Assignment 2

Student: Voicu Sara-Ioana

**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 main purpose of this project is to design and implement an application that tracks the activity of the Software Design laboratory. Relational databases should be used to store assignments, students, teachers, attendances, laboratories, and submission. For implementing the web application, I chose the Java programming language and the Spring Framework.

Other sub-objectives are:

* Analyzing the problem and requirements
* Implementing each task for the users of the application
* Connecting the application to a database and performing different operations on the data
* Sending requests and receiving responses by using the Postman API platform

# Functional Requirements

- Each user has a username, password, and a role

- Any task must be done by a user that logged in

- Allowing only some type of user to have the authority to execute a specific task

- The data from the tables contains assignments, students, teachers, attendances, laboratories, and submission

- Operations like create, update, delete and view the content should be performed

- Data is stored in a relational database

- MVC architectural pattern is used to organize the application

# Non-functional Requirements

- Using the JSON Web Token that represents a compact and self-contained way for securely transmitting information: useful for authorization, information exchange

- Using encryption algorithms for the passwords of the users

- Helping the users to fill in the mandatory fields, in case of invalid input

2. Use-Case Model

Use case: Performing operations on students, laboratories, assignments and attendances

Primary actor: Teacher

Main success scenario: The teacher logs into the application with a valid username and password. Then he creates a student that receives a specific token. He can perform CRUD operations on the data of the students which means that he can create a new student, update an already existing one, delete a student or see a list of all the students. The admin can do the same operations with laboratories, assignments and attendances. He can also grade the submissions of students.

Extensions: If the admin doesn’t introduce the right username and password, he cannot log in.

Diagram

Description automatically generated

Use case: View laboratories, assignments and create submissions

Primary actor: Student

Main success scenario: The student logs into the application with a valid username and password and by using the token that the teacher created for him/her. Then he can view a list of laboratories, a list of the assignments for a specific laboratory and can create a submission.

Extensions: If the admin doesn’t introduce the right username and password, he cannot log in.

Diagram

Description automatically generated

3. System Architectural Design

* 1. **Architectural Pattern Description**

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

* The **Model** encapsulates the application data and in general they will consist of POJO.
* The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
* The **Controller** is responsible for processing user requests and building an appropriate model and passing it to the view for rendering.

Diagram

Description automatically generated

* 1. **Diagrams**

**Component diagram:**

Diagram

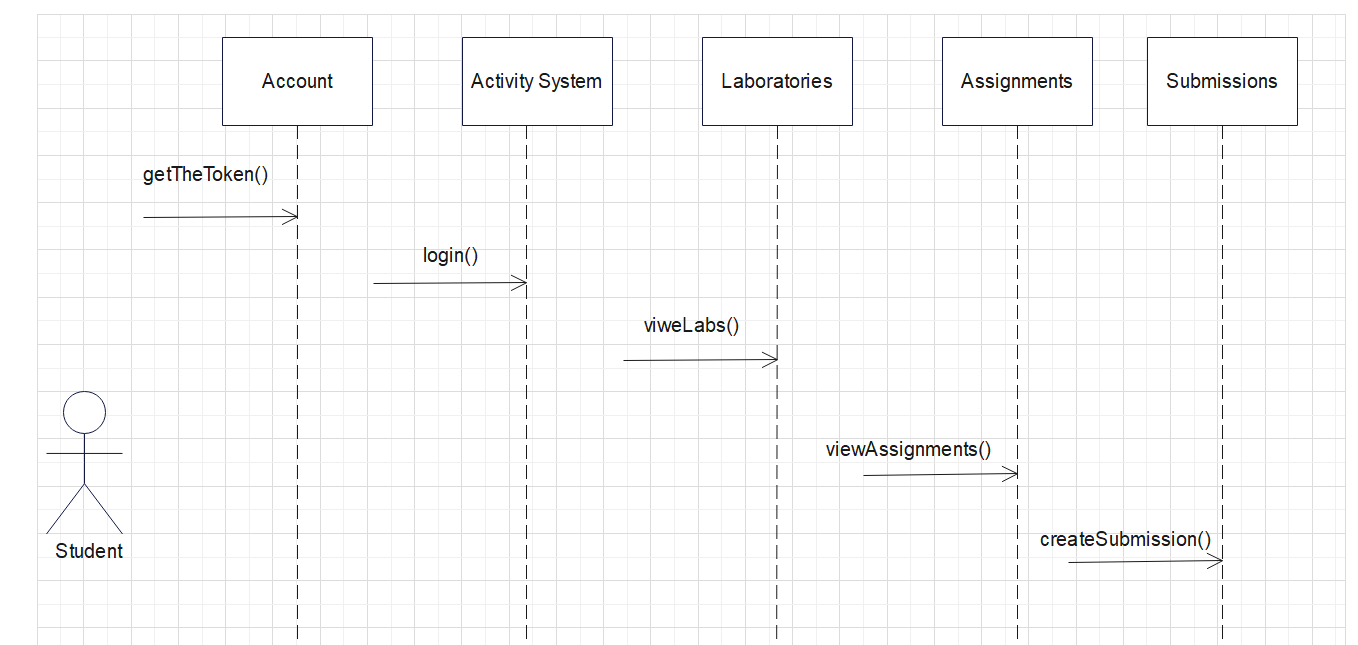
Description automatically generated

**Deployment diagram:**

Diagram

Description automatically generated

4. UML Sequence Diagrams

**

5. Class Design

**5.1 Design Patterns Description**

I have explained in chapter 3 the MVC architectural pattern that I used.

**5.2 UML Class Diagram**

*Graphical user interface

Description automatically generated*

6. Data Model

**Teacher:** contains an ID of type Integer, which is a primary key, a username and password of type String, an email address, a full name, a list of laboratories and a role, which can be only TEACHER.

**Student:** contains an ID of type Integer, which is a primary key, a username and a password of type String, an email address, a full name, a hobby, a group number, an attendance and a role, which can only be STUDENT.

**Submission:** contains an ID of type Integer, which is a primary key, a student name, an assignment name and a grade for the student.

**Laboratory:** contains an ID of type Integer, which is a primary key, a number, a date, a title, a description, a teacher, and a list of attendances, one for each student.

**Attendance:** contains and ID of type Integer, which is a primary key, a student and a list of laboratories, representing the laboratories that the student attends.

**Assignment:** contains ID of type Integer, which is a primary key, a deadline, a description, a name and a laboratory.

There are many one-to-many, many-to-many, one-to-one relationships between these models.

7. System Testing

There are some methods of validation that I implemented for the project. For example,

6+.the requests can be done only based on the token that is generated. If the token is for a student, he can perform only some actions. The same goes for the teacher.

The teacher cannot grade the assignment with a number less than one. In order to not have any magic string, the number is read from a file, as a minimum grade.

When performing the CRUD operations, we cannot delete a laboratory that is present in another table. A message like this will appear: “Laboratory couldn’t be deleted.”.

8. Bibliography

<https://www.youtube.com/watch?v=KxqlJblhzfI>

<https://www.youtube.com/watch?v=9SGDpanrc8U&t=3144s>

<https://www.youtube.com/watch?v=bxy2JgqqKDU&t=5364s>

<https://stackoverflow.com/>

<https://www.baeldung.com/spring-boot-data-sql-and-schema-sql>

<https://www.geeksforgeeks.org/spring-boot-crud-operations/>

<https://www.baeldung.com/jpa-many-to-many>

<https://www.baeldung.com/hibernate-one-to-many>

<https://www.baeldung.com/jpa-one-to-one>