Ensures secure data handling and compliance with privacy regulations.

### Project Management-Related Deliverables:

- **Project Plan and Gantt Chart:** Detailed timeline and milestones.
- **Documentation:** Comprehensive project documentation, including requirements, design, and testing plans.

### Product-Related Deliverables:

- **Functional Platform:** Fully developed and tested UniVerse platform with all specified features.
- **User Training and Support Materials:** Guides, tutorials, and support for end-users.

## 1.6 Gant chart

This project timeline is planned for the last detail so that each of its phases is completed on time. The detailed Gantt chart that outlines the schedule for the proposal phase can be seen in Figure 1.2 below.

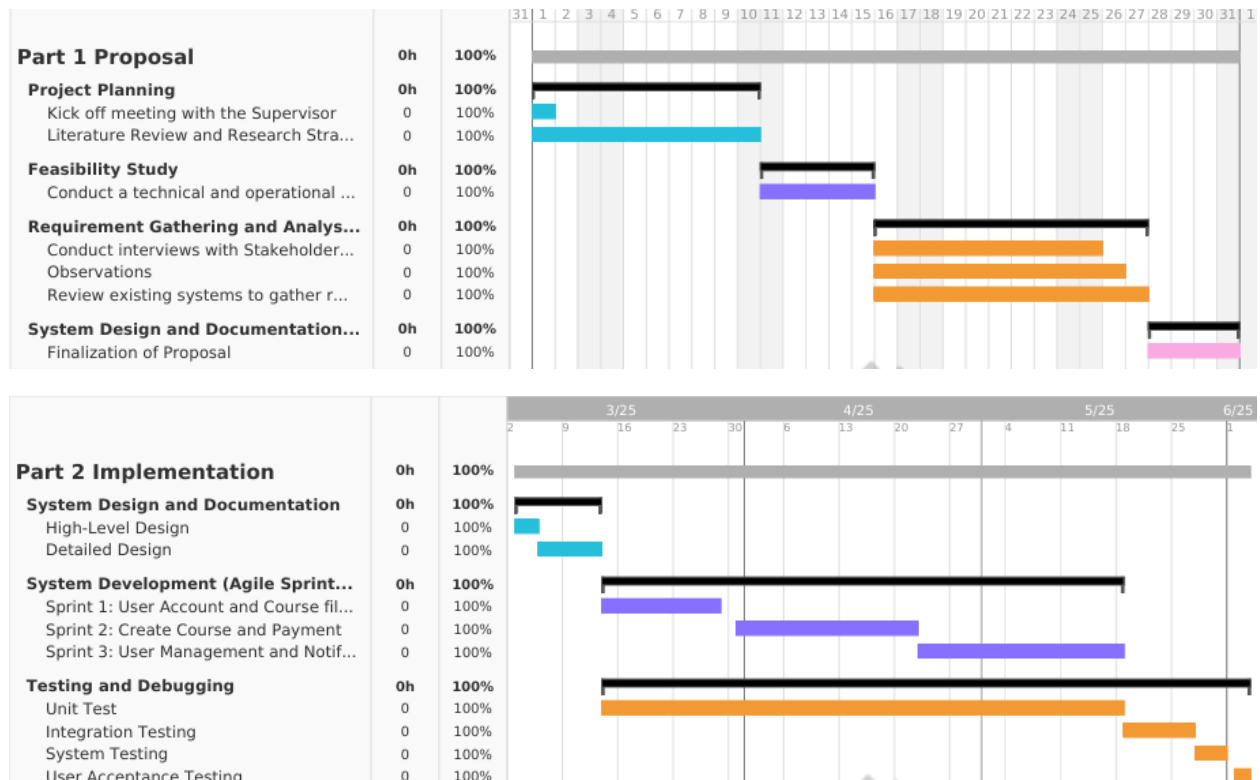


Figure 1.2 Gant chart

This timeline, represented in Figure 1.2, outlines key points at which literature review, feasibility study, and system design must be completed to ensure coherence in the fulfillment of the project's objectives.

## 1.7 Project Outline

The UniVerse project is structured into six comprehensive chapters, each addressing a key aspect of the system's development and implementation.

- **Chapter One: Introduction**

This chapter introduces the UniVerse platform, outlines the problem statement that highlights the financial and academic challenges faced by university students, and presents the main objectives of the project. It also includes the research strategy adopted and defines the scope of the system in terms of features available for students and tutors.

- **Chapter Two: Literature Review**

This chapter explores existing peer-to-peer tutoring platforms and e-learning systems, evaluating their strengths and shortcomings. It highlights gaps such as

lack of affordability, accessibility, or proper matching mechanisms, and justifies the need for a more student-centric and cost-effective platform like UniVerse.

- **Chapter Three: Methodology**

Here, the software development lifecycle and project management strategy are discussed. The Agile Scrum methodology is chosen for its flexibility and user involvement. This chapter also includes the feasibility study and requirement gathering processes conducted with potential users (students, tutors, and administrators).

- **Chapter Four: System Design**

This chapter presents the architecture and technical design of the UniVerse system using various UML diagrams such as use case diagrams, activity diagrams, and entity-relationship diagrams. It provides a visual and structural understanding of how the system functions.

- **Chapter Five: Implementation and Testing**

This chapter documents the implementation of core features across multiple sprints and discusses the tools and technologies used (e.g., React, .NET 8, and SQL Server). It also covers testing procedures including unit testing, integration testing, and user acceptance testing, with results demonstrating the system's reliability and performance.

- **Chapter Six: Conclusion and Future Work**

The final chapter summarizes the accomplishments of the project, reflects on the challenges faced, and evaluates whether the defined objectives were achieved. It also discusses future enhancements, such as AI-based tutor recommendations, mobile app development, or expansion to other universities.

# CHAPTER 2: Literature Review

## 2.1 Overview

The literature reviews relevant to the UniVerse platform, examining peer-to-peer tutoring systems from technological, educational, and economic perspectives. It provides insights into how similar platforms have been developed, the challenges they have faced, and the solutions they have implemented. The chapter is organized as follows: Section 2.2 discusses related work and the challenges encountered, and Section 2.3 summarizes the key findings and their relevance to UniVerse.

## 2.2 Related Work

This section provides a detailed comparison between UniVerse and other established online tutoring platforms, focusing on their strengths, weaknesses, and how UniVerse aims to improve upon them.

- **TeacherON:**

TeacherOn.com is a globally recognized platform trusted by thousands of students and teachers. Its primary mission is to make every teacher easily searchable, both in their local area and online, while keeping the platform free for both teachers and students. Additionally, the platform strives to provide students with a tutor within 24 hours of posting their requirements. [2]

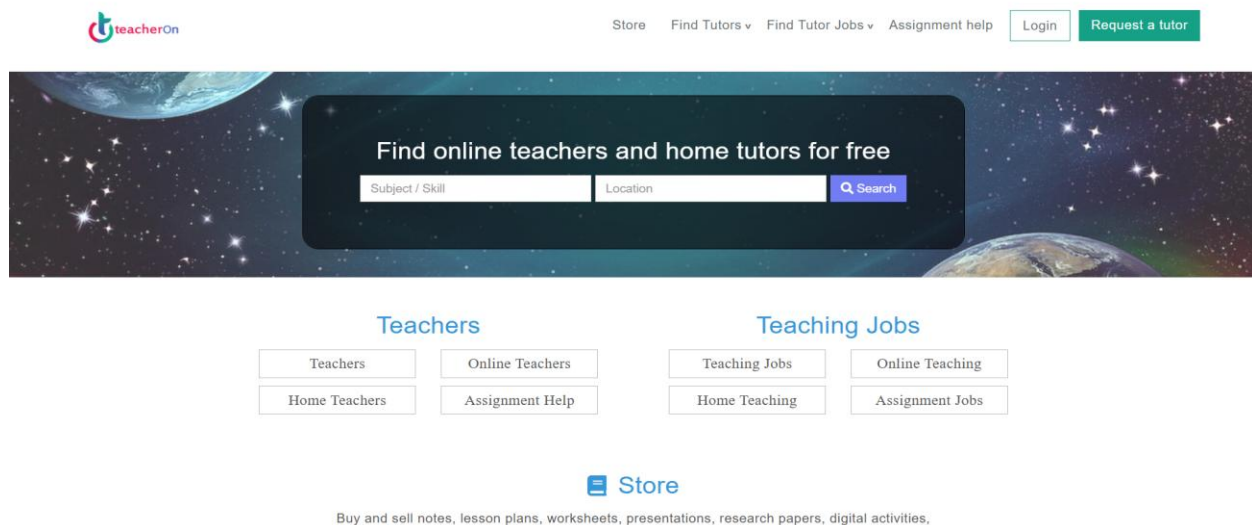


Figure 2.1: Main interface of the TeacherOn website

### Advantages of TeacherOn:

- **Global Reach:** TeacherOn connects tutors and students worldwide, offering a wide range of subjects across different academic levels.
- **Variety of Subjects:** The platform supports diverse disciplines, making it easy for students to find specialized help.
- **Flexible Tutor Registration:** Tutors can sign up with minimal restrictions, allowing those with expertise in a subject to offer lessons.

### Disadvantages of TeacherOn:

- **Cost for Communication and Lessons:** While listing a profile is free, students must pay both for lessons and to the platform for initial communication with tutors, which can be a financial burden.
- **Inconsistent Tutor Quality:** The lack of stringent verification processes means tutor quality can vary, potentially affecting the learning experience.
- **Algorithmic Limitations:** The platform's ranking algorithms may prioritize factors like profile completeness or keywords over actual teaching effectiveness, potentially leading to less competent tutors being ranked higher.
- **Overwhelming Options:** The wide range of subjects and tutors can make it challenging for students to quickly find the right match.

- Delayed Matching: TeacherOn aims to connect students with suitable tutors within 24 hours of posting their requirements.

- **JO Academy:**

Jo Academy, established in 2014, is the leading e-learning platform in Jordan, offering interactive education and training courses using advanced technologies. Its vision is to enhance education through technology, providing secure and innovative solutions for students, parents, and institutions. Jo Academy aims to support educational institutions and empower students to succeed in the evolving technological landscape.[3]



Figure 2.2: Main interface of the Jo Academy website

### Advantages of Jo Academy:

- Technology Integration: Jo Academy uses advanced technologies to offer interactive and engaging educational experiences.
- University, Major, and Course Segmentation: Allows students to find resources that match their academic needs.

### Disadvantages of Jo Academy:

- Pre-recorded Content: Limits opportunities for real-time engagement and personalized assistance.
- Full Course Structure: Does not allow for customization based on individual needs.

- **Complex Process for Tutors:** The rigorous requirements for becoming a tutor can slow down the process of expanding the tutor base.
- **Watad:**

Watad Educational Platform is an online learning platform aimed at simplifying education for students across Jordan by offering a modern, easy, and enjoyable learning approach. The platform covers all educational levels, from primary grades, through high school (Tawjihi), and up to university and beyond.[4]



Figure 2.3: Main interface of the Watad website

Watad is similar to Jo Academy in all advantages and disadvantages, but it does not offer the feature of university, major, and course segmentation that allows students to find resources matching their academic needs.

## 2.4 Summary

This section provides a summary of findings and includes a comparison table of features from related works, showing how UniVerse addresses existing gaps.

Table 2.1: Comparison of features across related platforms.

	Live Content	Technology Integration	Customization based on individual needs	Global	Flexible Tutor Registration	Segmentation
Teacher ON	✓	✗	✓	✓	✓	✗
JO Academy	✗	✓	✗	✗	✗	✓
Watad	✗	✓	✗	✗	✗	✗
<b>Universe</b>	✓	✓	✓	✗	✓	✓

The comparison shows that UniVerse stands out as a promising option in the field of e-learning, distinguished by its ability to personalize the learning experience for each student, which enhances their understanding and comprehension of the material. Additionally, UniVerse offers content segmentation and live content features. Although it does not have a broad global reach, the flexibility in its trainer registration system makes it an attractive choice.



# CHAPTER 3: Methodology

## 3.1 Overview

The **Agile Scrum** methodology was adopted for the development of the UniVerse platform due to its proven efficiency in delivering flexible, user-centered software solutions. Agile Scrum divides the development process into iterative cycles known as *sprints*, typically lasting 2–3 weeks. Each sprint focuses on delivering a functional increment of the platform, allowing the development team to adapt to evolving requirements and user feedback in real time (Schwaber & Sutherland, 2017).

The methodology emphasizes close collaboration between team members and stakeholders through regular ceremonies such as daily stand-ups, sprint planning meetings, sprint reviews, and retrospectives. These practices ensure transparency, continuous improvement, and alignment with user expectations. Given that UniVerse serves a diverse group of users—students and tutors—the need for flexibility and continuous refinement is critical.

By using Agile Scrum, the UniVerse team can quickly respond to usability concerns, integrate new ideas based on student or tutor input, and maintain high product quality throughout the development lifecycle. This adaptive approach significantly increases the likelihood that the final platform will meet the academic, social, and operational needs of its users.

## 3.2 Feasibility Study

To ensure the successful development and implementation of the UniVerse platform, a feasibility study is conducted to evaluate the project's viability from several critical perspectives. This study assesses technical, operational, economic, and legal aspects to determine whether UniVerse is a feasible and worthwhile endeavor.

- **Technical Feasibility**

Evaluate whether the technology and tools required for UniVerse are appropriate and available.

🔗 **Technology Stack:**

- Frontend: React

- Backend: ASP.NET Core web API
- Database: SQL Server
- Payment Gateway: Stripe or PayPal

#### **? System Requirements:**

- Server: Cloud-based hosting (e.g., AWS or Azure)
- Security: SSL/TLS encryption
- Scalability: Adaptable to varying loads
- APIs: RESTful APIs for frontend and backend communication

- **Operational Feasibility**

Determine if UniVerse can be effectively operated and maintained.

- User Experience:
  - Design: User-centered design principles to ensure ease of use for students, tutors, and administrators.
  - Testing: Extensive usability testing to gather feedback and refine the user interface.
- Support and Training:
  - Documentation: Comprehensive user manuals and online help resources.
  - Training: Development of training materials for users and administrators.

### **3.3 Methodology Process**

The Agile Scrum methodology process was quite appropriate for projects with continually evolving requirements for the engagement of different stakeholders, and it was on this basis that the UniVerse was developed. The working of the Agile Scrum methodology involves segmentation of a project into small-size iterations known as sprints, usually lasting 2 to 4 weeks. Every phase of such a process was executed in the following manner:

### 3.3.1 Requirements

The initial requirements for the UniVerse platform were gathered through interviews, observation, and analysis of similar systems. These requirements are divided into functional and non-functional categories and will be refined based on feedback from future questionnaires.

#### **Functional Requirements**

##### 1. Student Services

- 1.1 Register/Login: Students can create accounts, log in securely, and recover passwords.
- 1.2 Browse Courses: Students can explore available courses with filters and categories.
- 1.3 Enroll in Course: Students can enroll in free or paid courses.
- 1.4 Manage Profile: Students can update personal details, password, and view course history.

##### 2. Instructor Services

- 2.1 Register/Login: Instructors can register, log in, and verify their identity.
- 2.2 Create/Edit Course: Instructors can create, update, or delete courses.
- 2.3 Upload Content: Instructors can upload files, videos, and quiz materials.
- 2.4 Request Withdrawal: Instructors can request withdrawal of their earnings.
- 2.5 Manage Profile: Instructors can update their personal and payment information.

##### 3. Admin Services

- 3.1 Manage Users: Admins can add, edit, delete, or block users (students/instructors).
- 3.2 Moderate Courses: Admins can approve, reject, or deactivate submitted courses.

#### 4. Payment Handling

- 4.1 Process Payment: Handles transactions via third-party gateways (e.g., Visa, PayPal).
- 4.2 Send Confirmation / Failure: Notifies users of payment success or failure.
- 4.3 Update Instructor Earnings: Automatically updates instructor wallets after enrollments.

#### **Nonfunctional Requirements**

- Security: Role-based access, secure authentication, and data privacy protection.
- Usability: Clean, responsive UI using Bootstrap for easy navigation across all devices.

#### **Integrated Agile Scrum Process Overview**

The development of the UniVerse system followed the Agile Scrum methodology, ensuring adaptability and continuous stakeholder involvement. Requirements were placed into a dynamic product backlog, updated regularly based on evolving needs and feedback. At the beginning of each sprint (2–4 weeks), sprint planning was conducted to select top-priority features, define sprint goals, and estimate tasks. Development was incremental, with each sprint producing a functional module tested through unit, integration, and system tests.

After each sprint, reviews were conducted with stakeholders for feedback, and retrospectives were held to evaluate team performance. The system was deployed incrementally, enabling early use and direct feedback on core features. This feedback, combined with retrospective insights, helped refine the product backlog and guide the next sprint. The iterative nature of Agile Scrum allowed the project to remain aligned with real user expectations and continuously evolve toward a complete, user-centric solution.

Table (3.1) Agile Scrum for AAU Seminar Flow development

Number of sprints	Sprint 1	Sprint 2	Sprint 3
	<pre> graph TD     A[Sprint Planning] --&gt; B[Development]     B --&gt; C[Testing]     C --&gt; D[Review &amp; Retrospective]     D --&gt; E[Incremental Deployment] </pre>	<pre> graph TD     A[Sprint Planning] --&gt; B[Development]     B --&gt; C[Testing]     C --&gt; D[Review &amp; Retrospective]     D --&gt; E[Incremental Deployment] </pre>	<pre> graph TD     A[Sprint Planning] --&gt; B[Development]     B --&gt; C[Testing]     C --&gt; D[Review &amp; Retrospective]     D --&gt; E[Incremental Deployment] </pre>
Time of Sprint	2 weeks	3 weeks	4 weeks
Requirements Item/Backlog	Register/Login Create/Edit Course Browse Courses	Upload Content Filter Courses	Process Payment Update Earnings
Data Collection tool	Interviews, desk review and observation	Interviews, desk review and observation	Interviews, desk review and observation

The theme of incremental development is underscored in Table 3.1, where it points out that each sprint had functionalities to be used at that time, that is, user account creation, forming of groups, and reporting progress. This approach will ensure, for sure, the evolution of the system based on real-time feedback from the stakeholders.

# Chapter 4: System Design

## 4.1 Overview

The UniVerse architecture is documented through a series of well-defined design diagrams that collectively represent the system's structure, behavior, and user interactions. These diagrams provide clear insight into how the system functions, processes data, and facilitates communication between components. The documentation includes:

- Context Diagram – Defines the boundaries of the UniVerse system and its interaction with external entities such as Students, Instructors, and Admins.
- Data Flow Diagram (DFD) – Describes how information flows through the system's main processes, such as course registration, instructor onboarding, and course management.
- Use Case Diagram – Highlights the functional requirements of the system from the perspective of different users, detailing their interactions and goals.
- Sequence Diagrams – Depict the temporal flow of messages and events in key processes, such as when a student becomes an instructor or creates a course.
- Entity Relationship Diagram (ER Diagram) – Defines the logical structure of the UniVerse database by outlining key entities such as Users, Courses, Enrollments, and Instructors, along with their relationships. It ensures organized data storage and integrity across the system.

## 4.2 Context Diagram

The Context Diagram depicts the UniVerse boundaries and external interactions, showing how Students, Instructors, and Administrators interact with the system. It represents the system as a single process and illustrates all input and output flows, providing a clear overview of how the system interfaces with its external entities and what information is exchanged between them.

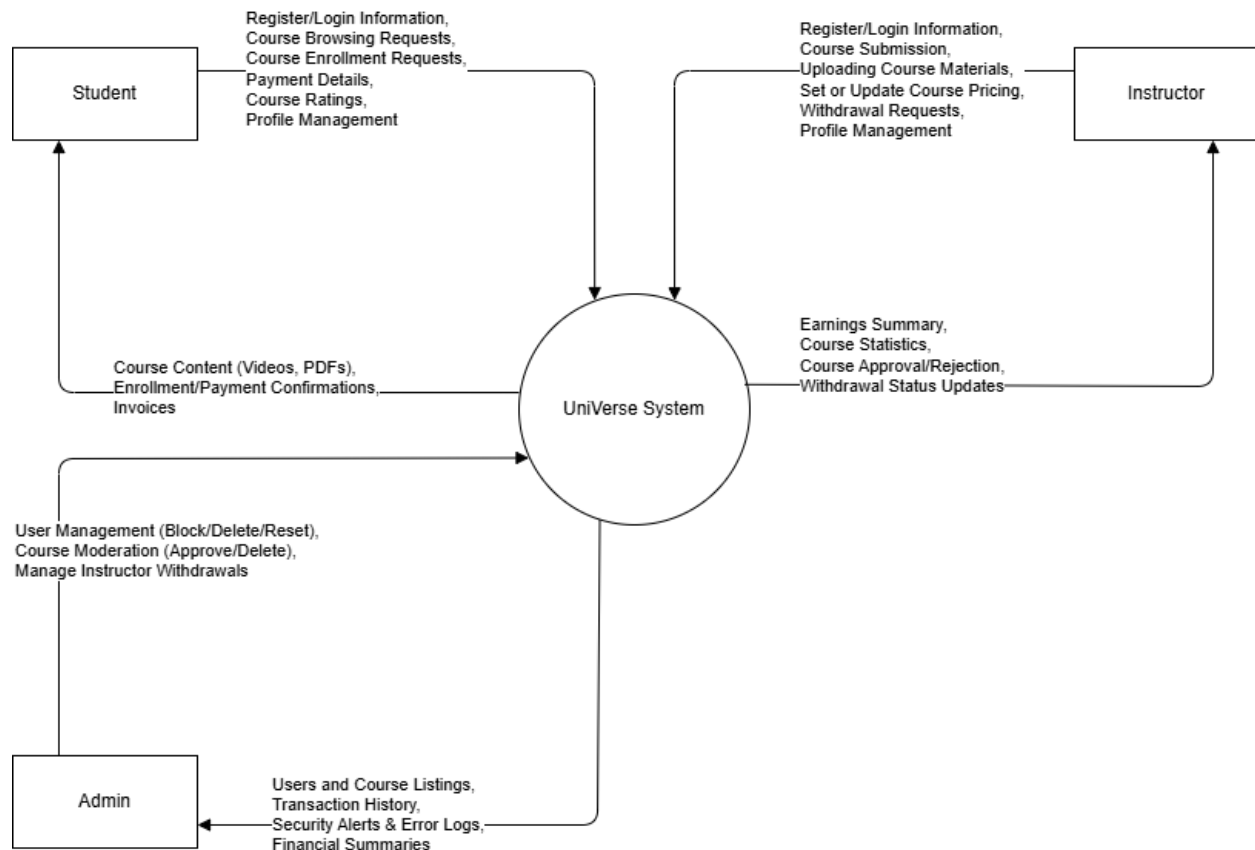


Figure 4.1 Context Diagram

## Data flow Diagram

Coming soooooooooooooooooooooon

## 4.3 Use Case Diagram

The Use Case Diagram represents the functional requirements of the UniVerse system by illustrating how various users interact with its features. It highlights the relationships between actors (Students, Instructors, and Admin) and their associated use cases, effectively showcasing the system's functionality from each user's perspective. In the diagram, dotted lines represent extend relationships, while solid lines represent include relationships.

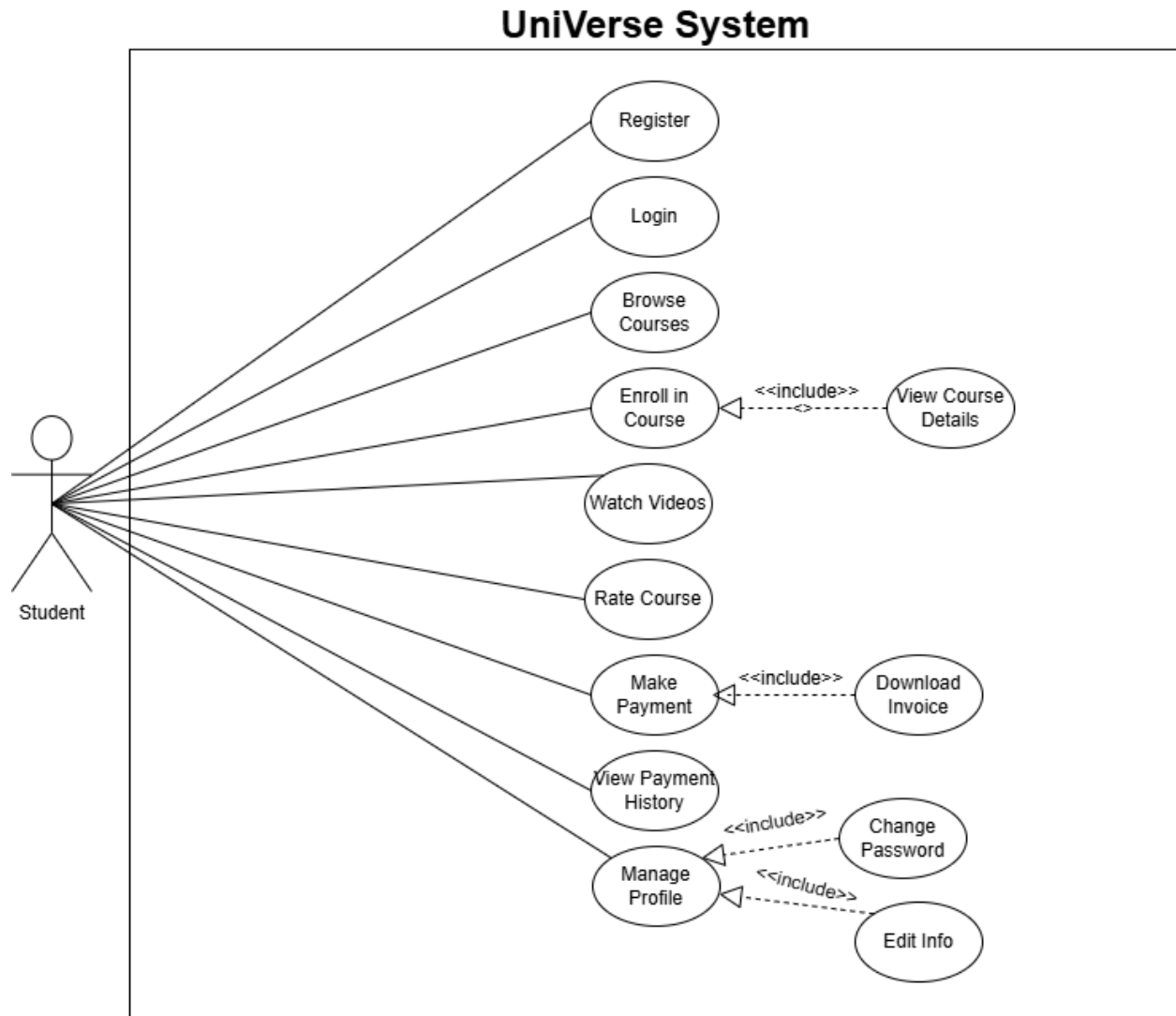


Figure 4.3 Student Use Case Diagram



## UniVerse System

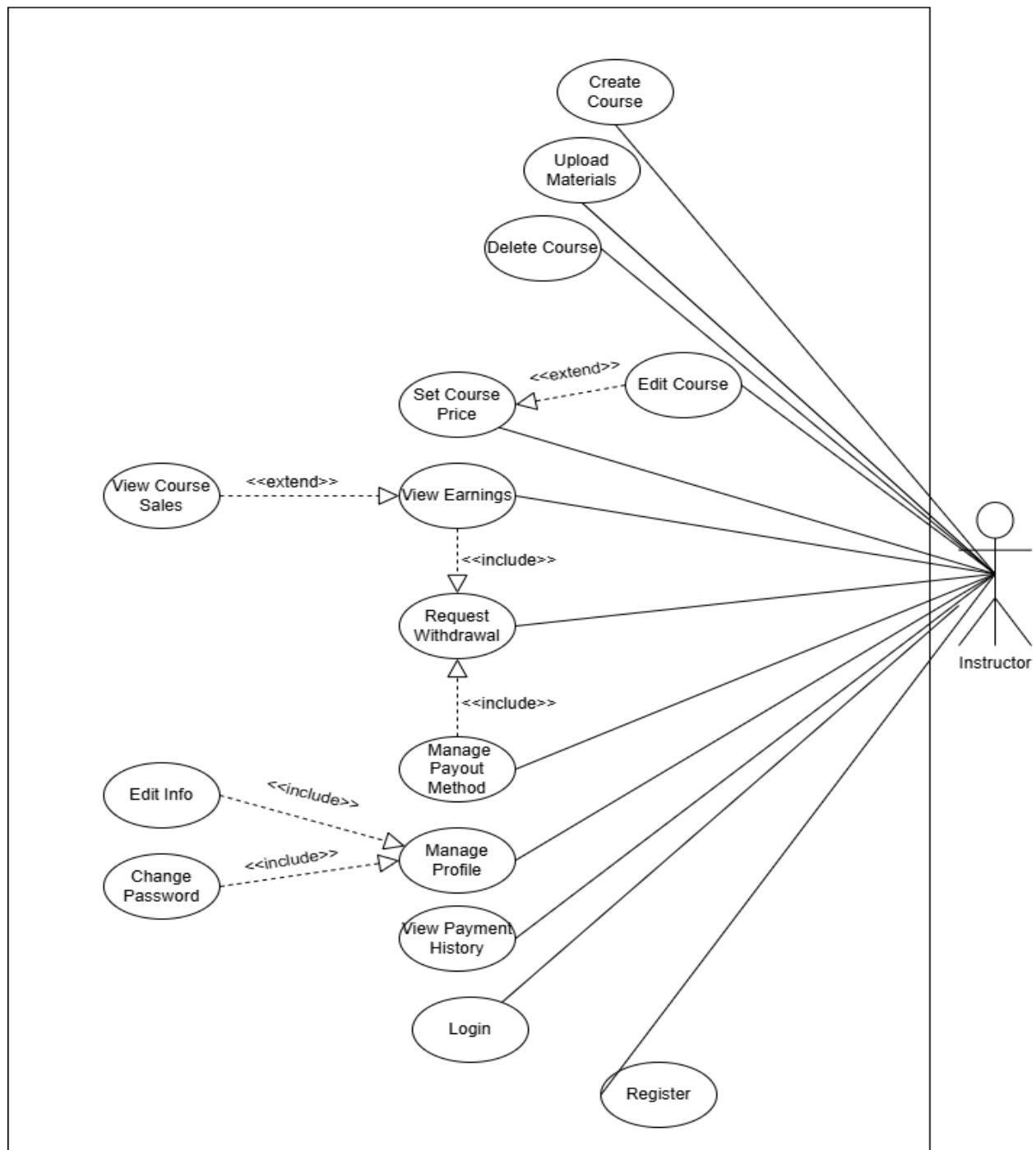


Figure 4.2 Instructor Use Case Diagram

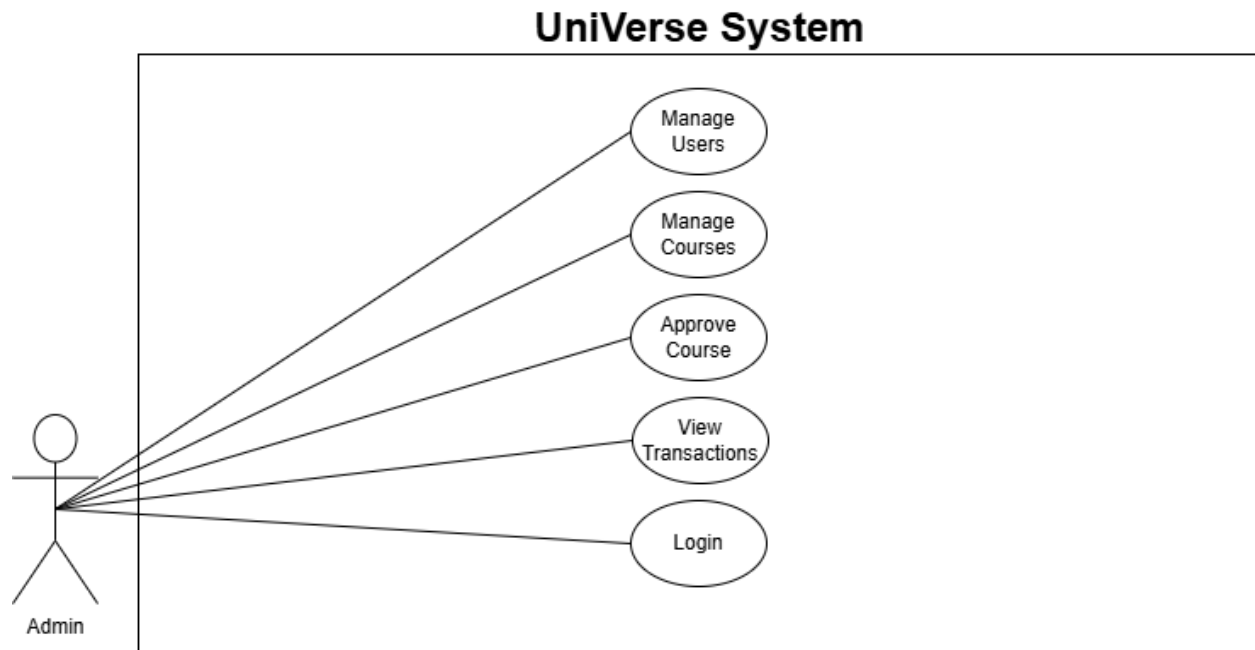


Figure 4.3 Admin Use Case Diagram

#### 4.4 Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) provides a high-level conceptual representation of the data model used within the UniVerse system. It illustrates the key entities involved—such as Students, Instructors, Courses, Enrollments, and Assignments—and the relationships between them. The purpose of this diagram is to define the structure of the database in a clear and logical manner, enabling efficient data organization and retrieval. By outlining the attributes and associations of each entity, the ERD helps ensure data integrity and supports the development of a robust backend architecture. This diagram serves as a foundation for database design and plays a crucial role in aligning system requirements with the underlying data model.

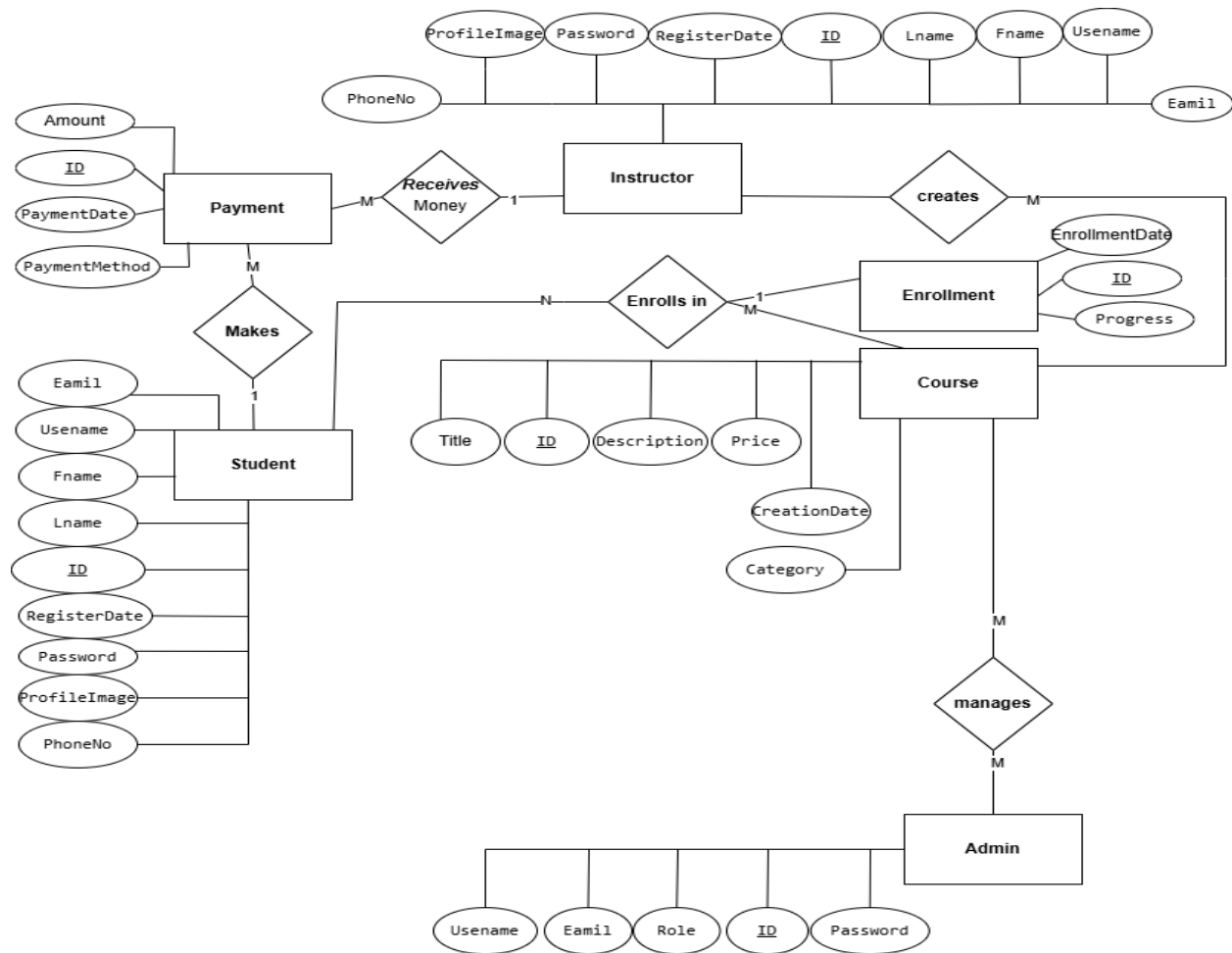


Figure 4.4 Universe ER diagram

## 4.4 Sequence Diagram

The Sequence Diagrams for the UniVerse platform illustrate the chronological interaction flow between users and the system during essential use cases. They depict how different actors—such as students and instructors—communicate with the system over time to achieve core educational tasks. These diagrams capture major processes, including student registration, course enrollment, and course creation. The first diagram represents the student registration sequence, highlighting actions such as submitting registration details, backend validation, and automatic login. The second diagram demonstrates the instructor’s interaction with the system when creating and publishing a course, showcasing content preparation, saving, and publishing functionalities. The third diagram visualizes the course enrollment process from the student’s perspective, including browsing available courses, selecting one, and completing enrollment. Together, these diagrams offer a clear

view of how UniVerse manages user interactions, simplifies educational workflows, and ensures an organized flow of operations between the frontend and backend components.

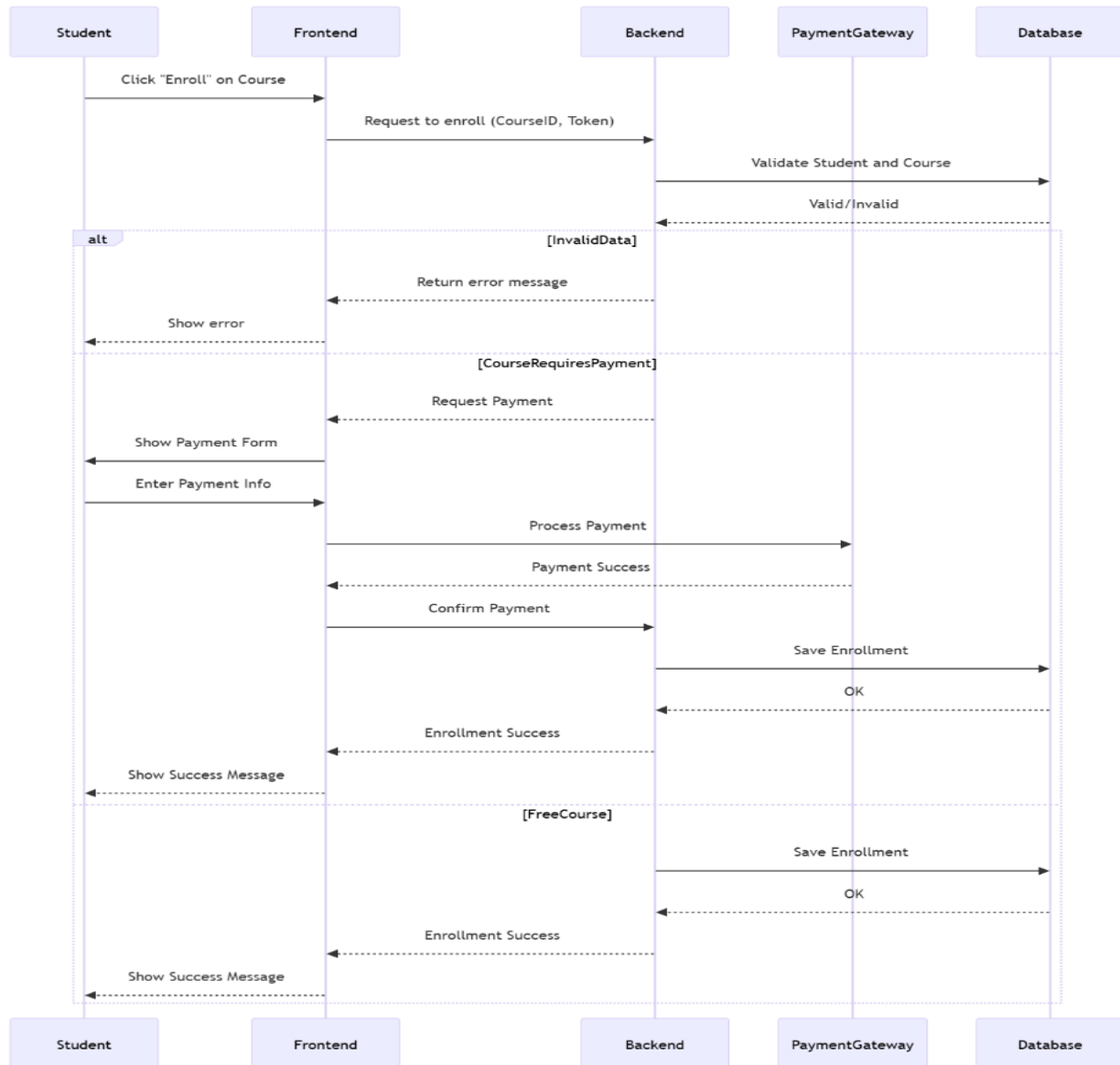


Figure 4.4 Student's course enrollment process sequence diagram

This Sequence Diagram illustrates the process a Student follows to enroll in a course within the UniVerse system. The interaction starts when the student clicks “Enroll” on a course. The system validates the student's identity and course availability. If the course requires payment, a payment process is triggered using a Payment Gateway. Once payment is

confirmed, the system saves the enrollment and displays a success message. If the course is free, the enrollment is saved directly without a payment step. Alternate flows handle invalid data or errors in payment.

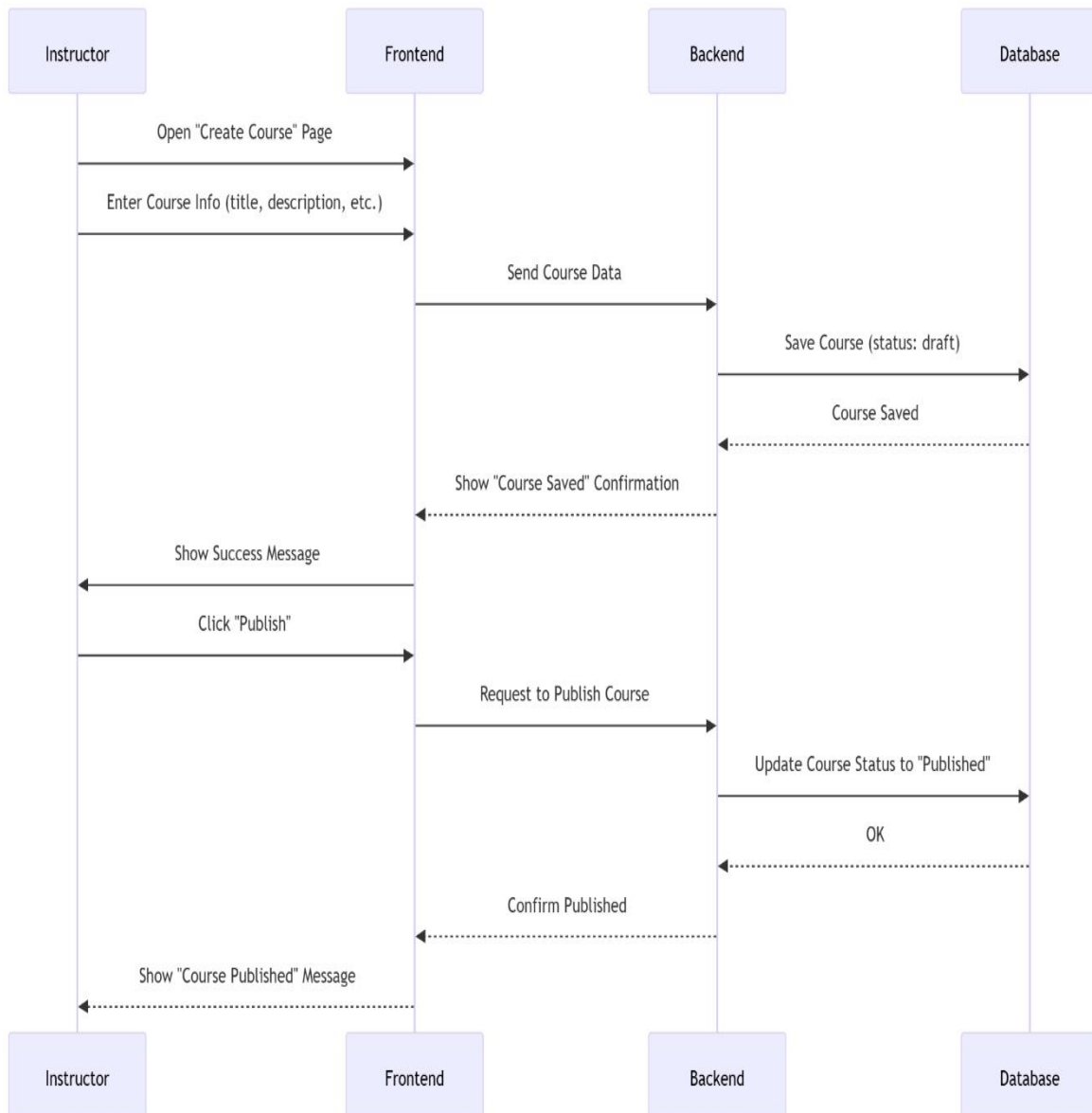


Figure 4.5 Instructor create course sequence diagram

This diagram demonstrates how an Instructor creates and publishes a course. The process begins when the Instructor accesses the course creation page and submits course details

such as title and description. The system saves the course in a draft status. After reviewing the course, the Instructor clicks “Publish,” prompting the backend to update the course status to “Published.” Feedback is provided to the Instructor after each stage, confirming both the save and publish operations.

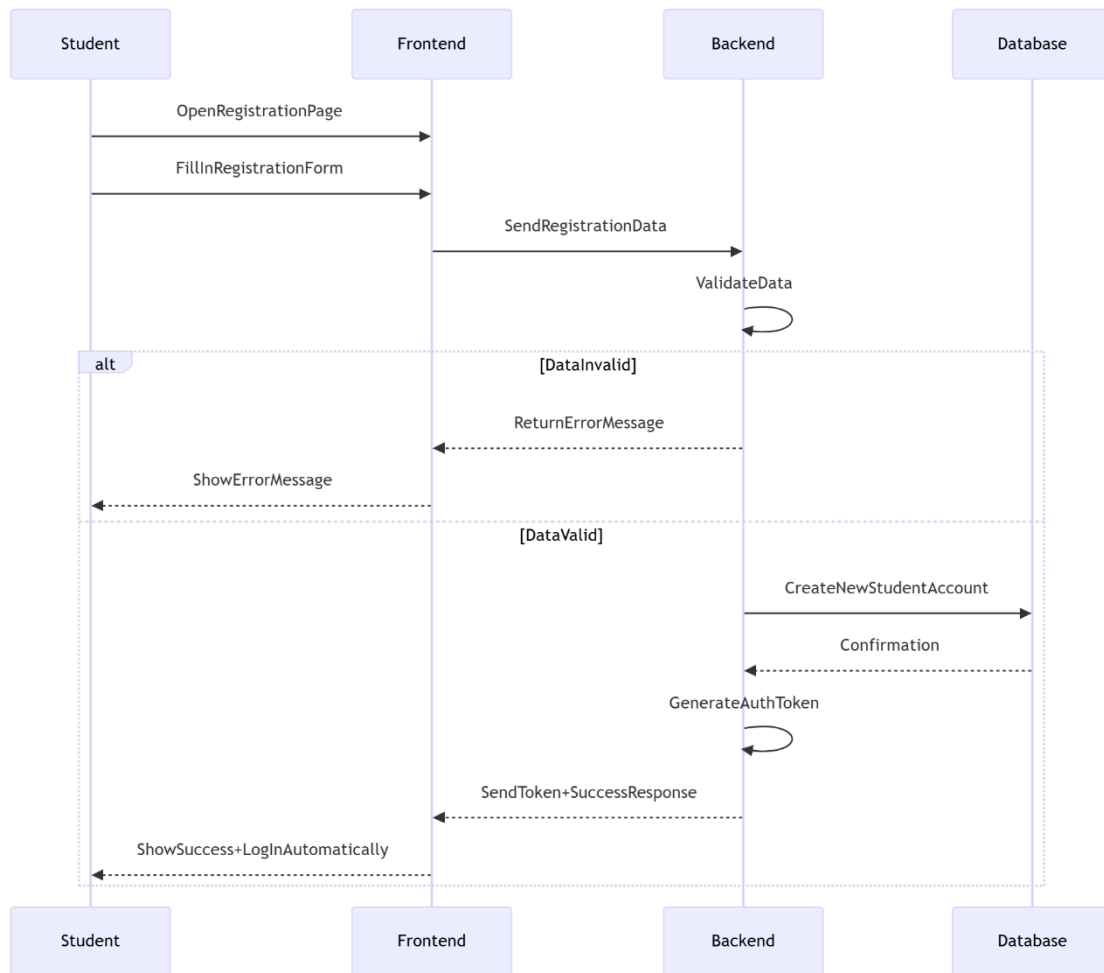


Figure 4.6 Student registration sequence diagram

This Sequence Diagram outlines the steps for registering a new Student in the UniVerse system. The Student opens the registration page, fills in the required details, and submits the form. The backend validates the data; if valid, it creates a new student account and generates an authentication token. A success response is sent back to the frontend, which displays a success message and logs the user in automatically. If the data is invalid, an error message is returned and shown to the user.

# Chapter 5: Implementation and Testing

## 5.1 Overview

The UniVerse system was tested using simple and practical manual tests. For unit testing, the login functionality was tested to ensure that the system accepts correct credentials and rejects invalid ones. For integration testing, the process of becoming an instructor was tested to verify that once a student submits a request, the system updates their role and allows access to instructor features. For system testing, the complete process of creating a course by an instructor and having a student enroll in it was tested end-to-end to ensure all components work together smoothly. These basic tests were sufficient to confirm the core

## 5.2 Testing methodologies

Testing methodologies are the ways used to check if the system works correctly. In UniVerse, we used manual testing like unit, integration, and system testing to make sure everything works as expected.

### 5.2.1 Unit Testing Results

Unit testing focuses on testing individual modules and components of the UniVerse system in isolation. This includes verifying that specific features like user login, instructor registration, and course creation behave as expected. Each component is tested separately to ensure it functions correctly before combining it with the rest of the system. Below is the test case:

Tabel 5.1 Unit Test Case

Field	Description
Test Case ID	UT001
Component	Login Module
Description	Verify user login with valid credentials
Pre-condition	User account exists in the system
Test Steps	<ol style="list-style-type: none"><li>1. Navigate to the login page</li><li>2. Enter valid email</li><li>3. Enter valid password</li><li>4. Click the login button</li></ol>
Expected Results	<ol style="list-style-type: none"><li>1. System validates credentials</li><li>2. User is successfully authenticated</li></ol>

	3. User is redirected to their dashboard
Pass/Fail Criteria	All steps are completed without errors, and the user reaches the correct dashboard

Figure 5.1 corresponds to the Unit Testing table and demonstrates the unsuccessful execution of a specific test case for the Login Module. The table outlines the test case ID, pre-conditions, steps, and expected outcomes, while the figure provides a visual representation of the system's response to valid and invalid inputs. Together, they confirm that individual components, such as login functionality, operate as intended in isolation, ensuring reliability before integration.

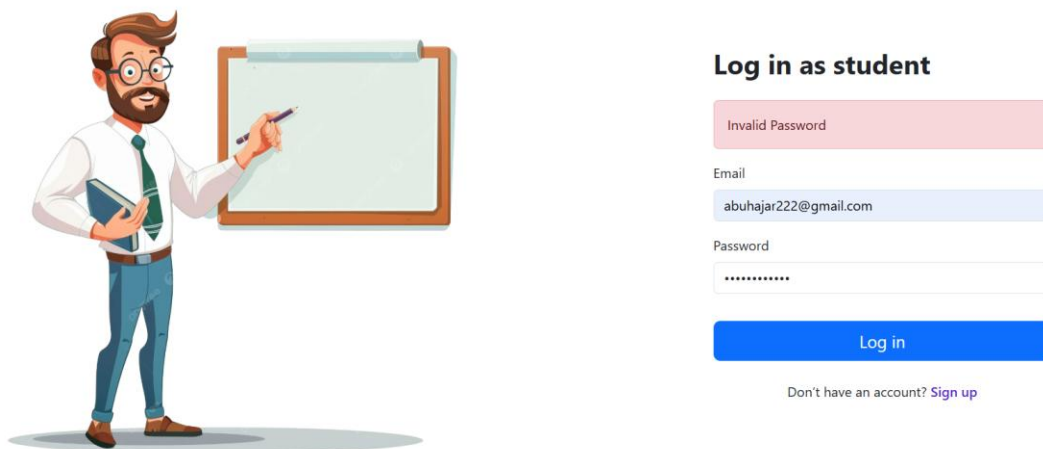


Figure 5.1 Student Login Test

### 5.2.2 Integration Testing Results

Integration testing examines how different modules of the UniVerse system work together. This phase focuses on verifying communication and data flow between connected components, particularly the process of converting a student into an instructor. Once a student submits a request to become an instructor, the system updates their role accordingly and grants access to instructor-specific features. This test confirms that the integration between the student module, instructor module, and authentication mechanisms functions correctly, with data passed accurately between them. Below is the test case:



Tabel 5.2 Integration Test Case

Field	Description
Test Case ID	IT001
Component	Student Module + Instructor Module + Authentication Module
Description	Test the complete workflow of converting a student into an instructor
Pre-condition	Student account exists and is logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Student clicks "Teach with Us" button</li> <li>2. System sends request to controller</li> <li>3. Modal prompts for instructor password</li> <li>4. Student submits password</li> <li>5. System creates instructor profile</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Student role updated in the database</li> <li>2. Instructor profile created</li> <li>3. Instructor dashboard becomes accessible</li> </ol>
Pass/Fail Criteria	Student successfully gains access to instructor features without logging out

The Integration Testing figure complements the associated test case table, which details the workflow between connected modules, such as the Student Module, Instructor Module, and Authentication Module. The table outlines the pre-conditions, step-by-step actions, and expected outcomes for the process of converting a student into an instructor. The figure illustrates the actual system interface used during integration testing. Together, they validate the seamless communication and proper data flow between modules, confirming the system's ability to consistently update roles and transition users between functionalities without errors.

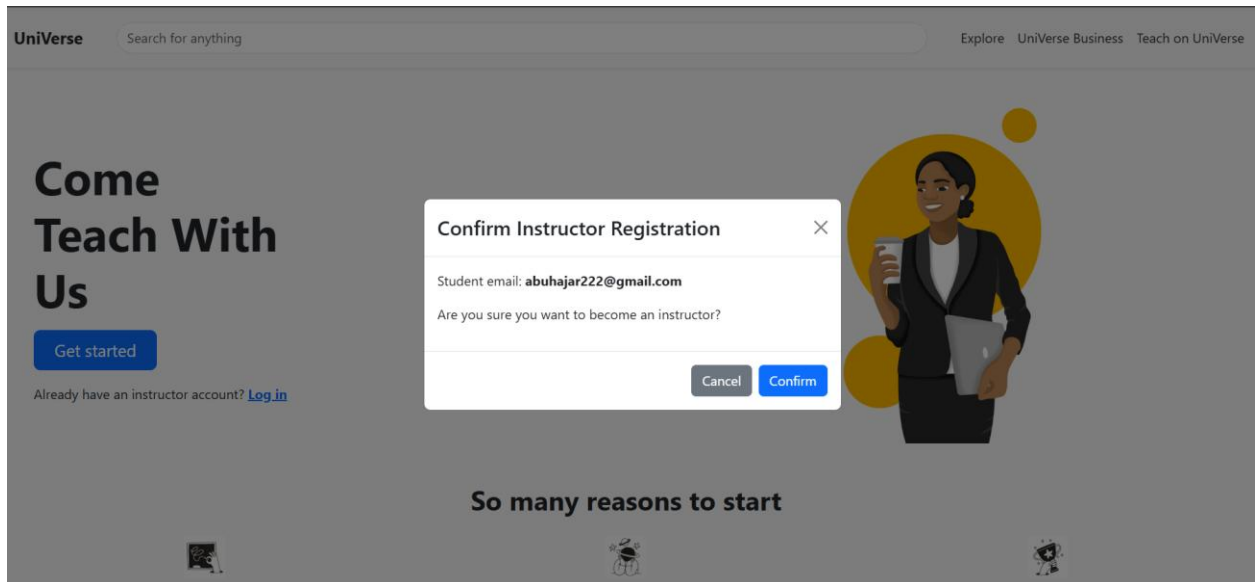


Figure 5.2 Student converting a student into an instructor integration Test

### 5.2.3 System Testing Results

System testing evaluates the complete UniVerse system as a whole, ensuring that all components function correctly in realistic, end-to-end usage scenarios. This includes testing comprehensive workflows such as an instructor creating a course, and a student browsing, enrolling, and interacting with that course. The objective is to verify that the entire system meets its functional requirements and that the integrated modules operate seamlessly together under real-world conditions. Below is the test case:

Tabel 5.2 System Test Case

Field	Description
Test Case ID	ST001
Component	Instructor Module + Course Module + Student Module
Description	Test the complete workflow of creating a course and student enrollment

Pre-condition	Instructor and student accounts exist and are logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Instructor creates a new course</li> <li>2. Course is published and visible</li> <li>3. Student browses available courses</li> <li>4. Student enrolls in the course</li> <li>5. Student accesses course content</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Course is saved and listed publicly</li> <li>2. Enrollment request is recorded</li> <li>3. Student is added to course participants</li> <li>4. Course content becomes accessible to student</li> </ol>
Pass/Fail Criteria	All steps completed successfully without error; student gains access to course content

Figure 5.3 illustrates a part of the system testing process in the UniVerse platform, specifically the workflow of course creation by an instructor. It demonstrates the actual system interface used during the test, showing how the instructor fills in course details and submits the form. This visual representation supports the test case by confirming that the system responds correctly to user inputs and that the course creation process functions as intended within the complete system workflow.

The screenshot shows the UniVerse platform interface. On the left is a dark sidebar with the UniVerse logo and a menu with items: Courses, Communication, Performance, Tools, and Resources. The main content area has a header 'Jump Into Course Creation' and a sub-header 'No courses found. Based on your experience, we think t...'. A modal window titled 'Create Course' is open in the center. It has a close button (X) in the top right corner. The form fields are: Title (Database), Description (This course introduces the fundamental concepts of database systems, including data modeling, relational), Price (5), Duration (hours) (2), Picture URL (https://www.google.com/imgres?q=picture&imgurl=https%3A), and Video Preview URL (https://www.youtube.com/watch?v=OYIzcXA3LxI). A blue 'Submit' button is at the bottom of the modal. In the background, there are sections for 'Managing Course' and 'Build Your Audience'.

Figure 5.3 Instructor Create Course

## 5.2.4 Acceptance System Results

User Acceptance Testing is the final phase of testing where the actual users (students, instructors, and administrators) test the system to verify it meets their requirements and business needs. During UAT, users perform their typical tasks in the system to ensure it supports their daily operations effectively. This testing phase helps identify any usability issues, missing functionality, or business process gaps before the system goes live. Below is the test case:

Tabel 5.2 Integration Test Case

Field	Description
Test Case ID	UAT001
Component	Student
Description	Test key student services: Registration/Login, Course Browsing, Enrollment, and Profile Management
Pre-condition	User has internet access and UniVerse platform is accessible
Test Steps	1. Register a new student account 2. Log in with valid credentials 3. Browse courses using filters 4. Enroll in a course 5. Update profile details and change password
Expected Results	1. Account created successfully 2. Login accepted 3. Course list filtered correctly 4. Enrollment confirmed 5. Profile updates saved
Pass/Fail Criteria	All tasks completed without errors Interfaces are user-friendly System responds within acceptable time

## 5.3 Discussion and Evaluation

The testing phase of the UniVerse system followed a structured approach incorporating unit testing, integration testing, system testing, and user acceptance testing (UAT). Unit testing focused on verifying individual functionalities such as login validation, course browsing, and profile updates. These tests confirmed that the core components operate as expected in isolation. Integration testing assessed the interaction between modules, particularly the flow from student registration to instructor conversion, and the interaction between course creation and student enrollment processes.

System testing validated full end-to-end workflows, ensuring that instructors could create courses and students could enroll, interact, and manage their learning experience seamlessly. In the UAT phase, actual users—students and instructors—engaged with the platform to evaluate usability and real-world functionality. Their feedback confirmed that the system supports essential educational operations such as account management, course access, and instructional roles effectively.

Overall, the test results confirmed the successful implementation of UniVerse’s core services. Minor improvement areas were noted, such as enhancing the responsiveness of course filtering and expanding instructor tools. However, these observations were categorized as enhancements rather than functional issues, indicating the system is stable and ready for production deployment with potential for continuous improvement based on user needs.

## Chapter 6: Conclusion and Future Work

### 6.1 Overview

This chapter presents a summary of the development and implementation of the UniVerse system, highlighting its achievements, contributions, and areas for future enhancement. By reviewing the overall progress and evaluating the final outcomes, we assess how effectively the system has met its original objectives—namely, simplifying the educational process for students and instructors through a unified digital platform. The chapter also reflects on the system’s practical value, usability, and identifies opportunities for further refinement to support continued growth and user satisfaction.

### 6.2 Summary about the Project

The UniVerse system was developed to provide a unified and user-friendly platform for managing the educational experience of both students and instructors. It implements essential functionalities such as secure user registration and login, course browsing and enrollment, instructor onboarding, course creation, and profile management. Designed using modern web technologies and a structured development approach, UniVerse streamlines key academic processes by centralizing interactions in one digital environment. The system enhances accessibility, flexibility, and engagement for users while laying a solid foundation for future expansion to support a broader range of educational services.

### 6.3 Achieved Objectives

The system has successfully achieved its primary objectives:

- Enabling secure and streamlined user registration, login, and profile management
- Providing intuitive browsing and filtering of available courses for students
- Allowing seamless enrollment in courses
- Implementing the ability for students to apply and transition into instructor roles
- Offering instructors tools to create, manage, and publish courses effectively

## 6.4 Main contributions of the work

The key contributions of this project include:

- Digitizing traditional manual processes related to course enrollment and management
- Saving time and effort for both instructors and students through automation and simplified workflows
- Providing improved tracking and monitoring of user activities, course engagement, and progress
- Enhancing data organization, accessibility, and consistency across the system
- Facilitating streamlined communication between students and instructors in a centralized platform
- Introducing a standardized and scalable approach to managing course content, user roles, and educational interactions

## 6.5 Limitation

The UniVerse project faced several limitations, mainly due to time pressure caused by overlapping exams, assignments, and projects from other courses. This limited the time available for development and thorough testing. Despite these constraints, core features were implemented successfully.

## 6.6 Future Work

- Developing a mobile application version
- Implementing advanced analytics and reporting
- Adding real-time chat functionality
- Developing a notification system
- Adding document version control
- Implementing automated backup systems
- Add Arabic language to the System

## References:

- [1] 1. Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
- [2] TeacherOn. (n.d.). About us. Available at: <https://www.teacheron.com/about-us>.
- [3] Jo Academy. (n.d.). About us. Available at: <https://www.joacademy.com/about>.
- [4] Watad. (n.d.). About us. Available at: <https://watad.me/about>.