**Stack Overflow**

Student: Presecan Alexandru

Group: 30434

Teaching assistant: Bindea Bogdan

1. **Introduction**

Stack Overflow is a well known website among programmers, used mainly to ask questions and receive answers for whatever difficulty the programmer might encounter. This project is a simplified version of Stack Overflow, preserving just the main functionalities such as posting questions and answers, tagging the questions, and an up/down vote system that affects the users’ score.

1. **Tech Stack**

Backend: ASP.NET

Frontend: Angular

Database: SQL Server

1. **Endpoints**

**ApplicationUsers**

/api/ApplicationUsers - GET

/api/ApplicationUsers/{id} - GET

/api/ApplicationUsers – POST

{

    "email": "testuser@yahoo.com",

    "userName": "TestUser",

    "password": "Password123!"

}

/api/ApplicationUsers/{id} – PUT

{

    "email": "testusernew@yahoo.com",

    "userName": "TestUserNew",

    "password": "Password123!",

"newPassword": "Password123!New"

}

/api/ApplicationUsers/{id} - DELETE

**Questions**

/api/Questions - GET

/api/Questions/{id} - GET

/api/Questions – POST

{

    "authorId": "authorId",

    "title": "test",

    "text": "test",

    "tags": ["tag1", "tag2", “tag3”]

}

/api/Questions/{id} – PUT

{

    "title": "testNew",

    "text": "testNew"

}

/api/Questions/{id} - DELETE

**Answers**

/api/Answers - GET

/api/Answers/{id} - GET

/api/Answers – POST

{

    "authorId": "authorId",

"questionId": "questionId",

    "text": "test"

}

/api/Answers/{id} – PUT

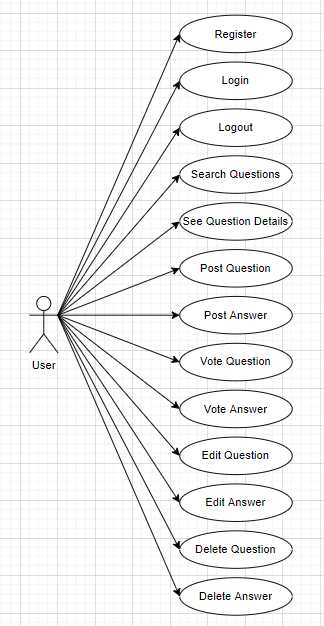
{

    "text": "testNew"

}

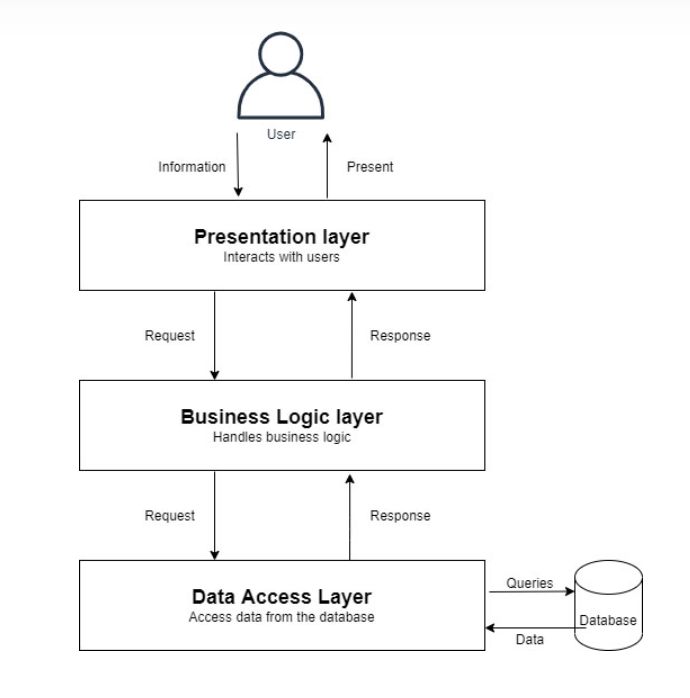
/api/Answers/{id} - DELETE

1. **Use cases diagrams**



User posts answer Case:

The user registers if not already registered, logs in, searches through questions and selects one that he/she is able to answer. He/She completes the answer and posts it. The user will gain or lose score points if other users vote his/her answer.

1. **Architecture**

Presentation Layer:

* Represents the client application (Angular SPA Application)
* Acts as an user interface
* Allows user interactions

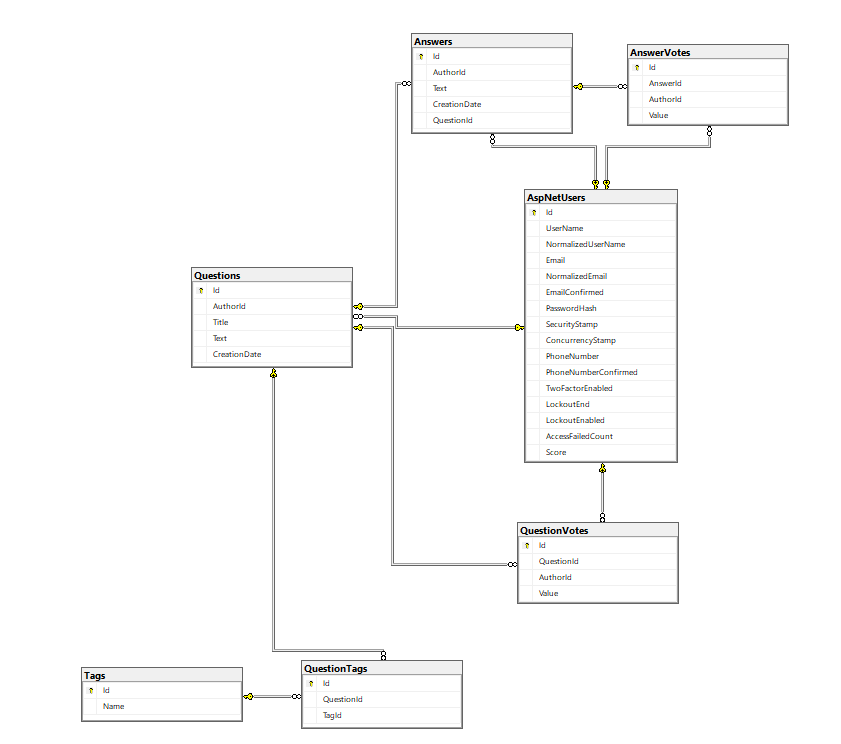
Business Logic Layer:

* Contains all the business logic of the application
* Composed out of API controllers and services

Data Access Layer:

* Used to access the database and perform transactions
* Most data access functionality is provided by the Entity Framework

1. **Database Diagram**

****

1. **Bibliography**

<https://docs.microsoft.com/en-us/aspnet>

<https://single-spa.js.org/docs/ecosystem-angular>