	University of the East **College of Engineering**		**Prepared for:** Engr. Marjon Umbay *Faculty, CpE Department*
	Subject code: NCP_2103	**Section:** 2CPE-1B	**Date:** October 13, 2025
Prepared by: Lopez, Gwen Angelynn M. Basconcillo, Karl Angelo P. Dela Cruz, Arabela Grace Y. Ballera, Ashlie Magdalen Nunez, Matthew Jeric O. Alaba, Mark Einjelo M.	**Midterm Project Proposal**		

SmartQuiz: Quiz Application

Introduction

SmartQuiz is a system quiz application designed to make creating, managing, and taking quizzes easier and more efficient for both teachers and students. It is made to help teachers create, manage, and check quizzes more easily, while giving students a faster and more enjoyable way to take them. It serves as a digital tool that makes classroom quizzes simpler, more organized, and more interactive for both teachers and students.

The main purpose of SmartQuiz is to support teachers in conducting quizzes without the need for manual checking or paper-based exams. It saves time in quiz preparation, grading, and monitoring of student performance. For students, the system provides an organized and interactive way to take quizzes, view results instantly, and track their progress.

Our target users are high school to college students and teachers. Because of this, we designed our user interface (UI) to suit their age group—modern, clean, and easy to use. We also researched suitable color palettes for the UI to make sure the design is visually pleasing and comfortable for users in this age range. This ensures that SmartQuiz is not only functional but also appealing to its intended audience.

From a programming point of view, SmartQuiz follows Object-Oriented Programming (OOP) principles such as encapsulation, inheritance, polymorphism, and abstraction. These help the system stay organized, secure, and easy to update in the future. It shows how modern programming can be used to build educational tools that are both useful and efficient.

Overall, SmartQuiz aims to combine technology and education to make learning more accessible, interactive, and effective. It encourages students to be more motivated, teachers to work more efficiently, and schools to use technology in a way that improves the whole learning experience.

System Features

1. Sign-Up and Login
   - Sign-Up Page: Allows new users, either Students or Teachers, to create an account and register in the system.
   - Login Page: Enables existing users to log in using their credentials and be redirected to their respective dashboards.

2. Teacher Features
   - Quiz Management: Teachers can create, view, edit, and delete quizzes easily.
   - Quiz Code: Can generate a quiz code to ensure that only authorized students can access a quiz.
   - Student List Management: Allows teachers to view their students, track scores, and manage quiz access.
   - Leaderboard / Ranking System: Displays the top-performing students in each class to encourage friendly competition.
   - Notifications: Enables teachers to send announcements or reminders directly to students.
   - Question Timer Setting: Allows teachers to set a time limit for each question or for the entire quiz.

3. Student Features
   - Take Quizzes: Students can answer multiple-choice, true or false, or identification questions.
   - Join via Quiz Code: Students can join a quiz by entering the unique code given by their teacher.
   - Question Timer: Each question is timed to test both knowledge and quick thinking.
   - Notifications: Students receive alerts about new quizzes or announcements from their teachers

4. Unique Feature: Quiz Battle

The Quiz Battle is a special feature in SmartQuiz that allows students to compete with others without needing a quiz code.

   - Students can join the battle anytime while it is open.
   - Once they finish, the system automatically checks their answers and time. The Leaderboard displays who achieved the highest score and who answered the fastest, making learning more exciting and competitive.

Application of Object-Oriented Programming Concepts

Encapsulation (hiding data, using getters/setters).

- Encapsulation is the practice of maintaining the privacy of data and granting restricted access only by way of public methods such as getters and setters.
- Every user, whether a teacher or a student, has personal information like their email address, password, and username. Only secure methods getUsername(), setPassword() can access or modify these.
- Private fields such as quizTitle, quizCode, questionList, and timer are included in quiz objects. To avoid unwanted modifications, they can only be altered or retrieved using the class methods.
- Sensitive data is likewise encapsulated in the Score and Leaderboard classes, which only certain operations (like updateScore()) can alter.

Inheritance (parent/child classes).

- allows one class to derive properties and behaviors from another class.
- Code reuse is encouraged by way of inheritance, which enables one class to inherit properties and functions from another class.
- The abstract class is called user
- Subclasses Teacher and student inherit its methods login(), logout() and properties.
- Additionally abstract, the question acts as a parent for:

- True or False Question
- Question with Multiple Choices
- Question of Identification

Polymorphism (same method, different behavior).

- Depending on whatever object calls the method, the same method name might act differently
- Polymorphism lets the same method name do various things depending on which object calls it.
- The behavior of checkAnswer() varies depending on the Question subclass:
- A boolean is compared via a TrueOrFalse question.
- The selected letter (A–D) is checked by the MultipleChoiceQuestion.
- The IdentificationQuestion examines strings without regard to case.
- Depending on whether the recipient is a student or a teacher, displayNotification() may display different information.
- In Teacher and Student, viewResults() operates differently.

Abstraction (general idea of questions).

- The goal of abstraction is to conceal underlying implementation details and only display important features.
- General roles or entities are represented by the abstract User and Question classes.
- Without disclosing data structures, QuizManager manages quiz operations createQuiz(), editQuiz(), and saveQuizzes().
- Results and messages are processed using internal logic that is abstracted by QuizResult and Notification.

The SmartQuiz system was built with high cohesion and low coupling to ensure a clear, organized, and efficient program structure.

Cohesion means each class has a single, focused responsibility. Every class was designed to perform one specific role such as managing users, handling quizzes, checking questions, or displaying results without mixing other functions. This clear division of tasks improves readability, reduces errors, and makes future updates easier since each class can be modified independently.

Low coupling ensures that classes are connected only through necessary and defined relationships. They communicate through specific methods without depending on each other's internal logic. For example, a teacher can create quizzes but not control how answers are checked, and students can take quizzes but not alter scoring or quiz settings. Because of this, changes in one part of the system like adding new quiz types or features won't affect the others.

This combination of strong cohesion and low coupling makes SmartQuiz flexible and easy to maintain, allowing the system to grow or improve without causing errors or conflicts in other components.

UML Class Diagram

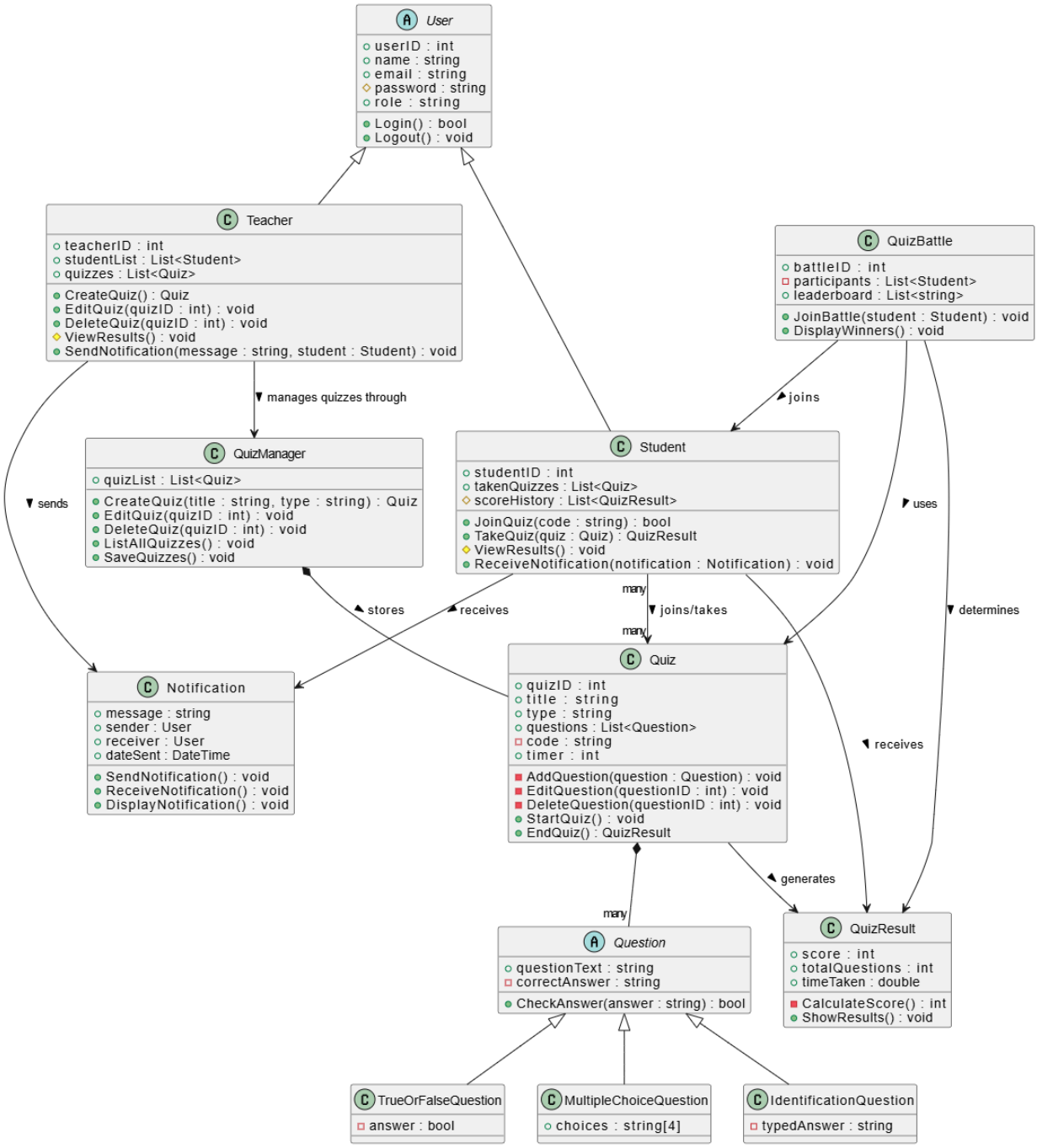

Figure 1. *SmartQuiz UML Diagram created by draw.io*

Explanation of the Diagram

Inheritance: Teacher and Student inherit from User

Composition: Quiz contains Questions, QuizManager controls Quizzes

Association: Users interact through Quizzes, Results, and Notifications

Polymorphism: Different Question types share checkAnswer( ) method

Encapsulation: Attributes are private, accessed via methods.

UI/UX Design Overview

Figma Link:

- https://www.figma.com/design/CKzZCiXJfmFZCZqrhqGMhk/SmartQuiz?node-id=237-2&p=f&t=8kZCJLdGNLikw4eb-0

Conclusion

The SmartQuiz system was created to make quizzes easier and faster for both teachers and students. It helps teachers manage quizzes without using paper and automatically checks answers, saving time and effort. For students, it provides a fun and organized way to take quizzes, view results instantly, and track their performance. The system also includes notifications and a quiz battle feature to make learning more interactive and exciting.

In terms of programming, SmartQuiz applies Object-Oriented Programming (OOP) principles such as encapsulation, inheritance, polymorphism, and abstraction. These principles keep the system organized, secure, and flexible. Each class in the UML diagram has a clear role — teachers handle quiz creation, students take quizzes, and the QuizManager controls quiz operations — showing how OOP helps divide tasks properly in the program.

Overall, SmartQuiz shows how technology can improve classroom learning. It promotes efficiency, motivation, and engagement among students while helping teachers save time and manage their classes better. With its well-designed structure and user-friendly interface, SmartQuiz can serve as a practical and modern tool that supports both teaching and learning.