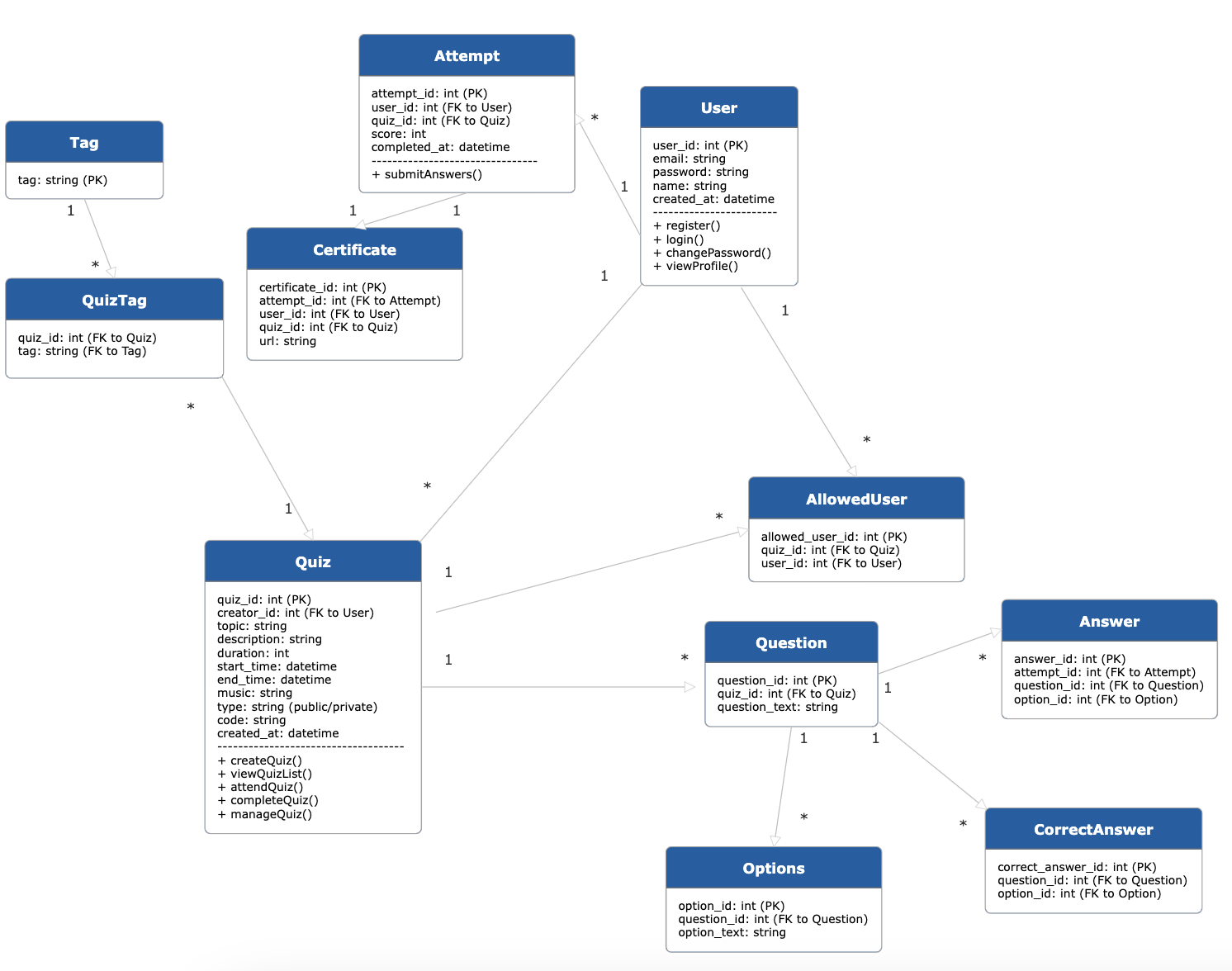
# **Quiz Application Documentation**

## **Introduction**

This documentation provides a detailed overview of the functionalities and requirements for the Quiz Application. The application allows users to register, login, create quizzes, attend quizzes, and manage their profiles. It also includes functionalities for quiz creators to manage their quizzes and view attendee results.

## **Class Diagram**



## **User Authentication**

### **User Registration**

* **Endpoint**: /register
* **Method**: POST
* **Parameters**:
  + email: User's email address (string, required)
  + password: User's password (string, required)
* **Description**: Allows users to register using their email and password.

### **User Login**

* **Endpoint**: /login
* **Method**: POST
* **Parameters**:
  + email: User's email address (string, required)
  + password: User's password (string, required)
* **Description**: Allows users to login using their email and password.

### **Change Password**

* **Endpoint**: /change-password
* **Method**: POST
* **Parameters**:
  + old\_password: User's current password (string, required)
  + new\_password: User's new password (string, required)
* **Description**: Allows logged-in users to change their password.

## **Quiz Management**

### **View Quiz List**

* **Endpoint**: /quizzes
* **Method**: GET
* **Parameters**:
  + topic: Filter by topic (string, optional)
  + tags: Filter by tags (string, optional)
* **Description**: Displays a list of quizzes. Public quizzes are enabled, and private quizzes are enabled only if the user has permission to attend.

### **Create Quiz**

* **Endpoint**: /create-quiz
* **Method**: POST
* **Parameters**:
  + topic: Quiz topic (string, required)
  + tags: Quiz tags (array of strings, predefined, required)
  + description: Quiz description (string, required)
  + questions: List of questions (array of objects, required)
    - Each question includes:
      * question\_text: Question text (string, required)
      * options: List of options (array of strings, 4 required)
      * correct\_answers: List of correct answers (array of strings, 1 or more required)
  + duration: Duration of the quiz (integer, required)
  + slot: Time slot for the quiz (object, required)
    - start\_time: Start time (datetime, required)
    - end\_time: End time (datetime, required)
  + music: Music for the quiz (string, predefined, required)
  + type: Quiz type (string, public/private, required)
  + allowed\_users: List of allowed users (array of emails, optional, required if private)
* **Description**: Allows users to create a quiz with various settings. A unique 6-digit code is generated for each quiz.

### **Attend Quiz**

* **Endpoint**: /attend-quiz
* **Method**: POST
* **Parameters**:
  + code: 6-digit quiz code (string, required)
* **Description**: Allows users to attend a quiz by entering the 6-digit code. Checks if the quiz is public or if the user has permission to attend.

### **Quiz Completion**

* **Endpoint**: /complete-quiz
* **Method**: POST
* **Parameters**:
  + quiz\_id: Quiz ID (string, required)
  + start\_time : StartTime (timestamp, required)
  + end\_time : StartTime (timestamp, required)
  + answers: User's answers (array of objects, required)
    - Each answer includes:
      * question\_id: Question ID (string, required)
      * selected\_answers: List of selected answers (array of strings, required)
* **Description:** Handles quiz completion. Checks if the user scored 80% or more and awards certificates accordingly. If the user completes and crosses 80% of the answers correct in 50% of the given time, a special certificate will be generated. Redirects to the home page after completion.

## **User Profile**

### **View Profile**

* **Endpoint**: /profile
* **Method**: GET
* **Description**: Displays the user's profile, including all quizzes attended and certificates received. Allows users to view correct answers and their mistakes.

### **Quiz Creator Management**

* **Endpoint**: /manage-quiz
* **Method**: GET/POST/DELETE
* **Description**: Allows quiz creators to view who attended their quizzes, their scores, and the leaderboard. Quiz creators can also modify or delete their quizzes.

### 

### **Entities and Relationships**

1. **User**
   * user\_id (PK)
   * email
   * password
   * password hash
   * name
   * created\_at
2. **Quiz**
   * quiz\_id (PK)
   * creator\_id (FK to User.user\_id)
   * topic
   * description
   * duration
   * start\_time
   * end\_time
   * music
   * type (public/private)
   * code
   * created\_at
3. **Question**
   * question\_id (PK)
   * quiz\_id (FK to Quiz.quiz\_id)
   * question\_text
4. **Option**
   * option\_id (PK)
   * question\_id (FK to Question.question\_id)
   * option\_text
5. **CorrectAnswer**
   * correct\_answer\_id (PK)
   * question\_id (FK to Question.question\_id)
   * option\_id (FK to Option.option\_id)
6. **QuizTag**
   * quiz\_id (FK to Quiz.quiz\_id)
   * tag (FK to Tag.tag)
7. **Tag**
   * tag (PK)
8. **AllowedUser**
   * allowed\_user\_id (PK)
   * quiz\_id (FK to Quiz.quiz\_id)
   * user\_id (FK to User.user\_id)
9. **Attempt**
   * attempt\_id (PK)
   * user\_id (FK to User.user\_id)
   * quiz\_id (FK to Quiz.quiz\_id)
   * score
   * completed\_at
10. **Answer**
    * answer\_id (PK)
    * attempt\_id (FK to Attempt.attempt\_id)
    * question\_id (FK to Question.question\_id)
    * option\_id (FK to Option.option\_id)
11. **Certificate**
    * certificate\_id (PK)
    * attempt\_id (FK to Attempt.attempt\_id)
    * user\_id (FK to User.user\_id)
    * quiz\_id (FK to Quiz.quiz\_id)
    * url