Software Requirements Specification
for
Online Course Management System

Prepared by:

Fabiha Tasnim (2254901051)
Anika Afrida (2254901063)
Sadia Afrin Nirjhar (2254901065)
Afsana Tanjum Anonna (2254901083)
Umme Farhana Sumaiya (2254901127)

Bangladesh University of Professionals
Software Testing and Maintenance Lab

23 September 2025

Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the Course Management Web Application. This document serves as a reference for all stakeholders, ensuring a common understanding of the system's features, constraints, and goals. The SRS will guide the design, development, testing, and maintenance phases of the project. It is intended to prevent ambiguity and misinterpretation by providing a precise description of the product.

1.2 Intended Audience

1.3 Intended Use

The Course Management Web Application will be used as an online platform to:

- Allow **students** to enroll in courses, download study materials, take quizzes, track their progress, communicate with instructors, provide feedback, and obtain certificates.
- Enable **instructors** to create, edit, and manage course content, and quizzes, as well as communicate with students and review their progress.
- Provide **administrators** with the tools to manage users, oversee course quality, handle reported content, and ensure smooth operation of the system.
- Support both **free and paid courses**, with integrated payment processing for premium content.

1.4 Product Scope

The Online Course Management System is a web-based platform that supports three roles: student, instructor, and admin. Students can enroll in courses, access materials, take quizzes, submit assignments, and download certificates. Instructors can create and manage courses, upload content, post assignments, and interact with students. Admins can manage users, monitor activity, and oversee courses. The goal is to provide a secure, easy-to-use, and scalable system that improves learning and course management.

1.5 Risk Definition

The system may face risks such as data breaches, server downtime, data loss, misuse of feedback, or payment errors. These risks can affect user trust and learning quality. To reduce impact, the system will include proper security, backups, and monitoring.

Chapter 2

Overall Description

2.1 User Classes and Characteristics

Students: Access free/paid courses, attend quizzes, interact with instructors via chat.

Require basic computer literacy and ability to use a modern web browser.

Instructors: Upload and manage course materials, create quizzes, grade assignments, interact with students.

Should be comfortable using web applications for content management.

Administrators: Manage users, oversee course quality, handle payments, ensure compliance with regulations.

Require higher-level privileges and technical knowledge to maintain platform integrity.

2.2 User Needs

Students: Easy enrollment, secure payments, reliable course access, quizzes with instant grading, and certificate issuance.

Instructors: Simple tools to upload/update course content, manage quizzes, and monitor student progress.

Administrators: Dashboard for user management, course moderation, payment tracking, and compliance enforcement.

2.3 Operating Environment

Frontend: Web-based platform accessible through Chrome, Edge, Firefox, and other modern browsers.

Backend: Node.js with Express framework.

Database: MySQL for storing user, course, and transaction data.

Hosting: Secure cloud/server environment with backup support.

Network: Requires stable internet connection for video streaming, chat, and downloads.

2.4 Constraints

- **Resource Constraints:**
  - Development must be completed within the allocated timeline and budget.
  - Limited development resources (small team may restrict the initial scope of features.
- **Technological Constraints:**
  - Backend must be developed using Node.js with MySQL as the database.
  - Frontend must use HTML, CSS, Bootstrap/Tailwind, JavaScript (React optional).
  - All features (enrollment, quizzes, chat, payments) must be compatible within the selected tech stack.
  - The system must ensure data integrity and ACID compliance via MySQL.
- **External Constraints:**
  - The system must comply with relevant data privacy and security laws.
  - Payment systems must comply with financial regulations and encryption standards.
  - Any third-party APIs (payment gateway, email, or video hosting) must be available and reliable.
- **Operational Constraints:**
  - Only registered students and instructors can access course-related features.
  - One-to-one chat must be restricted to authenticated users (student, instructor, admin, or both).
  - Course material uploads must follow file size/type limitations defined by the system.
- **Performance Constraints:**
  - The system must support at least 200 concurrent users during peak hours.
  - The platform should maintain a response time of under 2 seconds for standard operations (login, course navigation).
  - File uploads/downloads must be optimized to avoid server overload.

2.5 Assumptions

Technical Assumptions:

– Users will access the platform primarily through modern web browsers.

– A stable internet connection will be available to users for login, course access, video streaming, and downloading course materials.

– The hosting server will support Node.js runtime and MySQL database connectivity.

– Email services or third-party APIs for password recovery and certificate generation will be accessible and reliable.

Operational Assumptions:

– Students and instructors will have basic computer literacy to navigate the platform.

– Instructors will be responsible for uploading course materials, managing quizzes, and updating their profiles.

– Students are expected to provide accurate information during registration and payment processes.

– Quizzes will be automatically graded based on predefined benchmarks set by the instructors/admin.

Business Assumptions:

– Paid courses will use an integrated payment gateway, assumed to comply with local regulations.

– Certificates generated by the system will be considered valid by the authority.

– AI-powered features will be integrated in future iterations, not in the first release.

Technical Assumptions:

– Paid courses will use an integrated payment gateway, assumed to comply with local regulations.

– Certificates generated by the system will be considered valid by the institution/university authority.

– AI-powered features will be integrated in future iterations, not in the first release.

Environmental Assumptions:

– The system will be deployed in a secure cloud or server environment with regular backups.

– The platform will remain accessible across different geographic regions unless restricted by local laws.

Chapter 3

Requirements

3.1 Functional Requirements

3.1.1 Student Functional Requirements

User story: As a student, I want to create an account and log in safely so I can use the platform. I want to browse courses by category, enroll in free or paid courses, and download materials to study offline. I can search for courses with keywords. I want to submit assignments, take quizzes (and retake them if needed), and track my progress on a dashboard. I message instructors, rate and comment on courses, and download my course certificates and payment receipts as proof of my learning.

Functional Requirements for Students:

- Account creation and login/logout functionality with secure authentication.
- Browse and search courses by category.
- Course enrollment (free and paid) with access to materials.
- Assignment submission and storage.
- A quiz with retake functionality and result storage.
- Progress tracking dashboard.
- Messaging system to communicate with instructors.
- Rating and commenting on courses.
- Certificate and receipt generation for completed/paid courses.

3.1.2 Instructor (Teacher)Functional Requirements

As an instructor, I want to log in securely so that only I can access my account. I want to manage my profile to keep my information up-to-date. I want to create and manage courses so that students can enroll and learn. I want to post assignments for students to complete. I want to create and manage quiz questions to test students' knowledge. I want to communicate with students through messages so that I can answer their questions. I want to view course feedback so that I can monitor student progress and improve the quality of my courses.

Functional Requirements for Instructor:

- Secure login and profile management.
- Course creation, editing, deletion, and categorization.
- Upload course materials (PDFs, videos, slides).
- Post and manage assignments.
- Create and manage quiz questions.
- Messaging system to reply to student queries.
- View and analyze student feedback for courses.

3.1.3 Admin Functional Requirements

As an admin, I want to log in securely so that only authorized users can access my account. I want to manage student and instructor accounts to ensure proper access and maintain user information. I want to create, edit, delete, and categorize courses so that the platform is organized and easy to navigate. I want to monitor student and instructor activity to track engagement and performance. I want to review course feedback to maintain quality standards. I want to approve paid course enrollments to ensure valid transactions. I want to generate reports so that I can analyze platform usage and make informed decisions.

Functional Requirements for Admin:

- Manage student and instructor accounts (view, edit, delete, block).
- Secure login and role-based access control.
- Add, edit, delete, and categorize courses.
- Monitor course feedback, ratings, and student/instructor activity.
- Approve and verify paid course enrollments.
- Generate analytics and performance reports (most enrolled courses, top-rated courses, etc.).

3.2 Non Functional Requirements

The system should not only provide the required features but also meet quality, security, and performance standards. These requirements ensure that the platform is reliable, secure, and user-friendly.

3.2.1 Performance

The system should support at least 100 concurrent users without significant delays. Course material downloads and quiz submissions should respond within 2 seconds. Page loading times should remain consistent even during peak usage.

3.2.2 Security

All user passwords must be encrypted using industry-standard hashing (e.g., bcrypt). Unauthorized users should not access admin or instructor features. Session management should prevent session hijacking. Sensitive data such as payment information and certificates must be securely stored.

3.2.3 Safety

Data backup must occur regularly to prevent loss of student submissions, certificates, and course content. The system should handle errors gracefully to prevent data corruption.

3.2.4 Quality

The platform should be responsive and compatible with major browsers, including Chrome, Firefox, and Edge. The user interface should be intuitive and accessible. Proper labels, navigation, and error messages should guide users. Input validation must prevent invalid data entry and maintain data integrity.