CS 353 Database Systems

Project Proposal

Coding Practice and Interview System

Group 16

Tanay Toksoy | 21703919 - Sec 2

Ali Kemal Özkan | 21302087 - Sec 3

Osman Can Yıldız | 21302616 - Sec 3

Göktuğ Öztürkcan | 21702290 - Sec 2

2th March, 2020

Project Website

https://alikemalozkan.github.io/CodingPractice353/

Table of Contents

1. The Description of the Application System

Coding Practice and Interview System will provide a platform for companies and users. With using this platform, users will be able to practice coding, join tracks and companies will be able to interview and hire users.

For practice mode users can choose the subject and the difficulty of the questions they want to practice.

Tracks will be about certain subjects and will be composed of multiple questions each varying in difficulty. Each track will have it's own leaderboard. Users will be ranked according to the number of correct answers, the time it took to complete the track and the time complexity of their algorithms.

Developers will be able to create and submit their own questions and once they are checked by admins the questions will be published.

For the hiring process company representatives will be able to create their own set of questions and interview candidates for the hiring process.

This report is prepared for the proposal of the project. Requirements, how a database is going to be used, limitations explained clearly and E-R diagram of the system provided.

2. Why/How a Database is going to be used

This system is going to need a database since we will have to store data for various components. There are going to be different types of users ranging from normal users to companies. The database management system will help to store all the required information. Since the available questions are going to have categories, users will be able to search for their desired categories or even filter them. The developers will be able to make additions to the database since they will be allowed to ask their own questions while company representatives will be able to add interview questions. After answering a question, the user will get a score depending on their answer and their scores will also be stored in the database to display them in the leaderboard.

3. Requirements

3.1. Functional Requirements

3.1.1. Sign Up

People who want to become a member will sign up by creating an account with a unique username and password. Also they will share additional info about themselves. After the sign up process, they will create their profile by entering name, surname and email address.

3.1.2. Log in

If the user is already a member, the user will just need to enter their username and password to log in.

3.1.3. Log off

Users have the option to log off from the account.

3.1.4. Search & Filter & Choosing Category

Users will be able to search for a specific practice question to their needs. Filter part is used to filter the questions according to questions' status as solved or unsolved, questions' difficulty as easy, medium or hard and keyword that questions include. Users can also choose the category of the question for a quick filter.

3.1.5. Adding Question to Application

Developers can add the question to the application. Also company representatives can add interview questions for candidates for the hiring process. The questions are published if and only if admins give approval to the question.

3.1.6. Removing Question

After the admin checked the questions that are wanted to be added by developers or company representatives, the admin can remove the question according to its appropriateness.

3.1.7. Questions

All questions will be related to coding. After solving a question, developers can submit their answer and output will be checked according to at least 5 test cases.

3.1.8. Tracks

Tracks are basically question sets. It will include multiple questions with specific subject. Users will collect points from tracks and results will be published in the leaderboard.

3.1.9. Interviews

Companies may arrange interviews which consist of multiple questions. They can publish an interview through their company representatives. Results for interviews can be used for hiring process.

3.1.10. Leaderboard

Results of the tracks will bring points to developers. The points will be used for the leaderboard. That will create a competitive environment.

3.1.11. Profile

Every user will have a profile page in the application. Basic info and short bio of the user will be appearing on the profile page. Also rank, points info will be showed.

3.2. Non Functional Requirements

3.2.1. User Interface

User interface of the application will be user-friendly (easy to use) so that users can learn the interface within one minute without reading the user manual.

3.2.2. Privacy

The data of users will not be shared with third party apps.

3.2.3. Response Time

Response time of each action will be no longer than 3 seconds so that users can use the app effectively.

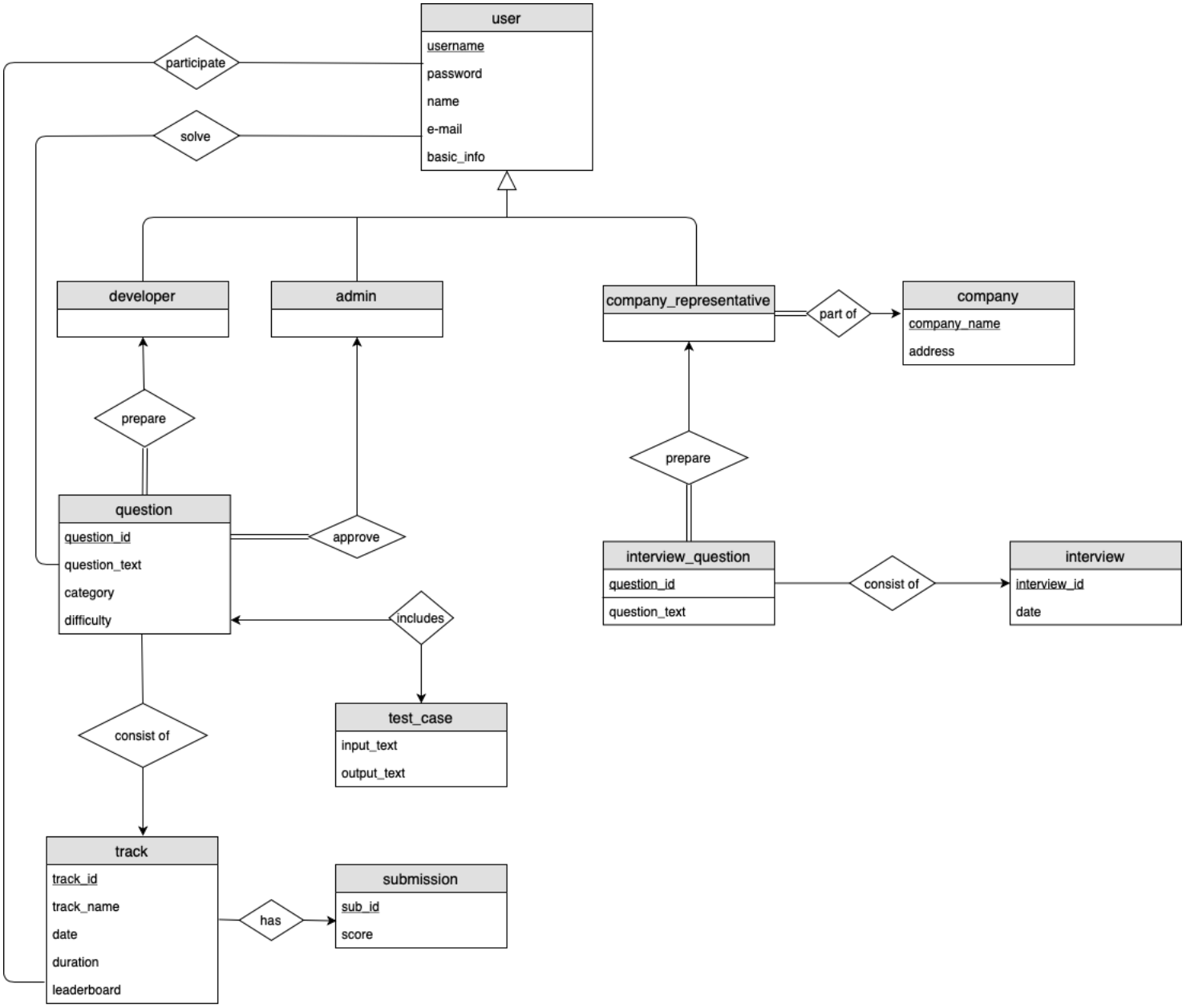
3.2.4. Reliability

System will not cause a data loss after the submission of a solution.

4. Limitations

- Authorization of users will prevent any disallowed access and manipulation of the data.
- Username will be unique for all users.
- Only admins can check questions before publishing.
- Questions will be related to only computer engineering.
- Only companies can arrange interviews through their representatives.

5. E-R Diagram


user
username
password
name
e-mail
basic_info
participate
solve
developer
admin
company_representative
part of
company
company_name
address
prepare
prepare
question
question_id
question_text
category
difficulty
approve
includes
interview_question
question_id
question_text
consist of
interview
interview_id
date
test_case
input_text
output_text
consist of
track
track_id
track_name
date
duration
leaderboard
has
submission
sub_id
score