StackOverflow

Intro

We will design and implement a simple version of StackOverflow.

- Our system will have only one type of user
  (two types if you also implement the bonus features)

- No actions should be possible if the user is not logged in

1) Feature 1

- Users shall be able to ask questions. Each question must have an author, title, text, creation date & time and one or more tags. If an appropriate tag does not exist, the user must be able to create one.

- The list of questions shall be displayed, sorted by creation date. The most recent question should be displayed first.

- The user must be able to filter questions by tag or via a text search. The text search should check the question title.

2) Feature 2

- Each question may be answered one or more times by any user (including the original author).

- Each answer must have an author, text and creation date & time.

- Answers may be edited or deleted by **their** author.

- When displaying a question individually, the list of answers must also be displayed.

3) Feature 3

- Users may vote questions and answers (upvote and downvote, like and dislike).

- Each user may only vote **once** on each question or answer. Users cannot vote on their own answers or questions (Like&Dislike).

- On each voted question or answer, the vote count must be displayed
  (vote count = upvote/like count - downvote/dislike count). The vote count can be negative.

- The answers for a question must be sorted by their vote count. Answers with the highest vote count must be displayed first.

4) Bonus Feature 1

- Based on upvotes and downvotes or likes and dislikes, the system must compute a user score with the following rules:
  - Each user starts with 0 points.
  - Users gain points when:
    - ➢ Their question is voted up (+5 points per vote),
    - ➢ Their answer is voted up (+10 points per vote).
  - Users lose points when:

        - ➢ Their question is voted down (-2 points per vote),
        - ➢ Their answer is voted down (-2 points per vote),
        - ➢ They down vote an answer of another user (-1 point).
- The user score shall be displayed next to the author's name on each question / answer.

5) Bonus Feature 2

- Moderators are users with special privileges. They shall be able to:
    - Remove questions or answers if inappropriate,
    - Edit any question or answer on the site,
    - Ban users from the site indefinitely in case of bad behavior.
- Banned users must see a message indicating that they were banned when trying to login and should be unable to perform any other actions.
- The banned users must receive an e-mail when they are banned.

TECHNICAL REQUIREMENTS:

- Store the data in a **relational** database
- Use a LAYERED Architecture
- Documentation **mandatory**

DELIVERABLES:

- Documentation(see template)
- Source code in a Version Control System (GitHub, Gitlab, etc…)
- SQL Script for generating the DB model and for populating it with some test data.

DEADLINE A1: W5, at the beginning of the lab 15:00/17:00, later penalties will be applied.

GRADING Assignment 1

Points	**Functionality**
+1p	Entity Mapping
+1p	CRUD user
+1p	CRUD question
+1p	CRUD answers of a questions
+1p	Layered Architecture
+0.5p	Clean DataBase Design
+0.5p	Testing
+1p	Clean Code, OOP Concepts, Coding Conventions
+0-3p	Documentation

DEADLINE A2: W8, at the beginning of the lab 15:00/17:00, later penalties will be applied.

GRADING Assignment 1

Points	**Functionality**
+1p	Entity Mapping / Similar
+1p	User pages
+1.5p	Answer/Question Pages
+0.5p	Usage of JSON for Display & Testing / Similar

+1p	Usage of Modules & Components/ Similar
+1p	Clean Code
+0.5p	Usage of Services
+0.5p	Testing
+0-3p	Documentation

BIBLIOGRAPHY:

- Tour page of StackOverflow: https://stackoverflow.com/tour
- ORM Explained: What are ORM Frameworks? - KillerPHP.com
- Layered Architecture 1. Layered Architecture - Software Architecture Patterns [Book] (oreilly.com)
- Hibernate Hibernate - Configuration (tutorialspoint.com)
- DAO Data Access Object Pattern (tutorialspoint.com)
- Entity Manager Operations JPA - Entity Managers (tutorialspoint.com)
- Entities relations Ultimate Guide - Association Mappings with JPA and Hibernate (thorben-janssen.com)
- Testing with Mockito https://site.mockito.org/
- Angular: https://app.pluralsight.com/course-player?clipId=aede2ad2-17b4-4ca9-8969-6153d659b049