UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Java (OODP)

Quiz Game

Student ID: 2407749

Student Name: Subash Bikram Tamang

Course Code: 5CS037

Table of Contents

Introduction

Final Portfolio project development is discussed in this report. The overall objective of the course is to develop a functional quiz program using Java with MySQL incorporated for managing data. A functional quiz program with user administration, question answering, and reporting is one of the expected deliverables.

Methodology

Water Fall model is used in my project. Where the process is being implemented step by step, and where each phase is completed before moving to next step where the water fall model is used. Some of the development tools include:

- Java programming of the backend logic
- Initialization of the GUI development
- MySQL database management
- Using JUnit for unit testing
- Eclipse IDE for coding and debugging

Implementation

1.1. Competitor Class Development

There is a ManagerPlayer class that handles quiz results and user information. The class has methods of handling user ID, calculating score, reading/writing player data, and handling player ID, username, and total/cumulative score.

1.2. MySQL and Arrays

The system employs a MySQL database for holding quiz questions and users' scores. The database design encompasses tables for queries, users, and scores. Java classes employ arrays for holding user answers and question sets dynamically. The score is determined using the correct answers in the database.

1.3. MySQL Integration and Reports

- The DatabaseSetup class creates a JDBC connection to MySQL.
- The ScoreManager class fetches and calculates scores from the database.
- performance.

1.4. Error Handling

To make the system more resilient:

- Try-catch blocks are employed to catch MySQL connection exceptions.
- Validation checks ensure that all input forms are valid.
- Consistent error messages simplify debugging and usability.

1.5. Testing

JUnit tests were developed for key features such as:

- AdminQnsPageTest.java tests question adding and form validation.
- Score computation tests are available in ScoreManager.
- Database connection tests ensure data retrieval consistency.

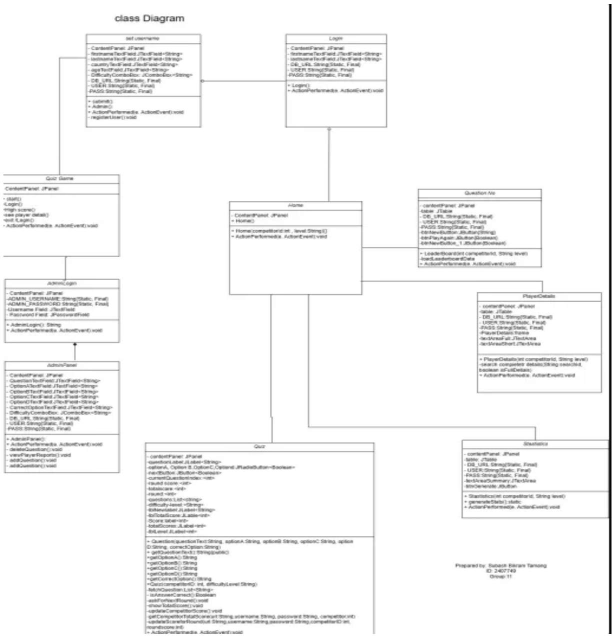
Javadoc Comments

Javadoc comments were included for significant classes and methods to allow for better code readability.
For example:

```
/**
 * Method saveQuestion.
 * Description: Saves a quiz question to the database.
 */
```

This documentation standardizes method explanations for better maintainability.

Class Diagram



Test Cases

Test Cases Include:

Test Case	**Description**	**Expected Output**
Add Question	Admin adds a new quiz question	Question stored in database
Score Calculation	Calculate total score	Correct total score displayed
Invalid Login	User enters incorrect credentials	Login failure message

Status Report

- Quiz question addition and retrieval.
- Login and scorekeeping for the user
- Integration with MySQL to save data

Certain user interface enhancements and report enhancements are needed.

Known Bugs and Limitations

- JavaDocs and module-info.java are out of sync, which is a JUnit dependencies problem.
- GUI Layout Changes: Some UI elements need better alignment.
- Improvements to Error Handling: Exception messages need to be more descriptive.

Conclusion

It effectively installs a quiz system that is attached to a database and can report it. May be involved in any potential upcoming promotions:

- Enhances the user interface through the modernized design structure
- Covering API support for the source of quiz questions from external platforms
- Increase performance in database questions

This report undertakes intensive evaluation of the project, details its actual undertaking and pinpoints areas of potential improvement.