

Mind Metrics (GUI based Quiz Application)

|  |  |  |  |
| --- | --- | --- | --- |
| Academic Year | Module | Assessment Number | Assessment Type |
| 2024-2025 | 5CS019 – Object Oriented Desing and Programming | A1 | Individual Report |

Student ID: 2407046

Student Name: Kamal Dhital

Group: L5CG5

Module Leader: Subash Bista

Tutor: Dhan Thapa

Submitted On: 2025/02/12

# Abstract:

This project focuses on the development of a Java-based Graphical User Interface (GUI) Quiz Application that adheres to SOLID principles of Object-Oriented Design. The application offers essential features such as user authentication, allowing secure registration and login, with role-based access control for admins and users. Admins can efficiently manage quiz content through CRUD operations on quiz questions, while users can take quizzes, track scores, and view their performance on a leaderboard. The system uses MySQL for data storage, enabling efficient management of user data, quiz questions, and scores. A key goal of this project was to ensure the application's modularity and scalability while maintaining a clean and maintainable codebase. The application also incorporates error handling and unit testing to ensure reliability and robustness. Despite minor issues such as quiz loading delays, the project successfully delivers a functional quiz platform. Future improvements are planned to enhance navigation, introduce quiz categorization, and add social features, further elevating user engagement.

Table of Contents

[Abstract: I](#_Toc190269904)

[1. Introduction 1](#_Toc190269905)

[Objective 1](#_Toc190269906)

[Expected Outcome 1](#_Toc190269907)

[2. Methodology 1](#_Toc190269908)

[2.1 Requirement Analysis 1](#_Toc190269909)

[2.2 Design 1](#_Toc190269910)

[2.3 Implementation 1](#_Toc190269911)

[2.4 Testing 2](#_Toc190269912)

[2.5 Tools & Technologies 2](#_Toc190269913)

[3. Core Development 2](#_Toc190269914)

[3.1 Class Development 2](#_Toc190269915)

[3.2 MySQL and Arrays 2](#_Toc190269916)

[3.3 Database Connection and Reports 2](#_Toc190269917)

[3.4 Error-Handling 2](#_Toc190269918)

[3.5 Testing 3](#_Toc190269919)

[Code Documentation 3](#_Toc190269920)

[Javadoc Usage 3](#_Toc190269921)

[System Design 3](#_Toc190269922)

[Class Diagram Overview 3](#_Toc190269923)

[6. Test Scenarios 4](#_Toc190269924)

[7. Project Status Report 5](#_Toc190269925)

[8. Future Enhancement 5](#_Toc190269926)

[9. Final Thoughts & Conclusion 5](#_Toc190269927)

[Appendix: III](#_Toc190269928)

# 1. Introduction

## Objective

This coursework focuses on designing and developing a Java-based GUI Quiz Application that adheres to the SOLID principles. The application features user authentication, allowing registration and login with role-based access. Admins can manage quiz content through a dedicated panel, performing CRUD operations on questions. Users can take quizzes, receive scores, and track progress via a leaderboard. The system ensures an intuitive, interactive experience with a user-friendly GUI. Built with modularity and scalability, the application maintains a clean, maintainable codebase, promoting flexibility and reusability while providing an engaging platform for both quiz administrators and participants.

## Expected Outcome

The expected outcome of this project is a fully functional and user-friendly quiz application with an intuitive graphical user interface (GUI). The system will allow users to register and log in securely, ensuring role-based access to either an admin panel or a user panel. Admins will have complete control over quiz content, enabling them to add, update, delete, and view questions seamlessly. Users will be able to play quizzes, receive instant feedback on their scores, and track their performance over time. Additionally, a leaderboard will display top-performing participants, fostering a competitive and engaging learning environment.

# 2. Methodology

The development process adhered to the stages of the Software Development Life Cycle (SDLC):

## 2.1 Requirement Analysis

During the requirement analysis phase, the key features of the application were identified to ensure a robust and functional system. The application supports user authentication, allowing users to register and log in securely. It also includes quiz management features, enabling CRUD operations for quiz questions, along with score tracking and a leaderboard to showcase top performers. Additionally, secure storage of user and quiz data was planned using a relational database structure, ensuring efficient management of user details, quiz questions, and scores.

## 2.2 Design

The system design was structured using UML Class Diagrams to visualize relationships between components, ensuring clarity and scalability. A strong emphasis was placed on modularity, reusability, and adherence to SOLID principles.

## 2.3 Implementation

The Graphical User Interface (GUI) was developed using Java Swing, providing an interactive and user-friendly experience. The system was integrated with a MySQL database using JDBC to ensure seamless data storage and retrieval. The implementation followed Object-Oriented Programming (OOP) principles, ensuring maintainability, scalability, and ease of future enhancements.

## 2.4 Testing

To ensure the reliability and correctness of the application, JUnit 5 was used for unit testing. Various modules were tested, including database operations, user authentication and validation, CRUD operations for quiz questions, and score calculations. These tests ensured that the system functioned as expected and met the predefined requirements.

## 2.5 Tools & Technologies

The application was developed using Java with MySQL as the database and JDBC for database connectivity. The Java Swing framework was used to build the GUI, while JUnit 5 ensured reliable testing. Git was employed for version control, and Javadoc was used for documentation to maintain clear and structured code references.

# 3. Core Development

## 3.1 Class Development

The Class was developed to manage user registration and login. It includes attributes such as full name, email, password, and role to distinguish between users. Upon authentication, users are directed to either the admin panel or the user panel based on their assigned role. Basic form validation ensures correct input handling, although the validation process is currently semi-validated and can be improved for better security and data integrity.

## 3.2 MySQL and Arrays

The application integrates MySQL Database to store and manage user or admin, questions and quiz data efficiently. The MindMetrics database consists of three key tables:

* account: Stores user information, including ID, full name, username, email, password, and role.
* quiz\_score: Maintains quiz performance records with ID, user ID, username, score, and quiz date.
* questions: Holds quiz data with ID, question text, multiple-choice options (A–D), correct choice, and difficulty level.

## 3.3 Database Connection and Reports

To enhance user experience, the application features a leaderboard that retrieves data from the quiz\_score table to display top performers. Additionally, reports are dynamically generated within the GUI, allowing users to view quiz statistics in tabular format. The Class communicates with other components to efficiently fetch and present relevant data.

## 3.4 Error-Handling

Error-handling mechanisms were implemented to prevent crashes and system failures during critical operations. Proper exception handling was introduced to manage database connectivity issues, authentication failures, and invalid quiz submissions. These measures ensure the application remains stable and resilient in real-world usage.

## 3.5 Testing

Comprehensive unit testing was conducted using JUnit5 to validate the system's core functionalities. Unit tests were written for:

* Adding and deleting quiz questions
* Database initialization and data retrieval
* Registration form validation
* Quiz participation and score calculation
* Updating and viewing quiz questions and scores

These tests helped ensure the application's reliability, functionality, and overall performance, making it robust and ready for deployment.

# Code Documentation

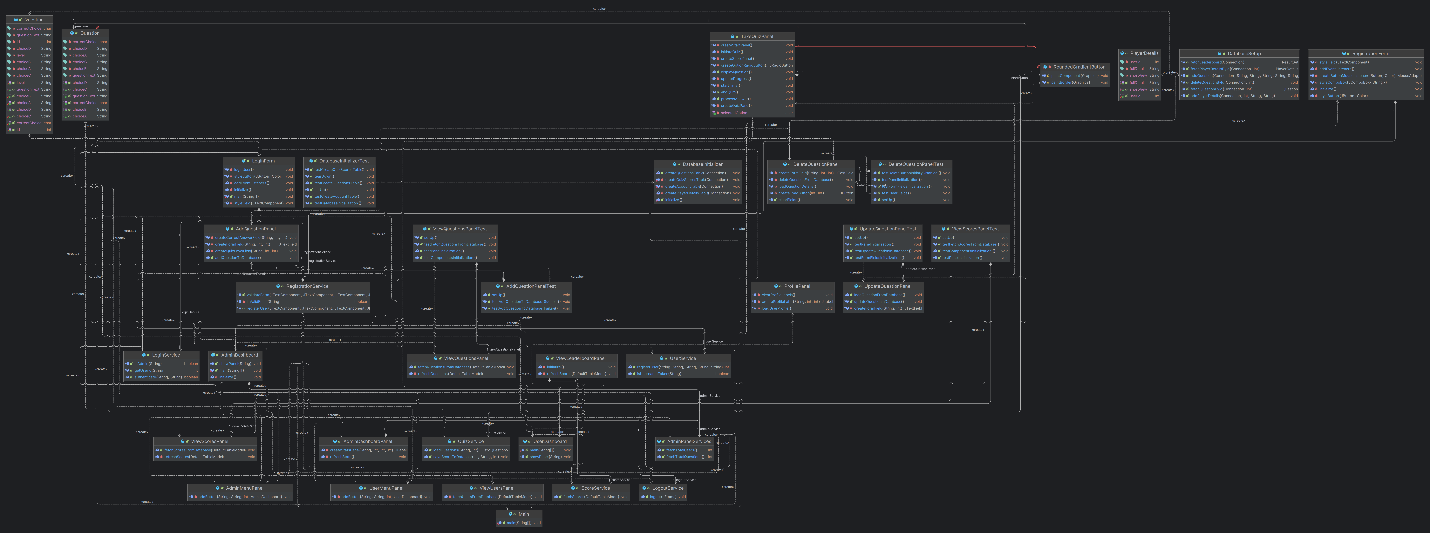
## Javadoc Usage

To ensure maintainability and code clarity, Javadoc comments were added to all major classes and methods. These comments provide detailed descriptions of each class's purpose, attributes, and functionality. The documented classes include DatabaseInitializer, DatabaseSetup, PlayerDetails, Question, AddQuestionPanel, AdminDashboard, AdminMenuPanel, DeleteQuestionPanel, LoginForm, ProfilePanel, RegistrationForm, TakeQuizPanel, UpdateQuestionPanel, UserDashboard, ViewLeaderboardPanel, ViewQuestionsPanel, ViewScoresPanel, ViewUsersPanel, and Main. This documentation serves as a comprehensive reference for developers, facilitating future modifications and enhancements.

# System Design

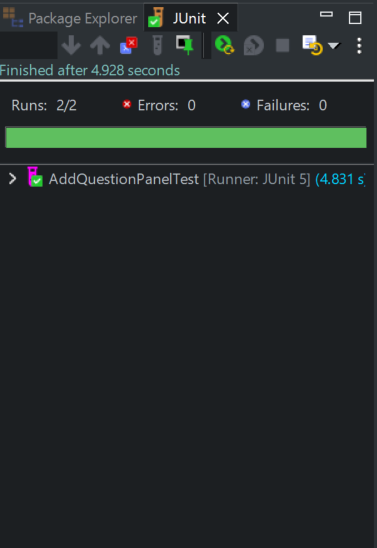
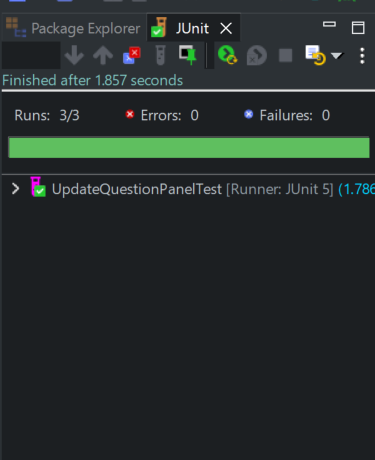
## Class Diagram Overview

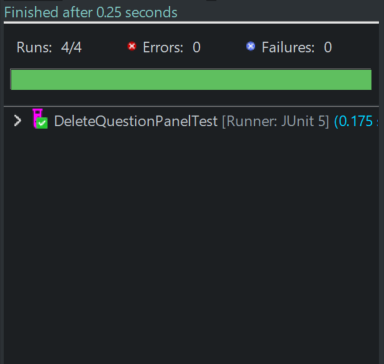
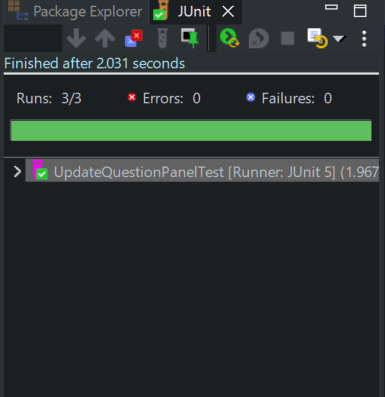
A UML Class Diagram was designed to illustrate the structure of the application by showcasing the relationships between its various classes. This diagram provides a detailed representation of how the classes interact with each other, including their attributes, methods, and dependencies. It serves as a visual blueprint, offering a clear understanding of the system's architecture and class responsibilities. This design approach ensures modularity and adherence to Object-Oriented Programming (OOP) principles, making the application maintainable and scalable.



# 6. Test Scenarios

Although a formal test table was not created, the system underwent extensive testing through predefined scenarios. These test cases covered user registration and login, CRUD operations for quiz questions, score calculations, retrieval of quiz results, and database connectivity. Additionally, error-handling mechanisms were validated to ensure system stability. This approach confirmed that the core functionalities of the application were implemented correctly and performed as expected in the test case.

# 7. Project Status Report

The application successfully meets all defined requirements, ensuring a functional and seamless user experience. The implemented features include user authentication, quiz management, leaderboard tracking, and CRUD operations, all of which operate efficiently. The system is fully functional, with all essential modules integrated and tested, making it ready for deployment and practical use.

# 8. Future Enhancement

It takes times to load next question even after click on next button while playing quiz. Future improvements will address this issue along with several planned enhancements, including the addition of a side panel for user navigation, categorization of quizzes into different subsections, and a “Follow a Friend” feature for social engagement. Additionally, a quiz recommendation system will be introduced to suggest quizzes based on users' past performance, enhancing the overall learning experience.

# 9. Final Thoughts & Conclusion

The Java GUI-based quiz application was successfully developed, fulfilling all specified requirements while adhering to SOLID principles and structured database connectivity. The project showcases strong software engineering principles, modular design, and effective implementation using Java technologies. Future enhancements will focus on improving navigation, expanding quiz categories, and incorporating social features to further engage users. The final outcome demonstrates a scalable and robust application, making it a valuable platform for interactive learning and quiz-based assessments.

# Appendix:

