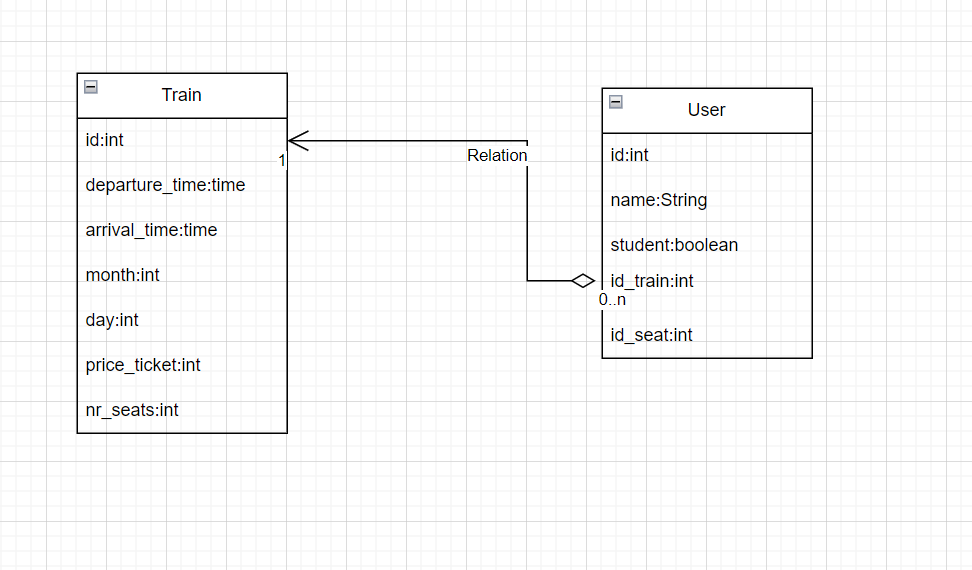
***Assignment 2 – software design 2024***

Vaida Diana-Laura

group 3

The UML diagram



My application is for managing a railway station online( like CFR). For the moment I have only two tables in a one to many relationship (the train and the user). A train can have more users(people going with that train).

The train has an id associated with it, the departure time, arrival time, month, day and the price ticket(standard).

The user has an id, name, the option of being a student or not, the id of the train and id of the seat. For the moment I have only 2 tables, later I will add more, tables for tickets, cities.

I have 4 main packets: Service, Repository, Entity and Controller, and the main class.

In the ‘application.properties’ from the ‘resources’ I connect to my database containing the tables. I checked if my CRUD operations work using PostMan. I tried to implement all the functionalities in the assignment.

In the future I want to add more functionalities – to calculate the specific price ticket (considering if a user is a student or not), checking if the train is already full (using the number of seats), adding a table for cities (what trains belong to what city).

Chapter 3 Division into packages

1. Utils: • It contains an utility class that provide common functionlity across the application. • Itcontains a static method getResponseEntity that generates a ResponseEntity with a custom response message and HTTP status. • The class is made final to prevent inheritance, and the constructor is private to prevent instantiation.

2. Entity: • This package contains entity classes that represent the structure of my database tables, such as User.

3. Service: • This package holds service interfaces that define the business logic operations for the following entities: UserService.

4. ServiceImpl: • Contains the implementations of the service interfaces defined in the service package, such as UserServiceImplementation.

5. Repository: • Holds repository interfaces responsible for database operations, such as UserRepos itory • The interface serve as bridges between the application’s business logic and the database. These interfaces extends the JpaRepository interface provided by Spring Data JPA, which offers powerful features for implementing database operations.

6. Controller: • It contains the following controller interfaces: UserController.

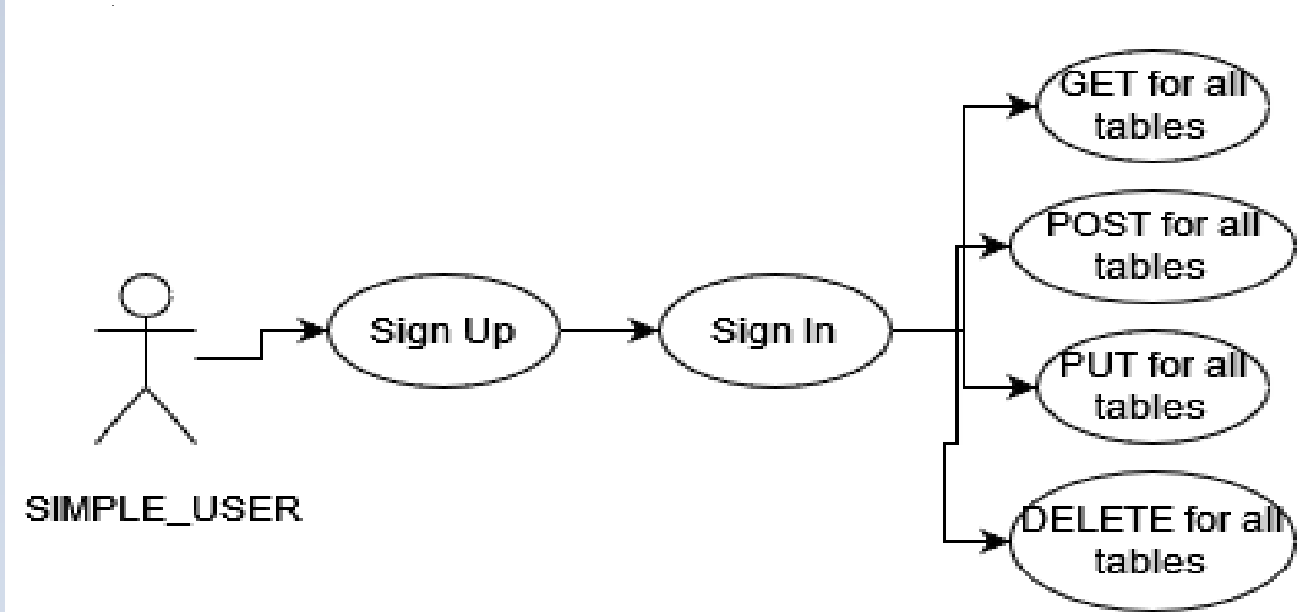
7. ControllerImpl: • Contains the implementations of controller interfaces defined in the controller pack age, such as UserControllerImplementation.

8. Constants: • Holds set of constants used throughout the application to maintain consistent mes saging and error handling.

9. JWT: • This package contain classes related to security configurations, such as JWT authentication and authorization, as seen in SecurityConfiguration, JWTFilter, and JWTUtil.

10. Wrapper: • It contains wrapper classes that encapsulate entities or DTOs (Data Transfer Objects), such as UserWrapper.

**Use case diagram**

****