**StackOverflow**

**Backend**

1. **Introduction**

The Question and Answer Platform is an innovative web-based application designed to foster knowledge exchange and collaboration among users. Its primary objective is to provide a seamless experience for users to ask questions and receive answers from a community of experts. The platform incorporates various features to ensure efficient question management and response handling.

One of the core functionalities of the platform is the ability for users to ask questions. Each question includes essential information such as the author, title, text, creation date and time, picture, and one or more tags. In case a suitable tag does not exist, users have the flexibility to create a new one. The platform showcases a list of questions sorted by their creation date, with the most recent questions displayed at the top. Users have the autonomy to modify or delete their own questions as needed. Furthermore, the platform empowers users to filter questions based on tags, perform text searches, view questions from specific users, or conveniently access their own questions.

To facilitate comprehensive knowledge sharing, the platform allows users, including the original question author, to provide answers to questions. Each answer includes details such as the author, text, picture, and creation date and time. Users possess the ability to edit or remove their answers if necessary. Additionally, when viewing individual questions, users can easily access a list of answers associated with that question. The answers are intelligently sorted based on their vote count, with the most highly-rated answers prominently displayed first. This ensures that users can quickly access the most relevant and helpful responses.

1. **Technology**

The backend of the Question and Answer Platform is developed using the Spring Boot framework and utilizes a MySQL database for data storage and retrieval. Spring Boot offers a robust and efficient development environment for creating RESTful APIs and managing the application's business logic.

Spring Boot is a popular Java framework that provides a streamlined way to build and deploy web applications. It offers several key advantages, including automatic configuration, dependency management, and a wide range of built-in features. With Spring Boot, developers can focus on writing business logic rather than dealing with complex configurations. It also integrates seamlessly with other Spring projects, making it an ideal choice for building scalable and maintainable applications.

The backend is responsible for exposing a set of RESTful APIs that allow seamless communication between the frontend and backend components. These APIs enable the frontend to perform various operations, such as creating and retrieving questions and answers, filtering and searching for questions, and managing user interactions. The APIs are designed following the REST architectural principles, ensuring a stateless and scalable communication mechanism.

The backend of the Question and Answer Platform relies on a MySQL database for storing and retrieving data. MySQL is a widely used open-source relational database management system known for its performance, scalability, and ease of use. It provides robust data storage capabilities and supports efficient querying of data using SQL. With MySQL, developers can ensure the reliability and consistency of data in the application.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Use Case Diagram**

The Use Case Diagram section provides an overview of the functional requirements of the Question and Answer Platform by visually representing the interactions between actors and the system. Use case diagrams capture the different ways in which actors interact with the system to achieve specific goals or tasks. This section outlines the primary actors involved in the system and illustrates the various use cases that they can perform.

A diagram of a person

Description automatically generated with low confidence

1. **Software Architecture**

The Question and Answer Platform has been designed following a layered architecture pattern. This architecture provides a structured and modular approach to developing web applications, ensuring separation of concerns and promoting maintainability, scalability, and code reusability. The application is divided into distinct layers, each responsible for specific functionalities and interactions.

A picture containing text, screenshot, font, number

Description automatically generated

* Presentation Layer

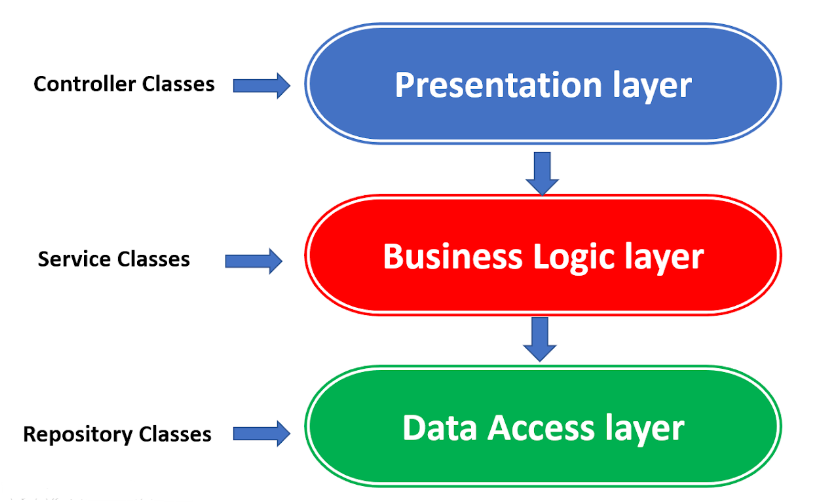
The presentation layer, also known as the user interface layer, is the topmost layer of the application. It is responsible for presenting information to the users and capturing their interactions. In the Question and Answer Platform, the presentation layer is implemented using Angular, a popular front-end framework. Angular allows for the creation of dynamic and responsive user interfaces, providing a seamless user experience.

* Application Layer

The application layer, also known as the business logic layer, acts as the intermediary between the presentation layer and the data layer. It contains the core business logic and rules of the application. In the Question and Answer Platform, the application layer is implemented using Spring Boot, a Java-based framework. Spring Boot provides a robust and efficient environment for developing RESTful APIs and managing the application's business logic.

* Data Layer

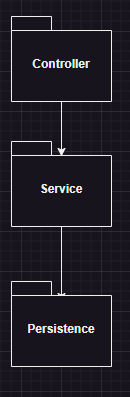
The data layer, also known as the persistence layer, is responsible for managing data storage and retrieval. It interacts with the underlying database to perform CRUD (Create, Read, Update, Delete) operations. In the Question and Answer Platform, the data layer is implemented using MySQL, a widely used relational database management system. MySQL offers reliable and efficient data storage capabilities, ensuring the persistence and integrity of application data.



1. **Package Diagram**

Following the Layered Architecture we have a delimitation into multiple packages, each one being responsible for a different layer in the architecture.

Along the three packages, there are also some other packages, each one being responsible for different functions, such as DTOs, Configuration



1. **Class Diagram**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Database Diagram**