Database Systems Project Proposal

Online Coding Platform

28.02.2022

Project Group No: 14

Ahmet Hakan Yilmaz - 21803399

Akin Kutlu- 21803504

Selbi Ereshova - 21901326

Zehra Erdem - 21801977

Overview

This is a system for coders, companies, and code editors. Coders can practice via coding challenges or non-coding questions. To assess their skill, they can participate in different contests and see their results and ranking in the system. Editors can create coding challenges and non-coding questions with different categories. Editors can organize a contest as a group or individually. Companies can sponsor these contests. Companies can also arrange interviews with coding challenges and non-coding questions and invite coders to interview via unique invitation codes. Coders can participate in these interviews, and the owner of the interviews can display the results. Companies can have mock interviews as well.

This report explains why a database system is required, what are the functional and non-functional requirements for this system, the limitations of the system, and finally, an Entity Relationship Diagram to represent the database that will be used.

Why Database System is Required

In this system, persistent data is needed. When the app is closed, the data must not be lost. Since this application will be used by several users and some of them will create questions and others will solve them, a considerable amount of data is needed to be stored and relations between data like who created which question, who solved which ones, and what is their results or which interview belongs to which company must be stored as well. Because of that, relational database systems will be used in this application. A relational database allows keeping different data types, the relationship between data, and querying vast amounts of data by joining different types or putting some predicates to queries. That gives flexibility that the application needs since there are too many relationships between data, and different kinds of querying will be needed during implementation.

Requirements

Functional requirements

1. **Login:** Every user will be able to login to the coding platform by entering their usernames and passwords.

2. **Register:** Users should register to the coding platform to use the website.

3. **Create challenge:** Editors can create coding challenges. Companies can also create challenges as part of their interview.

4. **Create non-coding questions:** Editors can create non-coding questions with multiple choice. Companies can also create non-coding open-ended or multiple-choice questions as part of their interview.

5. **Add solution:** Editors can add the solutions to questions that they created.

6. **Edit/delete question:** companies and editors will be able to delete or edit all questions that they have created.

7. **Create contest**: Editors can create contests consisting of coding challenges**.**

8. **Sponsor contests**: Companies can sponsor coding contests.

9. **Create interview**: Companies can create interviews from this platform. Interviews can include challenges and non-coding questions. Companies can also create mock interviews.

10. **Attend interview**: Coders that have an invitation from a company can attend the specific interview that the company has arranged.

11. **Solve challenge**: Coders can solve challenges by submitting their code for that challenge. They can solve challenges during practice, during contests, or interviews.

12. **Running test case**: When coders solve a challenge, they can run ready test cases, and they also can upload and run their own test cases.

13. **View submission**: A coder can see all their submissions. Moreover, companies can see submissions for their interviews.

14. **See solutions**: After a coder solves a question, the coder can see the solution. Another way to see a solution is to trade some points gained from challenges and contests.

15. **Attend contests**: Coders can attend contests.

16. **Gain points**: Coders will be able to gain points after solving coding challenges or participating in contests.

17. **View leaderboard for contests**: Users will be able to see the leaderboard of contests

18. **View ranking, view daily improvement**: Users will be able to see rankings which are determined according to their contest success and will be able to see their improvements with the solved questions via a graph.

19. **Sort challenges and questions**: Users will be able to sort the challenges according to different categories as they wish

20. **Give random questions:** Coders can select category and difficulty level to solve random challenges.

21. **Edit profile:** Companies, editors, and coders can edit their profiles. Users can change their personal information and their password.

22. **View profile of coders:** Users will be able to see the profiles of other users that include information such as name, school, age, rating, contest success.

Non-functional requirements

1. **Performance**: The system will respond to the user request in less than 3 seconds. This will ensure there is not much lagging, making it smoother for coders to code their answers.

2. **Usability:** The user interface will be straightforward with readable font, global navigation, and categorized pages. The user interface will contain icons and conventions adopted by many other applications, which will improve the user experience by familiarity [1], [2].

3. **Scalability:** Our application offers scalability for three different types of users. It will offer scalability as it is expected that a lot of people will use the application at the same time.

4. **Security:** As our website will be an important application for people who want to get jobs and companies who want to hire people, security is an important requirement for the application. Register includes verification processes to increase the security of everyone.

5. **Supportability:** Our application will be a web-based application, and it can run any machine with a web browser. Although it is easier to use the website on the computer, it can also be run on mobile phones.

Limitations

1. Points cannot be gained from interviews.

2. Editors can only create multiple choice non-coding questions.

3. A contest can consist of only coding challenges.

4. Solutions to coding challenges can be seen only after submitting a solution that passes all test cases or by trading with enough gained points.

5. A company cannot have coding challenges or non-coding questions that are not included in any of its interviews.

6. The result of any contest cannot be disregarded from the general rating.

7. In challenges, coders can gain points only by submitting their code.

8. To participate in interviews the coders should be registered in the system.

9. Coders cannot participate in a contest that has already passed.

10. Companies should communicate with coders via mail. There will not be any features for communication in this system.
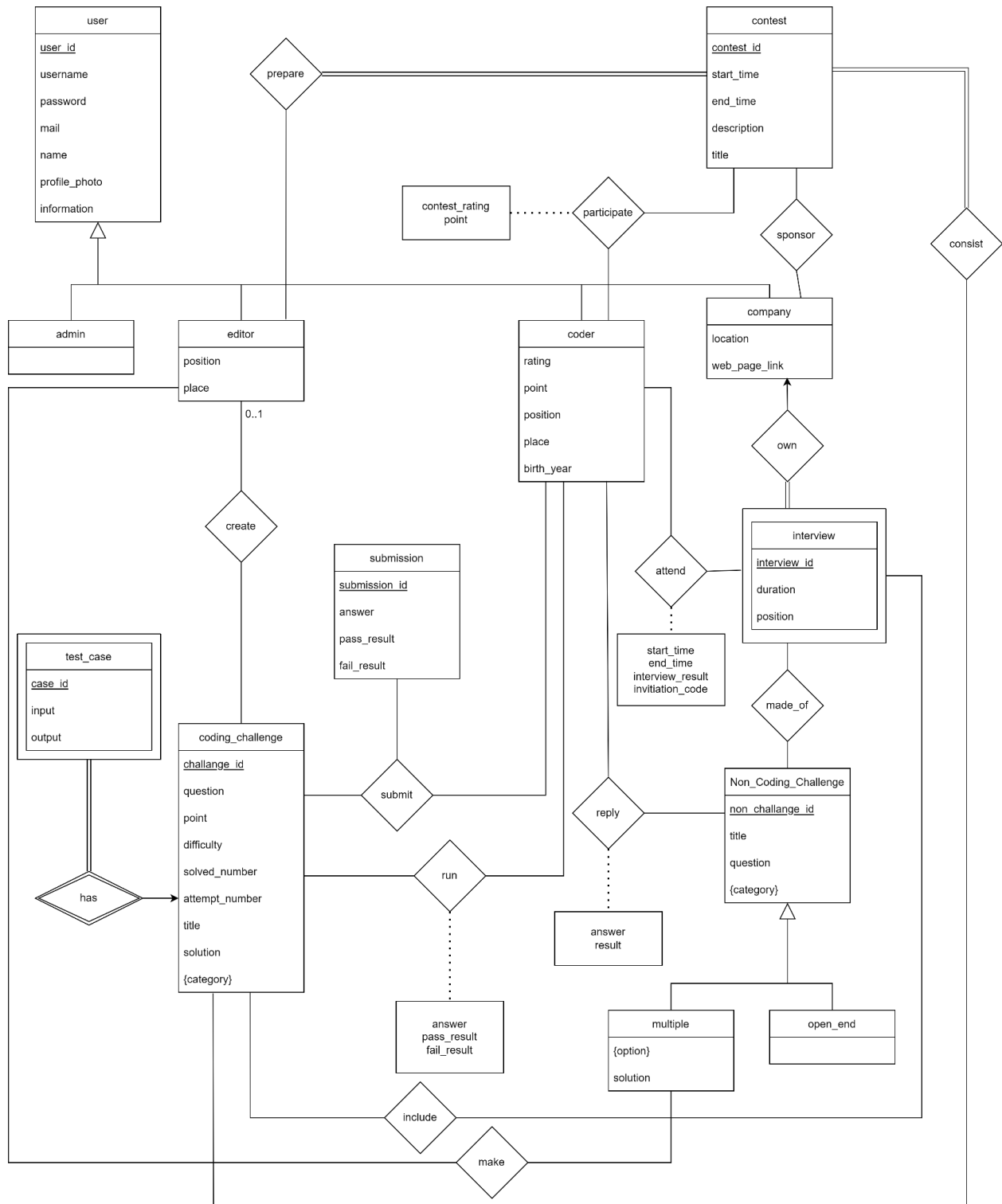
Entity Relationship Diagram

Figure 1: ER Model

The diagram (Figure 1) is drawn by using online draw.io [3].

Web Page Address

Glossary:

Rating: Our application will have a rating system that every coder has a rating which is determined according to their contest ratings.

Contest Rating: Contest rating is the rating that a user takes from a specific contest after participating.

Point: Our application will also have a point system that every coder can gain points with solving coding challenges. Coders can spend their points to see solutions of coding challenges if they can not solve the challenge.

References:

1. Leetcode.com. n.d. *LeetCode - The World's Leading Online Programming Learning Platform*. [online] Available at: https://leetcode.com/ [Accessed 27 February 2022].

2. HackerRank. n.d. *HackerRank*. [online] Available at: https://www.hackerrank.com/ [Accessed 27 February 2022].

3. "Diagrams.net - free flowchart maker and diagrams online," *Flowchart Maker & Online Diagram Software*. [Online]. Available: https://app.diagrams.net/. [Accessed: 27-Feb-2022].