

System Architecture Document for EduLearn LMS

Shivaramakrishna Aravapally, Surya Prakash Reddy Pachika

September 2024

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
2	System Architecture	2
2.1	Overview	2
2.2	Architecture Diagram	2
2.3	Components	2
2.3.1	User Service	2
2.3.2	Teacher Service	3
2.3.3	Student Service	3
2.3.4	Assignment Service	3
2.3.5	Material Service	3
2.3.6	Video Service	3
2.3.7	Session Service	3
3	Component Interactions	3
3.1	Service Communication	3
3.2	Data Flow	3
4	Design Considerations	4
4.1	Scalability	4
4.2	Security	4
4.3	Reliability	4
5	Conclusion	4

1 Introduction

1.1 Purpose

The purpose of this document is to describe the system architecture of the Learning Management System (LMS) application. It provides a high-level overview of the components, their interactions, and the overall structure of the system.

1.2 Scope

This document covers the architecture of the LMS application, focusing on the design principles, major components, and their interactions. It is intended for developers, system architects, and other stakeholders involved in the design and development of the LMS.

2 System Architecture

2.1 Overview

The LMS application is built on a microservices architecture, where each service is responsible for a specific piece of functionality. The system is designed to be scalable, reliable, and secure, ensuring that it can handle a large number of concurrent users and data.

2.2 Architecture Diagram

The following diagram illustrates the overall architecture of the LMS application. Each component is represented as a separate microservice that interacts with others through RESTful APIs.

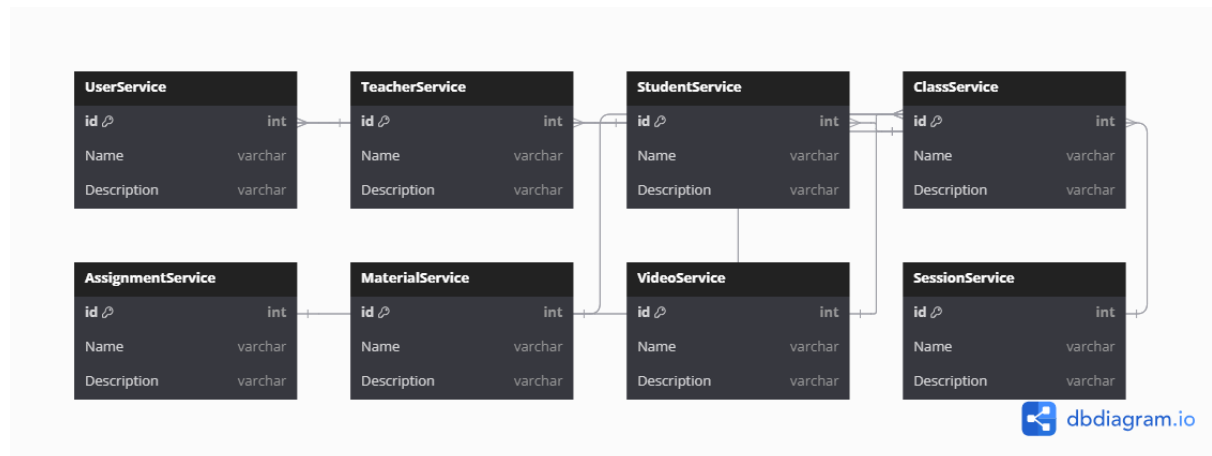


Figure 1: LMS System Architecture Diagram

2.3 Components

2.3.1 User Service

The User Service is responsible for handling user authentication, registration, and profile management. It stores user credentials and roles and interacts with other services to provide authentication and authorization.

2.3.2 Teacher Service

The Teacher Service manages all teacher-related functionalities, including class creation, student management, and assignment distribution. It interacts with the Student Service, Assignment Service, and Material Service.

2.3.3 Student Service

The Student Service handles student-related functionalities such as assignment submission, viewing materials, and accessing class information. It interacts with the Teacher Service and Assignment Service.

2.3.4 Assignment Service

The Assignment Service manages the creation, distribution, submission, and grading of assignments. It stores assignment details and tracks submissions from students.

2.3.5 Material Service

The Material Service handles the uploading and management of learning materials such as PDFs, documents, and presentations. It ensures that materials are accessible only to authorized users.

2.3.6 Video Service

The Video Service manages the uploading and streaming of video content. It provides access to video lectures and other multimedia content linked to specific classes.

2.3.7 Session Service

The Session Service is responsible for scheduling and managing class sessions, including online lectures and meetings. It provides students with access to session details and links.

3 Component Interactions

3.1 Service Communication

The microservices in the LMS application communicate with each other using RESTful APIs. Each service exposes a set of endpoints that other services can call to perform specific actions. For example, the Teacher Service can call the User Service to authenticate a teacher before allowing them to create a class.

3.2 Data Flow

Data flows between the various microservices in the LMS application are designed to ensure smooth interactions and consistency across the platform. When a user, such as a teacher or student, takes an action, the relevant service handles the request and communicates with other services as needed to complete the process.

For instance, when a teacher creates a new class, the Teacher Service processes the creation request and sends the class details to the Class Service, where the class information is stored. The Class Service then interacts with the Student Service to update students' dashboards with information about the new class. If the teacher adds assignments to the class, the Teacher Service again coordinates with the Assignment Service to store the assignment details, ensuring that the assignment is linked to the correct class and is accessible to the enrolled students.

Similarly, when a student submits an assignment, the Student Service collects the submission data and sends it to the Assignment Service, where the submission is recorded. The Assignment Service then communicates with the Teacher Service to notify the teacher of the new submission, updating their dashboard with the necessary details for review.

In managing educational content, such as materials and videos, the Teacher Service handles the upload process and forwards the content to the Material Service and Video Service, respectively. These services store the content and ensure it is linked to the appropriate class. Students can then access this content through the Student Service, which retrieves the relevant materials and videos based on their enrolled classes.

Additionally, the Session Service plays a critical role in scheduling and managing class sessions. When a teacher schedules a session, the Teacher Service sends the session details to the Session Service, which stores the information and updates the students' dashboards to reflect the upcoming session.

Throughout the application, data flows are designed to maintain consistency and synchronization across services. This is achieved through RESTful API communications, where each service provides specific endpoints that other services can call to perform actions or retrieve data. The architecture ensures that each service performs its role independently while remaining integrated with the overall system, providing a seamless experience for both teachers and students.

4 Design Considerations

4.1 Scalability

The LMS application is designed to scale horizontally. Each microservice can be deployed independently, allowing the system to handle increased load by adding more instances of a specific service.

4.2 Security

Security is a key consideration in the LMS architecture. All communications between services are secured using SSL/TLS. User data is protected through encryption and role-based access control.

4.3 Reliability

The system is designed for high availability and reliability. Services are deployed in a clustered environment, ensuring that the system remains operational even if some instances fail.

5 Conclusion

This document provides a high-level overview of the LMS application architecture. The microservices architecture ensures that the system is scalable, reliable, and secure, meeting the needs of a modern educational platform.