Project: Java-Based Learning Management System (LMS)

Project Overview:

The LMS project is required to be a web-based application for managing and organizing online courses, and assessments from the perspective of students and instructors. It should support a range of user needs, each with specific features. The primary functionalities will include course creation and management, user management, assessments, and performance tracking.

Key Components:

1. User Management

- **User Types**: Admin, Instructor, Student.
- **Admin**: Manages overall system settings, creates users,and manages courses.
- **Instructor**: Creates courses, manages course content, adds assignments and quizzes, grades students, removes students from courses.
- **Student**: Enrolls in courses, accesses course materials, take quizzes, hand in assignments, view assignments and quiz grades.
- **Features**:
    - User Registration and Login (role-based access).
    - Profile Management (view/update profile information).

2. Course Management

- **Course Creation**:
    - Instructors can create courses with details like title, description, duration, ... etc.
    - Instructors can upload media files (videos, PDFs, audio, ... etc.).
    - Course consists of a number of lessons to be attended by students.
- **Enrollment Management**:
    - Students can view available courses and enroll in these courses.
    - Admins and Instructors can view the list of enrolled students per course.
- **Attendance Management**
    - Instructors can generate an OTP per lesson to maintain the students attendance.
    - Students can select the lesson to attend and enter the OTP received from the instructor.

3. Assessment & Grading

- **Assessment Types**: Quiz, Assignment
- **Quiz Creation**:
    - Instructors can create quizzes with different types of questions (MCQ, true/false, short answers).
    - Instructors can create a questions bank per course.
    - Randomized question selection for each quiz attempt.
- **Assignment Submission**:

    - Students can submit assignments by uploading files for review by Instructors.
- **Grading and Feedback**:
    - Instructors can grade assignments.
    - Students receive automated feedback after quizzes and manual feedback on assignments.

4. Performance Tracking

- **Student Progress Tracking**:
    - Instructors can track quiz scores, assignment submissions, and attendance.

5. Notifications

- **System Notifications**:
    - Students can check their notifications after they login to the system where they can find notifications for enrollment confirmation, graded assignments, and course-related updates.
    - Notifications should be handled such that the students can choose to view only the unread notifications or all notifications.
    - Instructors can check their notifications after they login to the system such as getting notifications for the students who enroll on their courses.

5. Bonus

- **Role-Based Access Control**:
    - Using Spring Security for authentication and authorization.
    - Restrict access permissions so that they are granted based on role type.
- **Performance Analytics:**
    - Admins and Instructors can generate excel reports on student performance (including grades and attendance).
    - Visual representations (charts) of progress, performance, and course completion.
- **Email Notifications:**
    - Similar to the system notifications, students can get email notifications for enrollment confirmation, graded assignments, and course-related updates.

Technical Requirements:

Backend

- **Java** with Spring Boot for RESTful API services.
- **MySQL** or **PostgreSQL** or **SQLite** or similar database management tool.

Integration, Testing & Deployment

- **JUnit** for unit testing.
- **Git** is mandatory for version control (each member in the team should contribute and commit to the GitHub repository).

Milestones:

Phase 1: (Due Date: Friday 29th of November, 2024 @ 11:59 PM)

Deliverables:

1. Software Architecture Design document(use the IEEE format provided) . Your design document should contain:

1) **A clear specification of the application's architecture, including all its interfaces, components, connectors, and constraints; Mention clearly the architecture style you used.**

2) **Use at least two architecture view to represent and document your design**

3) **A description of your design process and a thoughtful reflection on your design, including the rationale behind it (telling a story about your design). Evaluate your design using at least 6 of your defined scenarios.**

** **Explicitly mention any assumptions you have made.**

**** Your project customer (whom you can check requirements with) and coach is your TA.**

Phase 2: (Due Date: Sunday 24th of December, 2024 @ 11:59 PM)

The complete implementation (source code) should be delivered.

Submission Details:

1. You should work in teams from 5 to 6 students from the same lab or with the same TA.
2. NO LATE submissions and NO email submissions will be accepted.
3. Cheating is not tolerated and will be given negative grades.
4. For each project phase, you should submit ONE zip file including a text file having your names and IDs and the zip file should be with the below naming convention including one student ID: ASWE_PhaseNumber_GroupNumber_ID1
   (example: ASWE_P1_CS1-2_20200001)
5. You SHOULD NOT copy any code from the internet or from your colleagues. It will be detected and considered as a cheating case.