Project Documentation

On

Learning Management System (LMS)

By Khairul Basar

Intern ID: 16

Table of Contents

1. Introduction

The **Learning Management System (LMS)** is a web-based platform designed to streamline online education and training. It enables students to enroll in courses, access materials, submit assignments, and track progress. Instructors can create courses, upload content, and evaluate student performance, while admins oversee user management, analytics, and system configuration.

2. Objectives

- Provide a platform for online learning and course management.
- Enable students to enroll in courses, access materials, and track progress.
- Allow instructors to create and manage courses, assignments, and quizzes.
- Facilitate interaction between students and instructors through discussion forums.
- Generate certificates upon course completion.
- Provide an admin dashboard for system management and analytics.

3. Functional Requirements

Functional requirements define **what the system should do**.

3.1 Student Features

- **Authentication**:
    - Register, log in, update profile, and reset password.
- **Course Interaction**:
    - Browse and enroll in courses.
    - Access course materials (videos, PDFs, documents).
    - Submit assignments and take quizzes.
    - Track progress (completed modules, grades).
- **Communication**:
    - Participate in discussion forums.
    - Receive notifications (new assignments, grades).
- **Certification**:
    - Download certificates upon course completion.

3.2 Instructor Features

- **Authentication**:
  - Register, log in, and update profile.
- **Course Management**:
  - Create, update, and delete courses.
  - Upload course materials (videos, PDFs, documents).
- **Assessment**:
  - Create assignments and quizzes.
  - Grade submissions and provide feedback.
- **Interaction**:
  - Engage in discussion forums.

3.3 Admin Features

- **Course Management**:
  - Approve/delete courses and categories.
- **Analytics**:
  - View system metrics (users, courses, enrollments).
  - Generate reports (course completion rates, user activity).

3.4 Core System Features

- **User Authentication**: Secure login using Spring Security + JWT.
- **Course Management**: CRUD operations for courses and materials.
- **Enrollment & Progress Tracking**: Track student enrollment and progress.
- **Assignments & Quizzes**: Create, submit, and grade assessments.
- **Discussion Forums**: Real-time interaction via WebSocket/SSE.
- **Notifications**: Alerts for deadlines, grades, and updates.
- **Certification**: Automated certificate generation.
- **Admin Dashboard**: Manage users, courses, and analytics.

4. Non-Functional Requirements

Non-functional requirements define **how the system should perform**.

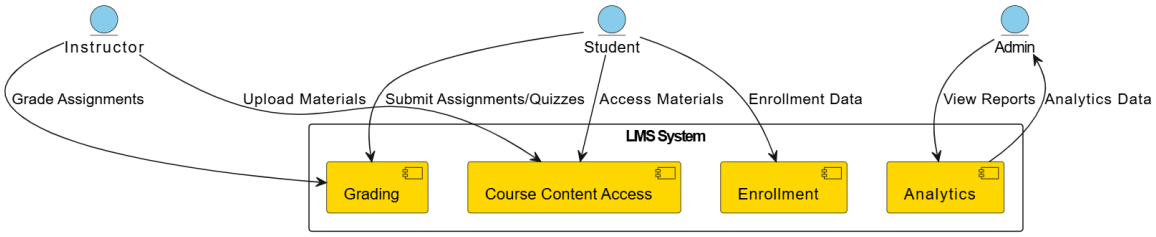
Category	Requirements
Performance	- Handle 1,000 concurrent users. - Page load time <2 seconds.
Scalability	- Use AWS/Kubernetes for horizontal scaling.
Security	- Encrypt passwords (AES-256). - Prevent SQL injection/XSS attacks.
Usability	- Intuitive UI with responsive design (desktop, tablet, mobile).
Reliability	- 99.9% uptime. - Automated backups for data recovery.
Maintainability	- Modular code structure. - Comprehensive developer documentation.
Compatibility	- Support Chrome, Firefox, Safari, Edge. - Accept PDF, MP4, DOCX files.
Accessibility	- WCAG 2.1 compliance (alt text, captions).

5. Structured Analysis

5.1 System Components

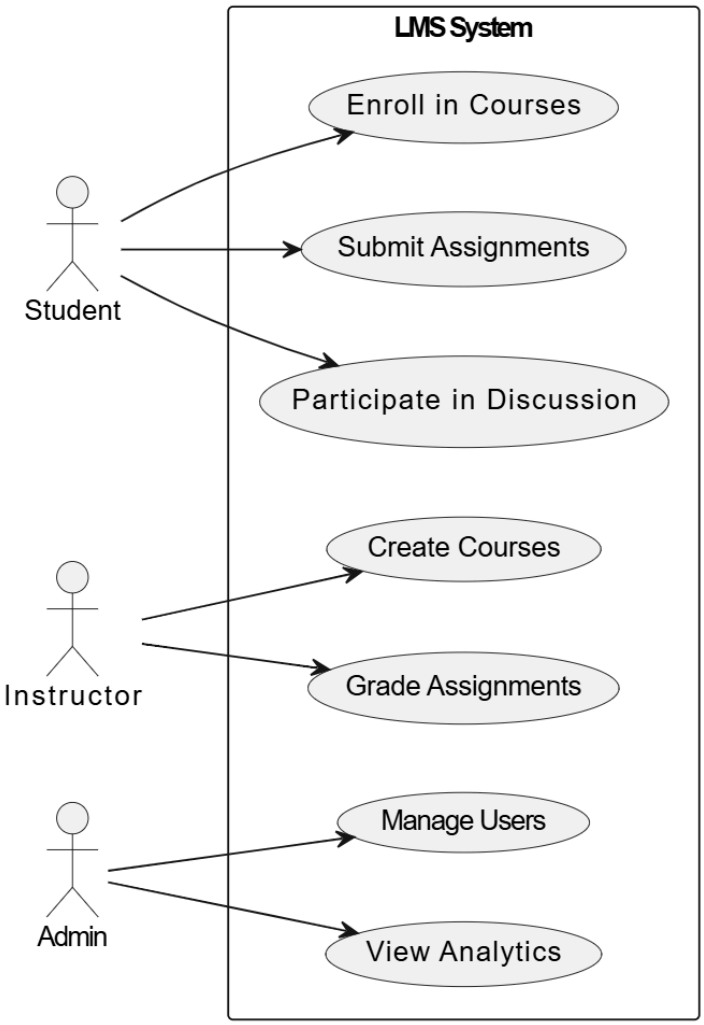
1. **User Management Module**: Handles authentication and role-based access.
2. **Course Management Module**: Manages courses, materials, and categories.
3. **Enrollment Module**: Processes course enrollment and tracks progress.
4. **Assignment/Quiz Module**: Creates, submits, and grades assessments.
5. **Discussion Forum Module**: Enables real-time communication.
6. **Notification Module**: Triggers alerts for key events.
7. **Certification Module**: Generates course completion certificates.
8. **Admin Dashboard**: Centralized system management and analytics.

5.2 Data Flow Diagrams (DFD)

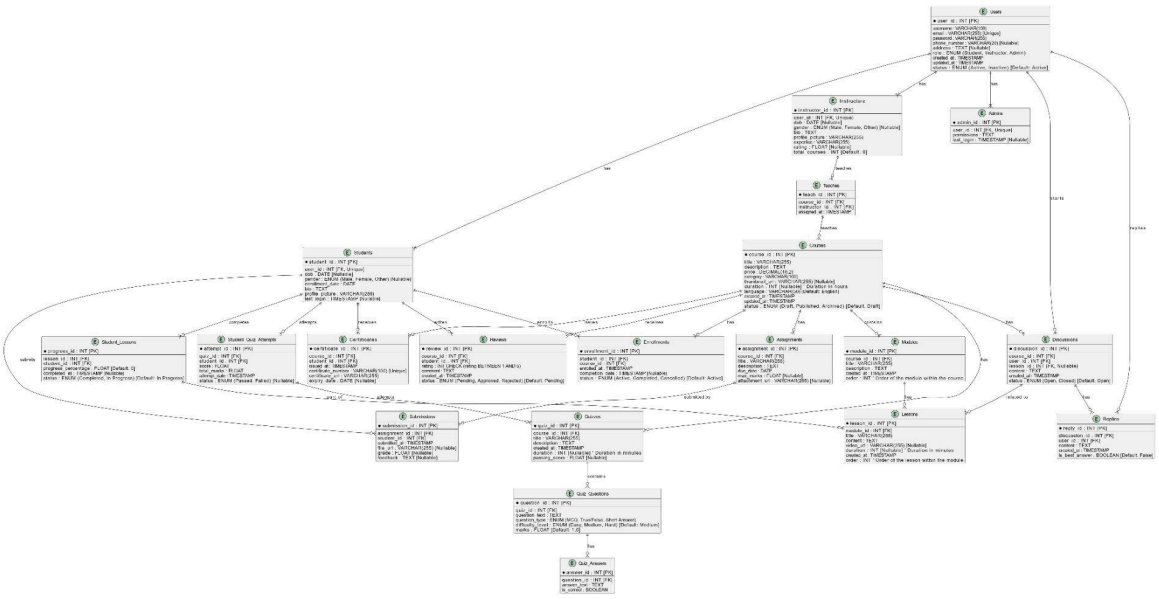


5.3 Use Case Diagram

- **Actors**: Student, Instructor, Admin.



5.4 Entity-Relationship Diagram (ERD)



6. Technology Stack

Component	Technologies
Backend	Java Spring Boot
Frontend	React/Thymeleaf, HTML, CSS
Database	MySQL/PostgreSQL
Authentication	Spring Security + JWT
File Storage	AWS S3
Real-Time Updates	WebSocket/Server-Sent Events (SSE)
Deployment	Docker, Kubernetes, AWS/Render

7. Timeline

Phase	Duration	Deliverables
Project Setup	1 week	Requirement analysis, DB design, Spring Boot setup.
User Authentication	2 weeks	Secure login, registration, password reset.
Course Management	2 weeks	CRUD operations for courses and materials.
Enrollment & Progress	1 week	Enrollment system and progress tracking.
Assignments & Quizzes	2 weeks	Creation, submission, and grading features.
Discussion Forums	1 week	Real-time forum integration.
Notifications	1 week	Alert system for updates and deadlines.
Certification	1 week	Automated certificate generation.
Admin Dashboard	1 week	User/course management and analytics.
Testing & Deployment	2 weeks	Unit/integration testing, deployment to AWS.
Total	**14 weeks**	MVP ready for launch.

8. Expected Outcomes

- A **fully functional LMS** with user authentication, course management, assignments, quizzes, forums, notifications, and certification.
- A **scalable and secure system** capable of handling 10,000+ users.
- An **admin dashboard** for managing users, courses, and analytics.