**Assignment 1 – documentation**

**Software design**

Pașca Eduard-Marian

**Introduction**

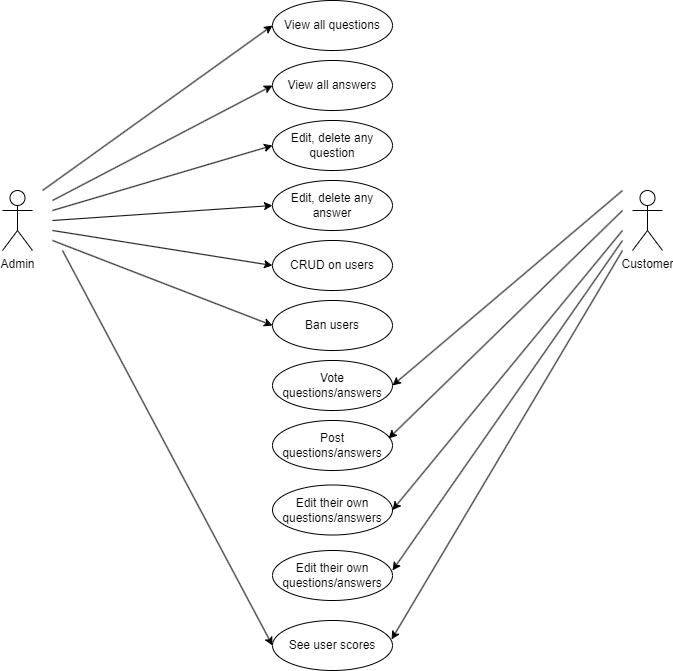
The project is meant to provide a platform where users can interact with each other via programming related answers & questions. The key features are represented by access to all the questions and answers posted by the users, the ability to post new questions or answer questions posted by other users.

**Technology**

The project is based on Java + Springboot for the backend part, VueJS for the frontend part, and MySQL for the database part.

**Use case diagram**

The use case diagram presents the main features which can be accessed by each type of user, as shown below:

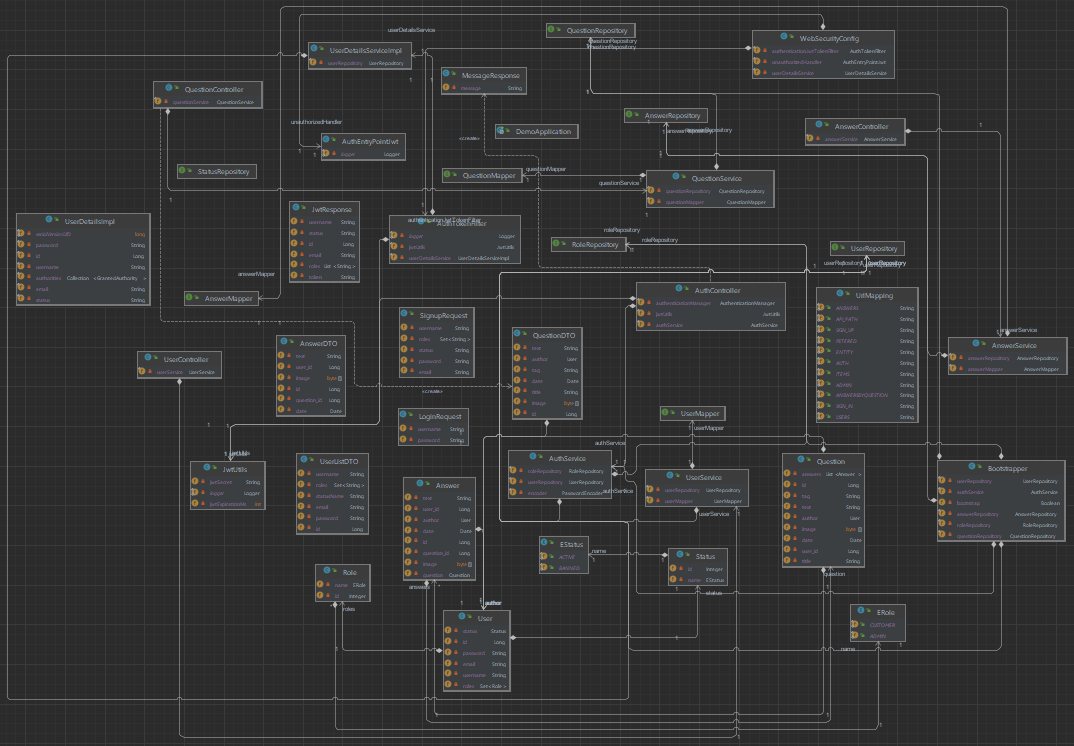


**Architecture**

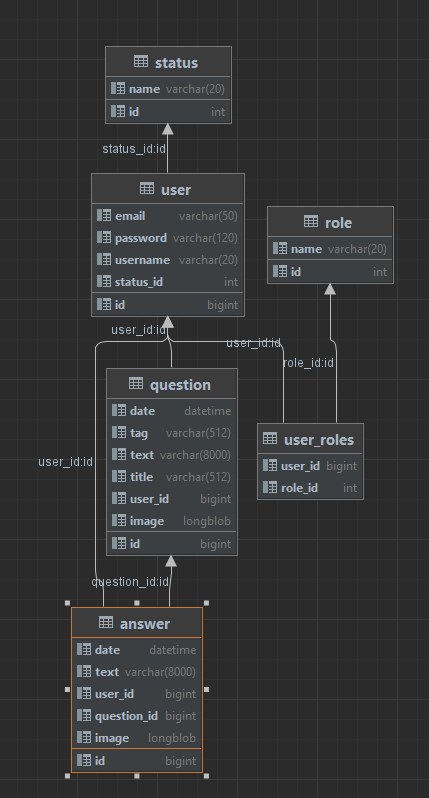
Layered architecture will be used, consisting of Data Logic, Business Logic, Presentation Logic:

* The Data Layer contains the models for the basic classes in the database, such as User, Question, Answer
* The Presentation Layer will contain all the views used in the platform, such as endpoints used by the users to see the questions posted by everyone, the answers to each question; the admins will be able to see details about all the users, all the questions, and all the answers
* The Business Logic Layer will contain the logic needed in order to process the CRUD performed by each user on their own answers and questions, CRUD performed by the admins on questions/answers/user accounts, addition of new answers/questions by each user

**Class diagram**

This diagram presents all the classes and the relationships between them:

**Database diagram**

****

**Endpoints**

There are endpoints for each entity, created via the controller; they process requests sent by the frontend, and execute the server-side computations needed to send the proper data back when rendering the information on the frontend.

* http://localhost:8080/api/answers/answersByQuestion/{{id}}

-input: id of the question for which to retrieve the answers

-output: a list of AnswerDTOs which is used by the frontend to render the table of answers for a particular question seen by a basic user

* http://localhost:8080/api/answers/{{id}}

-input: id of the answer to be retrieved

-output: AnswerDTO which is used to render a page for a single answer

* http://localhost:8080/api/items/admin

-input: none

-output: list of QuestionDTOs in order to render the table with all the questions seen by the admin

* http://localhost:8080/api/users

-input: none

-output: list of UserDTOs in order to render the table with all the users seen by the admin

* http://localhost:8080/api/users/{{id}}

-input: id of a specific user

-output: UserDTO containing the details of that specific user, allowing the admin to edit or delete, based on the type of request that is sent