Software Requirements Specifications Document

Learning Management System (LMS)

Prepared By

Jeevansree E

Table of Contents

1. Introduction

1.1 Purpose

This SRS document specifies the functional and non-functional requirements for an Online Learning Management System (LMS) designed to manage courses, batches, users, and learning materials. The system will automate course delivery, performance monitoring, and communication, replacing the current manual processes involving WhatsApp and separate links for tests and videos.

1.2 Scope

The LMS will:

- Manage courses, batches, and users.
- Allow Super Admins to create, assign, and manage courses.
- Enable Admins to edit their own courses, upload materials, and monitor student performance.
- Allow Users (Students) to access assigned courses, learning materials, and view performance analytics.
- Automate the delivery of tests, videos, and reports previously managed through WhatsApp and external links.

1.3 Definitions, Acronyms, and Abbreviations

- **LMS** : Learning Management System
- **Super Admin** : System Owner / Manager
- **Admin** : Teacher / Instructor
- **User** : Student
- **Batch** : Group of students for a specific year/month

2. Overall Description

2.1 Product Perspective

The LMS will be a centralized, web-based system accessible via browsers. It will support:

- **Super Admin**: Full control over courses, batches, and user assignments.
- **Admin** : Manage own courses, upload materials, monitor student performance.
- **User** : Access learning materials, view analytics, attempt tests.

2.2 Product Functions

Super Admin:

- Create and delete courses.
- Assign teachers to courses and batches.
- Monitor system-wide activities and analytics.

Admin :

- Edit own courses and materials.

- Upload videos, tests, and other resources.
- View student scores, analytics, and performance reports.

User :

- Access assigned courses and materials.
- View personal performance analytics.
- Attempt tests and assessments.

2.3 User Characteristics

User Role	Characteristics
Super Admin	System owner, manages overall platform.
Admin	Responsible for course delivery and student monitoring.
User	Learner, accesses materials and assessments.

2.4 Constraints

- Web-based application; requires internet access.
- Secure authentication for all users.
- Scalability for multiple batches, courses, and users.
- Integration of test and video delivery within the system.

2.5 Assumptions and Dependencies

- Internet connectivity is required.
- Cloud hosting and storage are used for scalability.
- Existing tests and videos will be migrated into the LMS.

3.Functional Requirements

3.1 User Roles and Permissions

The system shall support three types of users:

- Super Admin: Can create, manage, and delete courses, assign teachers (admins), and assign students to batches.
- Admin (Teacher): Can manage their own courses, create assignments/tests, upload learning materials, monitor student performance, and access reports.
- User (Student): Can view assigned courses, access learning materials, submit assignments, take tests, and view their own scores and performance reports.

3.2 Course and Batch Management

- The system shall allow the Super Admin to create, monitor, and delete courses.
- The system shall allow the Super Admin to assign teachers (Admins) to specific courses.
- The system shall allow the Super Admin to assign students (Users) to specific batches based on the year and month of admission (e.g., B1, B2, B3).
- The system shall allow the Super Admin to define n number of batches per year/month.

3.3 Learning Materials Management

- The system shall allow Admins and Super Admins to upload learning materials such as PDFs, videos, presentations, and external resource links for their respective courses.
- The system shall allow Users to access and download/view course materials for their enrolled courses.
- The system shall organize learning materials under relevant courses and batches for easy access by students.

3.4 Assignment Creation and Management

- The system shall allow Admins to create assignments.
- The system shall allow Users to access their assignments, instructions, deadlines, and submission options for enrolled courses.

3.5 Test Management

- The system shall provide a secure test environment that prevents tab switching and copying.
- The system shall auto-submit tests upon timeout and deny re-entry after submission.
- The system shall allow Admins (Teachers) to create online tests/assessments.
- The system shall allow Admins to add questions, correct answers, and allocate marks for each question.
- The system shall generate analytics and reports such as total scores, highest and lowest marks, averages, and performance graphs.
- The system shall allow Students to view their own scores and feedback after test completion.

3.6 Conversation Forum

- The system shall provide a course-specific forum for discussions between students and the course admin.
- Students shall be able to post questions and reply to threads.

3.7 Question of the Day and Leaderboard

- The system shall allow the Super Admin to upload a daily question for all students.
- Students shall be able to submit their answers.
- The system shall automatically evaluate and update a leaderboard.

3.8 Code Compiler for Practice

- The system shall integrate an in-browser code editor with a backend execution engine.
- The compiler shall support at least one programming language for student practice.

- The system shall display output/errors for the code submitted by the student.

3.9 Admin–Super Admin Chat Interface

- The system shall allow Admins to send real-time or asynchronous messages to the Super Admin.
- The system shall maintain a history of conversations between Admin and Super Admin.

3.10 Performance Analytics and Reports

- The system shall allow Admins and Super Admins to view performance analytics for each batch, course, and student.

4. Non-Functional Requirements

4.1 Performance

- The system shall support at least 500 concurrent users with minimal latency.
- API response times should not exceed 2 seconds under normal conditions.

4.2 Scalability

- The LMS shall support the onboarding of multiple batches without needing architectural changes.
- All services shall be horizontally scalable.

4.3 Security

- The system shall ensure secure login using JWT authentication.
- All sensitive data such as passwords details shall be encrypted.
- The protected test environment shall disable right-click, copy/paste, and tab switch detection.

4.4 Usability

- The user interface shall be intuitive, accessible, and responsive across devices.
- Features like video progress tracking and code compilation shall be easy to use.

4.5 Availability

- The system shall maintain 99.9% uptime.
- All system-critical features (like test and submission) shall have monitoring and fallback mechanisms.

4.6 Compatibility

- The LMS shall work on the latest versions of Chrome, Firefox, and Edge.

5.Tech Stack & Tools

Frontend:

- **React.js** (UI)
- **Tailwind CSS** or **Bootstrap** (styling)
- **Socket.IO** (for chat/forum)

Backend:

- **Node.js + Express** (REST APIs)
- **Firebase** (Real-time components like chat/forum)
- **JWT** (Authentication)

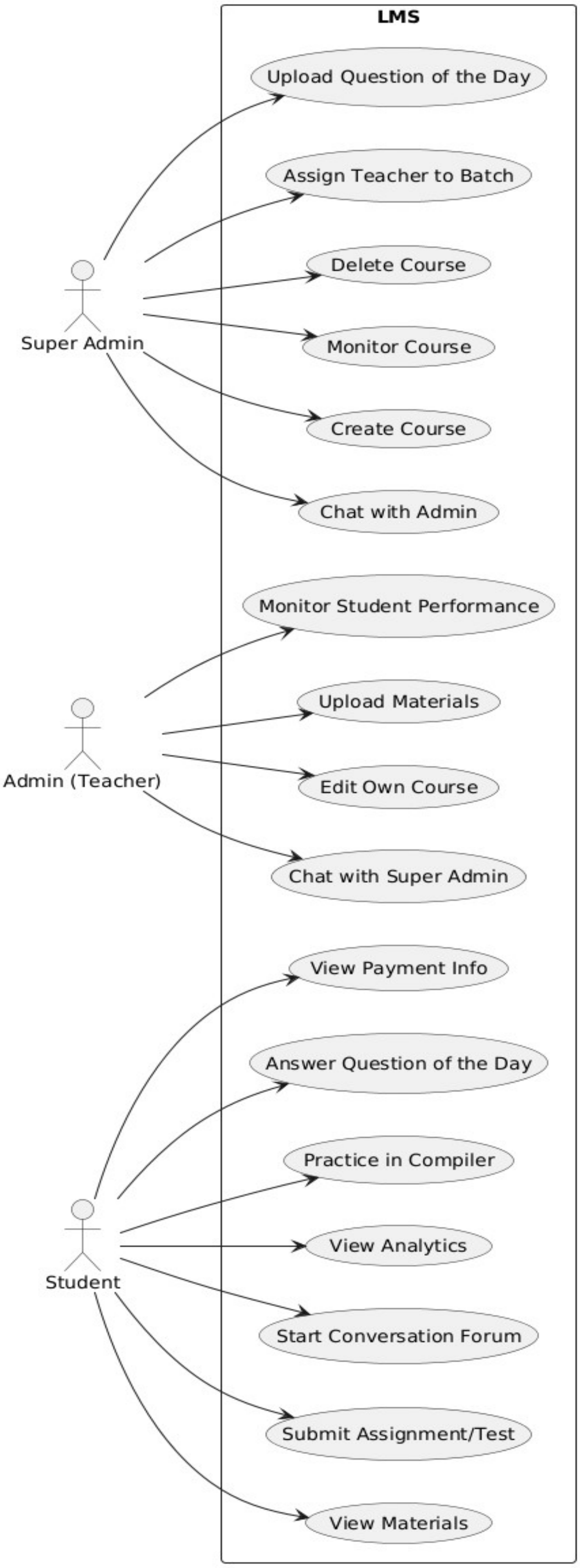
Database:

- **MongoDB** (Courses, Users, Batches, etc.)
- **Firebase Firestore** (Chat, real-time forum)

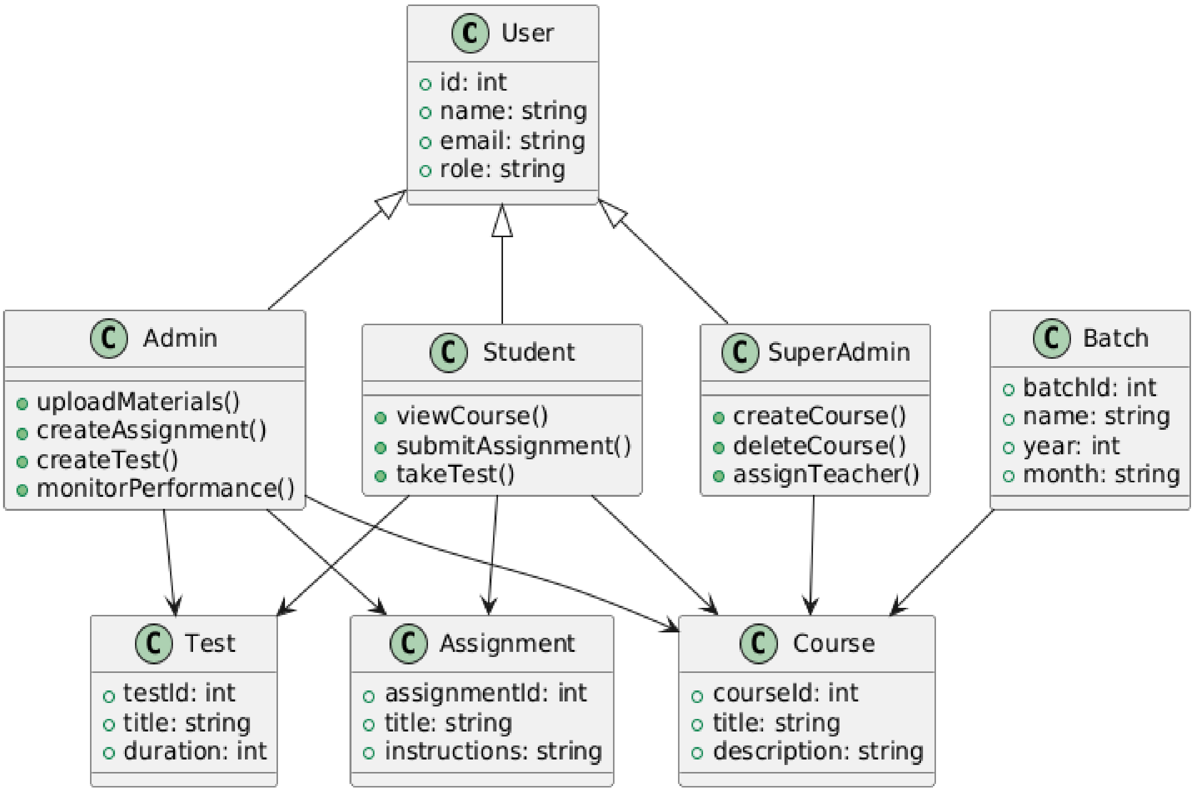
Storage:

Firebase Storage (for videos, certificates)

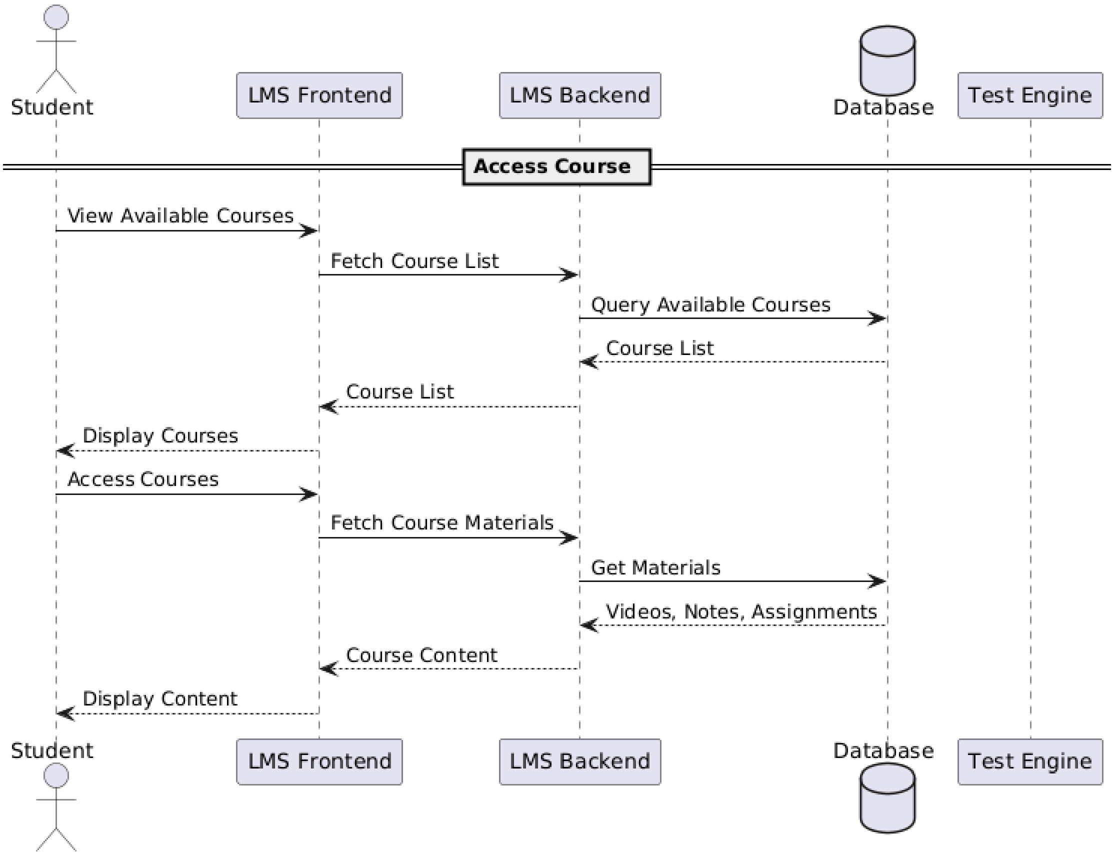
6. Use Case

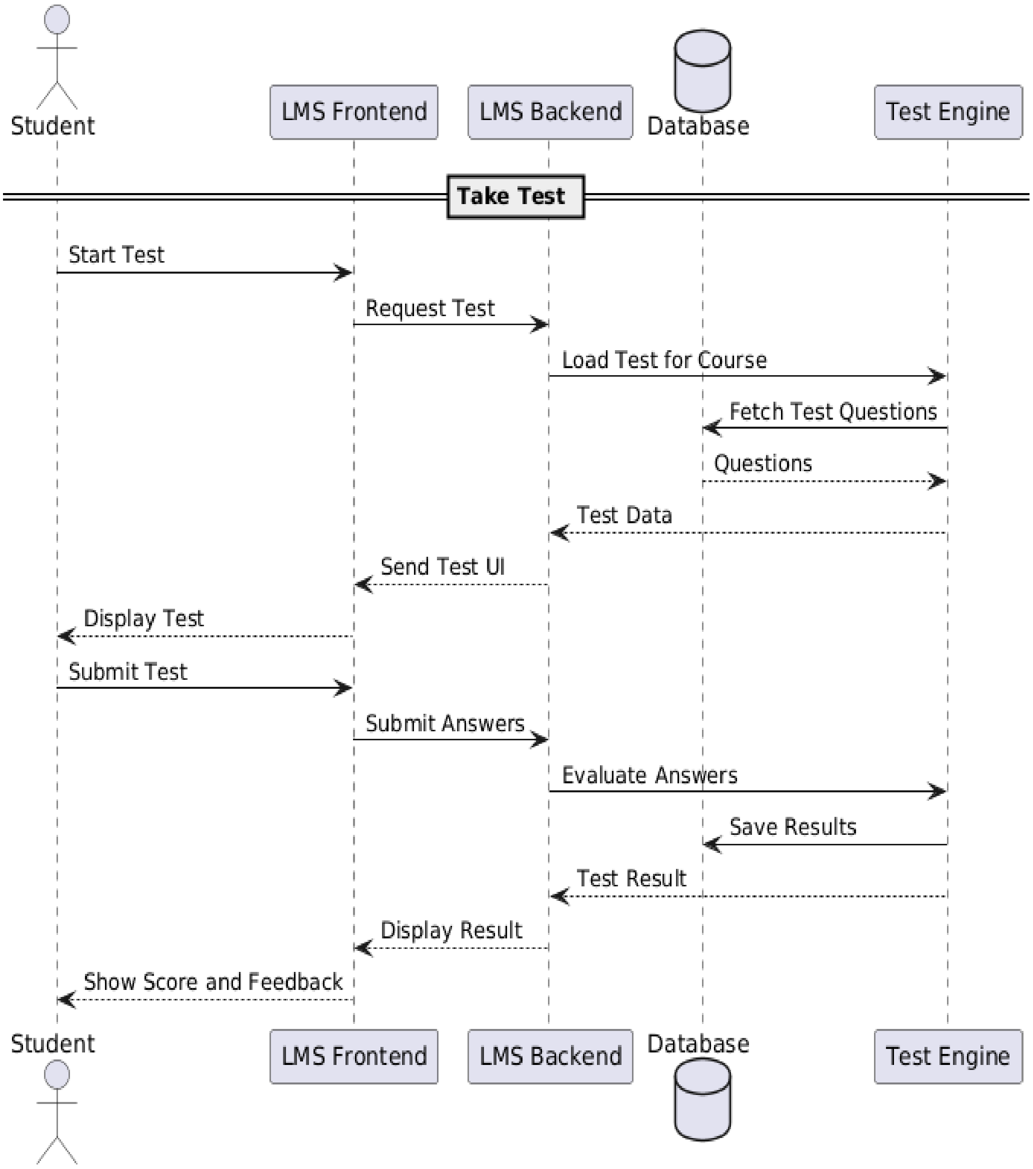

LMS
Upload Question of the Day
Assign Teacher to Batch
Delete Course
Monitor Course
Create Course
Chat with Admin
Super Admin
Monitor Student Performance
Upload Materials
Edit Own Course
Chat with Super Admin
Admin (Teacher)
View Payment Info
Answer Question of the Day
Practice in Compiler
View Analytics
Start Conversation Forum
Submit Assignment/Test
View Materials
Student

7. Class Diagram



8. Sequence Diagram





9.Conclusion

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements, as well as the system features and diagrams for the Learning Management System (LMS). The system is designed to streamline learning activities by enabling Super Admins, Admins (Teachers), and Students to interact within a centralized platform. By automating course management, content delivery, assessments, and performance tracking, the LMS aims to enhance the overall learning experience, eliminate manual processes, and provide a scalable solution for educational institutions.