# Document

**Team Details:-**

Team Member-1: K.Meghana (284717)

Team Member-2: G.Kausthub Rao (284748)

1. Scope:

Functional Requirements

1. Task and Assignment Management:

- Allow students to create, organize, and manage tasks and assignments.

- Provide features to set deadlines and receive notifications/reminders.

2. Progress Tracking:

- Track and display the progress of individual assignments and overall coursework.

- Provide visual indicators (e.g., progress bars, completion percentages) to show students their progress.

- Allow students to mark tasks as completed or pending.

3. Resource Management:

- Providing the study materials

- Enable categorizing of resources for easy retrieval.

4. Communication and Collaboration:

- Include messaging features for communication with peers and teachers.

5. User Feedback and Support:

- Implement a feedback mechanism for students to share their learning experiences and issues.

Non-Functional Requirements

1. Usability:

- Ensure the application has an intuitive and user-friendly interface.

- Optimize the application for both desktop and mobile devices.

- Provide consistent and easy navigation throughout the platform.

2. Performance:

- Ensure quick loading times and responsiveness, even with a large number of users.

3. Reliability:

- Ensure high availability and uptime, especially during crucial periods like exams or assignment submissions.

4. Scalability:

- Design the system to accommodate increasing numbers of users, resources, and data.

- Support horizontal and vertical scaling to handle growing demands.

5. Interoperability:

- Ensure compatibility with different operating systems and browsers.

- Support multiple languages and regional settings for a diverse user base.

- Facilitate easy integration with other educational tools and platforms.

Tech Stack:

* Java
* Spring Boot
* Angular
* HTML
* CSS
* JavaScript/ TypeScript

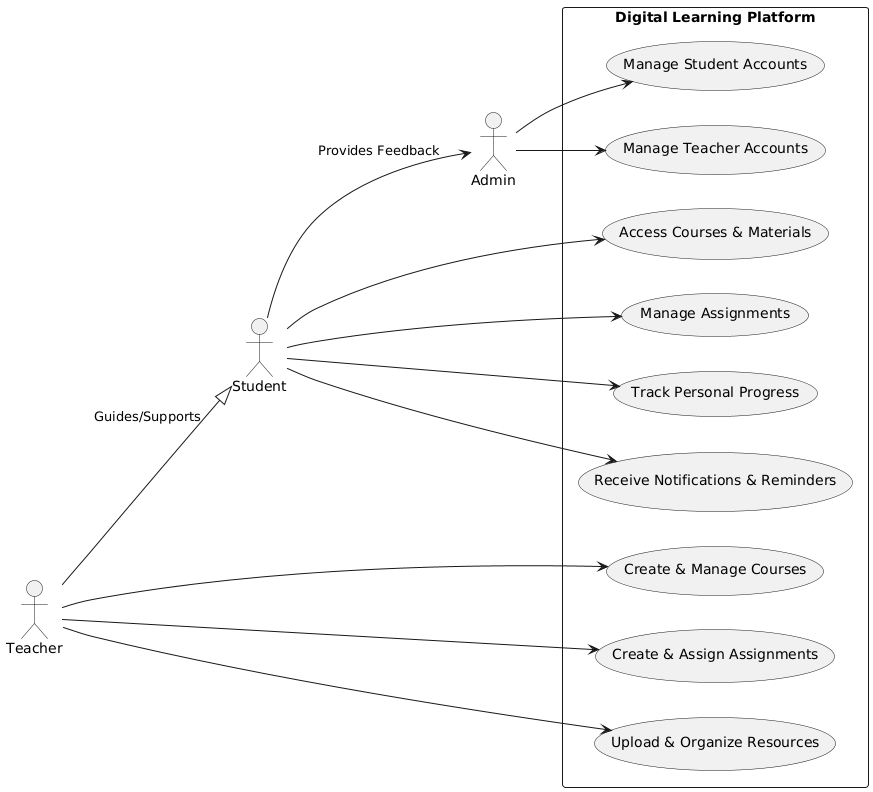
**Microservices Architecture:-**

**A diagram of a computer program

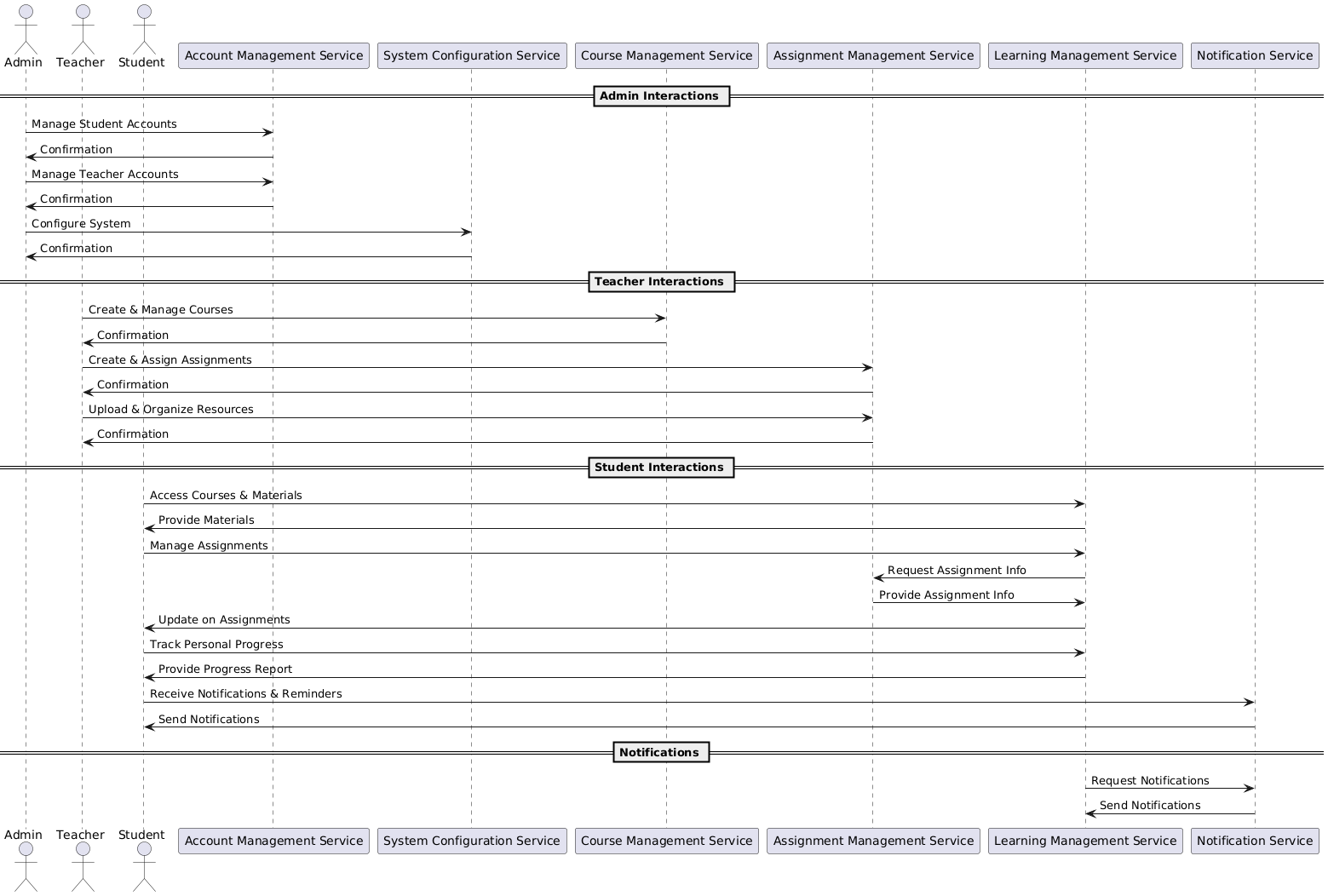
Description automatically generated**

**High Level Design: -**

**Use Case Diagram:-**

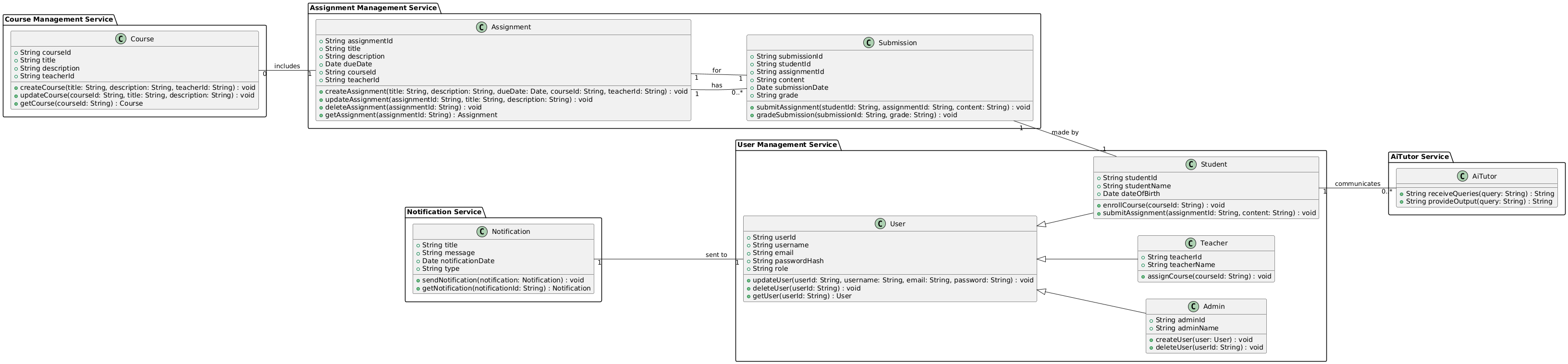


**Sequence Diagram: -**

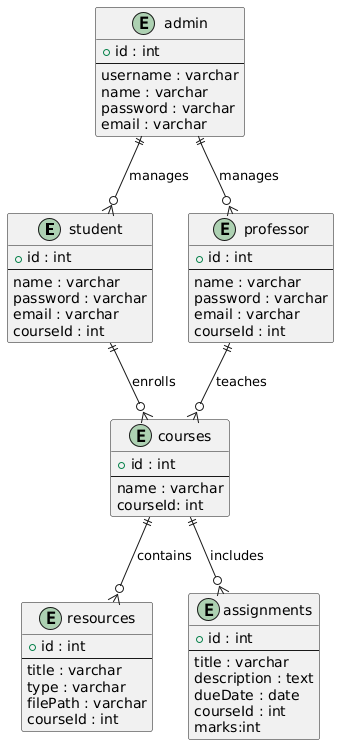


**Low Level Design: -**

**Class Diagram**



**ER/DataBase Diagram:-**



**Component Diagram: -**

