**Functional Requirements:**

1. **Students**
   * Students and teachers must be able to register and log in using unique credentials.
   * Students can view available quizzes and submit answers.
   * Students can select a quiz, answer questions, and submit it for scoring.
   * Students can view their scores after submitting a quiz.
2. **Teachers**
   * Teachers must be able to add, edit, and remove questions in a quiz.
   * Teachers can create, edit, and delete quizzes.
   * Teachers can view the performance of all students who participated in their quizzes.
3. **General**:
   * Each question must support multiple-choice answers.
   * The system must save the student’s score history.
4. **Database Integration**:
   * All data related to users, quizzes, and scores must be stored in a local SQL database.

**Non-Functional Requirements:**

1. **Performance**:
   * The application must support up to 100 concurrent users without significant latency.
2. **Usability**:
   * The UI must be intuitive and responsive for both desktop and mobile devices.
   * Students and teachers should be able to navigate the system with no more than three clicks to access core functionality.
3. **Security**:
   * Passwords must be encrypted before storage.
   * Teachers should have access to their quizzes and associated scores only; students should not access other students’ scores.
4. **Reliability**:
   * The application must recover from database connection issues within 5 seconds.
   * Data integrity must be maintained even in case of system crashes.
5. **Maintainability**:
   * Code must follow standard naming conventions and be well-documented.

**OUR DATABASE:**

**Tables:**

* Teachers: used as login info storage and profile of teachers.
* Students used as login info storage and profile of students.
* Quizzes used to store quiz meta data and as history of all quizzes.
* Questions used to store the question, the options and the correct option.
* StudentScores uses a quiz id, student id as foreign keys to act as linking table, also stores score.

**Detailed Tables View:**

**Teachers**

* TeacherID (Primary Key) - Unique identifier for teachers.
* Name - Teacher's name.
* Email - Teacher's email.
* Password - Encrypted password.

**Students**

* StudentID (Primary Key) - Unique identifier for students.
* Name - Student's name.
* Email - Student's email.
* Password - password

**Quizzes**

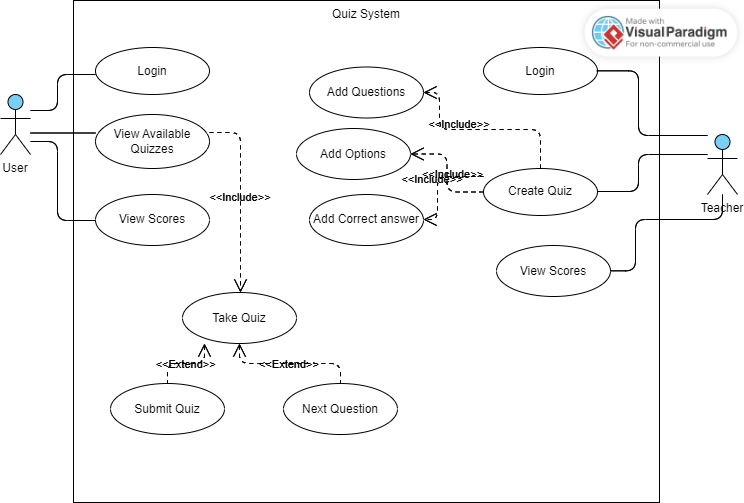
* QuizID (Primary Key) - Unique identifier for quizzes.
* Title - Title of the quiz.
* Description -Description of the quiz.
* TeacherID (Foreign Key) - Links to Teachers.

**Questions**

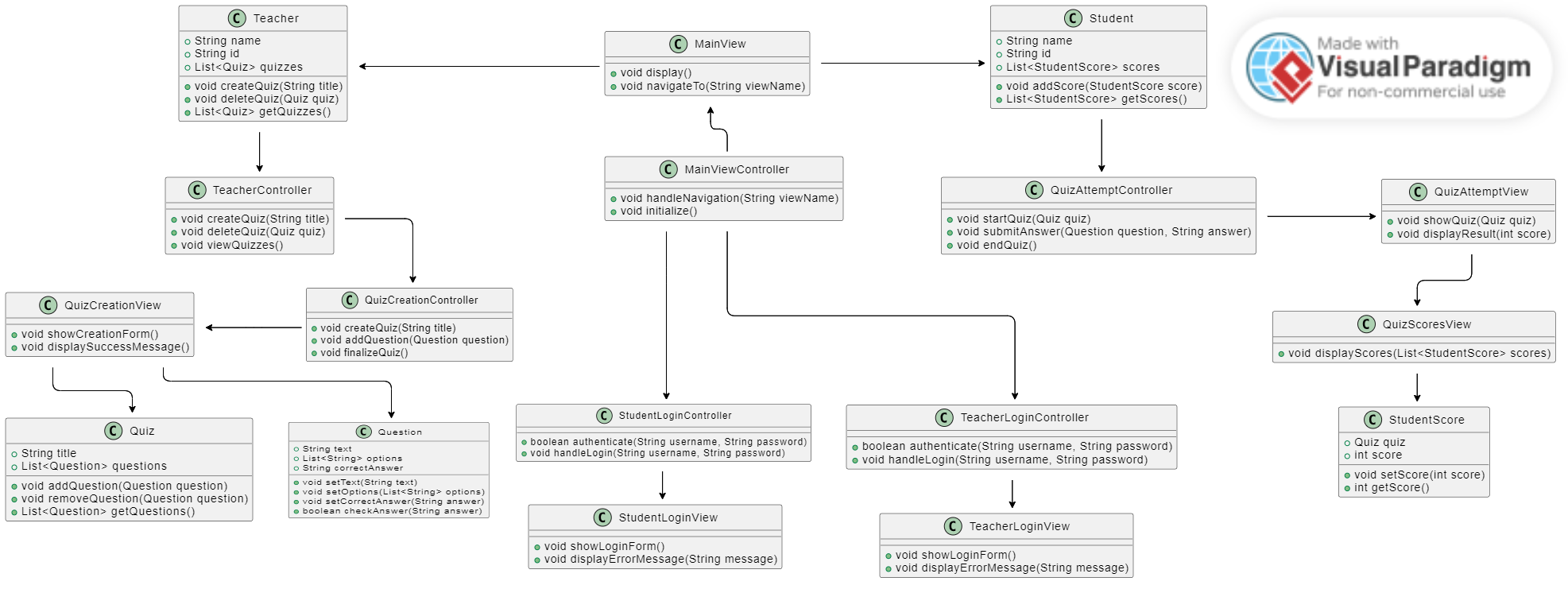
* QuestionID (Primary Key) - Unique identifier for questions.
* QuizID (Foreign Key) - Links to Quizzes.
* QuestionText -Text of the question.
* OptionA - First answer option.
* OptionB - Second answer option.
* OptionC - Third answer option.
* OptionD - Fourth answer option.
* CorrectAnswer-For automated checking.

**StudentScores**

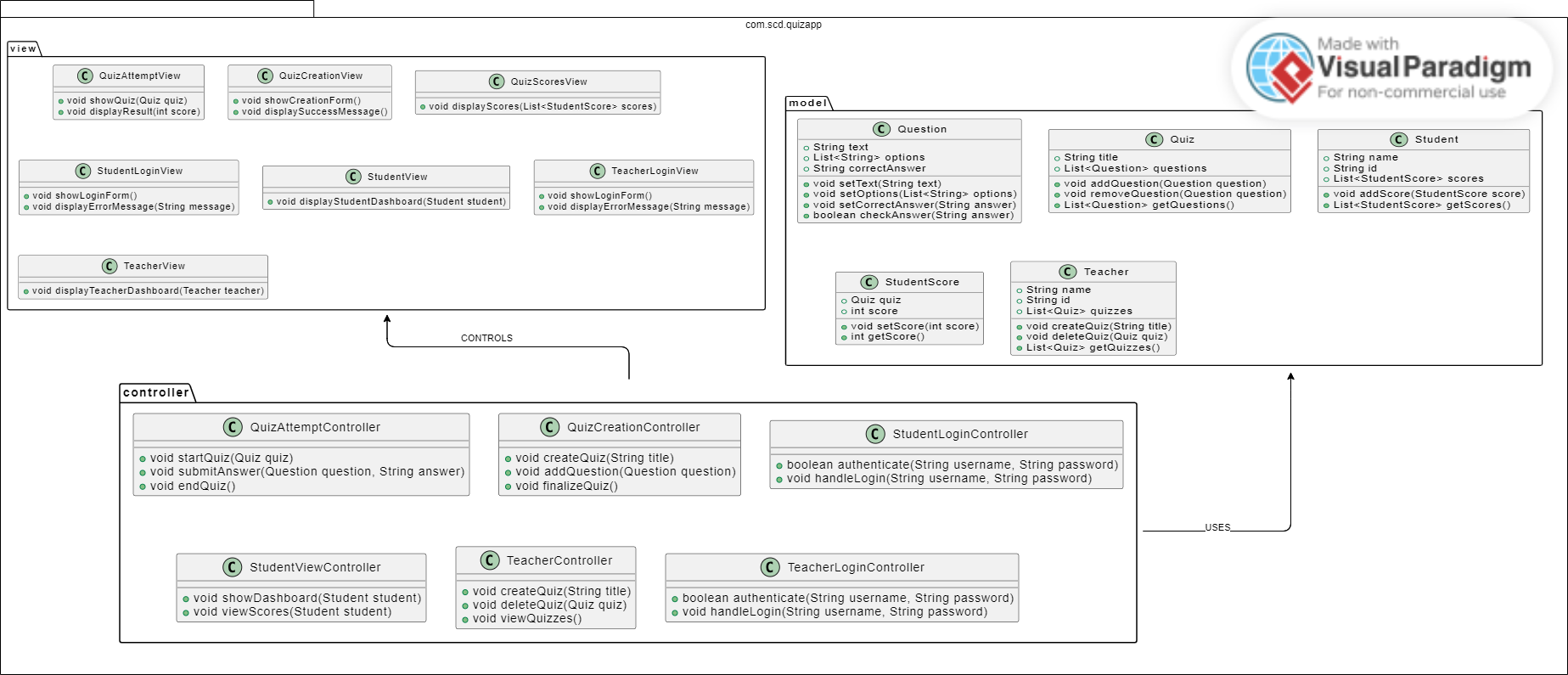
* ScoreID (Primary Key) - Unique identifier for scores.
* StudentID (Foreign Key) - Links to Students.
* QuizID (Foreign Key) - Links to Quizzes.
* Score -opbatined marks

**USE CASE:  
**

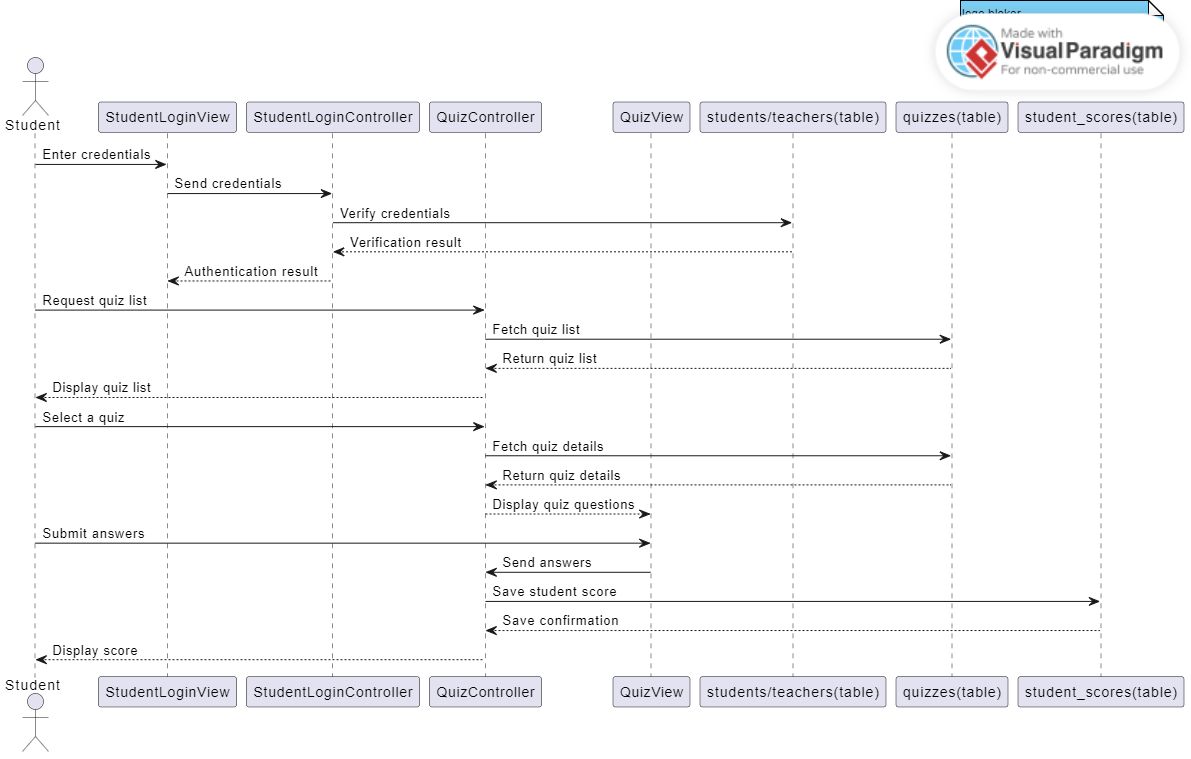
**CLASS DIAGRAM:**

****

**PACKAGE DIAGRAM:**

****

**SEQUENCE DIAGRAM:**

****

**Some Class Functionalities to keep in mind:**

 **Question**:

Will be used to generate an object for each separated question the teacher makes.

 **Quiz**:

It’s an Object that primarily holds list of <Questions>, title, and Quiz information received from the database.

 **Student**:

Manages student information such as ID, name, and login credentials.

 **StudentScore**:

An Object for scores for quizzes taken by students, including quiz ID and score details. All of this data will come from database.

 **Teacher**:

Stores teacher data like ID, name, and login credentials.

### ****Controller Classes****:

### ****MainViewController****: The Navigator of the whole application.

### ****QuizCreationController****: Facilitates the teacher to create a quiz and save it into the database.

### ****StudentLoginController****: Authenticates student credentials and manages session data.

### ****TeacherController****: Handles teacher-specific operations like managing quizzes and viewing scores.