Tracking laboratory activity

Analysis and Design Document

Student: Marin Andreea

**Group: 30431**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

# The objective of this assignment was to allow us to become familiar with MVC architectural pattern, services, repository and unit tests by designing and implementing an application for tracking the laboratory activity for the Software Design laboratory. The application should have two types of users (teacher and student) which must provide a username and a password to use the application.

# Functional Requirements

The application presents the following functional requirements:

* Teacher users can perform CRUD operations on students’ information. A token is generated when a new student is created, that will be used by the student to register
* Teacher users can perform add/delete/edit operations on laboratory classes
* Teacher users can perform CRUD operations on attendance for each lab
* Teacher users can perform CRUD operations on assignments
* Teacher users can grade the submitted assignments individually
* Student users can register using the token generated by the teacher
* Student users can view a list of all laboratory classes
* Student users can view the assignments for a laboratory class
* Student users can create/update/delete an assignment submission

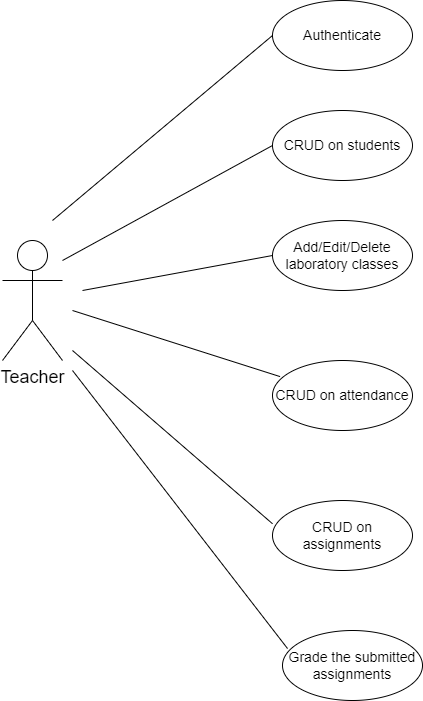
# Non-functional Requirements

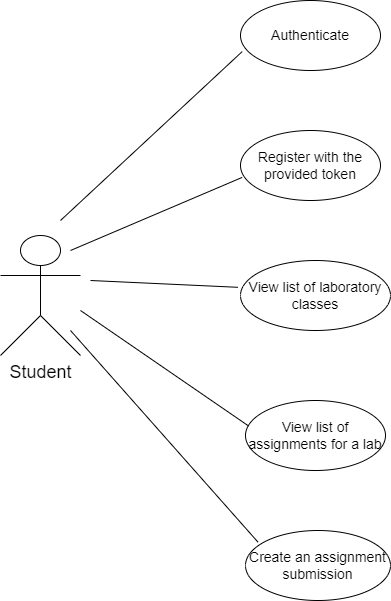
The application presents the following non-functional requirements:

- The system requires authentication

- The system is easy to work with

2. Use-Case Model





**Use-case:** Register using the generated token

**Level:**

**Primary actor:** Student

**Main success scenario:** The teacher introduces his credentials. If the login is successful, the teacher is able to create a new student by introducing the information of the new student except from password and token. After creating the student, a token is generated and the token is saved for the created student. Then, the student registers by introducing the username, the provided token and the password that will be associated to this account.

**Extensions:** The system will notify that the credentials or information about the new student introduced by the teacher are wrong and the login or creation of a new student were unsuccessful.

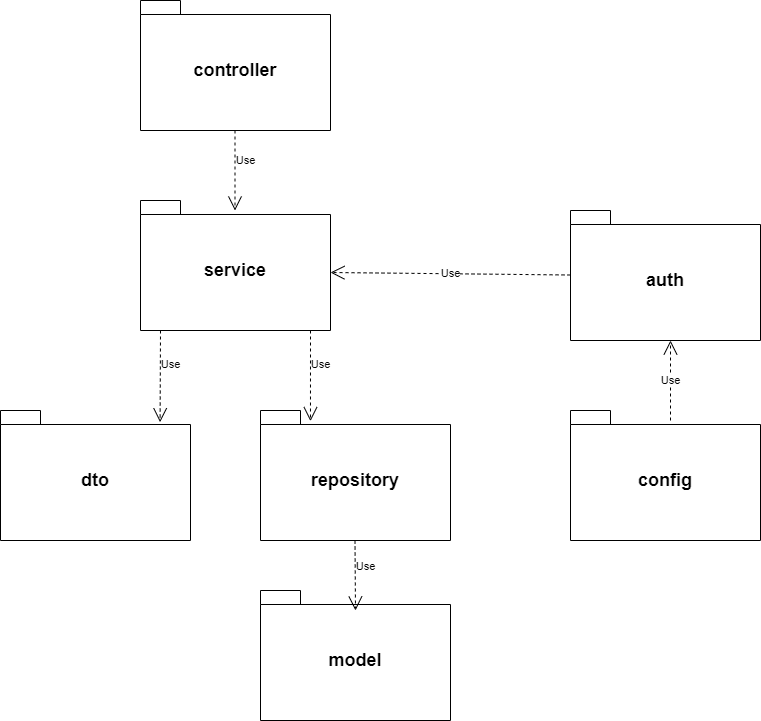
3. System Architectural Design

**3.1 Architectural Pattern Description**

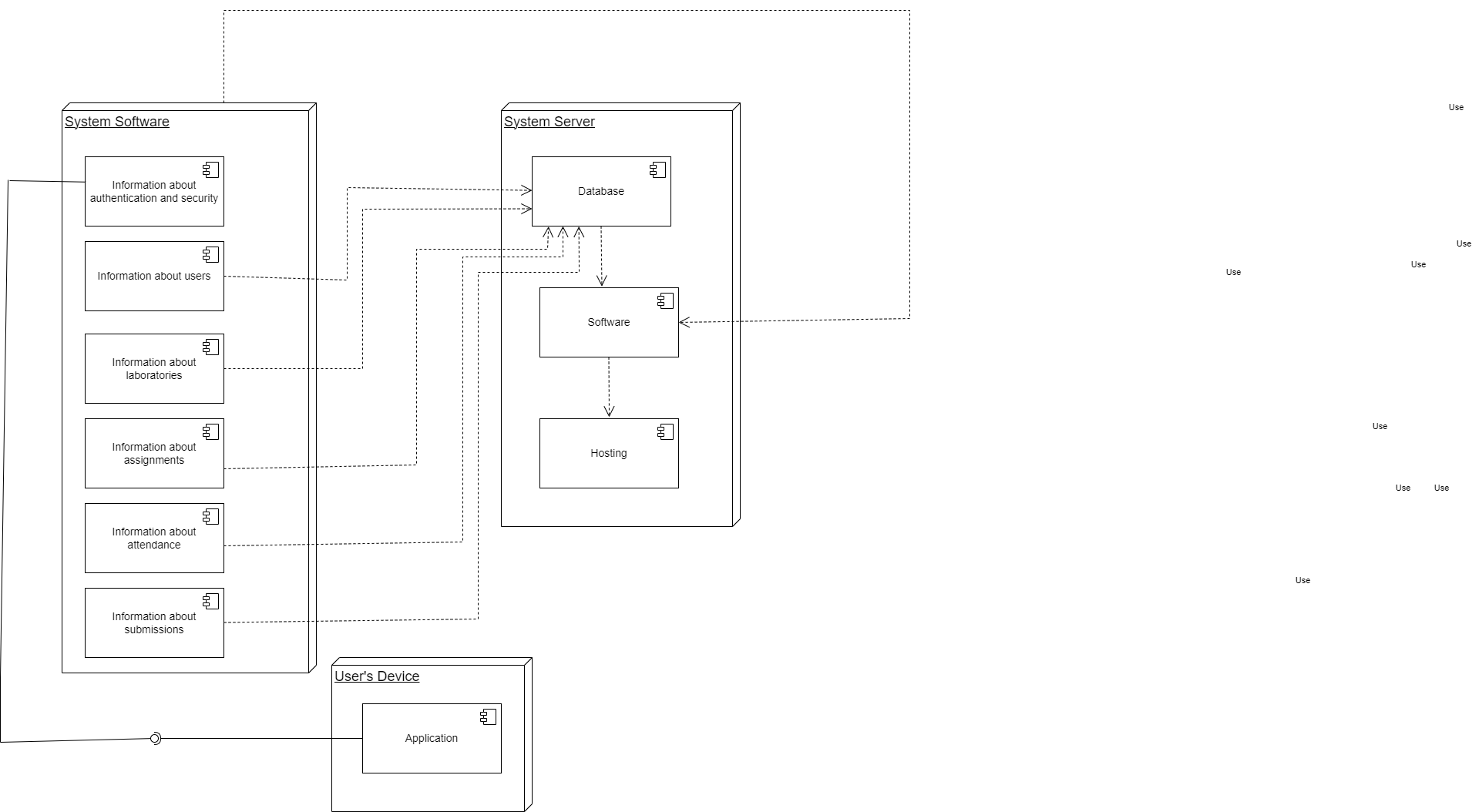
The architectural pattern used is MVC, but without the view part, which means the architectural pattern used is layered architecture. It is also known as an n-tier architecture and describes an architectural pattern composed of several separate horizontal layers that function together as a single unit of software. A major characteristic of this framework is that layers are only connected to the layers directly below them. It comes with the advantage that we can make changes in one layer without affecting the other layers.

**3.2 Diagrams**

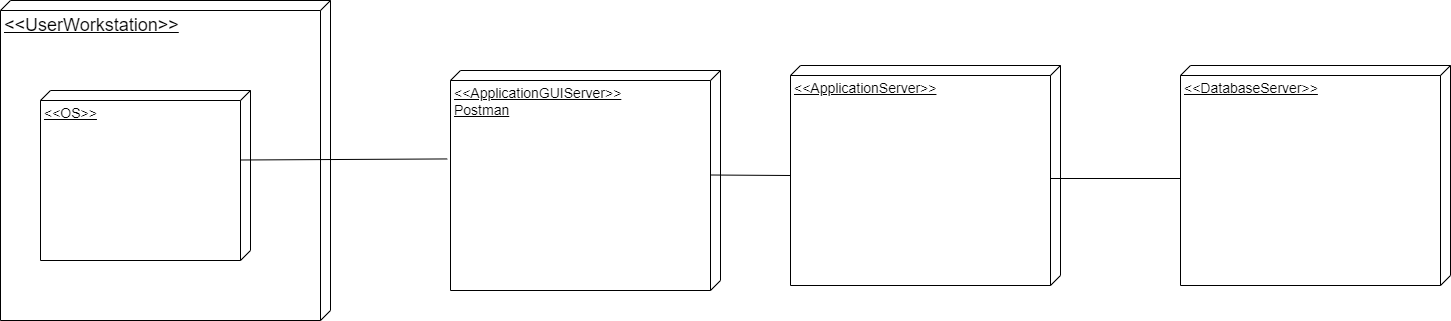
* Package Diagram



* Component Diagram

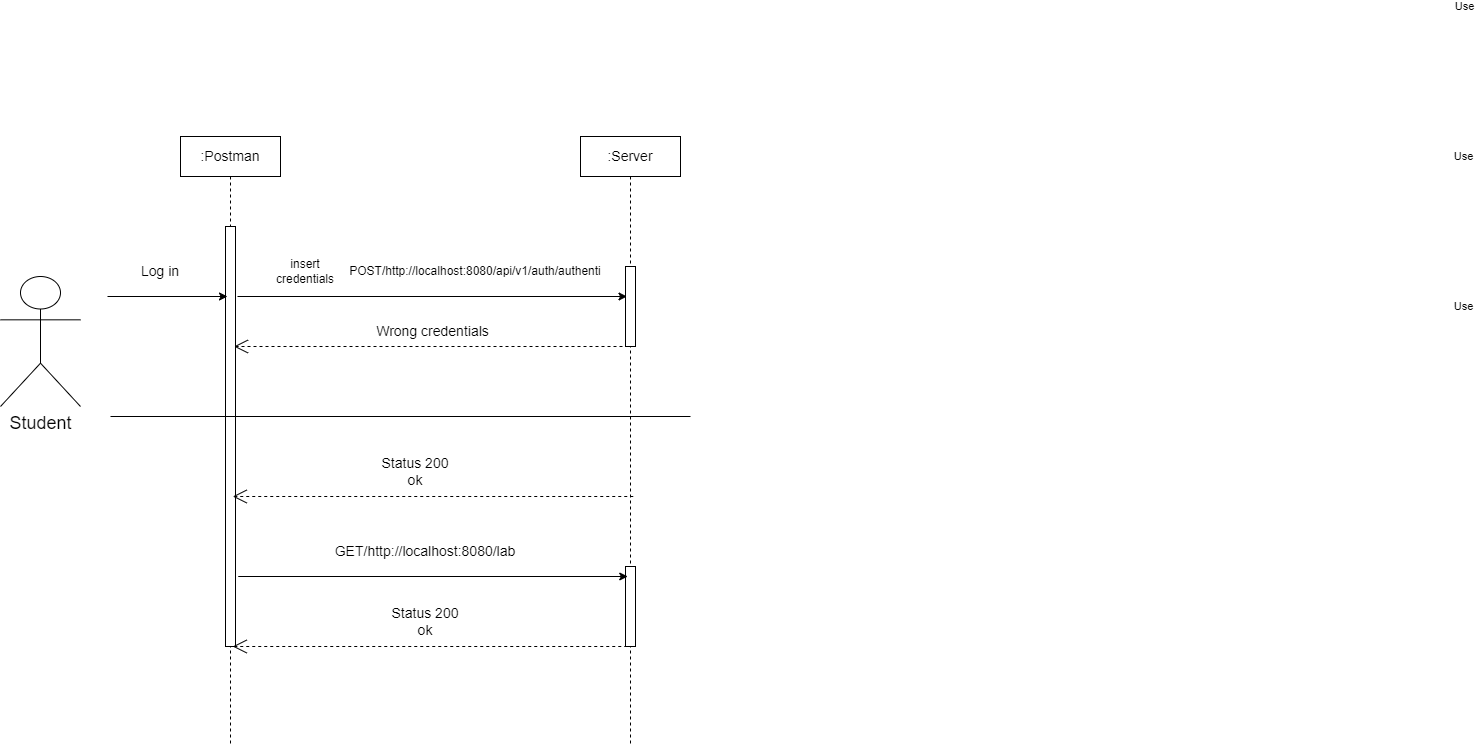


* Deployment Diagram



4. UML Sequence Diagrams

UML Sequence Diagram for viewing all the laboratory classes



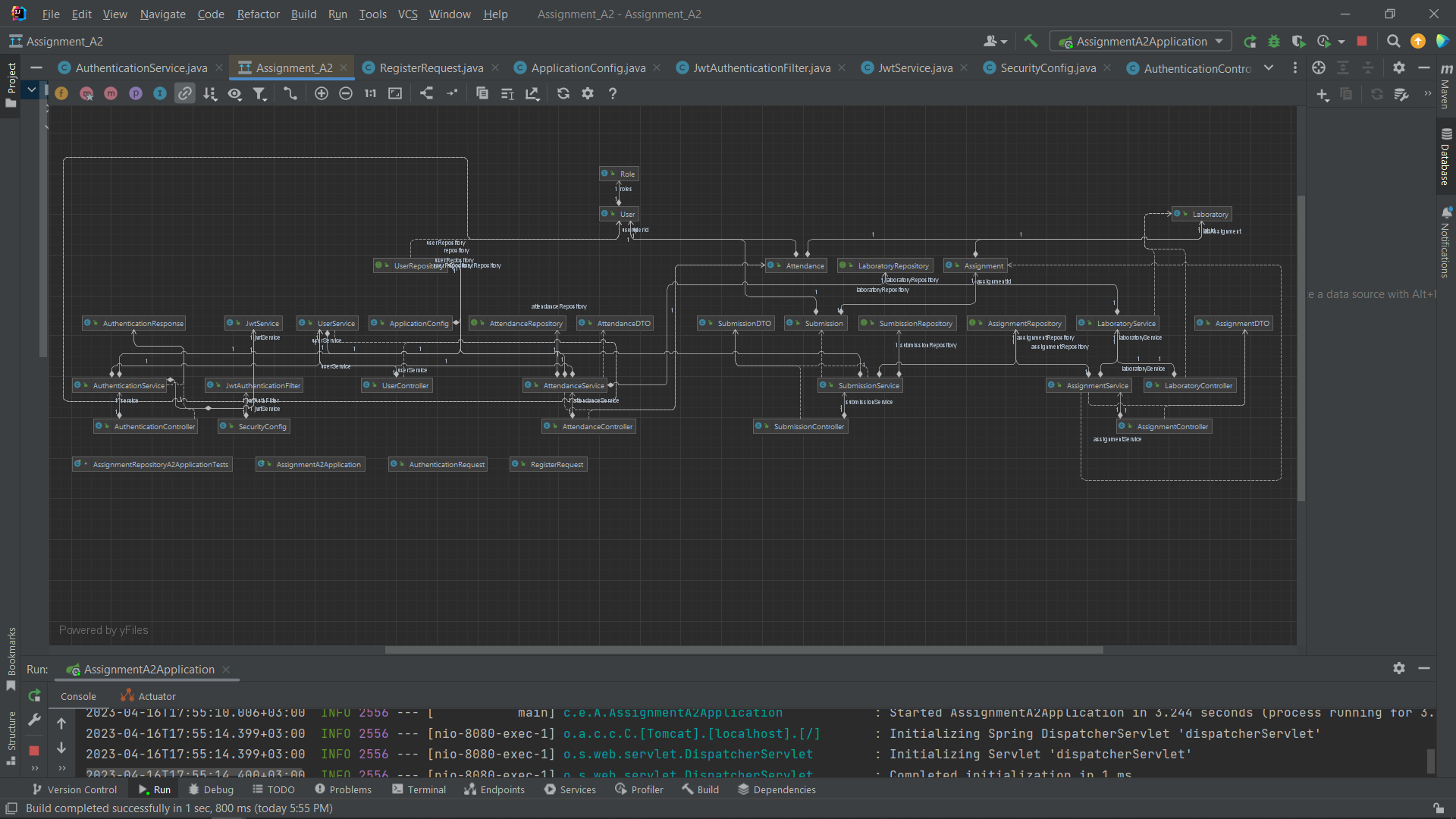
5. Class Design

**5.1 Design Patterns Description**

In software engineering, dependency injection is a design pattern in which an object or function receives other objects or functions that it depends on. A form of inversion of control, dependency injection aims to separate the concerns of constructing objects and using them, leading to loosely coupled programs.

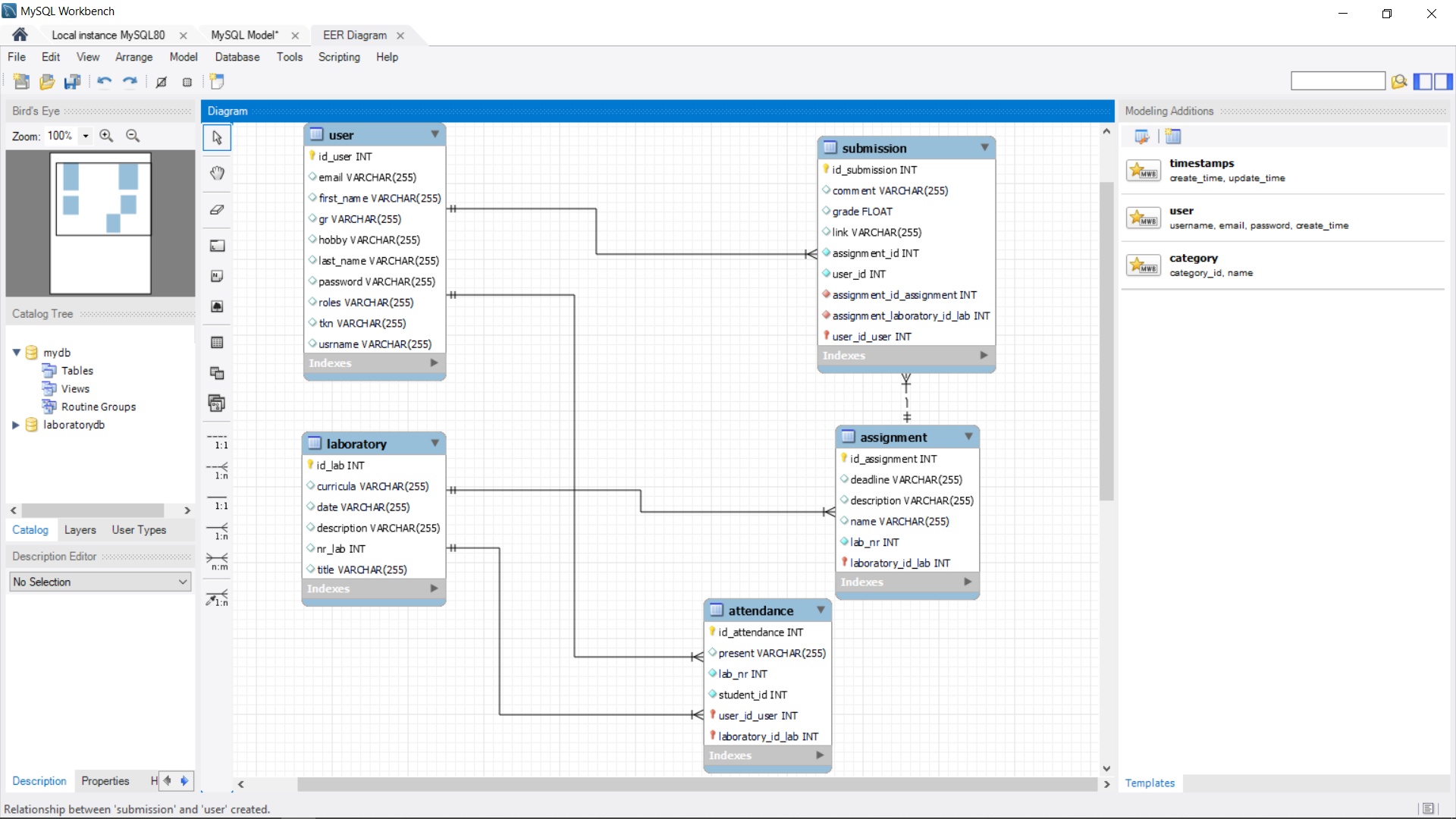
**5.2 UML Class Diagram**

The dependency injection pattern ensures that an object or function which wants to use a given [service](https://en.wikipedia.org/wiki/Service_(systems_architecture)) should not have to know how to construct those services.



6. Data Model

The system implements five data models: User, Laboratory, Assignment, Attendance and Submission. The User data model contains an id, information about the user (firstName, lastName, email, gr, hobby), token (used by the student to register), username, password and a role (TEACHER or STUDENT). The username and password will be used for authentication. The laboratory data model contains information such as laboratory number, curricula, description, date and title. Assignments can be created for each laboratory only by the Teacher. The attendance and assignment data models have references to the laboratory data model. When it comes to Submission data model, it has a reference to the Assignment data model and only students can create/update/delete an assignment submission. The teacher can only grade each submission individually.



7. System Testing

In order to ensure that the application functions correctly I have a flow of operations that were tested:

1. Login as Teacher
2. Create Student, save token,
3. Create Laboratory
4. Create Assignment for lab created at 3
5. Register as student with username, password and token
6. View list of laboratory classes
7. View assignments for a lab
8. Submit assignment for lab
9. Login as Teacher
10. Grade submitted assignment

8. Bibliography

* <https://www.youtube.com/playlist?list=PLqq-6Pq4lTTbx8p2oCgcAQGQyqN8XeA1x>
* <https://www.youtube.com/watch?v=KxqlJblhzfI>
* <https://www.youtube.com/watch?v=9SGDpanrc8U>
* https://www.baeldung.com/spring-boot-data-sql-and-schema-sql