App name: School Management System

Functional requirements

1. User Registration and Authentication: Users (students, teachers, administrators) should be able to register and create an account. They should be able to log in to their account using their logins and passwords.
2. Role-Based Access Control: The system should authorize users based on their roles (student, teacher, administrator), and provide different levels of access.
3. Course Management: Administrators should be able to add, update, or delete courses, as well as create, update or delete learning sessions. Teachers should be able to assign themselves to a course and learning sessions. Students should be able to sign up for courses and learning sessions.
4. Schedule Viewing: Administrators, teachers and students should be able to view the respective schedules.

Non-functional requirements

1. Performance: The application should be able to support at least 100 users concurrently. The response time should be quick, even during peak usage times.

2. Reliability: The application should be available for use at all times, with minimal downtime.

3. Usability: The application should be user-friendly, with an intuitive interface that makes it easy for users to find and use features.

4. Maintainability: The application should be easy to maintain and update. The code should be well-structured.

Use Cases

1. User Registration

- Actor: Student, Teacher, Administrator
- Description: The actor visits the registration page, fills in the required information, and submits the form to create an account.

2. User Authentication

- Actor: Student, Teacher, Administrator
- Description: The actor visits the login page, enters their credentials, and submits the form to log in to their account.

3. Role-Based Access Control (Authorization)

- Actor: Application
- Description: The system checks the role of the logged-in user and provides access to features and functionalities based on the user's role.

4. Course Management

- Actor: Administrator, Teacher
- Description: The administrator adds, updates, or deletes courses and learning sessions. The teacher assigns themselves to a course and learning sessions.

5. Student Course Enrollment

- Actor: Student
- Description: The student views available courses and learning sessions and signs up for them.

6. Schedule Viewing

- Actor: Student, Teacher, Administrator
- Description: The actor views their respective schedules.

Classes

1. User.

- Abstract class representing any user of the system.
- Attributes: userId, firstName, lastName, username, password, role.
- Methods: login(), logout(), and register().

2. Student.

- Extends User.
- Additional attributes: enrolledCourses (a list of Course objects).
- Additional methods: enrollCourse(Course course), dropCourse(Course course), viewSchedule().

3. Teacher.

- Extends User.
- Additional attributes: assignedCourses (a list of Course objects).
- Additional methods: assignCourse(Course course), removeCourse(Course course), and createSchedule(Course course).

4. Administrator

- Extends User.
- Additional methods: addCourse(Course course), updateCourse(Course course), and deleteCourse(Course course).

5. Course:

- Attributes: courseName, courseId, Teacher, Schedule.

6. Schedule:

- Attributes: days of week, time, location.

Relationships

- Student can be enrolled in multiple Courses.
- Teacher can be assigned to multiple Courses.
- Course is taught by one Teacher.
- Course has one Schedule.