**Documentation Assignment**

Lacatus Arpad-Alex

Group 30433

**1.Introduction**

This assignment consists of a simpler version of the website StackOverflow. We will have 2 types of users (regular user and moderator). Some key feature include: users not being able to use the app without being logged, an user can post a question, answer a question, upvote/downvote and delete its own questions and answers. Questions have title, tags, descriptions, and pictures while the answers only descriptions and pictures. The passwords will be stored encrypted in the database.

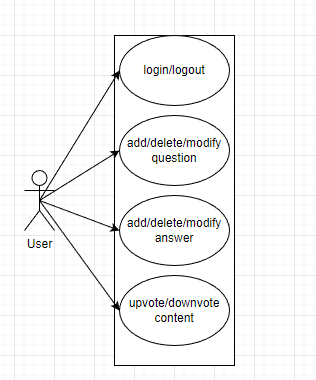
**2.Technology**

The main technologies used are Java Swing and SQL. Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs. Structured Query Language (SQL) is a standardized programming language that is used to manage relational databases and perform various operations on the data in them. The IDE we are using is IntelliJ IDEA, an Integrated Development Environment (IDE) for JVM languages designed to maximize developer productivity.

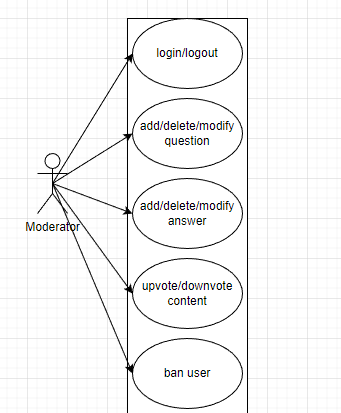
**3.Use case Diagram**

Use case diagram summarizes the details of our system’s users (or actors) and their interactions between them and with our system.

User use case:



Moderator use case:

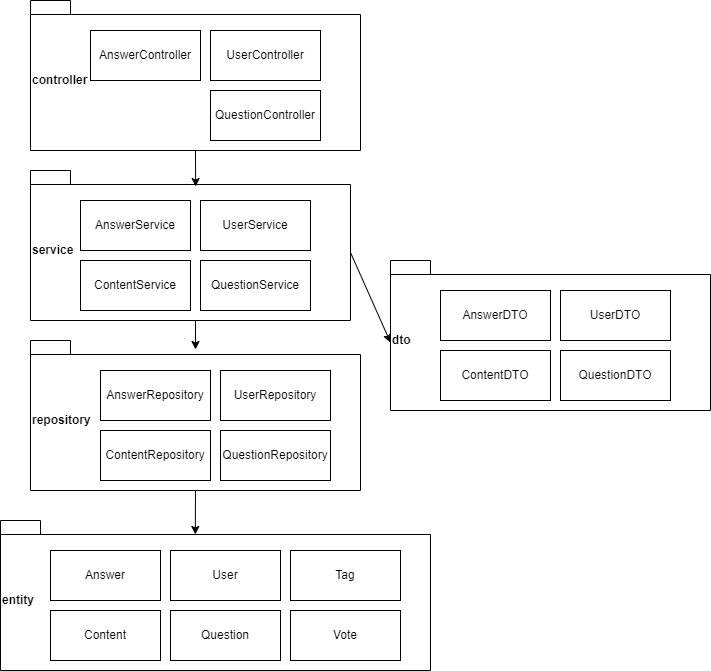


**4.Architecture**

For the architecture we will use a layered architecture which is comprised of controller, service, repository, model, DTO, GUI.

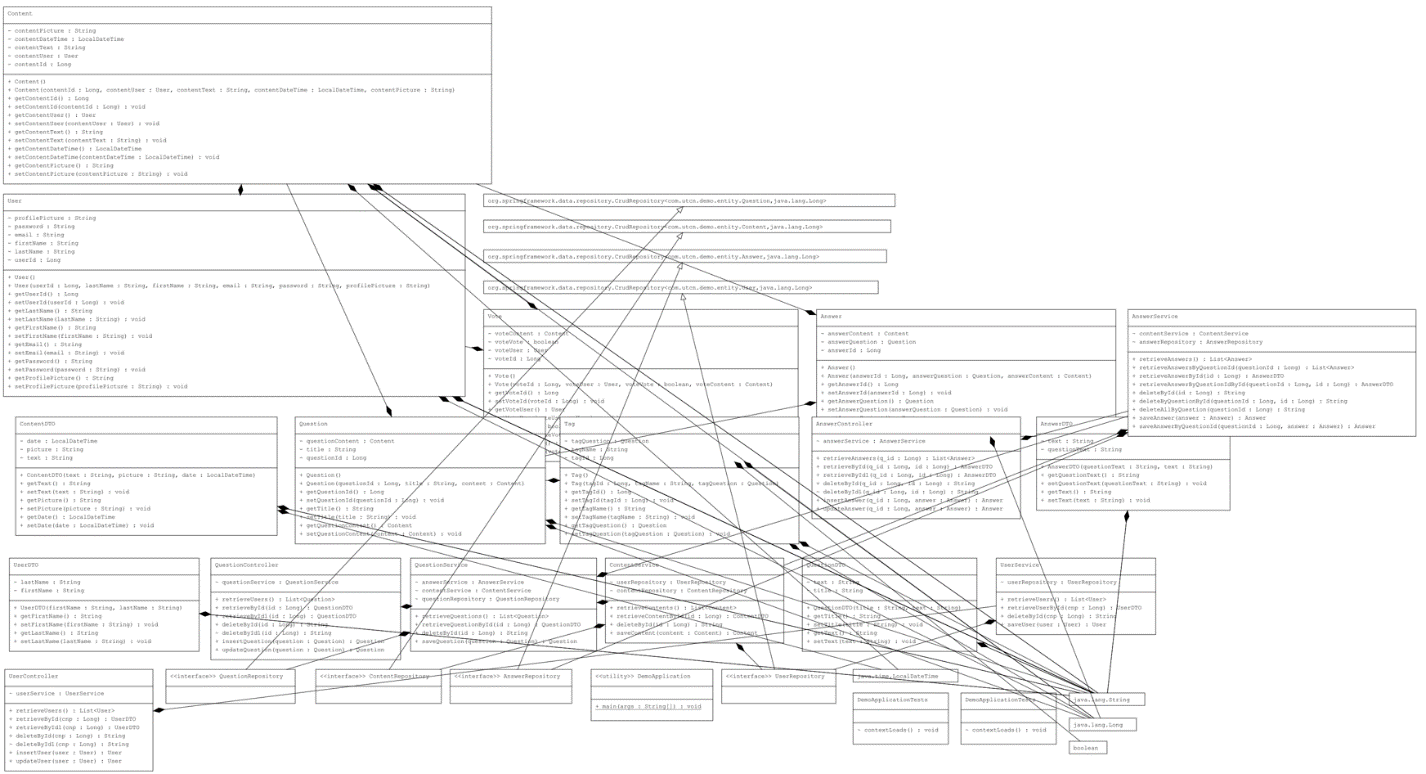
Controller is the one responsible for the request handling, it manages the requests and dispatches responses for each request. The service layer encapsulates the application's business logic, controlling transactions and coordinating responses in the implementation of its operations. The repository layer’s main goal is to isolate domain objects from details of the database access code and to minimize scattering and duplication of query code. Model layer is the one responsible for the implementation of the business logic. DTO has the responsibility of carrying data between processes. GUI layer constitutes the connection between the user and our application by providing tools to the user for interacting with the app.

**5.Package diagram**

The package diagram is a diagram that provides a way of visualizing the connections between the packages of our project.

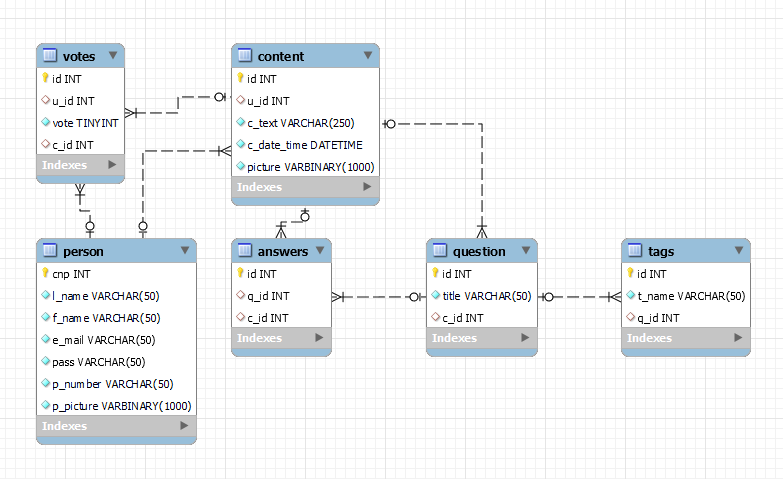
**6.Class diagram**

The class diagram is a diagram that is used to represent the connections between the classes in our application.

****

**7.Database diagram**

Here we have the diagram that describes visually the connections between the tables of our database.

****

**8.Endpoints requests**

Our application has endpoints for user, question or answer. These endpoints follow the CRUD principle so for each of them we have create, read, update, delete. The endpoints are as follows: for the create we have “insertQuestion” , for the read by id we have “getById/{id}” or just “getById”, for the delete we have “deleteById/{id}” and for the update we have “updateQuestion”. To note that for working with different services we have slight modifications question -> answer or user. For each controller these endpoints are preceded by a specification. For the questions we have “/questions” for the users we have “/users” and for the answers we have “/questions/getById/{cnp}/answers”. We used a different approach for the answers in order to bind them to a specific question. For the insert operation we have post mapping and we request a body that requires the object we work with(i.e. question/user/answer), for the read we get our id from a path variable, the delete works the same and as for update works similarly as the create and requires the same body.

**9.Architecture**

This part of the application focuses on the frontend. As such I used as framework Angular. Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications. Some key features of Angular are data binding, routing, and efficient use of reusable components.

In order to make the skeleton and face of our application we use html and scss. These two used together give us the main part of our frontend. With the use of html we were able to create buttons , forms , divisions, lists that with the help of scss were styled according to our needs. Some html elements were also attributed dynamically through typescript. For example the questions and answers page were filled dynamically with dummy data provided in a typescript file (to note that the dummy data is saved in an array , and this data will be replaced eventually with the data from the server.

This portion of the app is divided into 2 main parts heading and content. The head is always the same for our app and consists of the logo, which is a hyperlink to the home page, a search bar and a userprofile link. The second part is the content which through the help of routing can have many forms. These forms are split in mainly 3 user (login, signup, userprofile), question (questions, addquestion), answer (answers, addanswer).

**10.Routing**

The routing is handled with the help of router-outlet which through the function routelink or href routes our app to the desired page mentioned in app.module. With the help of Angular that handles the routing part by providing the class router the URL is change to the desired one and the content is mapped to the component that needs to be shown for the specific URL.

Here we can see the current routes in our application:

A screen shot of a computer code

Description automatically generated with medium confidence

In order to navigate the app we can not that we have the following routes linked to out content:

Login -> questions

-> signup signup -> questions

-> login

Questions -> answers

-> addquestion Answers -> addanswer

Using the heading stay always the same we can go to the homepage(login) and userprofile.