**United International University**

A report on

**Software Requirement Specifications**

**System Analysis and Design**

CSE 3411

Fall 2024

Section A

Submitted to

Mr. Md. Nafis Tahmid Akhand

Lecturer, Dept. of CSE

Submitted by

Samira Ali- 011221015

Tamanna Yousuf- 011221016

Syed Mahiul Ahsan - 011221021

Submission Date

09-02-2025

**Contents**

1. **Introduction 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

2. **Overall Description 4**

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 4

2.4 Constraints 4

2.5 Assumptions and Dependencies 4

3. **Systems Features 5**

3.1 Functional Requirements 5

3.2 Non-functional Requirements 5

4. **System Architecture 5**

4.1 High-Level Design 5

4.2 Component Descriptions 5

5. **Use Cases 6**

5.1 Scenarios 6

5.2 User Stories 7

**6. Validation and Verification 7**

6.1 Testing Requirements 7

6.2 Acceptance Criteria 7

**7. Appendices 8**

7.1 Glossary 8

7.2 Additional Information 8

**1.Introduction**

**1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the E-Learning Management System (ELMS). This document serves as a guide for stakeholders, including project managers, developers, testers, and end-users, ensuring alignment on the project's objectives and deliverables.

**1.2 Scope**

The ELMS will provide a platform for online learning, enabling educators to deliver courses and students to access educational materials. Key features include course management, user authentication, progress tracking, multimedia support, and reporting. The system will support web and mobile platforms, ensuring accessibility for a diverse user base.

**1.3 Definitions, Acronyms, and Abbreviations**

**Definitions:**

* **E-Learning Management System (ELMS): A platform for managing and delivering educational content, tracking progress, and facilitating interaction between instructors and students.**
* **Learning Management System (LMS): A type of ELMS focused on structured course delivery and tracking.**
* **Asynchronous Learning: Self-paced learning without real-time interaction.**
* **Synchronous Learning: Real-time, live interaction between instructors and students.**

**Acronyms and Abbreviations:**

* **ELMS (E-Learning Management System): The platform for managing e-learning.**
* **LMS (Learning Management System): A focused type of ELMS for course management.**
* **SCORM (Sharable Content Object Reference Model): A standard for creating e-learning content.**
* **API (Application Programming Interface): A protocol for integrating tools or systems with the ELMS.**

**References**

* Project Charter Document
* IEEE Standards for SRS Documentation
* Similar LMS platforms for benchmarking

**2.Overall Description**

**2.1 Product Perspective**

The ELMS will be an independent system integrated with third-party tools for video conferencing, payment processing, and content hosting. It is designed to complement traditional learning by offering a robust digital solution.

**2.2 Product Functions**

* User Registration and Authentication
* Course Creation and Management
* Interactive Quizzes and Assignments
* Multimedia Content Support
* Progress Tracking and Reporting
* Communication Tools (e.g., messaging, forums)

**2.3 User Characteristics**

* **Students**: Individuals seeking educational content. They require an intuitive and engaging interface.
* **Instructors**: Professionals managing courses. They need robust tools for course creation and interaction.
* **Administrators**: Personnel overseeing the system. They need advanced reporting and management features.

**2.4 Constraints**

* Must comply with data protection regulations (e.g., GDPR, CCPA).
* Should support a minimum of 10,000 concurrent users.
* Mobile application must be compatible with Android and iOS.

**2.5 Assumptions and Dependencies**

* Users have stable internet access.
* The project depends on third-party APIs for video conferencing and payment gateways.

**3. System Features**

**3.1 Functional Requirements**

**User Registration and Login:** Allow users to securely register and log in to the platform.

**Course Management:** Enable instructors to create, update, and delete courses.

**Progress Tracking:** Provide real-time tracking of student progress in enrolled courses.

**Communication Tools:** Support forums and direct messaging between students and instructors.

**Assignment Management:** Facilitate assignment submission by students and grading by instructors.

**Notifications:** Send alerts for course updates, deadlines, and announcements.

**User Account Management:** Allow administrators to manage user accounts (add, update, deactivate) and monitor system activity.

**3.2 Non-functional Requirements**

**Performance:** Ensure the system responds to user actions within 2 seconds.

**Security:** Implement robust encryption for data storage and transmission.

**Usability:** Provide an intuitive and accessible interface that complies with usability standards.

**Scalability:** Support concurrent use by up to 10,000 users without performance degradation.

**Reliability:** Guarantee 99.9% uptime, minimizing disruptions to system availability.

**Compatibility:** Ensure the system works seamlessly across devices and browsers.

**4. System Architecture**

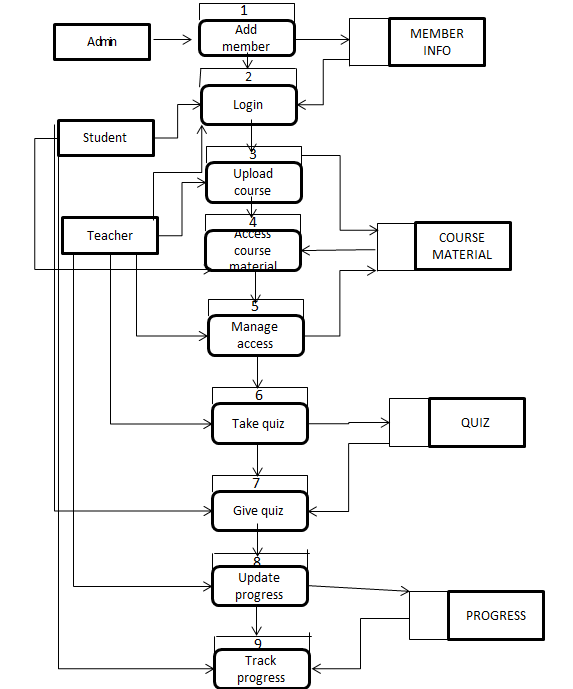
**4.1 High-Level Design**

The system will follow a modular architecture, with separate layers for presentation, business logic, and data storage. It will use a RESTful API to enable communication between components.

**4.2 Component Descriptions**

* **Frontend**: Web and mobile interfaces built using **React** and **Flutter.**
* **Backend**: Server-side logic implemented in **Node.js.**
* **Database**: A relational database (e.g., **MySQL**) for storing user and course data.

**Dataflow diagram:**

****

**5.Use Cases**

**5.1 Scenarios**

**User Registration and Authentication**

Users can register or log in to access the platform’s services. Registration stores user details securely, and the system verifies credentials to ensure authorized access for students, instructors, and administrators.

**Enroll in a Course**

Students can browse and enroll in courses after viewing details like descriptions and schedules. Successful enrollment updates their records and grants access to materials, with notifications confirming the process.

**Upload Course Content**

Instructors can upload and organize content such as videos and documents to create or update courses. The system ensures materials are accessible and organized for students.

**Submit Assignments**

Students can submit assignments through the platform. The system tracks deadlines, confirms submissions, and provides access for instructors to review and grade.

**Grade Assignments**

Instructors review assignments, assign grades, and provide feedback. Grades are recorded and shared with students for transparency.

**Track Progress**

Students can view progress reports summarizing grades and completed work, helping them identify areas for improvement and stay motivated.

**Communication Tools**

Students and instructors communicate via messaging, forums, or announcements to resolve queries and share updates.

**Manage User Accounts**

Administrators can add, update, or deactivate accounts and monitor system activity to ensure authorized access.

**Receive Notifications**

The system sends notifications about course updates, deadlines, and announcements, helping users stay informed and organized.

**View Performance Reports**

Students can access reports showing grades and course progress, helping them evaluate learning outcomes and prepare for assessments.

**5.2 User Stories**

**Student Enrollment:**

Includes: User Authentication

Student selects a course, enrolls, and gains access to course materials.

**Course Management:**

Includes: Content Delivery

Instructor creates, updates, deletes, and organizes courses.

**Content Delivery:**

Instructor uploads and organizes various types of content (videos, documents, etc.).

Assessment and Grading:

Instructor creates quizzes, grades assignments, and provides feedback.

Student submits assignments and takes quizzes.

**Communication:**

Student and Instructor communicate through messaging, forums, and announcements.

**Progress Tracking:**

Student views their progress in courses and receives performance reports.

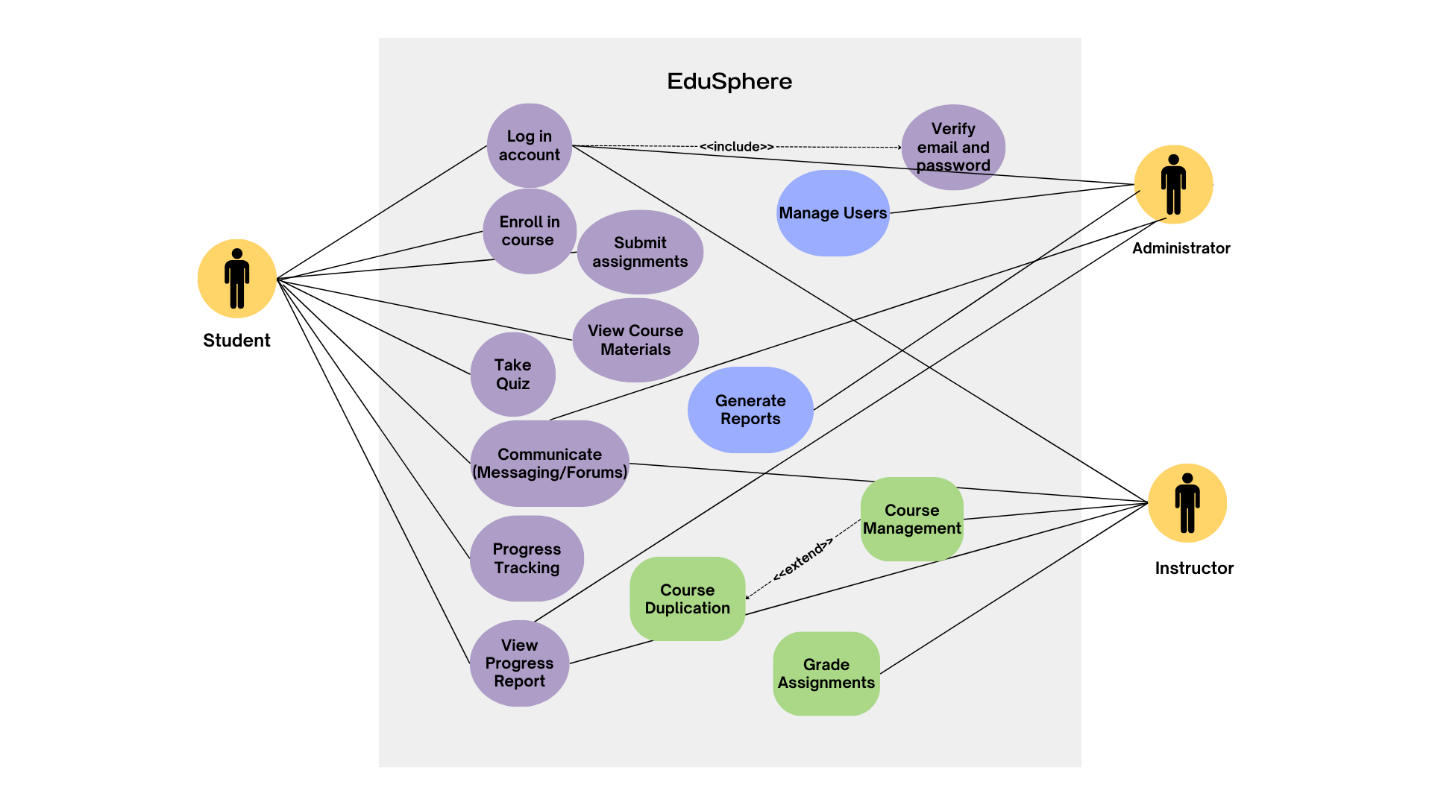
**User Authentication:**

Student, Instructor, and Administrator log in and access the system securely.

**System Administration:**

Administrator manages user accounts, monitors system performance, and handles administrative tasks.

**Use Case diagram:**



**6.Validation and Verification**

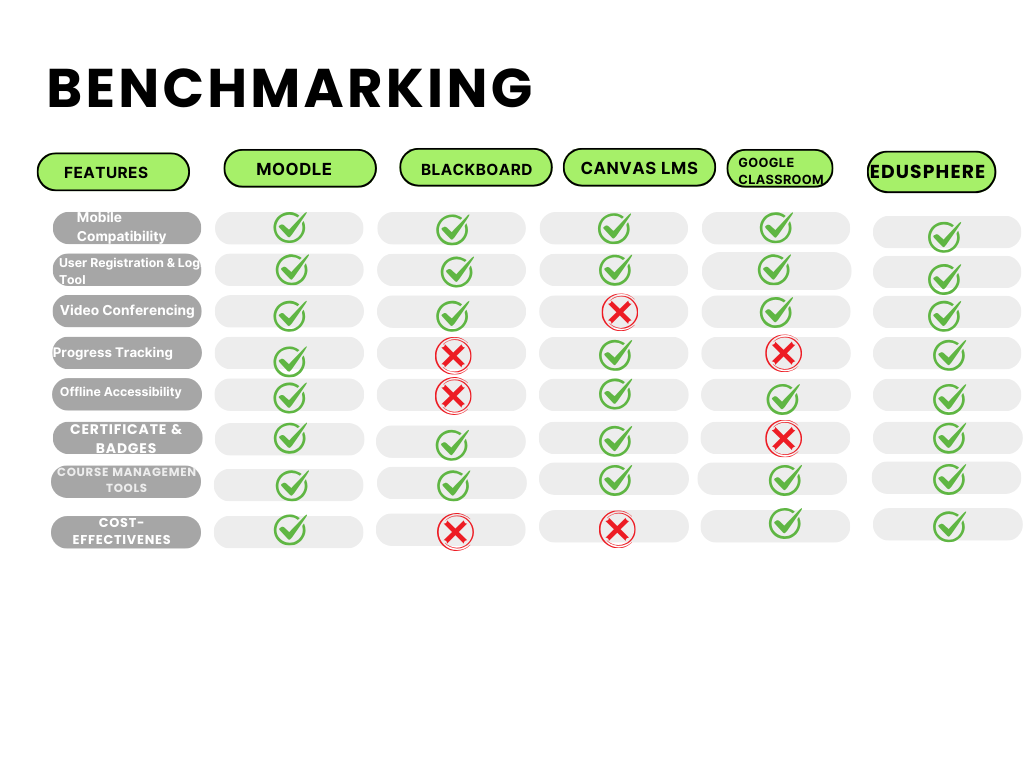
**6.1Testing Requirements**

* Unit Testing for individual components.
* Integration Testing to ensure smooth interaction between modules.
* User Acceptance Testing to validate system usability.

**6.2Acceptance Criteria**

* The system must successfully handle 10,000 concurrent users.
* All functional requirements must be fully implemented and tested.
* **Test case-:**

For the Student and Instructor Dashboards, various test cases will be executed to ensure functionality and reliability. One test case will verify whether students can accurately view their enrolled courses, including details like course progress, grades, and deadlines. The expected outcome is that all relevant data is displayed accurately for each student. Additionally, the instructor dashboard will be tested for the ability to manage courses, including uploading new content, viewing student progress, and responding to queries. Another test case will ensure that dashboards dynamically update with new changes, such as newly added courses or assignments. The expected outcome is real-time updates with accurate data displayed for both students and instructors. This testing aims to ensure a seamless user experience and maintain data accuracy across the platform.



**7.Appendices**

**7.1 Glossary**

* Concurrent Users: Users accessing the system simultaneously.
* Course Management: Tools for creating, updating, and managing courses.
* Data Encryption: Secure encoding of data to protect against unauthorized access.
* Frontend: The user interface component for accessing the system.
* Backend: The server-side logic responsible for processing requests and data.
* User Authentication: The process of verifying a user's identity.
* Progress Tracking: Monitoring and reporting on user performance.
* Multimedia Support: The ability to handle videos, PDFs, and other educational content.

**7.2 Additional Information**

* Estimated Project Completion: 6 months.
* Maintenance Plan: Regular updates and patches post-deployment.