**Software Requirement Specification (SRS)**

**Project: Learning Management and Assessment Scheduling System**

**Table of Contents**

1. Introduction  
   1.1 Purpose  
   1.2 Scope  
   1.3 Definitions, Acronyms, and Abbreviations  
   1.4 References  
   1.5 Overview
2. System Overview
3. Functional Requirements  
   3.1 User Roles and Authentication  
   3.2 Course Management  
   3.3 Assessment Scheduling  
   3.4 Student Enrolment and Attendance  
   3.5 Mark Management  
   3.6 Report Generation  
   3.7 Error Handling and Validation
4. Non-functional Requirements  
   4.1 Performance Requirements  
   4.2 Reliability and Availability  
   4.3 Security  
   4.4 Maintainability  
   4.5 Portability
5. System Features  
   5.1 Database Design  
   5.2 Interface Requirements
6. Use Case Diagram
7. Database Schema Diagram
8. Entity-Relationship (ER) Diagram
9. **Introduction**

**1.1 Purpose**

This document provides a comprehensive outline for the Learning Management and Assessment Scheduling System, a console-based application designed to manage courses, assessments, student enrolment, and reporting.

**1.2 Scope**

The system aims to support administrators, professors, and students with course and assessment management, grade recording, and report generation to improve academic record-keeping and assessment tracking.

**1.3 Definitions, Acronyms, and Abbreviations**

* **LMS**: Learning Management System
* **SRS**: Software Requirement Specification
* **ER Diagram**: Entity-Relationship Diagram

**2. System Overview**

The Learning Management and Assessment Scheduling System will manage course information, assessments, student enrolments, attendance, and performance data. It will generate reports based on student and course data to assist faculty and students in monitoring academic progress.

**3. Functional Requirements**

**3.1 User Roles and Authentication**

* **User Authentication**: Implement login functionality with role-based access (Admin, Professor, Student).
* **Roles and Permissions**: Admins can manage courses and users, professors can schedule assessments and manage grades, and students can view performance and enroll in courses.

**3.2 Course Management**

* **Admin Functions**: Add, update, delete courses.
* **Course Details**: Course code, name, syllabus, prerequisites.

**3.3 Assessment Scheduling**

* **Professor Functions**: Schedule exams, quizzes, and define types, dates, and time slots.
* **Assessment Types**: Exams and quizzes.

**3.4 Student Enrollment and Attendance**

* **Enrollment Management**: Students can enroll, and enrollment details are stored.
* **Attendance Tracking**: Track student attendance in courses.

**3.5 Mark Management**

* **Marks Storage**: Record student marks for assessments.
* **Performance Calculation**: Calculate and store overall student performance.

**3.6 Report Generation**

* **Report Types**: Student performance reports, course-wise statistics, attendance records.
* **Export Options**: Option to view and export reports in common formats.

**3.7 Error Handling and Validation**

* **Error Handling**: Catch exceptions like database errors, input validation errors.
* **Data Validation**: Ensure all inputs are in correct format and fields are complete.

**4. Non-functional Requirements**

**4.1 Performance Requirements**

The system must handle multiple concurrent users without performance degradation.

**4.2 Reliability and Availability**

The system should be available 99% of the time and ensure data persistence and reliability.

**4.3 Security**

Passwords should be stored using secure hashing, and sensitive data access must be restricted to authorized users only.

**4.4 Maintainability**

The codebase should be modular and well-documented to simplify future maintenance and updates.

**4.5 Portability**

The system should be compatible with common operating systems like Windows, macOS, and Linux.

**5. System Features**

**5.1 Database Design**

The database will store data across multiple tables: Users, Courses, Students, Enrollments, Assessments, and Marks. Relationships between tables will support foreign keys for referencing.

**5.2 Interface Requirements**

The system will feature a console-based interface that allows users to enter commands for each operation.

**6. Use Case Diagram**

A diagram of a learning management

Description automatically generated

* Depicts user interactions, including admin course management, professor assessment scheduling, student enrollment, and report generation.

**7. Database Schema Diagram**

**A computer screen shot of a diagram

Description automatically generated**

* Illustrates tables and relationships for courses, students, assessments, marks, enrollments, and user roles.

1. **Entity-Relationship (ER) Diagram**

**A screenshot of a computer

Description automatically generated**

* ER Diagram showing entities (e.g., Courses, Students, Enrollments) and relationships between them.