| Academic Year | Module | Assessment Type |
|---|---|---|
| 2025 | Object Oriented Design and Programming | Report |

# Java Quiz Application – Project Report

| | |
|---|---|
| Student Id | : [2432224] |
| Student Name | : Natasha Shrestha |
| Section | : L5CG24 |
| Module Leader | : Subash Bista |
| Tutor | : Anil Poudel |
| Submitted on | : 02-12-2025 |

# Introduction

The Java based quiz applications is basically an interactive system that enables the users to sign up, log in, participate in quizzes, and track their scores. Additionally, an admin panel is incorporated to facilitate question management and user performance monitoring.

The application is developed using:

- Java Swing for the graphical user interface (GUI)
- MySQL for data storage and retrieval
- JDBC (Java Database Connectivity) for database interaction

This report provides an overview of the system's architecture, features, database design, and potential improvements.

## Systems Overview

The system consists of two primary user roles:

Regular Uses:

- Register and log in
- Participate in multiple-choice quizzes
- View and track their scores

Administrators:

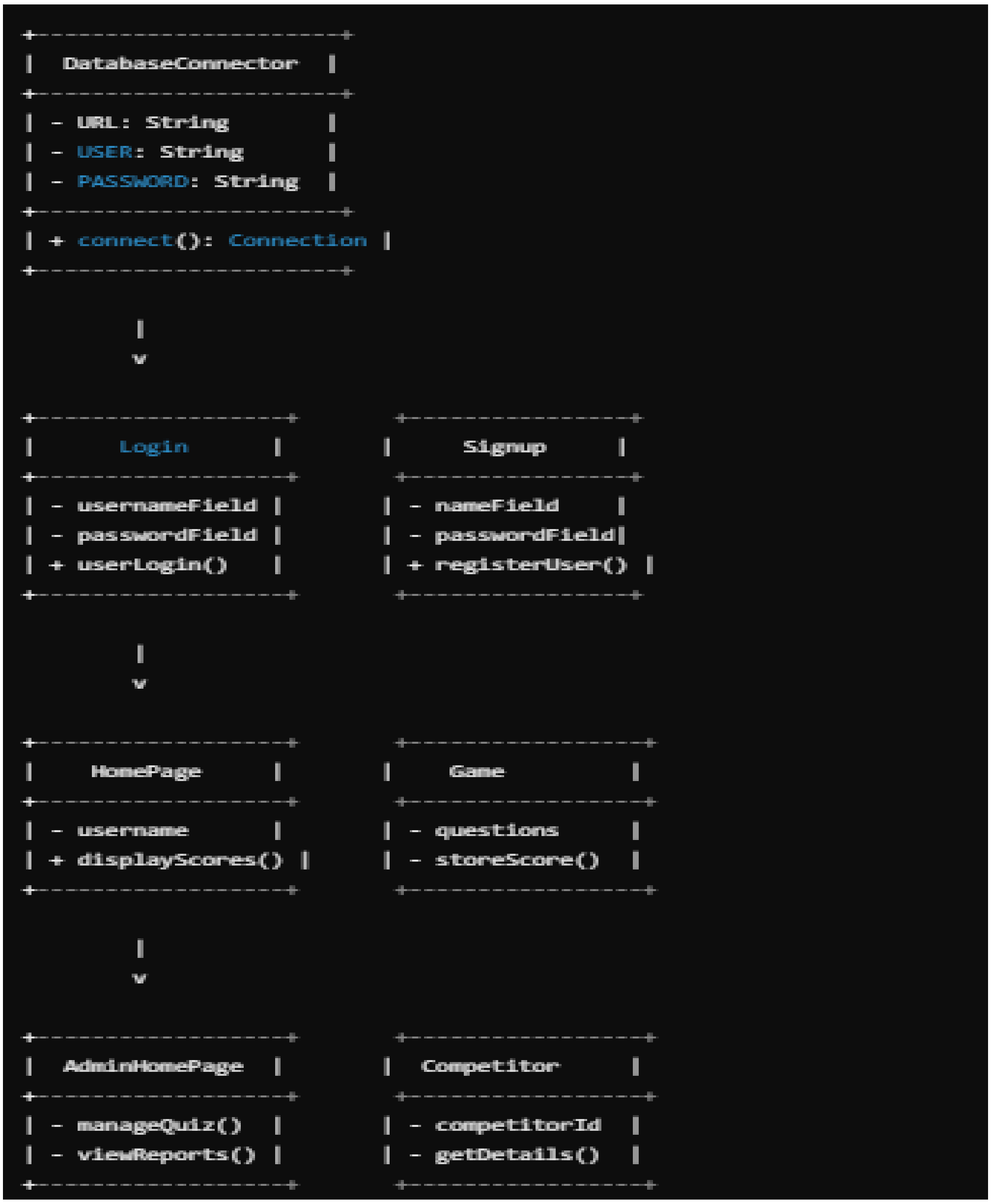
- Add, modify, and delete quiz questions
- Monitor user performance through score reports

System Flow

- User Registration – New users create an account with a name, password, and country.
- User Authentication – Users log in with their credentials.
- Quiz Participation – Users answer a series of multiple-choice questions.
- Score Calculation – The system evaluates the answers and records the user's score.
- Score Tracking – Users can view their past performance, and administrators can access overall statistics.

## UML Diagram

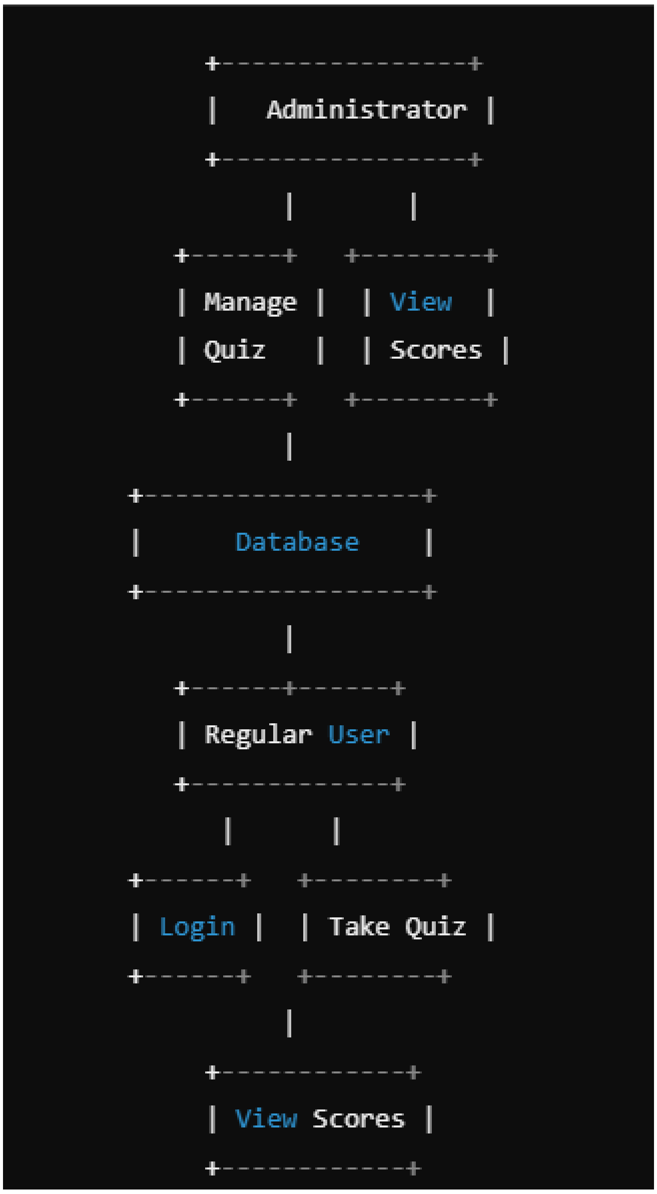
The Class Diagram below represents the system’s core components and their relationships:



```
+-----------------------+
|   DatabaseConnector   |
+-----------------------+
| - URL: String         |
| - USER: String        |
| - PASSWORD: String    |
+-----------------------+
| + connect(): Connection |
+-----------------------+

        |
        v

+-------------------+         +-----------------+
|       Login       |         |     Signup      |
+-------------------+         +-----------------+
| - usernameField   |         | - nameField     |
| - passwordField   |         | - passwordField|
| + userLogin()     |         | + registerUser() |
+-------------------+         +-----------------+

        |
        v

+-------------------+         +------------------+
|     HomePage      |         |    Game          |
+-------------------+         +------------------+
| - username        |         | - questions      |
| + displayScores() |         | - storeScore()   |
+-------------------+         +------------------+

        |
        v

+-------------------+         +------------------+
|   AdminHomePage   |         |  Competitor      |
+-------------------+         +------------------+
| - manageQuiz()    |         | - competitorId   |
| - viewReports()   |         | - getDetails()   |
+-------------------+         +------------------+
```

## Use Case Diagram

The Use Case Diagram highlights the system’s key functionalities and user interactions:



```
     +----------------+
     |   Administrator |
     +----------------+
          |       |
   +------+   +--------+
   | Manage |  | View   |
   | Quiz   |  | Scores |
   +------+   +--------+
          |
+------------------+
|      Database    |
+------------------+
          |
   +------+------+
   | Regular User |
   +-------------+
      |      |
+------+   +--------+
| Login |  | Take Quiz |
+------+   +--------+
          |
     +------------+
     | View Scores |
     +------------+
```

# Implementing Details

Database Design
The system uses MySQL as its database, structured as follows:

- Users Table: Stores user information.

```
CREATE TABLE Users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(255) NOT NULL UNIQUE,
   password VARCHAR(255) NOT NULL,
   country VARCHAR(255) NOT NULL
);
```

- Scores Table:
  Records quiz scores associated with users.

```
CREATE TABLE Scores (
   id INT PRIMARY KEY AUTO_INCREMENT,
   user_id INT NOT NULL,
   score INT NOT NULL,
   FOREIGN KEY (user_id) REFERENCES Users(id)
);
```

- questions Table:
  Stores quiz questions and multiple-choice options.

```
CREATE TABLE astrology_quiz (
   id INT PRIMARY KEY AUTO_INCREMENT,
   question TEXT NOT NULL,
   option_a VARCHAR(255) NOT NULL,
   option_b VARCHAR(255) NOT NULL,
   option_c VARCHAR(255) NOT NULL,
   option_d VARCHAR(255) NOT NULL,
   correct_answer VARCHAR(255) NOT NULL,
   difficulty VARCHAR(50)
);
```

## Key Functionalities

1.User Login

- Retrieves user credentials from the Users table.
- Authenticates users and redirects them to the Home Page.

2. User Registration (Signup.java)

- Validates user details and prevents duplicate registrations.
- Stores user data in the Users table.

3. Quiz Functionality (Game.java)

- Fetches quiz questions dynamically from the astrology_quiz table.
- Evaluates user responses and calculates scores.
- Stores scores in the Scores table.

4. Admin Panel (AdminHomePage.java)

- Enables administrators to add, delete, and update questions.
- Provides score reports to track user performance.

# Conclusion

The Java Quiz Application successfully integrates user authentication, quiz functionality, and database-driven score tracking into a seamless experience. Using Swing for the UI, MySQL for data storage, and JDBC for database interaction, the system ensures smooth quiz execution and user performance tracking.

This project can be expanded to incorporate AI-based question recommendations, multiplayer quizzes, and cloud-based storage.