**Fee Management System**

The document provided outlines an assignment for implementing a microservice in a Spring Boot application for managing student fees.

Assignment Overview:

Objective: Develop a microservice `feesms` to manage student fees.

Functionality\*\*:

  1. An API to fetch all fees paid by a student.

  2. An API to pay fees for a student.

- \*\*Integration\*\*: Implement APIs in the existing students` service to interact with the new fees microservice.

**Steps to Implement:**

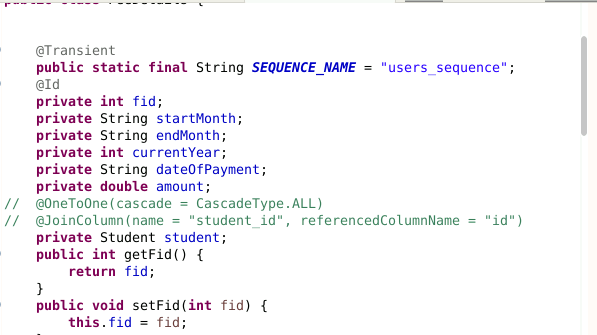
1. **Set Up the Microservice**:

   Create a new Spring Boot project for ‘feesms’.

   Define the necessary dependencies (e.g., Spring Web, Spring Data JPA).

2. **Define the Data Model:**

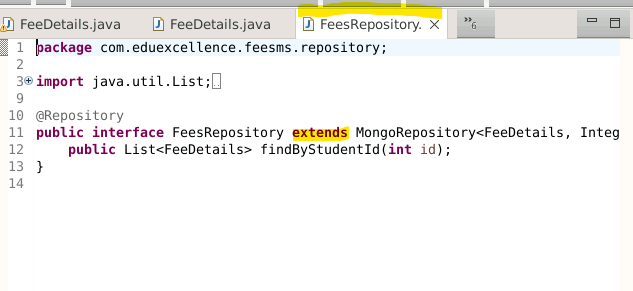
   Create an entity class for `Fee` with attributes like `id`, `studentId`, `amount`, `paymentDate`, etc.



    Use JPA annotations to map the entity to a database table.

3. **Create the Repository**:

   Define a `FeeRepository` interface extending `MongoRepository` to handle database operations.



4. **Implement the Service Layer**:

   - Create a `FeeService` class to handle business logic, such as fetching fees and processing payments.

A screenshot of a computer program

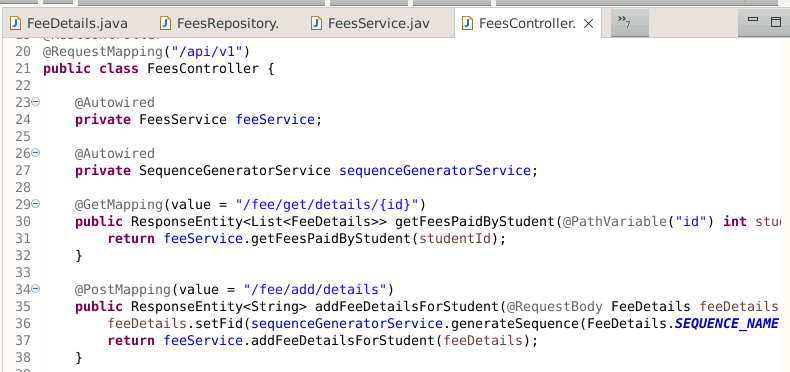
AI-generated content may be incorrect.

5. **Develop the Controller**:

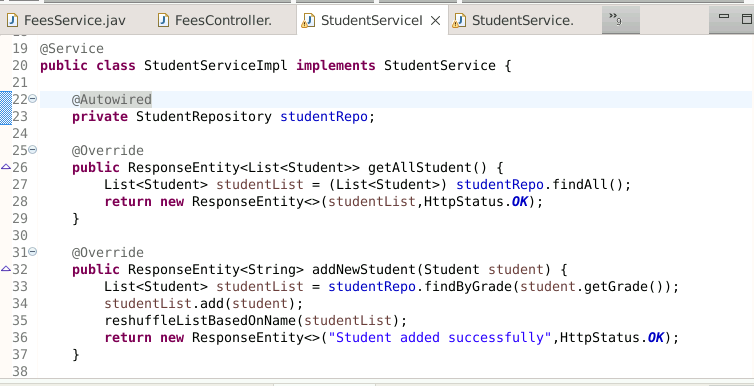
    Implement a `Fee Controller` with endpoints:

     GET :/fee/get/details/i{d}: Fetch all fees paid by a student.

     POST: /fees/add/details: Pay fees for a student.

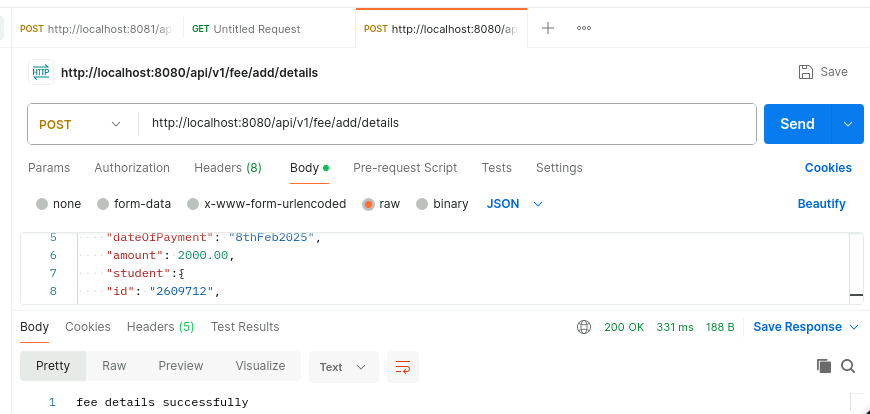


6. **Integrate with `students` Service:**



7. \*\*Testing\*\*:

   - Write unit and integration tests to ensure the functionality works as expected.

Post Operation  
  


A screenshot of a computer

AI-generated content may be incorrect.

An API to fetch a fee paid by a student

