

Software Design Specification

<Learning Management System>

Internal Advisor:

- Project Supervisor / Faculty Member
- Head of Computer Science / IT Department
- Academic Mentor
- Lab Instructor (if practical sessions involved)

External Advisor:

- Industry Professional in E-Learning / LMS Development
- Software Engineer / Developer with experience in web applications
- IT Consultant specializing in educational systems
- Educational Technology Expert

Project Team:

- Tuba
- Emaan Fatima
- Ayesha Hassan

Project Manager's Signature

Document Information

Category	Information
Customer	UOS – Department of Computer Science
Project	Learning management system
Document	Software Design Specification
Document Version	1.0
Identifier	SDS_Project_Group_08_LMS
Status	Draft
Author(s)	Tuba,Emaan Fatima,Ayesha Hassan
Approver(s)	Tuba,Emaan Fatima,Ayesha Hassan
Issue Date	November 19, 2025
Document Location	Stored in project GitHub Repository
Distribution	<ol style="list-style-type: none">1. Dr.Illyas2. Tuba,Emaan Fatima,Ayesha Hassan3. UOS – Department of Computer Science

Definition of Terms, Acronyms and Abbreviations

Term	Description
SDS	Software Design Specification
UOS	University of Sargodha
LMS	Learning management system

Table of Contents

1. Introduction.....	5
1.1 Purpose of Document	5
1.2 Project Overview	5
1.3 Scope	5
2. Design Considerations	5
2.1 Assumptions and Dependencies	5
2.2 Risks and Volatile Areas	6
3. System Architecture.....	6
3.1 System Level Architecture	6
3.2 Sub-System / Component / Module Level Architecture.....	6
3.3 Sub-Component / Sub-Module Level Architecture (1...n).....	6
4. Design Strategies	7
4.1 Strategy 1...n.....	7
5. Detailed System Design.....	7
6. References	7
7. Appendices.....	8

1. Introduction

1.1 Purpose of Document

This document describes the design of a Learning Management System (LMS). It is intended for internal and external advisors and the development team. It guides the implementation, architecture, and system requirements. The project uses **Object-Oriented Design (OOD) methodology** for modular and reusable design.

1.2 Project Overview

The Learning Management System (LMS) is a software application that manages courses, students, and instructors, allowing online learning, assignments, and progress tracking. It enables efficient course management and performance evaluation. The system will be developed using **Object-Oriented Design (OOD)** for modularity, scalability, and easy maintenance.

1.3 Scope

The LMS system will:

- ❖ Allow instructors to create, update, and manage courses and learning materials.
- ❖ Enable students to enroll in courses, access materials, submit assignments, and track progress.
- ❖ Support user management, including roles for students, instructors, and administrators.
- ❖ Provide assessments, quizzes, and performance reports.

The LMS system will NOT:

- ❖ Handle offline classroom management or physical attendance.
- ❖ Provide advanced AI-based tutoring or personalized learning recommendations.
- ❖ Manage financial transactions or payment processing.

2. Design Considerations

The Learning Management System (LMS) is divided into three main layers:

User Interface, Business Logic, and Database. It includes modules for User Management, Course Management, Assignments & Assessments, Notifications, and Reporting & Analytic. Each module has sub-components, such as registration, course creation, submission, grading, and notifications. Modules interact to provide seamless functionality while ensuring security, scalability, and performance. The system is web-based, supports multiple devices, and includes error handling and backup mechanisms.

2.1 Assumptions and Dependencies

- ❖ Users have basic computer and internet knowledge to access and use the LMS effectively.
- ❖ The system depends on stable web servers, databases, and internet connectivity for proper operation.
- ❖ Any third-party tools or libraries used (for quizzes, notifications, or analytic) must be compatible with the LMS framework.
- ❖ Design decisions depend on current technology standards, such as web-based platforms and responsive design for multiple devices.

- ❖ Changes in technology or infrastructure may require adjustments in the system design.

2.2 Risks and Volatile Areas

- ❖ New features or updates requested by stakeholders may impact design.
- ❖ Increasing number of users or courses may require architectural adjustments.
- ❖ New regulations or security threats may require design modifications.
- ❖ The system will be designed modularity to allow easy updates, maintenance, and integration of new features without major redesigns
- ❖ Updates in web technologies, database systems, or third-party libraries could affect compatibility.

3. System Architecture

The Learning Management System is divided into logical components that work together to support user roles, course activities, assessments, and reporting. The architecture ensures separation of concerns so that each part handles a specific responsibility while interacting with others efficiently.

3.1 System Level Architecture

- ❖ The LMS is decomposed into three main elements: User Interface Layer, Application/Business Logic Layer, and Database Layer.
- ❖ The UI Layer communicates with the Business Logic Layer, which processes requests and interacts with the Database Layer.
- ❖ External interfaces include authentication services and notification services (email/alerts).
- ❖ The system runs on a web-based platform accessible via multiple devices.
- ❖ Error handling is implemented globally to ensure stable performance and user experience.

3.2 Sub-System / Component / Module Level Architecture

The main modules of the LMS are:

- ❖ **User Management Module** (login, registration, role assignment)
- ❖ **Course Management Module** (course creation, content upload, enrollment)
- ❖ **Assignment & Assessment Module** (submissions, quizzes, grading)
- ❖ **Notification Module** (email & system alerts)
- ❖ **Reporting & Analytic Module** (progress reports, activity logs)

3.3 Sub-Component / Sub-Module Level Architecture (1...n)

Each module contains further sub-units:

- ❖ User Management → Login, Registration, Profile, Role Permission
- ❖ Course Management → Course Setup, Material Upload, Enrollment Tracking
- ❖ Assignments & Assessments → Submission Handler, Quiz Engine, Grade Calculator

- ❖ Notifications → Email Sender, System Alert Manager
- ❖ Reporting & Analytic → Student Progress Tracking, Course Performance Metrics

4. Design Strategies

The LMS uses a modular and object-oriented design to keep the system scalable, maintainable, and easy to extend. The system separates interface, logic, and database layers, ensuring secure data storage and supporting multiple users at the same time with stable performance.

4.1 Strategy 1...n

The design enables future upgrades, code reuse, simple user interaction, secure data handling, and smooth simultaneous access.

- ❖ **Future extension:** New LMS features can be added easily.
- ❖ **System reuse:** Common functions like login and notifications reused.
- ❖ **UI paradigm:** Simple and intuitive interface.
- ❖ **Data management:** Secure centralized database.
- ❖ **Concurrency:** Supports many active users simultaneously.

5. Detailed System Design

A detailed design should include the following:

- *Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.*
- *Detailed Sequence diagram with parameter list*
- *State Transition Diagram*
- *Logical data model (E/R model)*
- *Physical data models*
- *Detailed GUIs*

6. References

This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained (This section is like the bibliography in a published book).

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2025-Proposal	Project Proposal	Oct 20, 2025	<Give the path of your Project repository/Folder>

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2025-FS	Functional Specification	Oct 20, 2025	<Give the path of your Project repository/Folder>

7. Appendices
