The Java Spring GUI Mathematical and Logical Aptitude Quiz

Object Oriented Analysis and Design using Java (UE20CS352)

Mini Project

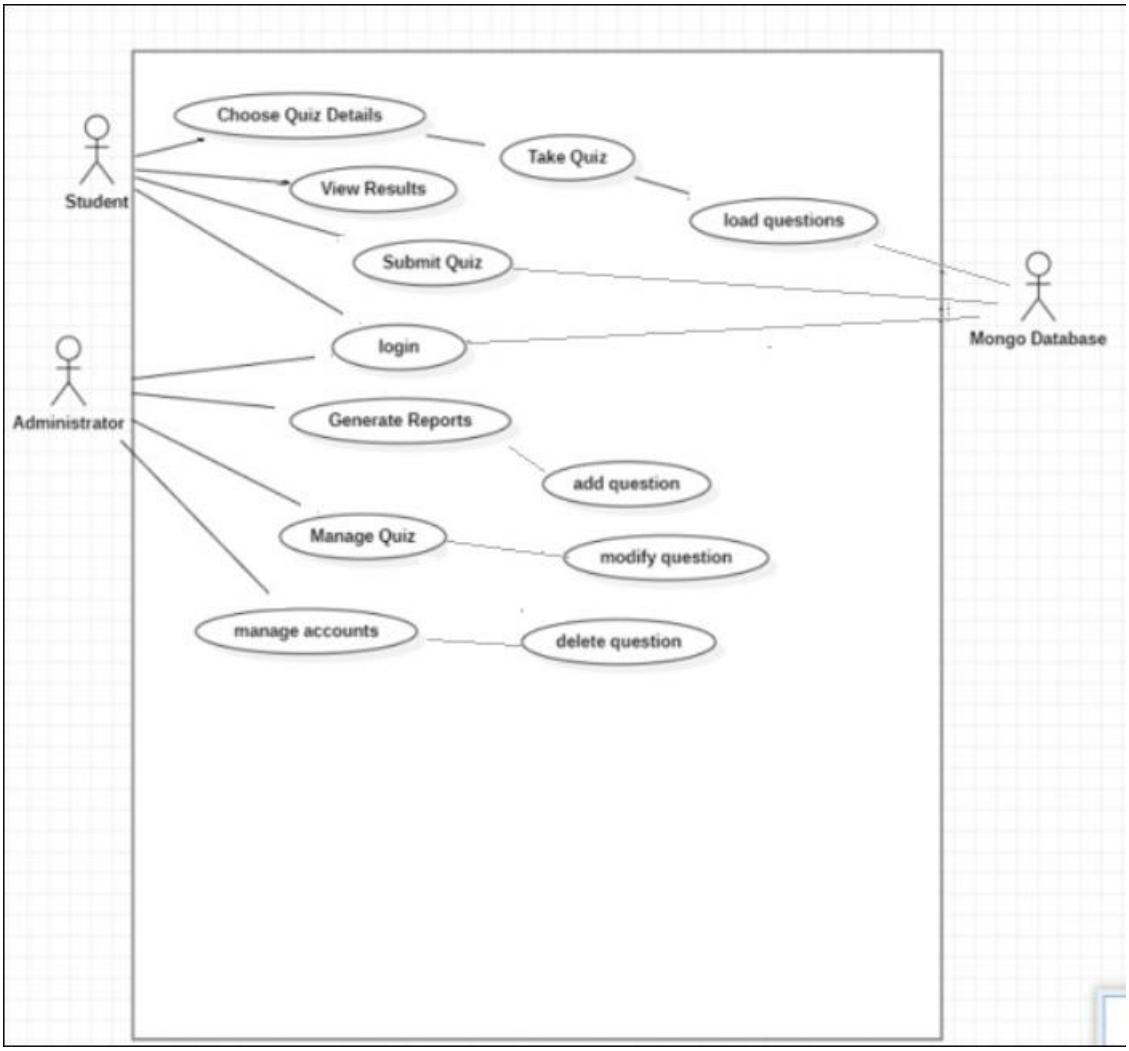
Team members:

Hemachandran S	***PES2UG20CS139***
Lohit Siriki	***PES2UG20CS180***
M Manonmana Udupa	***PES2UG20CS181***
Md. Taseen Atehar	***PES2UG20CS191***

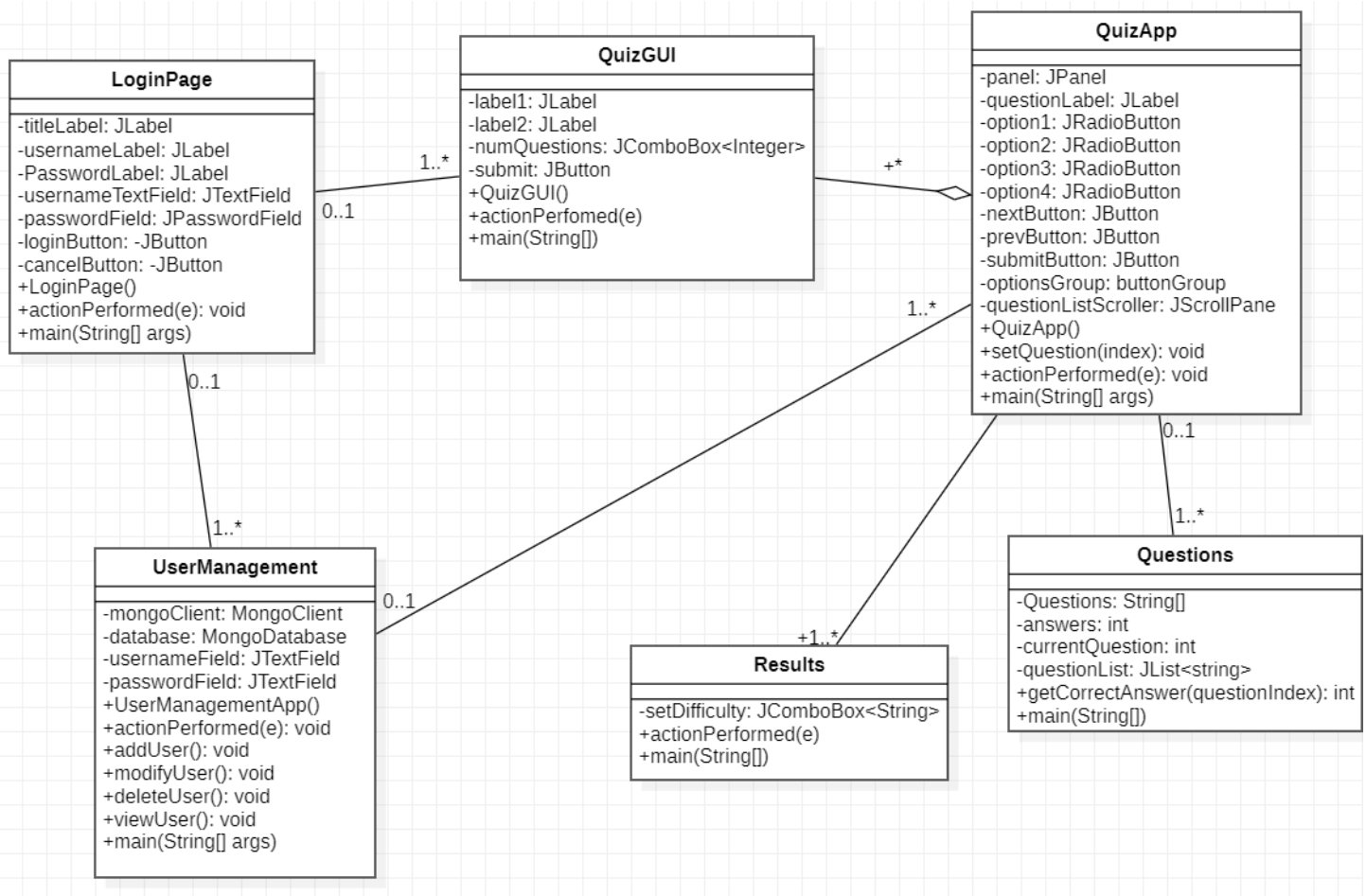
Problem Statement:

The Java Spring GUI Mathematical and Logical Aptitude Quiz is a mini project designed to help candidates prepare for job interviews by testing their mathematical and logical reasoning abilities. The quiz is built using the Java programming language and the Spring framework for GUI development, and is part of an object-oriented design and programming course. The purpose of this project is to provide a valuable resource for candidates looking to improve their mathematical and logical reasoning skills and prepare for job interviews in fields that require these skills.

Use Case Diagram:



Class Diagram:



Architecture Patterns, Design Principles, and Design Patterns Used:

- **Model-View-Controller (MVC) Architecture:** The MVC pattern is used to separate the application logic from the user interface. The model represents the data and business logic, the view represents the user interface, and the controller acts as the intermediary between the model and the view. This pattern is used to improve maintainability, reusability, and testability of the application.

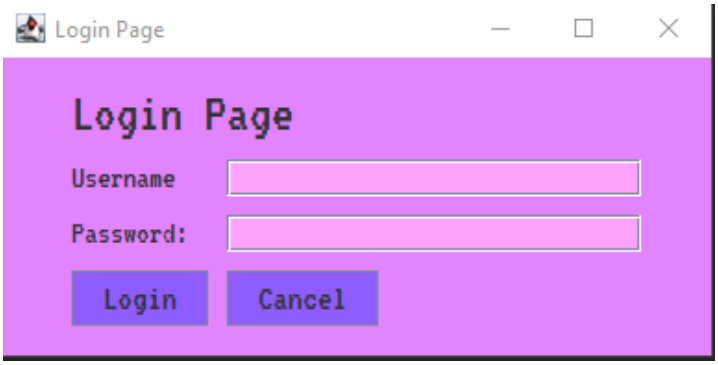
- **Dependency Injection (DI) Principle:** The DI principle is used to reduce the coupling between the application components. In our project, Spring framework's dependency injection feature is used to inject the required objects into the components instead of creating them manually. This improves modularity, scalability, and testability of the application.

- **Singleton Design Pattern:** The Singleton pattern is used to ensure that there is only one instance of a class created at runtime. In our project, the MongoDB database connection object is implemented using the Singleton pattern to ensure that only one database connection is created and shared across the application.

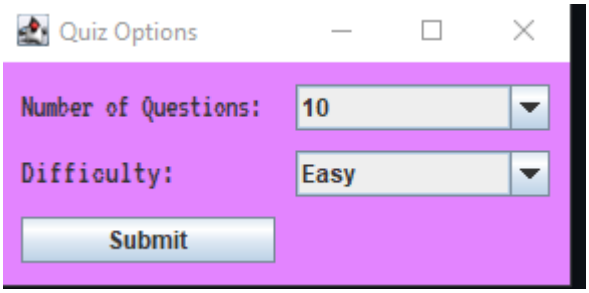
Overall, these patterns and principles are used to improve the scalability, maintainability, and testability of the application. The MVC architecture ensures separation of concerns, DI principle reduces coupling between components and Singleton pattern ensures a single database connection.

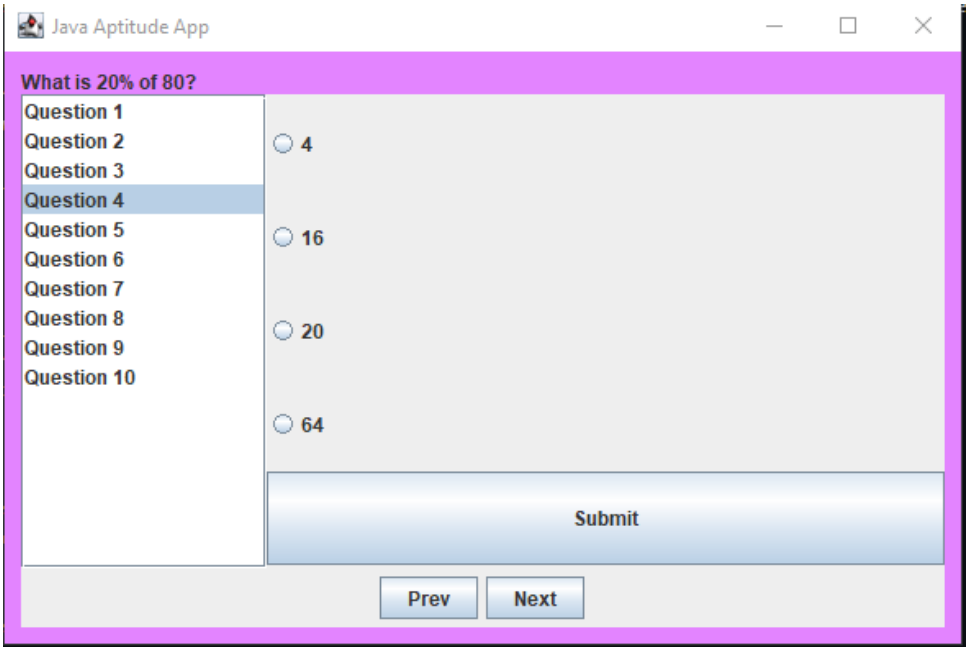
Output Screenshots:

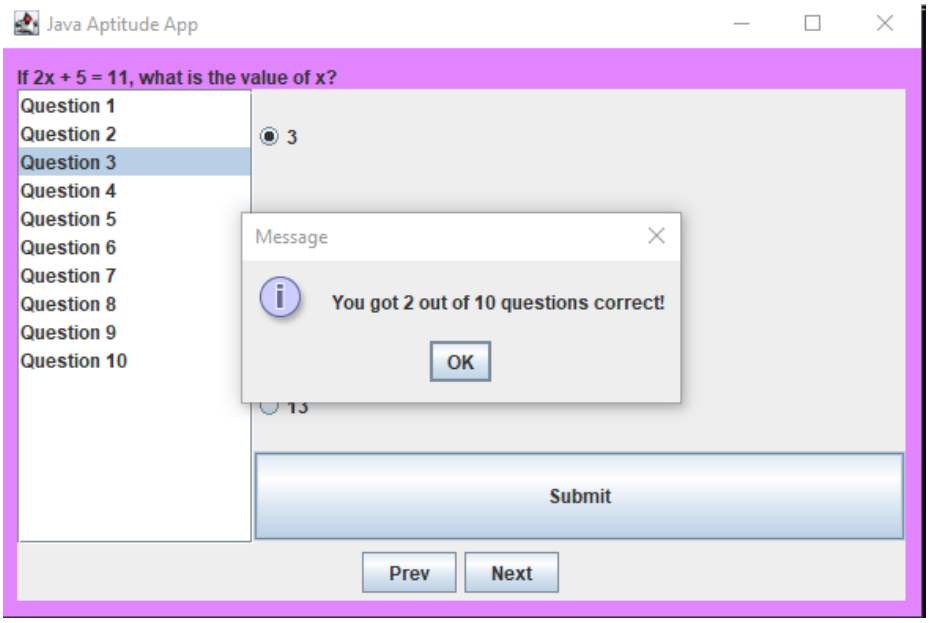
Login Window:



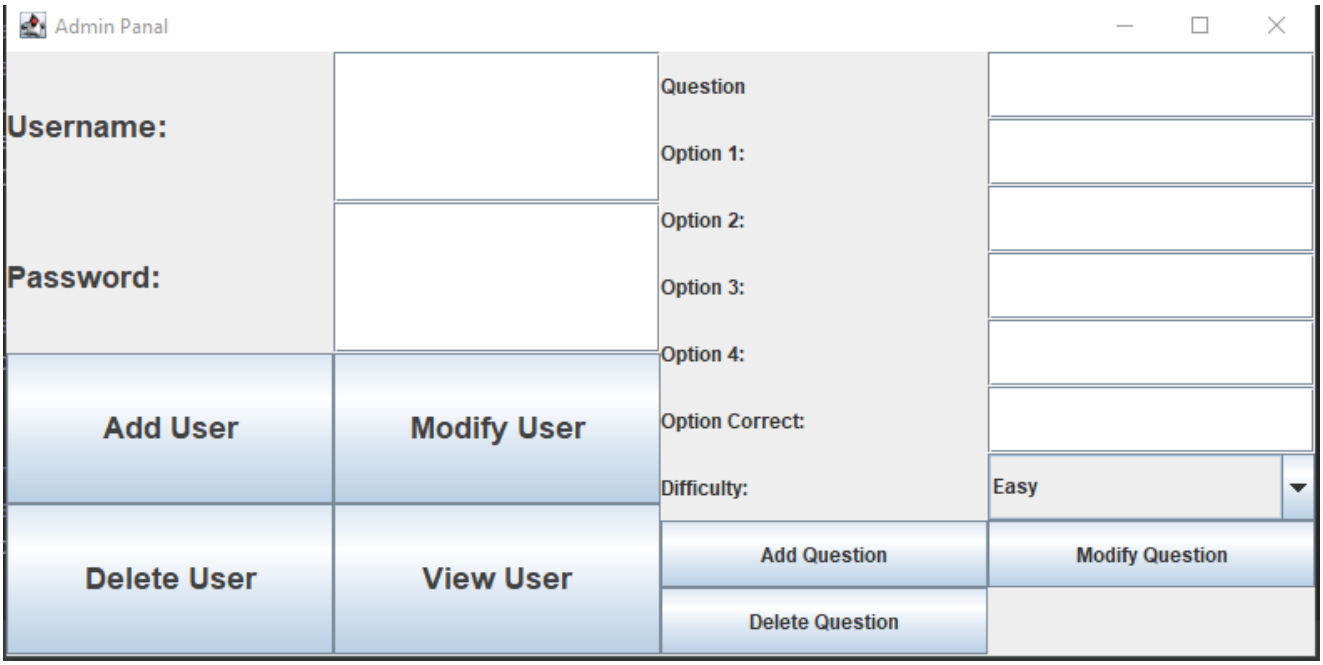
User Interface:







Admin interface:



GitHub repo link:

https://github.com/Seen3/Java_Aptitude

Individual Contributions:

Md. Taseen Atehar: Quiz GUI, Quiz Application,
M Manonmana Udupa: Questions, Result
Lohit Siriki: User Management
Hemachandran S: Login Page