Learning Management System (LMS) Documentation

# Introduction

This LMS allows users (students, instructors, and administrators) to manage educational content, track progress, and evaluate learning outcomes. This document outlines key use cases, describing the interaction between users and the system.

## User Roles:

- Administrator: Manages the system, users, and permissions.  
- Instructor: Creates and manages courses, assignments, and grades.  
- Student: Enrolls in courses, submits assignments, and tracks progress.

**Project Architecture will microservices**-userManagementService  
-courseManagementService  
-enrollmentService  
-contentDeliveryService  
-quizAssignmentService  
-progressTrackingService  
-discussionForumService  
-notificationService  
-billingPaymentService  
-reportingAnalyticsService  
-adminManagementService  
-configService  
-gatewayService  
-discoveryService

Standards :::::

-same naming convention for all the project

-same branches naming

-take care of time complexity (prevent nested loops as much as can )

-prevent too much joins

-no redundant data

-clean coding (write solid code)

-no methods without unit testing

-coding style as per standards of spring

-to be implemented sonarqube

-to be implemented jenkins

# 1. Use Case: User Authentication

Actors:

- Student, Instructor, Administrator

Description:

Users must authenticate before accessing the LMS features.

Pre-conditions:

- User must have valid credentials.

Main Flow:

1. User navigates to the login page.  
2. User enters email and password.  
3. System verifies the credentials.  
4. If valid, the user is directed to their dashboard.

Post-conditions:

- The user is successfully logged in and has access to their role-specific features.

# 2 Use Case: User Registeration

Actors:

- Student, Instructor, Administrator

Description:

Users must signup before accessing the LMS features.

Pre-conditions:

- no pre conditions

Main Flow:

1. User navigates to the signup page.  
2. User enters mobile and password and name ….  
3. System send otp.  
4. If otp valid, the account created successfully.

Post-conditions:

- The user is successfully signed up.

# 3. Use Case: Course Management (Instructor)

Actors:

- Instructor

Description:

Instructors can **create**, **edit**, **view** and **delete** courses.

Pre-conditions:

- Instructor must be logged in.

Main Flow:

1. Instructor navigates to "Create Course" page.  
2. Fills in course details (title, description, schedule).  
3. Submits the form.  
4. System saves the course and makes it available for students.

Post-conditions:

- The course is available for student enrollment.

# 4. Use Case: Course Enrollment (Student)

Actors:

- Student

Description:

Students can browse and enroll in available courses.

Pre-conditions:

- Student must be logged in.  
- Courses must be available for enrollment.

Main Flow:

1. Student browses the course catalog.  
2. Selects a course to enroll in.  
3. Clicks "Enroll" button.  
4. System registers the student for the course.

Post-conditions:

- The student is enrolled in the course and can access its content.

# 5. Use Case: View Courses (Student)

Actors:

- Student

Description:

Students can browse available courses.

Pre-conditions:

- Student must be logged in.  
- Courses must be available for enrollment.

Main Flow:

1. Student browses the course catalog.

# 6. Use Case: Assignment Submission (Student)

Actors:

- Student

Description:

Students submit assignments for courses they are enrolled in.

Pre-conditions:

- Student must be enrolled in the course.  
- Assignment must be created and active.

Main Flow:

1. Student navigates to the course's assignments section.  
2. Selects the assignment to submit.  
3. Uploads the assignment file or fills in the required text.  
4. Clicks "Submit."  
5. System validates and confirms submission.

Post-conditions:

- Assignment is submitted and timestamped.

# 7. Use Case: Take Exam (Student)

Actors:

- Student

Description:

Students take exams for courses they are enrolled in.

Pre-conditions:

- Student must be enrolled in the course.  
- exam must be created and active.

Main Flow:

1. Student navigates to the course's exam section.  
2. Selects the exam to take.  
3. fills in the required text.  
4. Clicks "Submit."  
5. System validates and confirms submission.

Post-conditions:

- exam is done and timestamped.

# 8. Use Case: Grading Assignments (Instructor)

Actors:

- Instructor

Description:

Instructors can view and grade submitted assignments.

Pre-conditions:

- Assignment must have student submissions.

Main Flow:

1. Instructor navigates to the course's assignments section.  
2. Selects the assignment to grade.  
3. Reviews student submissions.  
4. Enters grades and feedback.  
5. Submits the grade.

Post-conditions:

- The student's grade is updated and visible to them.

# 9 Use Case: Grading exam (Instructor)

Actors:

- Instructor

Description:

Instructors can view and grade submitted exams.

Pre-conditions:

- exam must have student submissions.

Main Flow:

1. Instructor navigates to the course's exams section.  
2. Selects the exam to grade.  
3. Reviews student submissions.  
4. Enters grades and feedback.  
5. Submits the grade.

Post-conditions:

- The student's grade is updated and visible to them.

# 10. Use Case: Progress Tracking (Student)

Actors:

- Student

Description:

Students can view their progress in enrolled courses.

Pre-conditions:

- Student must be enrolled in courses with progress data (e.g., grades, completed assignments).

Main Flow:

1. Student navigates to "My Progress" section.  
2. System displays course-wise progress (completed assignments, grades).  
3. Student reviews their progress.

Post-conditions:

- Student is aware of their progress.

# 11. Use Case: System Management (Administrator)

Actors:

- Administrator

Description:

Administrators manage users, courses, and system settings.

Pre-conditions:

- Administrator must be logged in with elevated permissions.

Main Flow:

1. Administrator accesses the admin dashboard.  
2. Selects "Manage Users" or "Manage Courses."  
3. Creates, edits, or deletes users/courses as needed.  
4. Updates system configurations (e.g., session timeouts, enrollment policies).

Post-conditions:

- System settings or entities are updated.

# 12. Use Case: Notifications and Alerts

Actors:

- Student, Instructor, Administrator

Description:

System sends notifications to users about course updates, deadlines, or announcements.

Pre-conditions:

- User must be subscribed to notifications or have pending deadlines.

Main Flow:

1. System detects an event (new assignment, approaching deadline, etc.).  
2. System sends notification (email, SMS, or in-app).  
3. User receives the notification.

Post-conditions:

- User is informed of the event.

# 13. Use Case: Assessment Management (Instructor)

Actors:

- Instructor

Description:

Instructors can **create**, **edit**, **view** and **delete** exams or assignment.

Pre-conditions:

- Instructor must be logged in.

Main Flow:

1. Instructor navigates to "Create Exam" page.  
2. Fills in Exam details (choose Questions).  
3. Submits the form.  
4. System saves the exam and makes it available for students.

Post-conditions:

- The course is available for student enrollment.

# 14. Use Case: Choices Management (Instructor)

Actors:

- Instructor

Description:

Instructors can **create**, **edit**, **view** and **delete** choices.

Pre-conditions:

- Instructor must be logged in.

Main Flow:

1. Instructor navigates to "Create choices" page.  
2. Fills in choices details (Option text , is true ).  
3. Submits the form.  
4. System saves the exam and makes it available for students.

Post-conditions:

- The course is available for student enrollment.

# 15. Use Case: Questions Management (Instructor)

Actors:

- Instructor

Description:

Instructors can **create**, **edit**, **view** and **delete** questions.

Pre-conditions:

- Instructor must be logged in.

Main Flow:

1. Instructor navigates to "Create questions " page.  
2. Fills in choices details (question text , type).  
3. Submits the form.  
4. System saves the exam and makes it available for students.

Post-conditions:

- The course is available for student enrollment.

# Conclusion

These are common use cases for a Learning Management System. Each use case outlines a specific interaction between users and the system, detailing how the system responds to user actions. This documentation helps stakeholders understand system capabilities and design future enhancements.