CS353 Database Systems

Online Coding Platform
Project Proposal

<u>Group 12</u>

Mert Atakan Onrat	21802520
Egemen Öztürk	21801601
Yiğit Emiray Dinç	21704275
Okan Alp Ünver	21702632

Contents

1. Introduction

We are designing an online platform that users can conduct technical job interviews. Our Online Coding platform will have three primary users: users, editors, and companies. Editors will have the chance to prepare coding challenges, coding competitions, and non-coding questions for their goods. Users on our platform will participate in these contests and have the right to solve coding-related or non-coding questions while facing the challenges. On the other hand, companies will be able to make preparations for the benefit of the interviews. Companies will have the right to prepare coding challenges and non-coding questions similar to editors. In addition, companies can also sponsor these challenges. Both challenges and questions will have categories, and these categories will have a dynamic system design.

The purpose of the project proposal is to summarize the application we are developing, limitations, non-functional and functional requirements, a website link to our platform, and how the system will be integrated using our database's E/R model.

The proposal begins with a Project Description that includes a summary of the project and an approach to demonstrate how the database would be used in this project. It will then explain why we need a database for the Online Coding Platform and how we are preparing ourselves to use SQL for the project's database.

The proposal continues with functional and non-functional requirements. This section will look at functional and non-functional requirements and provide clarification for our requirements. Functional requirements are essential for building the site's properties and characteristics. The nonfunctional criteria will concern our system's scalability, security, performance, economics, and usability. Then, there is a part that examines the platform's limitations in the limitations section.

In the end, there is an E/R Diagram of the Online Coding Platform, which is the basis of the system's database.

2. Project Description

This application will be an online coding platform for all users. Companies and editors will be the primary content makers, just like Hackerrank. Ordinary users and editors, and companies will use the application. The system will have personal IDs and passwords to access the procedure quickly and securely. Names, surnames, and dates of birth will be kept. In addition, users will have achievements, education, and submission information. Editors will have prepared questions, contests, and challenges and have an id like every user. Categories will have the two most necessary information: id and name. Companies will have access to interviews and coding contests. Coding contests will have information for a brief description just as a Person entity.

3. Limitations

For this database design we mainly focused on the description that was given. That is, as a nature of such preparation, it is accepted that some of the requirements that are not asked in the description are added by us. Moreover, the main concern of ours was to keep every structure simple and lightweight as much as possible to add or remove main constructs of our design in case of fatal flaw. As we learn more about the design we come up with a solution that is flexible and easy to manipulate. Limitations can be understood and be explained by the following Requirements part since above mentioned requirements are also the design’s limitations as a lower bound needed to be satisfied.

4. Requirements

4.1 Functional Requirements

4.1.1 User

User can participate in coding contests.

User can try to solve coding challenges.

User can try to answer non-coding questions.

User can attend an interview.

4.1.2 Editor

Editor can prepare coding contests.

Editor can prepare coding challenges.

Editor can prepare non-coding questions.

Editor can prepare interviews.

4.1.3 Company

Company can sponsor coding contests.

Company can prepare interviews.

Company can prepare coding challenges.

Company can prepare non-coding questions.

4.1.4 Coding contest

Coding contests are attended by Users.

Coding contests are prepared by Editors.

Coding contests may have coding challenges.

Coding contests may have non-coding questions.

Coding contests can have sponsor companies.

4.1.5 Coding challenge

Coding challenge can form coding contests.

Coding challenge can be (un)solved by Users.

Coding challenge are prepared by Editors.

Coding challenge may be part of an interview.

4.1.6 Non-Coding Question

Non-coding challenge can form coding contests.

Non-coding challenge can be (un)solved by Users.

Non-coding challenge are prepared by Editors.

Non-coding challenge may be part of an interview.

4.1.7 Interview

Interview can be prepared by companies.

Interview can be participated by Users.

Interview can have coding challenges.

Interview can have non-coding questions.

4.2 Non-Functional Requirements

4.2.1 Scalability

The initial thought of our design had no limits on entity count. Thus, most of the design is conceptualized by countless numbers of users. However, this is not a smart way to implement a design that probably has a relatively lesser number of users. Hence, we figured out a design that can be easily scalable to more users while keeping in mind that we will probably have a lesser number of users.

4.2.2 Security

As in the Scalability part, our initial plan is to set a database that will be kept on our computers such that if our computers are secure then the data is secure. However, we know this is not a smart way to achieve security. Then we come up with a solution such as encrypting all the data that we store. Why we did not agree on this idea will be explained deeply in the Performance part. After refusing such a structure we thought that what we do will be tested by us so that we left Security requirements in outer space.

4.2.3 Performance

For the initial plan it is considered that we must design a structure that is simple and lightweight so that even wrong SQL queries ought to result as light as possible. Hence, our design arose. However, this is for sure not the best solution in performance vise. To keep lightweight we separated most of the entities and connected them with profound relationships.

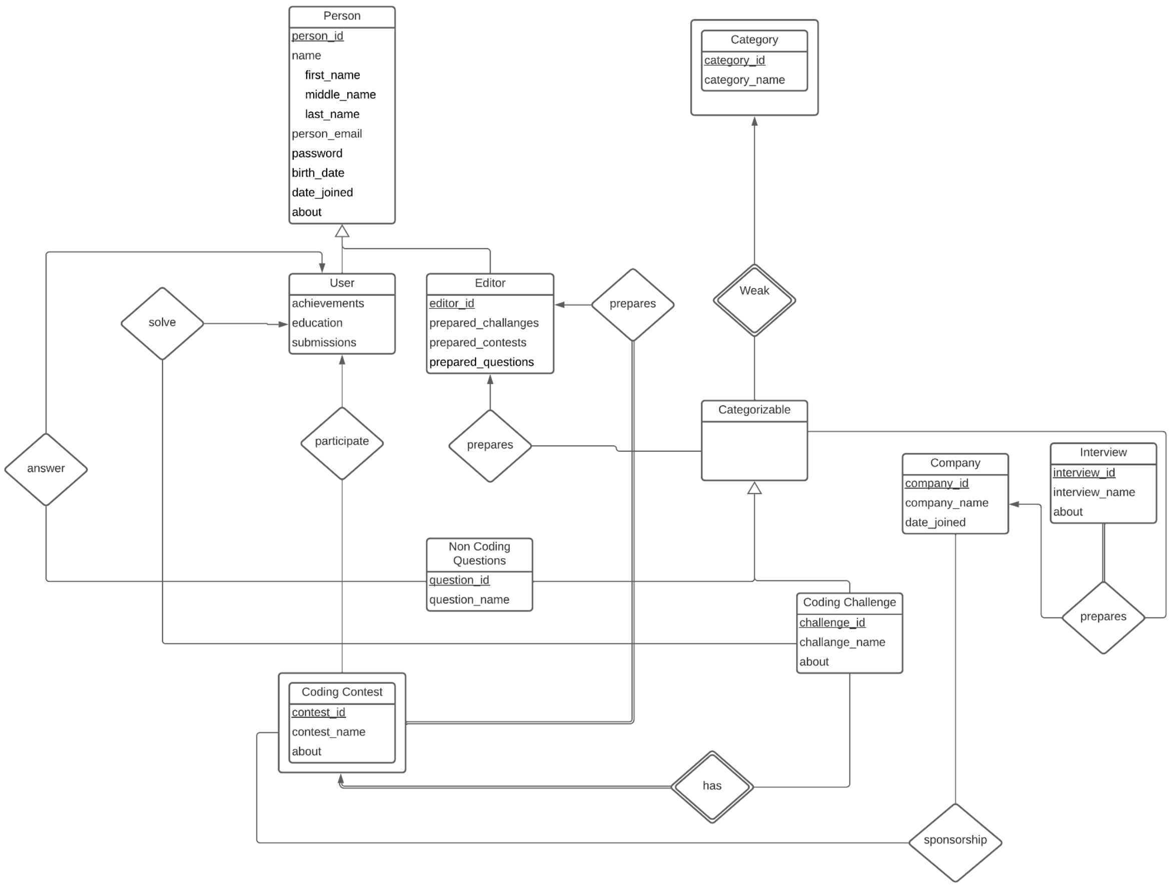
4.2.4 Economics

Economics vise this design is what we could get the best such as if we would have access to specialized computers for database needs, we would be able to design at most a slightly better version of design. Hence, we could get better performance and higher storage space if we could access it although it is not required directly, it could be worse economically.

4.2.5 Usability

Usability of this design will stand for itself at the end of this semester. For now, we can only expect it to be usable. We want to achieve such consideration with simple UI.

5. Entity Relationship Diagram



6. Website

For the upcoming news and current status of the project please refer to the following project webpage.

https://cs353ocp.github.io/cs353ocp/