A black and red logo

Description automatically generated

Nicholas Ifeanyi

SD Assignment 1

Group 30432

This is a simple assignment to simulate the functionality of the stack overflow website where we have users who make questions and answers, and the users interact with each other. In this iteration I have implemented the CRUD operations for the user, question and answer entities to be specified below

**1. User.java**

Purpose:

* Represents a user in the system.

Attributes:

* **cnp**: The unique identification number of the user.
* **username**: The username of the user.
* **password**: The password of the user.
* **role**: The role of the user (e.g., admin, regular user).
* **email**: The email address of the user.
* **questions**: A list of questions authored by the user.

Relationships:

* One-to many relationship with questions.
* One-to-many relationship with answer.

Usage:

* Used to store user information and their relationship with questions.

**2. Question.java**

Purpose:

* Represents a question posted by a user in the system.

Attributes:

* **id**: The unique identification number of the question.
* **author**: The user who posted the question.
* **title**: The title of the question.
* **question**: The content of the question.
* **creationDate**: The date when the question was created.

Relationships:

* Many-to-one relationship with users.
* One-to-many relationship with answers

Usage:

* Stores information about questions posted by users.

To create documentation for the provided files, you can follow a structured format that includes descriptions of each component, its purpose, and how it fits into the overall system. Below is an example of how you can document each file:

**1. User.java**

Purpose:

* Represents a user in the system.

Attributes:

* **cnp**: The unique identification number of the user.
* **username**: The username of the user.
* **password**: The password of the user.
* **role**: The role of the user (e.g., admin, regular user).
* **email**: The email address of the user.
* **questions**: A list of questions authored by the user.

Relationships:

* Many-to-one relationship with questions.

Usage:

* Used to store user information and their relationship with questions.

**2. Question.java**

Purpose:

* Represents a question posted by a user in the system.

Attributes:

* **id**: The unique identification number of the question.
* **author**: The user who posted the question.
* **title**: The title of the question.
* **question**: The content of the question.
* **creationDate**: The date when the question was created.

Relationships:

* Many-to-one relationship with users.

Usage:

* Stores information about questions posted by users.

**3. Answer.java**

Purpose:

* Represents an answer to a question posted by a user in the system.

Attributes:

* **id**: The unique identification number of the answer.
* **author**: The user who posted the answer.
* **question**: The question to which the answer is posted.
* **answer**: The content of the answer.
* **creationDate**: The date when the answer was created.

Relationships:

* Many-to-one relationships with users and questions.
* Many-to-one relationship with user.

Usage:

* Stores information about answers posted by users to specific questions.

**4. UserRepository.java, QuestionRepository.java, AnswerRepository.java**

Purpose:

* Provide interfaces for interacting with the database tables (Users, Questions, Answers).

Usage:

* Extends JpaRepository to inherit CRUD operations and query methods.

**5. UserDTO.java, QuestionCreateDTO.java, QuestionUpdateDTO.java, AnswerCreateDTO.java, AnswerUpdateDTO.java**

Purpose:

* Data Transfer Objects (DTOs) used for transferring data between client and server.

Usage:

* Represent input/output data structures for creating, updating, and retrieving entities.

**6. UserService.java, QuestionService.java, AnswerService.java**

Purpose:

* Implement business logic for handling user, question, and answer operations.

Usage:

* Perform CRUD operations and business logic on user, question, and answer entities.

**7. UserController.java, QuestionController.java, AnswerController.java**

Purpose:

* Define RESTful endpoints for user, question, and answer operations.

Usage:

* Handle HTTP requests, call corresponding service methods, and return responses.

The query of this operations will be handled by postman to see the interaction with the database.