3. Solution

3.1) Solution - 1

Our First Approach

Our initial solution was to implement the Interactive Quiz System using a basic structure focusing on essential functionalities. The primary features included a login system, adding quizzes, deleting quizzes, and viewing all quizzes. We aimed to build a foundation before adding more complex features.

Reason for Not Selecting This Solution

This approach had significant drawbacks. The application only supported adding quizzes without the ability to add questions and answers, making the system incomplete. Managing multiple users and quizzes through a CLI was unmanageable. Given these limitations, we decided that this solution would not be suitable for our project goals and needed further enhancement.

However, this solution did not meet all the **constraints** and requirements of the project:

1. **Application Type:** The application satisfied the desktop application constraint by providing a text-based interface. However, it did not offer an interactive user experience, especially in terms of question and quiz management.
2. **Question Types:** The application did not support multiple question types, such as multiple choice and true/false. This limitation made the system not helpful for handling multiple quizzes.
3. **Real-Time Scoring:** The application did not provide real-time scoring upon quiz completion. Users can not receive immediate feedback on their performance.
4. **Compatibility:** The application was compatible with Java 11 or later.

Summary of Available Features

- **Login System:** Basic user authentication to secure access.
- **Add Quiz:** Users could add quizzes to the system.
- **Delete Quiz:** The system allowed users to delete existing quizzes.
- **View All Quizzes:** Users could list all available quizzes.

3.2) Solution - 2

Our Second Approach

The second solution added more of the basic functionalities in the first solution by introducing additional features such as the ability to create, edit, and delete quizzes, add users, and manage quizzes with a structured approach. This version included the use of several classes to handle different parts of the quiz system which made the code more readable and easier to maintain.

Reasons for Not Selecting This Solution

While this solution addressed many of the limitations of the first solution, it still fell a little short in some of the areas. This Solution While this solution was a good improvement over Solution 1, it still had some limitations. Specifically, it lacked a direct method to update an existing quiz. Users had to delete the old quiz and create a new one to update a quiz, which was not very user-friendly.

This solution met all the constraints and requirements of this project:

1. **Application Type:** The system was a desktop application, focusing on providing an interactive user experience.
2. **Question Types:** The system supported multiple question types, including multiple choice and true/false.
3. **Real-Time Scoring:** The system provided real-time scoring upon quiz completion.
4. **Compatibility:** The system was compatible with Java 11 or later.

Summary of Available Features (requirements)

- **Login System**: Improved user authentication with support for admin and regular users.
- **Add Quiz**: Admins could add quizzes with multiple questions, improving the basic quiz addition of the first solution.
- **Edit Quiz**: Admins could edit existing quizzes - quiz titles and questions.
- **Delete Quiz**: Admins could also delete quizzes.
- **View All Quizzes**: Both admins and regular users could view all the available quizzes, with detailed information on questions and options.
- **Take Quiz**: Users could take quizzes and receive immediate scores, improving feedback system.
- **Add User**: Admins could add new users which supports better user management.