

**Stack Overflow Assignment**

Software Design

Tudor-Mihai Iakkel

Group 30432

Table of Contents

1. **Introduction**

Stack Overflow is a very well-known platform where hardware and software engineers come together to solve different issues related to coding, logic, weird bugs and errors and much more. This assignment is trying to mimic the basic functionalities of Stack Overflow, while also adding some new features of its own.

1. **Tech stack**

Throughout the assignment, we will use several technologies which include:

* Java Programming Language (JDK 21)
* Spring Boot
* Angular
* MySQL Database
* Postman (for testing purposes)

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". The following dependencies of Spring Boot will be used in the assignment:

* Spring Web – imperative for building web applications
* Spring Data JPA – simplifies data extractions in SQL
* JDBC API – connects a client to database queries
* MySQL Driver – connects the application to a MySQL Database

Angular is a TypeScript-based web application framework which will be used to define and implement the front-end part of the assignment.

Postman is an application that allows the testing of web APIs. This software will come in very handy for testing the CRUD operations of the database with the GET, POST, PUT and DELETE requests.

1. **Design**
   1. **Architecture**

The assignment will hold a layered architecture resembling the MVC (Model-View-Controller) architecture. The application is split as follows:

* Sub-server
  + Responsible for the authentication of a user
  + Uses JWT tokens for secure authentication
  + Can create a new user via registering
  + Can login a user if it already exists in the database
  + Can update a user’s details/attributes
  + The POST and PUT requests of a user are tested within this sub-server
* Java Application
  + Entity Package – holds the model classes of the application: User, Question, Answer
  + Repository Package – holds an interface for each previously-defined model; they extend the CrudRepository interface, useful for SQL queries
  + Service Package – each entity has a service class; establishes the connection between the application and the database through the necessary queries; the Controller package is highly dependent on this package
  + Controller Package – responsible for connecting the web endpoint to the application; a path is defined in each class with the help of the @RequestMapping annotation, as well as all 4 requests annotations (@GetMapping, @PostMapping, @PutMapping, @DeleteMapping).
  1. **Diagrams**

Class Diagram

A screenshot of a computer program

Description automatically generated

Database Diagram

A computer screen shot of a computer flow chart

Description automatically generated

Package Diagram

A diagram of a software application

Description automatically generated