Proctoring Software

Team: Welcome_to_IIIT

Introduction

Our mini-world is a generalized proctoring platform for conducting tests. This has been inspired by the current situation of the COVID-19 pandemic where universities need to conduct tests online. The environment is restricted only to conducting proctored tests with the help of a manual proctor, facilitating an examiner to check the responses, and eventually, release the scores.

Purpose

The database will help in storing the following data in an organized manner:

1. User details
2. Users' platform settings
3. Test details
4. Scores obtained
5. University details

This is a non-exhaustive list. The DBMS will also facilitate the analysis of students' performance across multiple tests, and even with each other.

Users

The users of the database can be categorized into:

1. **Examinee** who takes tests.
2. **Examiner** who is responsible for creating tests and checking responses.
3. **Proctor** whose job is to monitor the examinees while they are giving the test.
4. **University** which schedules tests.
5. **Support Staff** who help with issues related to the platform faced by other users.

Applications

1. University creates accounts for examinee, examiner.
2. University schedules tests.
3. Proctors can sign up on the platform.
4. Users can change their settings on the platform (dark/light mode, notification preferences, et cetera).
5. Users view their test scores.
6. Examinee can log in at the test's time and write the test.
7. Proctors monitor the examinees when they are writing the test. They can go back to the proctoring data to review an examinee's candidature.
8. Examiners can set tests by adding questions, answers, and can evaluate the performance of candidates.
9. Support Staff are able to view all the data and help users with any issue that they are facing.

Database Requirements

Assumptions

1. A student gives a test only once.
2. A question can only belong to one test. The question must be recreated if it needs to be used in another test.
3. User ID consists of a prefix (pr/er/ee for proctor, examiner, examinee) followed by their ID.

Entities

1. **User**
    a. Name
    b. User ID ***(Primary Key) (Composite Attribute)***
        i. User Type
        ii. ID
    c. Mobile Number ***(Alternate Key)***
    d. Email ID ***(Alternate Key)***

    Sub Classes: Examinee, Examiner, Proctor

2. **Test**
    a. Test ID **(*Primary Key)***
    b. Scheduled Time
    c. Time Duration
    d. Total Questions ***(Derived Attribute)***
    e. Maximum Marks ***(Derived Attribute)***

3. **University**
    a. University ID **(*Primary Key)***
    b. Name

4. **Proctoring Data *(Weak entity)***
    a. Student ID ***(foreign key)***
    b. Proctor ID ***(foreign key)***
    c. Test ID ***(foreign key)***
    d. Link to Audio
    e. Link to Video

5. **Question *(Weak entity)***
    a. Test ID ***(foreign key)***
    b. Marks Allocated
    c. Details ***(Composite Attribute)***
        i. Question Text
        ii. Link to Question Image ***(Multi-valued attribute)***
    d. Correct Answer ***(Multi-valued attribute)***

6. **Settings *(Weak entity)***
    a. User ID ***(Foreign Key)***
    b. Audio
    c. Video
    d. Theme

Weak Entities

1. Proctoring Data
2. Question
3. Settings

Relationships

1. ***Examination***
    An examinee takes a test which is monitored by a proctor. This generates the proctoring data.
    a. Entities: Examinee, Proctor, Test, Proctoring Data
    b. Degree: 4
    c. Attributes of relationship: Score
    d. Cardinalities:
        i. Examinee : Proctor = (M : N)
        ii. Examinee : Test = (M : N)
        iii. Examinee : Proctoring Data = (1 : N)
        iv. Proctor : Test = (M : N)
        v. Proctor : Proctoring Data = (M : N)
        vi. Test : Proctoring Data = (1 : N)

2. ***Belongs to***
    User(Examinee, Examiner) belongs to a University.
    a. Degree: 2
    b. Entities: Examinee, Examiner, University
    c. Cardinalities:
        i. User(Examinee, Examiner) : University = (N : 1)

3. ***Has***
    A user has his/her settings for the platform.
    a. Degree: 2
    b. Entities: User, Settings
    c. Cardinalities:
        i. User : Settings = (1 : 1)

4. ***Creates***
   An examiner creates questions for a test.
   a. Degree: 3
   b. Entities: Examiner, Question, Test
   c. Cardinalities:
      i. Examiner : Question = (1 : N)
      ii. Examiner : Test = (M : N)
      iii. Question : Test = (N : 1)

5. ***Schedules***
   A university schedules tests by providing the test time and test duration.
   a. Degree: 2
   b. Entities: University, Test
   c. Cardinalities:
      i. University : Test = (1 : N)

n>3 Relationship

The relationship ***Examination*** is of the 4th degree.

Subclass

The User entity type has 3 subclasses: Examinee, Examiner, Proctor

Special Attributes

1. Composite: *User ID* (of *User* entity type), *Details* (of *Question* entity type)
2. Multi-valued: *Link to Question Image*, *Correct Answer* (both belong to *Question* entity type)
3. Derived: *Total Questions, Maximum Marks* (both belong to *Test* entity type)
   a. *Total Questions* is derived by querying the Question table for all questions with a Test ID.
   b. *Maximum Marks* is derived by summing up the marks allocated to each question of the above query.

Bonus

1. In the *User* entity type:
   a. Candidate Keys: User ID, Mobile Number, Email ID
   b. Primary Key: User ID
   c. Alternate Keys: Mobile Number, Email ID
   d. Super Key: (Name, Mobile Number) may act as a super key.

2. The *User* entity type is related to itself in the relationship *Examination.* The *Proctor* monitors the *Examinee*, both of which are subclasses of *User*. The role is **‘supervision’**.

Functional Requirements

Retrievals

1. **Selection**: Retrieve all the tests scheduled by a particular university.
2. **Projection**: Retrieve all examinees who have scored above 80% in a particular test.
3. **Aggregate**: Find the average score of a particular test.
4. **Search**: Search questions with a particular keyword by comparing the keyword with the Question Text.
5. **Analysis**:
    a. Name of examinees who belong to a particular university and have given a particular test.
    b. For a particular test, publish the proctoring data details if more than 50% of the examinees monitored by a particular proctor have scored more than 90% marks.

Modifications

1. **Insert**:
    a. Insert the mobile number of a user and check if that is 10 digits long.
    b. Create a question and verify if all question images are of a valid image format.
2. **Modify**:
    a. Modify users’ settings.
    b. Update the answer key, and recalculate scores of users who gave that test.
    c. Modify the time duration of a test.
3. **Delete**:
    a. Delete users when they leave the university.

Rohan Chowdary V M	*2020101130*
Radheshyam Thiyagarajan	*2020115009*
Priyanshul Govil	*2021121002*