

QuizWhiz an Android Application

Submitted By

Student Name	Student ID
MD. ABDUL HAMIM	221-15-5264
MD. NOWAZESH KOBIR RIFAT	221-15-5247
MD. RAFIUL ISLAM	221-15-5320
RABEYA NUPUR	221-15-5576
MD. MAHAMUDUL HASAN MASUD	221-15-5206

QuizWhiz AN ANDROID APPLICATION PROJECT REPORT

This Report Presented in Partial Fulfillment of the course
**CSE312: Database Management System Lab in the Computer
Science and Engineering Department.**



DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

December 9, 2024

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Mr. Nokimul Hasan Arif, Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Mr. Nokimul Hasan Arif

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Submitted by

<div>Hamim</div> <div><hr/>MD. ABDUL HAMIM Student ID: 221-15-5264 Dept. of CSE, DIU</div>	
<div>Rifat</div> <div><hr/>MD. NOWAZESH KOBIR RIFAT Student ID: 221-15-5247 Dept. of CSE, DIU</div>	<div>Rafiel</div> <div><hr/>MD. RAFIUL ISLAM Student ID: 221-15-5320 Dept. of CSE, DIU</div>
<div>Nupur</div> <div><hr/>RABEYA NUPUR Student ID: 221-15-5576 Dept. of CSE, DIU</div>	<div>Masud</div> <div><hr/>MD. MAHAMUDUL HASAN MASUD Student ID: 221-15-5206 Dept. of CSE, DIU</div>

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:.

Table 1: Course Outcome Statements

CO's	Statements
CO1	Define and Relate classes, objects, members of the class, and relationships among them needed for solving specific problems
CO2	Formulate knowledge of object-oriented programming and Java in problem solving
CO3	Analyze Unified Modeling Language (UML) models to Present a specific problem
CO4	Develop solutions for real-world complex problems applying OOP concepts while evaluating their effectiveness based on industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1,EP3
CO2	PO2	C2	KP3	EP1,EP3
CO3	PO3	C4, A1	KP3	EP1,EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1,EP3

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

Declaration	i
Course & Program Outcome	ii
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Feasibility Study	2
1.5 Gap Analysis	3
1.6 Project Outcome	3
2 Proposed Methodology/Architecture	5
2.1 Requirement Analysis & Design Specification	5
2.1.1 Overview	6
2.1.2 Proposed Methodology/ System Design	6
2.1.3 UI Design	7
2.2 Overall Project Plan	7
3 Implementation and Results	8
3.1 Implementation	8
3.2 Performance Analysis	9
3.3 Results and Discussion	9
4 Engineering Standards and Mapping	11
4.1 Impact on Society, Environment and Sustainability	11
4.1.1 Impact on Life	11
4.1.2 Impact on Society & Environment	11
4.1.3 Ethical Aspects	11
4.1.4 Sustainability Plan	11
4.2 Project Management and Team Work	11
4.3 Complex Engineering Problem	11
4.3.1 Mapping of Program Outcome	11
4.3.2 Complex Problem Solving	12
4.3.3 Engineering Activities	12

5 Conclusion	13
5.1 Summary	13
5.2 Limitation	13
5.3 Future Work	13
References	14

Chapter 1

Introduction

We have covered the project's goal, motivation, feasibility study and gap analysis in this chapter.

1.1 Introduction

QuizWhiz is a mobile quiz application developed using Java for the Android platform. The primary goal of the application is to help users test their knowledge in various categories while providing a simple and engaging user interface. The app is targeted toward students, professionals, and anyone who wants to improve their skills through interactive quizzes. The application's tagline, "To Improve Yourself, take a Step Forward," reflects the purpose of encouraging users to continuously challenge themselves and track their progress.

1.2 Motivation

The project was motivated by the need to create an engaging, interactive, and user-friendly quiz application that facilitates learning and knowledge enhancement. With the increasing demand for digital education tools, QuizWhiz was designed to provide users with a platform to test their knowledge, track their progress, and improve in various subject areas.

1.3 Objectives

1. **Improve Learning By Means Of Interaction:** To offer a platform for users to test their knowledge on different topics with interactive quizzes, enabling active participation and interaction.
2. **Gamify the Learning Process:** Feature these elements of game design (such as scoring, difficulty levels, and progress tracking) to make users enjoy or become motivated in their learning activities.
3. **Improve Knowledge Retention:** Instant Feedback on Answers and Scores – Help users consolidate learning by providing high-level recall feedback.
4. **Scalability and Customization:** Set the stage for future improvements: more quiz topics, random question banks or even a multiplayer mode.

5. **Offline Accessibility:** Allow users to access quiz content without an active internet connection — making the app more flexible and user friendly.

1.4 Feasibility Study

1. Technical Feasibility:

- The project leverages Java, a robust and widely-used programming language, alongside Android development tools like Android Studio and SQLite, ensuring a reliable development environment.
- Availability of resources like your Lenovo V5 Pro laptop, adequate development tools, and knowledge of Java and Android development supports the successful implementation.
- The app's lightweight design ensures compatibility with devices like your Xiaomi Poco M6 Pro 5G, providing smooth performance even on mid-range devices.

2. Economic Feasibility:

- Development costs are minimal, as open-source tools like Android Studio and free libraries are utilized.
- The project requires no additional hardware investments, as it relies on existing resources.
- Potential for monetization through ads, premium features, or educational partnerships makes the app economically sustainable in the future.

3. Operational Feasibility:

- The app is simple and user-friendly, with a clean interface to ensure users can easily navigate and take quizzes.
- With gamified elements like scoring and progress tracking, QuizWhiz motivates users to engage regularly.
- Designed to meet the growing demand for mobile learning tools, ensuring relevance and usability in the target market.

4. Schedule Feasibility:

- The project was completed within a manageable timeline, showcasing your ability to set and meet realistic goals.
- Future enhancements can be planned in phases to allow continued development and improvement while maintaining user satisfaction

[1].

1.5 Gap Analysis

1. Current State (Existing Features):

- A functional quiz application built with Java, featuring interactive quizzes and scoring systems.
- Basic database integration using SQLite to manage and store questions and scores.
- A user-friendly interface designed for mobile devices.
- Offline accessibility for added convenience.

2. Desired State (Ideal Features):

- **Dynamic Question Bank:** Integration of a feature to fetch questions from an online server for more diverse and regularly updated quizzes.
- **Customizable Quiz Topics:** Allow users to select specific categories or difficulty levels to tailor the experience.
- **Progress Analytics:** Provide detailed performance reports and improvement suggestions based on quiz results.
- **Multiplayer Mode:** Enable users to compete with friends or global players for a more engaging experience.
- **Monetization Options:** Add features like premium content or ad-based revenue streams for future scalability.
- **Enhanced UI/UX Design:** Incorporate modern design principles and animations to enhance user engagement.

1.6 Project Outcome

1. Role-Based Functionality:

- Teachers can manage quizzes by adding questions categorized by subject and difficulty.
- Students can participate in quizzes and access a secure and intuitive environment tailored to their needs.

2. Secure User Management:

- Secure sign-up and login mechanisms prevent duplicate accounts and unauthorized access.
- Persistent session management ensures users remain logged in until they explicitly log out.

3. Quiz Integrity:

- Screenshot prevention during quizzes enhances security and prevents question leakage.

- A countdown timer ensures time-bound responses, fostering discipline and focus during quizzes.

4. User-Friendly Interface:

- The application features a simple and intuitive design, enabling teachers and students to navigate easily.
- Role-based dashboards provide clear and organized access to functionalities.

5. Practical Implementation:

- The application provides a functional solution for educational environments, ensuring secure and organized quiz management.

These outcomes collectively demonstrate the app's ability to address real-world educational challenges effectively.

Chapter 2

Proposed Methodology/Architecture

Here we have discussed the requirements, functionality, architecture, methodology and overall plan of the project.

2.1 Requirement Analysis & Design Specification

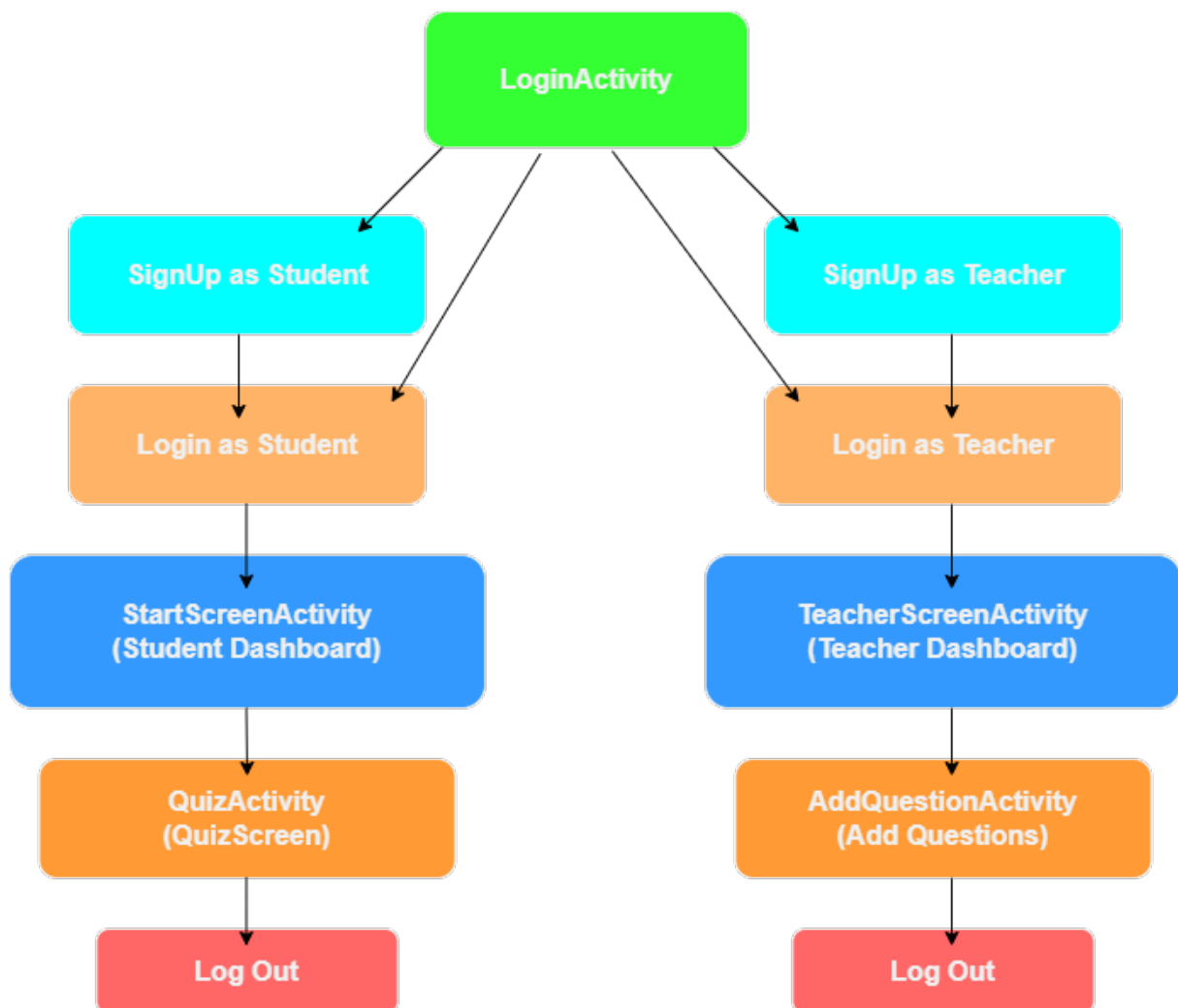


Figure 2.1: Block Diagram

2.1.1 Overview

1. Requirement Analysis: Functional Requirements:

- Users can take quizzes and view scores.
- Questions and answers are stored and retrieved from an SQLite database.
- Real-time feedback on quiz answers.

Non-Functional Requirements:

- User-friendly interface.
- Offline functionality for quizzes.
- Lightweight and compatible with mid-range Android devices.

2. Design Specification: User Interface:

- Clean, intuitive layout with easy navigation.
- Screens for quiz questions, results, and settings.

System Architecture:

- **Frontend:** Java-based Android application with XML for UI design.
- **Backend:** SQLite database for storing and managing quiz data.

Data Flow:

- **Input:** User selects quiz and answers questions.
- **Processing:** Validate answers and calculate scores.
- **Output:** Display results and store scores locally.

2.1.2 Proposed Methodology/ System Design

Agile Methodology

- **Iterative Development:** The project was developed step-by-step, focusing on completing individual features like quizzes, scoring, and database integration.
- **Flexibility:** Allowed for changes and improvements during development, such as tweaking the UI or adding new features.
- **User-Centric Approach:** Focused on creating a user-friendly and engaging experience, aligning with Agile's emphasis on customer satisfaction.
- **Continuous Testing:** Tested features like question display, scoring, and database integration throughout the development process.



Figure 2.2: Agile Methodology

2.1.3 UI Design

We have used Figma Software for the design of our project. Here we designed our plan and then started the development work. Here is the link of our UI design:

<https://www.figma.com/design/2l3qo0XVUGS0VhneO85KN1/QuizWhiz?node-id=0-1>

2.2 Overall Project Plan

First, we planned our project as a team work. Then based on the plan, we divided our project into 15 steps. Then we designed them all separately and came up with a final design. And above all, we started the development work. Finally, through our test we got a result of the project and we kept a plan for what else can be done about its deployment in the future.

Chapter 3

Implementation and Results

Here We discussed the overall implementation, performance and results of this project.

3.1 Implementation

- **Requirement Analysis:** Understanding user needs and defining app features.
- **Design:** Creating wire frames, user interface design, and defining the structure of the app.
- **Development:** Writing code in Java, integrating the SQLite database, and implementing features like login, quiz categories, and score tracking.
- **Testing:** Thorough testing of functionality, usability, and performance.
- **Deployment:** Launching the app on the Google Play Store (In Future).
- **Maintenance:** Regular updates and improvements based on user feedback.
- **Database Used-SQLite:** SQLite is used in this project to manage the storage of quiz data, including questions, categories, and scores. It provides a lightweight, server less database solution embedded in the application.
- **How the Database Works**

1. Tables:

Categories Table:

- Stores quiz categories (e.g., Programming, Math).
- Each category has a unique ID and name.

Questions Table:

- Stores quiz questions.
- Includes question text, options, correct answer, difficulty level, and category ID as a foreign key.

Shared Preferences (Separate from SQLite): Used to store user-specific data like:

- Name and email for auto-login.
- Latest score and high score.

2. Key Database Concepts Used:

- **Primary Key:** Ensures each record (category or question) is uniquely identifiable
- **Foreign Key:** Links questions to categories.
- **CRUD Operations:**

1. **Create:** Adding new categories and questions.
2. **Read:** Fetching questions for a specific category and difficulty.
3. **Update:** Updating high scores when a user achieves a new high score.
4. **Delete:** (Optional) Removing outdated questions or categories if needed.

3.2 Performance Analysis

We ran our app on different devices through emulators and tested its performance. However, in this case we faced two challenges but we also we solved it.

- **Maintaining user engagement:**

Introducing high score tracking and the ability for users to add their own questions ensures continued engagement.

- **Managing the database efficiently:**

Using SQLite to manage local data storage, ensuring quick access and efficient data handling.

3.3 Results and Discussion

Result: The developed Quiz App provides a functional and secure platform for both teachers and students, achieving the following key outcomes:

1. User Management:

- Secure sign-up and login functionality for both teachers and students, with persistent user data storage.
- Role-based navigation ensures teachers and students access their respective dashboards seamlessly.
- Duplicate sign-ups are prevented, and only validated users can log in.

2. Quiz Functionality:

- Students can participate in quizzes with multiple-choice questions categorized by difficulty and subject.
- A countdown timer prevents indefinite time usage per question.
- Screenshots during quizzes are disabled for enhanced security

3. Teacher Dashboard:

- Teachers can add questions categorized by subject and difficulty, ensuring a dynamic and expandable question bank.

4. Persistence:

- User sessions remain active unless explicitly logged out, improving user experience and reducing repetitive actions.

Discussion:

- **Usability:** The app is intuitive and user-friendly, allowing users to navigate effortlessly between sign-up, login, and dashboard functionalities.
- **Security:** The use of FLAG_SECURE prevents unauthorized sharing of quiz content through screenshots or screen recordings.
- **Role-based Access Control:** By differentiating features for teachers and students, the app ensures that functionality aligns with user roles.
- **Data Persistence:** Leveraging Shared Preferences for user data storage enhances reliability but may not scale for a larger user base or require advanced data querying.

Chapter 4

Engineering Standards and Mapping

Here, we have discussed the mapping and standards of our project.

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

The app will always serve as the right guide for a student and teacher.

4.1.2 Impact on Society & Environment

Our app will not have any negative impact on society and its sustainability will depend on our work.

4.1.3 Ethical Aspects

The app will work in a positive way from an ethical perspective.

4.1.4 Sustainability Plan

It's sustainability depend on having enough users and it's budget in the future.

4.2 Project Management and Team Work

As we mentioned at the beginning, our project is made up of active team members and everyone on the team contributes equally. Since this is a lab-based project, there is no set budget for this.

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

In this section, provide a mapping of the problem and provided solution with targeted Program Outcomes (PO's).

Table 4.1: Justification of Program Outcomes

PO's	Justification
PO1	Justification of PO1 attainment
PO2	Justification of PO2 attainment
PO3	Justification of PO3 attainment

4.3.2 Complex Problem Solving

In this section, provide a mapping with problem solving categories. For each mapping add subsections to put rationale (Use Table 4.2). For P1, you need to put another mapping with Knowledge profile and rational thereof.

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range of Conflicting Require- ments	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent of Stake- holder Involve- ment	EP7 Inter- dependence
✓	✓	✓	✓	X	✓	✓

4.3.3 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of re- sources	EA2 Level of Interac- tion	EA3 Innovation	EA4 Consequences for society and envi- ronment	EA5 Familiarity
✓	✓	✓	X	✓

Chapter 5

Conclusion

Here, we have discussed the over all summary, limitations and future work of the project.

5.1 Summary

adjustbox This project is a role-based Quiz Management Application designed for teachers and students, offering secure and user-friendly functionalities. Teachers can add categorized questions, while students can take quizzes with features like a countdown timer and screenshot prevention to ensure integrity. The app implements secure sign-up and login with persistent user sessions, preventing duplicate accounts and maintaining role-based navigation. With its intuitive interface, session management, and emphasis on security, the app addresses practical educational needs while being scalable for future enhancements such as real-time quizzes and performance analytics.

5.2 Limitation

There are many some limitations here, such as:

- There is no leaderboard, which means there will be no competition.
- There is no image upload system.
- Not online-based.

However, these limitations will be addressed through further development in the future.

5.3 Future Work

The current design and functionality set the foundation for future enhancements, including cloud-based data storage for questions, additional quiz categories, and the introduction of more gamified elements like leaderboards and achievement badges.

1. **Database Integration:** Replace SharedPreferences with a scalable database (e.g., SQLite, Firebase) to handle larger datasets, enhance querying capabilities, and enable synchronization across devices.

2. **Performance Analytics:** Provide students with detailed performance analytics, such as scores over time, question-specific feedback, and comparison with peers.
3. **Real-Time Features:** Introduce real-time online quizzes where students can compete with peers and Enable teachers to monitor quiz progress live.
4. **Additional User Roles:** Add new roles like administrators for managing users, quizzes, and analytics.
5. **Multimedia Questions:** Support for multimedia questions, such as images or audio clips, to make quizzes more interactive and engaging.

References

- [1] Florian Walther. *Coding In Flow*. Germany, 2017.
 - [2] Luca Ardito, Riccardo Coppola, Giovanni Malnati, and Marco Torchiano. Effectiveness of kotlin vs. java in android app development tasks. *Information and Software Technology*, 127:106374, 2020.
 - [3] Suhas Holla and Mahima M Katti. Android based mobile application development and its security. *International Journal of Computer Trends and Technology*, 3(3):486–490, 2012.
 - [4] Yanfi Yanfi and Pualam Dipa Nusantara. Ui/ux design prototype for mobile community-based course. *Procedia Comput. Sci*, 216(2022):431–441, 2023.
 - [5] Syeda Mariam Muzammal and Munam Ali Shah. Screenstealer: Addressing screenshot attacks on android devices. In *2016 22nd International Conference on Automation and Computing (ICAC)*, pages 336–341. IEEE, 2016.
- [2, 3, 4, 5]