Software Requirements Specification

For

Edura (E-Learning Platform)

Version 1.0 approved

Prepared by: Hafsa Shahid

<organization>

Dated:16-04-2025

Table of Contents :

Revision History **2**
1. Introduction **3**
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 3
1.5 References 4
2. Overall Description **4**
2.1 Product Perspective 4
2.2 Product Functions 4
2.3 User Classes and Characteristics 5
2.4 Operating Environment 5
2.5 Design and Implementation Constraints 6
2.6 User Documentation 6
2.7 Assumptions and Dependencies 6
3. External Interface Requirements **7**
3.1 User Interfaces 7
3.2 Hardware Interfaces 7
3.3 Software Interfaces 8
3.4 Communications Interfaces 8
4. System Features **9**
4.1 System Feature 1 9
4.2 System Feature 2 (and so on) 10
5. Other Nonfunctional Requirements **11**
5.1 Performance Requirements 11
5.2 Safety Requirements 11
5.3 Security Requirements 11
5.4 Software Quality Attributes 11
5.5 Business Rules 11
6. Other Requirements .....
Appendix A: Glossary .....
Appendix B: Analysis Models .....
Appendix C: To Be Determined List .....

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed overview of the **Edura** e-learning platform, version 1.0. Edura is an online education system designed to facilitate teaching and learning by providing a virtual environment for course management, content delivery, student assessment, and communication. This SRS outlines the functional and non-functional requirements for the platform, and serves as a foundation for development, testing, and maintenance activities. This document focuses on the core system functionalities and does not cover auxiliary services such as third-party integrations at this stage.

1.2 Document Conventions

This document follows standard IEEE SRS formatting guidelines. The following conventions are used throughout:

- **Bold text** indicates section headings and important terms.
- *Italic text* is used for emphasis or references to other documents.
- Monospaced text is used for code snippets or technical terms.
- Requirement priority levels are explicitly stated and do not assume inheritance from higher-level requirements. Each requirement is labeled with a unique identifier for easy traceability.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following stakeholders:

- **Developers**: To implement system functionality based on defined requirements.
- **Project Managers**: To plan, schedule, and manage the development process.
- **Testers**: To create test cases and ensure the software meets its requirements.
- **Clients and End Users**: To understand the capabilities of the system.
- **Documentation Writers**: To prepare user manuals and help content.

Readers should begin with Section 1 (Introduction) to understand the product and document structure, then proceed to Section 2 (Overall Description) for a high-level understanding of system features, followed by Section 3 (Specific Requirements) for detailed functionalities.

1.4 Product Scope

Edura is an e-learning platform aimed at streamlining digital education for academic institutions and independent educators. It allows instructors to create and manage courses, upload learning materials, conduct quizzes and assignments, and interact with students through messaging and forums. Students can enroll in courses, track their progress, and access educational content anytime and anywhere. The platform is accessible via web browsers and is designed with a user-friendly interface to enhance learning engagement and flexibility.

Edura aligns with the strategic goal of providing accessible and scalable education solutions in a digital-first world.

1.5 References

- IEEE Standard 830-1998: *IEEE Recommended Practice for Software Requirements Specifications*
- Edura Vision and Scope Document, Version 1.0, March 2025
- User Interface Style Guide – Edura UI Standards, Version 1.2
- ISO/IEC 25010:2011: *Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE)*
- https://www.edura.com/docs (internal documentation and API reference)

2. Overall Description

2.1 Product Perspective

Edura is a new, self-contained e-learning platform developed to meet the growing demand for flexible and accessible digital education. It is not a follow-on or replacement system, but rather a standalone product designed from scratch. The platform integrates multiple core components such as user management, content delivery, assessment tools, and communication features within a single cohesive system.

Edura interacts with external systems such as email services for notifications, cloud storage for file handling, and optional integration with Learning Management Systems (LMS) via APIs. A high-level architectural diagram is included below to show the major subsystems:

High-Level Architecture:

- **User Interface (Web Portal)**
- **Application Server**
- **Database Server**
- **External APIs (e.g., Cloud Storage, Email Gateway)**

2.2 Product Functions

At a high level, Edura provides the following major functionalities:

- **User Registration and Authentication** (students, instructors, and administrators)
- **Course Creation and Management** by instructors
- **Enrollment Management** for students
- **Content Upload and Delivery** including PDFs, videos, presentations, etc.
- **Assignments and Quizzes** creation, submission, and grading

- **Progress Tracking** and performance analytics for students
- **Internal Messaging System** for communication between users
- **Discussion Forums** for collaborative learning
- **Notifications and Announcements** system
- **Administrative Control Panel** for system-wide configuration

2.3 User Classes and Characteristics

The platform identifies the following primary user classes:

- **Students**
    - Use the system regularly to access course material, submit assignments, take quizzes, and communicate with instructors.
    - Varying levels of technical expertise; the system should be intuitive and easy to use.
- **Instructors**
    - Use the system to create and manage courses, upload content, grade assignments, and interact with students.
    - Expected to have moderate technical expertise.
- **Administrators**
    - Manage users, oversee course data, monitor system usage, and configure global settings.
    - Require advanced system access and control, with technical knowledge of system configuration.
- **Guests (Optional)**
    - May browse public courses or content previews.
    - No access to core platform features without registration.

2.4 Operating Environment

Edura is a web-based platform accessible via modern browsers on various operating systems and devices. The target environment includes:

- **Client Side**:
    - Web browsers: Chrome, Firefox, Safari, Edge (latest versions)
    - Devices: Desktops, laptops, tablets, and smartphones
- **Server Side**:
    - Operating System: Ubuntu 22.04 LTS or equivalent
    - Web Server: Apache/Nginx

- Backend: Node.js or Python (Django/Flask)
- Database: PostgreSQL or MySQL
- Cloud Services: AWS, Azure, or equivalent (optional)

2.5 Design and Implementation Constraints

- Must comply with standard web security practices (SSL, data encryption, input validation)
- Must support responsive design for mobile compatibility
- Backend development must use either Python or JavaScript (Node.js)
- Database must support SQL-based relational models
- All features must follow GDPR and local data privacy laws
- Deployment will be containerized using Docker
- APIs should be RESTful to allow potential future mobile app integration

2.6 User Documentation

The following user documentation will be provided:

- **User Manual** for students and instructors
- **Administrator Guide** for system configuration and management
- **Quick Start Guide** for first-time users
- **Online Help System** embedded within the platform
- **Tutorial Videos** and walkthroughs for common tasks
- All documentation will be provided in PDF and HTML formats and accessible through the Help section of the platform.

2.7 Assumptions and Dependencies

- It is assumed that users will have internet access and a modern web browser.
- The system depends on third-party services for:
  - Email notifications (e.g., SMTP provider)
  - Cloud storage for large file hosting
- The system may rely on open-source libraries for certain components (e.g., calendar widgets, charts)
- Changes in education policies or data protection laws could affect requirements.
- The development team is assumed to have access to the necessary development tools and environments.

3. External Interface Requirements

3.1 User Interfaces

The Edura platform will provide a **graphical web-based interface** accessible via modern browsers. The user interface is designed to be **responsive**, **intuitive**, and **accessible** across desktop and mobile devices. It will follow standard web UI/UX conventions, ensuring consistency and ease of use.

General UI Characteristics:

- **Navigation bar** on all screens (Home, Courses, Profile, Notifications, Help).
- **Standard components**: Buttons, modals, dropdowns, and input forms.
- **Icons** from a consistent icon library (e.g., Font Awesome or Lucide).
- **Error Handling**: Inline validation with clear error messages, toast notifications.
- **Accessibility**: Keyboard navigation support, alt text for images, ARIA roles.
- **Help Access**: A persistent Help button that provides tooltips and links to documentation.

Layout Conventions:

- Responsive grid system using flexbox or Tailwind CSS.
- Dark/light mode toggle for user preference.
- Profile dropdown on the top-right corner for quick access to user settings.

Sample Interfaces (documented separately):

- **Login/Signup Screens**
- **Dashboard Overview**
- **Course Page with Modules**
- **Quiz Interface**
- **Grading Panel (Instructor Only)**

Note: A full UI design document and style guide will accompany the system as a separate deliverable.

3.2 Hardware Interfaces

Edura does not directly interface with specific hardware devices. However, it assumes access to:

- **Standard computing hardware**: desktops, laptops, tablets, and smartphones.
- **Peripheral support (optional)**:
    - Keyboard and mouse (desktop)

- Touch input (mobile/tablet)
- Microphone/camera (for future live session support)

- **No special hardware drivers** or proprietary hardware integration is required.

3.3 Software Interfaces

Component	Type	Purpose	Version/Protocol
Web Browser	Client	User interaction	HTML5, CSS3, JS (ES6+)
Operating System	Server-side	Host application server	Ubuntu 22.04 LTS or later
Backend Framework	Application	Application logic and routing	Node.js 20+ / Django 4+
Database System	Data Storage	Store user data, courses, assessments	PostgreSQL 14 / MySQL 8
SMTP Server	Email	Sending password resets, alerts	SMTP (TLS/SSL)
Cloud Storage API	File Storage	Upload and retrieve course materials	AWS S3 / Firebase Storage
REST API	Interface	Client-server communication	JSON over HTTPS
Authentication Library	Security	User login and session management	JWT / OAuth 2.0

Data Sharing

- JSON is used for data interchange between client and server.
- Secure REST APIs will handle authentication, data access, and updates.
- Static assets (videos, PDFs) will be stored and retrieved via signed URLs from cloud storage.

3.4 Communications Interfaces

Edura will support communication through the following channels and standards:

Communication Protocols:

- **HTTP/HTTPS**: Primary protocol for all web requests. HTTPS is mandatory for secure data transmission.

- **SMTP**: Used for email services such as password reset, notifications, and user verification.
- **WebSockets** *(future feature)*: May be used for real-time notifications or chat functionality.
- **RESTful API**: JSON-based APIs for client-server communication.

Security:

- **TLS encryption** for all communication channels.
- **OAuth 2.0 / JWT** tokens for secure authentication and session management.
- **CORS policy** configured to allow trusted domain access only.

Message Formatting:

- Emails will be sent in both HTML and plain-text formats.
- REST API responses will be standardized with the structure:

4. System Features

4.1 User Registration and Authentication

4.1.1 Description and Priority

This feature allows users to sign up, log in, and manage sessions securely.
Priority: High
Benefit: 9 | **Penalty**: 8 | **Cost**: 3 | **Risk**: 5

4.1.2 Stimulus/Response Sequences

- **User** clicks "Sign Up" → System displays registration form.
- **User** fills and submits valid data → System creates account, redirects to dashboard.
- **User** enters invalid info → System displays inline error messages.
- **User** logs in → System authenticates and grants access.

4.1.3 Functional Requirements

- **REQ-1**: System must allow users to register with email, password, and role (student/instructor).
- **REQ-2**: Passwords must be encrypted before storing in the database.
- **REQ-3**: System must validate email format and password strength.
- **REQ-4**: System must provide session management using JWT.
- **REQ-5**: System must restrict access based on user role after login.

4.2 Course Management

4.2.1 Description and Priority

Allows instructors to create, update, and organize courses. **Priority**: High

4.2.2 Stimulus/Response Sequences

- Instructor selects "Create Course" → System prompts for course info.
- Instructor submits → Course appears in their dashboard.

4.2.3 Functional Requirements

- **REQ-6**: Only instructors can create and manage courses.
- **REQ-7**: System must allow uploading of course material (PDF, video, links).
- **REQ-8**: System must allow instructors to organize content into module

---

4.3 Quiz and Assessment Module

4.3.1 Description and Priority

Students can take quizzes; instructors can create and grade them. **Priority**: Medium

4.3.2 Stimulus/Response Sequences

- Instructor creates a quiz → System stores with course module.
- Student takes quiz → System auto-grades MCQs and stores scores.

4.3.3 Functional Requirements

- **REQ-9**: System must support MCQs and short-answer questions.
- **REQ-10**: System must auto-grade MCQs.
- **REQ-11**: Quiz must have start/end time and attempt limits.

4.4 Progress Tracking

4.4.1 Description and Priority

Students can track their performance and progress in enrolled courses. **Priority**: Medium

4.4.2 Stimulus/Response Sequences

- Student visits dashboard → System displays progress bar and grades.

4.4.3 Functional Requirements

- **REQ-12**: System must display quiz scores, course completion status.
- **REQ-13**: Students should view grades for individual quizzes/modules.

---

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- System must support at least **100 concurrent users**.
- All pages must load in under **3 seconds** on a standard internet connection.
- File uploads (up to 100MB) must complete in **under 10 seconds**.

5.2 Safety Requirements

- Data backup must occur **daily** and be stored on a secure cloud service.
- Any system failure should be logged and notified to the admin within **5 minutes**.
- System must prevent duplicate submissions during quizzes.

5.3 Security Requirements

- Passwords must be **hashed using bcrypt** or equivalent.
- System must **auto-logout** inactive users after 30 minutes.
- **Role-based access control** must be implemented across all routes.
- All APIs must use **HTTPS only**.

5.4 Software Quality Attributes

- **Availability**: 99.5% uptime guaranteed.
- **Usability**: Interface must be user-friendly for non-technical users.
- **Portability**: System should run on any modern browser.
- **Maintainability**: Codebase must follow modular structure and naming conventions.
- **Reliability**: All critical operations must have fail-safes.

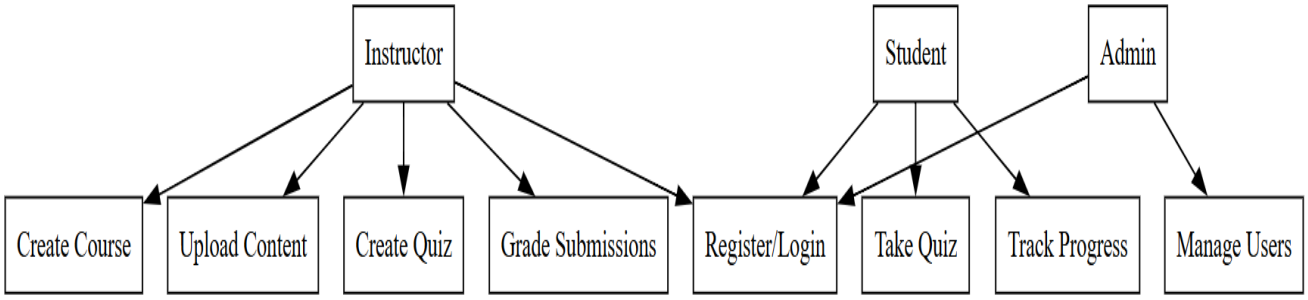
5.5 Business Rules

- Only instructors can create and manage content.
- Students can only access quizzes and content in enrolled courses.
- Admin can create user accounts and manage platform-wide content.

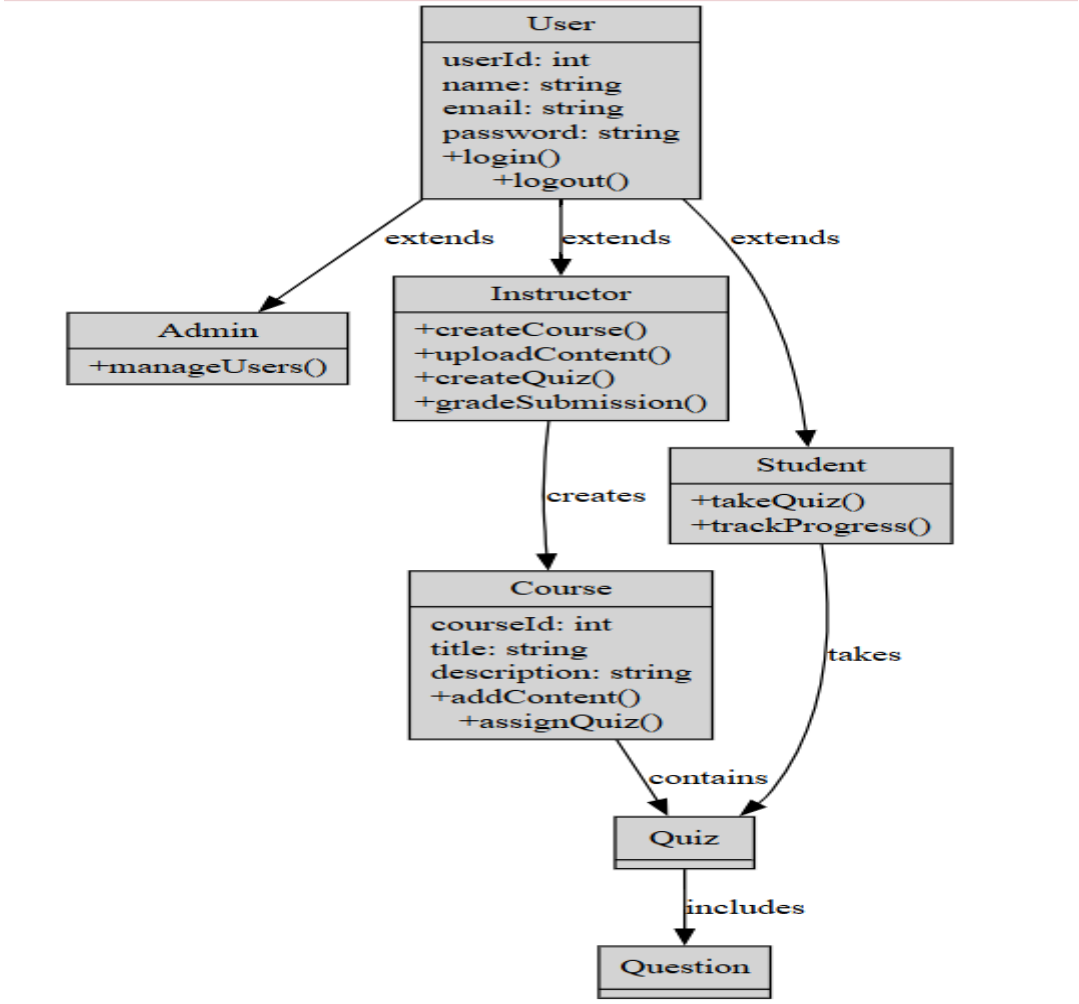
- Quiz attempts must be logged with timestamps.

6-Diagrams:

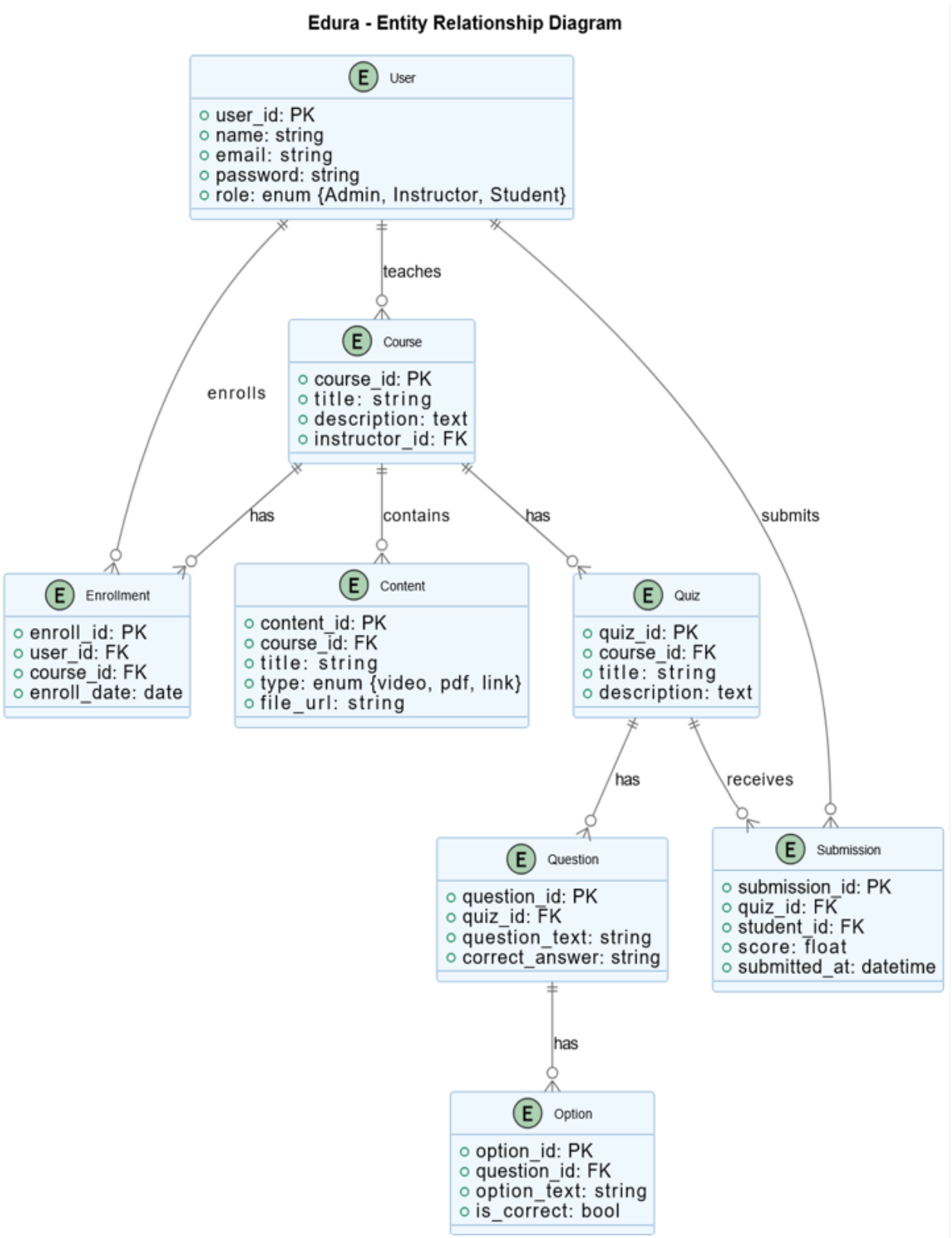
6.1-Use-Case Diagram:



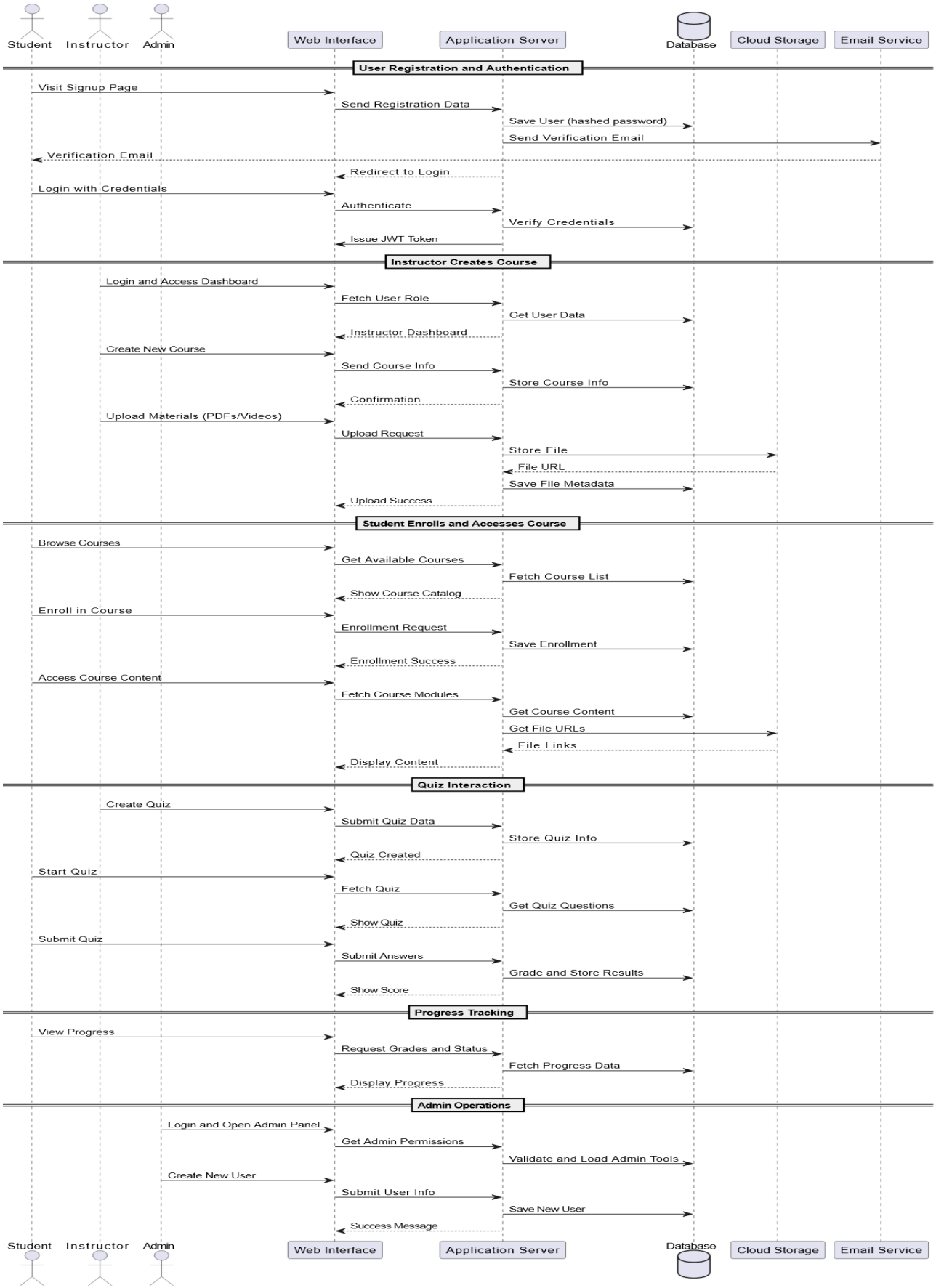
6.2-Class Diagram:



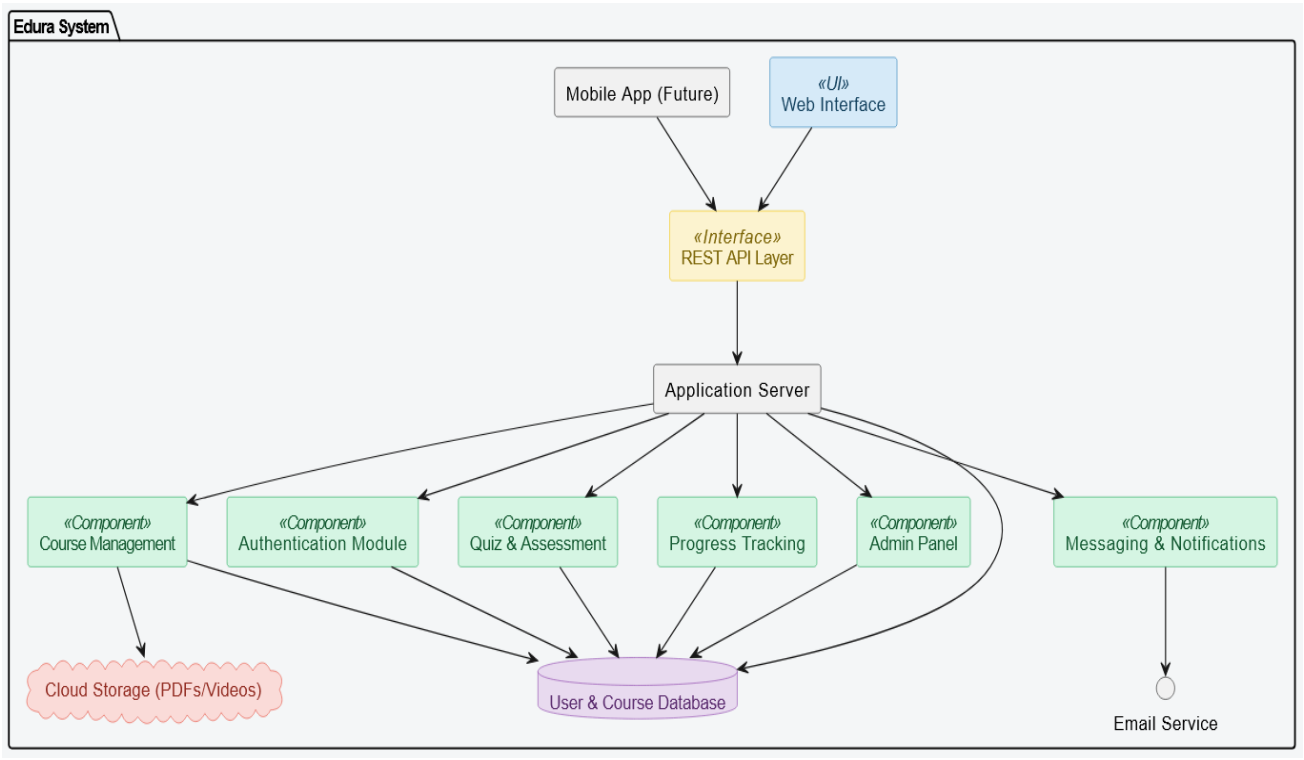
6.3-Entity Relationship Diagram:



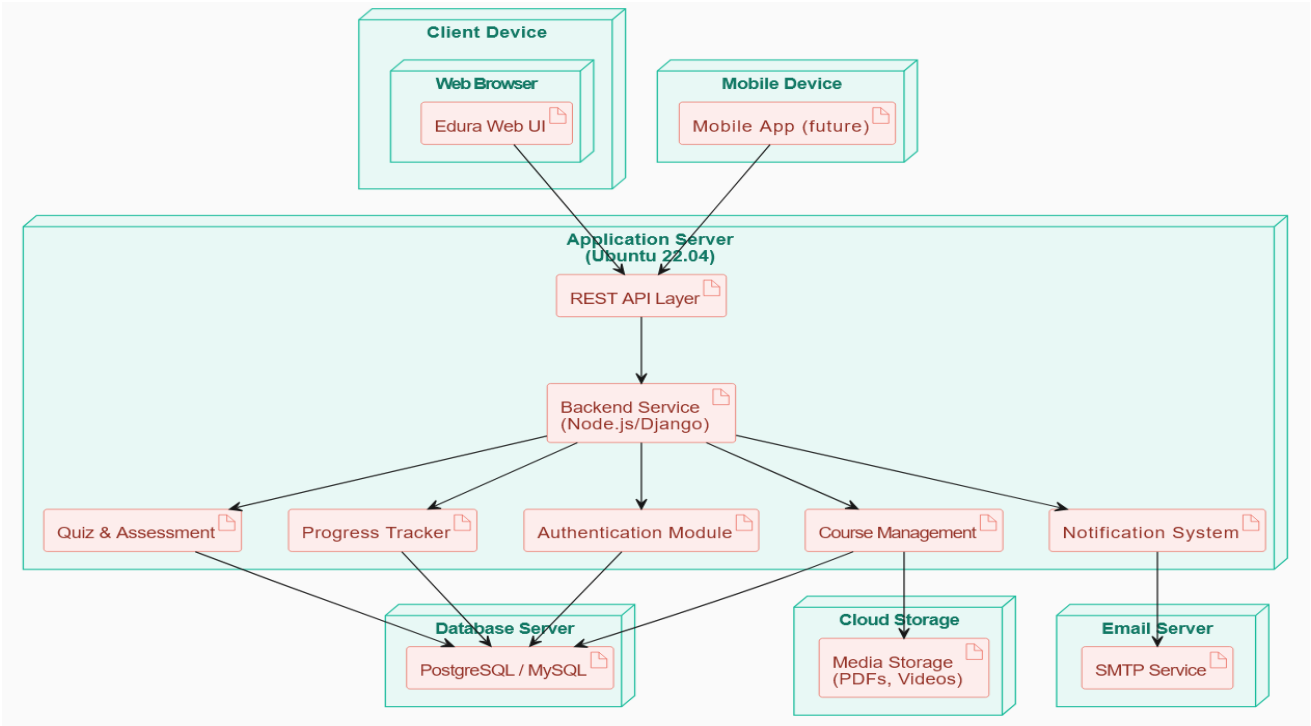
6.4-Sequence Diagram:



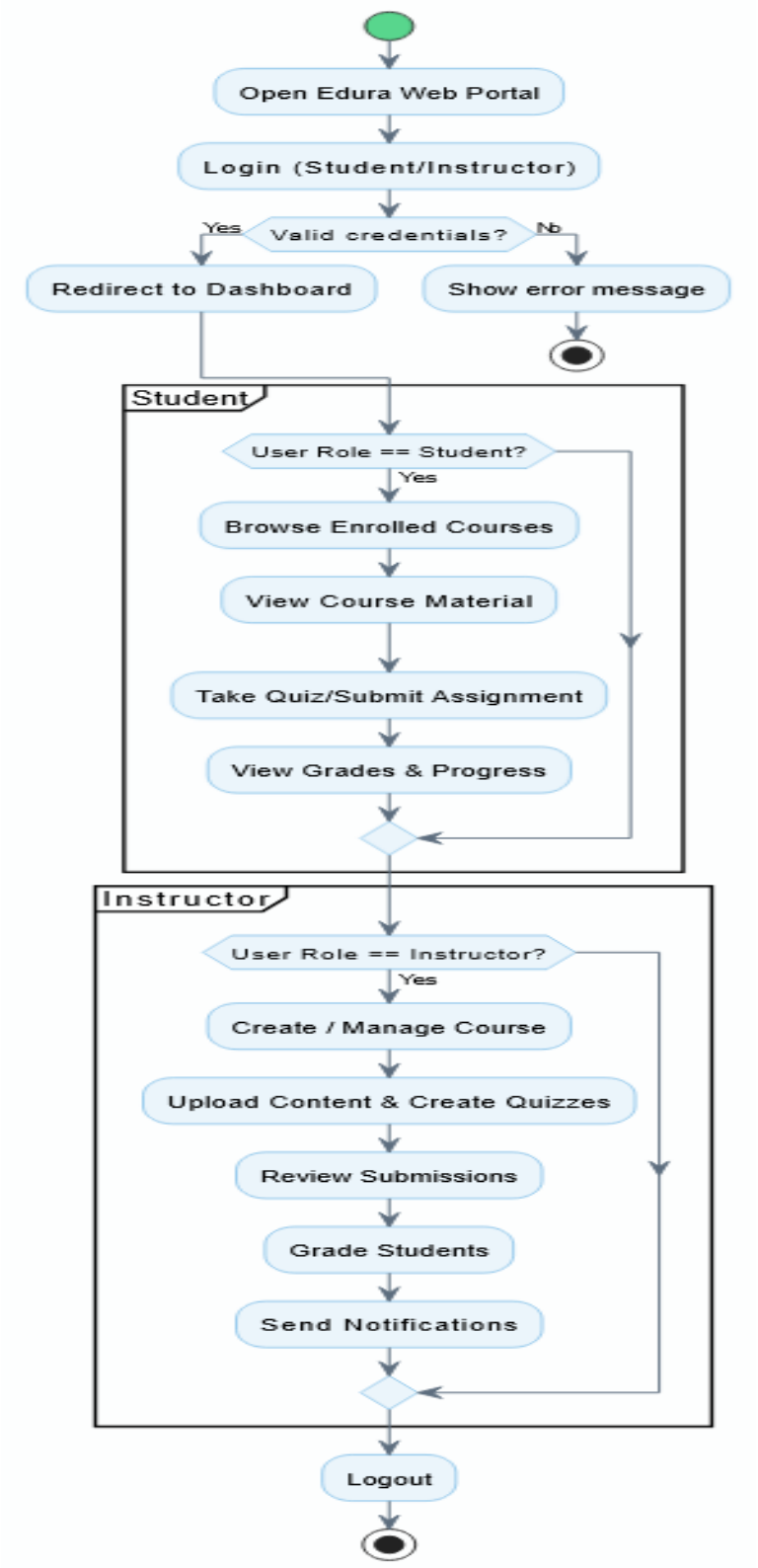
6.5-Component Diagram:



6.6- Deployment Diagram:



6.7- Activity Diagram:



Appendix:

Appendix A: Glossary

Term / Acronym	Definition
Edura	The e-learning platform being developed in this project.
UI	User Interface – the graphical layout through which users interact with the platform.
JWT	JSON Web Token – used for secure session management.
REST API	A set of web service endpoints that follow REST architecture for client-server communication.
GDPR	General Data Protection Regulation – legal framework governing data privacy in the EU.
LMS	Learning Management System – a system that helps deliver, track, and manage training/education.
MCQ	Multiple Choice Question – a type of assessment supported in quizzes.
ARIA	Accessible Rich Internet Applications – a set of attributes to improve accessibility.
WCAG	Web Content Accessibility Guidelines – a set of recommendations for making web content more accessible.

Appendix B: Analysis Models

The following diagrams have been included in the system specification:

1. **Use Case Diagram** – Shows interactions between users and system functions (e.g., register, manage course, take quiz).
2. **Class Diagram** – Illustrates the classes (e.g., User, Course, Quiz) and relationships.
3. **Entity Relationship Diagram (ERD)** – Depicts database entities and how they relate.
4. **Sequence Diagram** – Demonstrates sequence of operations like user login and quiz submission.
5. **Component Diagram** – Shows the modular structure of the system.
6. **Deployment Diagram** – Specifies the physical deployment architecture, including cloud and client-server components.

7. **Activity Diagram** – Represents workflow activities for core processes like course enrollment.

Appendix C: To Be Determined (TBD) List

The following items are identified as TBD and require further clarification:

#	Description
TBD-1	Exact list of supported languages for internationalization.
TBD-2	Selection of specific cloud storage service (AWS S3 vs Firebase).
TBD-3	Final UI/UX wireframes and color themes.
TBD-4	Real-time features like chat or video calls – whether and how to integrate.
TBD-5	Payment integration for course monetization (if applicable).
TBD-6	Detailed analytics dashboard for instructors (KPIs to track).

Conclusion:

The *Edura E-Learning Platform* is envisioned as a comprehensive, scalable, and user-friendly solution for modern digital education. This Software Requirements Specification (SRS) has provided a detailed overview of the platform's goals, system features, user interactions, functional and non-functional requirements, and external interfaces.

By defining clear system functionalities—such as course management, student assessments, progress tracking, and communication tools—this document lays the foundation for the successful design, development, and deployment of Edura. The inclusion of nonfunctional requirements like performance, security, and usability ensures that the platform will meet user expectations while complying with legal and technical standards.

Furthermore, supporting materials such as diagrams, a glossary, and the TBD list help ensure that all stakeholders have a shared understanding of the project. Remaining open items identified in the TBD list will be addressed during future development phases to finalize design decisions and implementation strategies.

This SRS will serve as a key reference throughout the lifecycle of Edura, from development and testing to deployment and maintenance. By adhering to the specifications outlined here, the project team will be better equipped to deliver a reliable, engaging, and accessible e-learning platform that empowers educators and learners alike.