**Java Programming II**

**Phase ..-Assignment …**

# **Million Winner**

## **Supervised by**

## Dr. Sahar Ismail

## **Prepared by:**

|  |
| --- |
| Abdalrhman Ahmed |
| Shahd Aboelkramat |
| Sara Magdy |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student Name | Task | Percentage of achievement |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

|  |  |
| --- | --- |
| Table of Contents | Page NO. |
| Abstract |  |
| Chapter 1: Introduction | 5 |
| 1.1 Problem definition. | 6 |
| 1.2 Solution approach. | 7 |
| 1.3 Used technology. | 8 |
| Chapter 2: System Design | 9 |
| 2.1 Use case diagram. |  |
| 2.2 Class diagram. | 10 |
| 2.3 Flowchart |  |
| **Chapter 3: Implementation** |  |
| 3.1 GUI Implementation. |  |
| 3.2 classes implementation |  |
| 3.3 Data store Implementation |  |
| Chapter 4: Conclusion |  |
| 4.1 Conclusion. |  |
| 4.2 Result’s screenshot. |  |

.

**Abstract**

…..

**Chapter 1**

**Introduction**

**Chapter 1**

**Introduction**

1. **1 Problem Definition**

Develop a quiz application that allows users to choose specific category for the questions they want to answer the application should present a set of questions based on the selected category and calculate the users score based on their correct answer if the user answer a question incorrectly the application should stop presenting further questions.

**Requirements:**

* **User interface:** the quiz application should have a user-friendly interface to display the questions and user answer and allow user to select the category from predefined list.
* **Question bank:** the quiz application has a question bank that stores the questions, answer option and the correct and incorrect answer for each category questions**.**
* **Category selection:** the application display categories as options for user to select from before starting the quiz.
* **Question presentation:** once user select a category the application should present questions one by one from the category each question includes question statement and multiple answer options.
* **Scoring system:** the application keep track the users score as they progress through quiz for every correct answer the score increases by predefined value if the user answer question incorrectly the quiz should end, and the final score will be displayed**.**
* **Termination:** if user answer incorrectly the final score displayed, user not allowed to continue answering questions after incorrect response.
* **Restart option:** after terminating quiz user have option to restart quiz or choose another category**.**

**Functional flow:**

* Display the username and password inputs for signing-up and login button if the user has an account.
* Display the username and password inputs for logging-in and sign-up button if the user does not have an account.
* Display Start button, Scoreboard button, and logout button.
* If the user selects the Scoreboard button it displays the top 10 users based on the Score.
* If the user selects logout button it will display the login page.
* If the user selects Start button it will display the available categories to the user.
* Give the user ability to select a category.
* Fetch questions from question bank based on selected category.
* Present the first question to the user.
* Accept users answer as input.
* Check users answer.
* If answer is correct update the score and proceed to the next question.
* If answer incorrectly terminate the quiz and display the final score.
* Provide an option to restart or choose another category.

1. **2 Solution Approach**

**The quiz application follows a simple modular architecture with the following key components:**

**1. User Class:**

- Manages user information such as username, password, and score.

**2. Question Class:**

- Represents a quiz question with the question itself, answer options, and the correct option index.

**3. MillionWinner Class:**

* + Orchestrates the flow of the application.
  + Manages user authentication, quiz categories, questions, and game logic.

**4. Scoreboard Class:**

- Manages the display and retrieval of user scores.

**User Authentication:**

* + Users are authenticated during the sign-up and login processes.
  + Passwords are securely stored.

**Quiz Gameplay:**

* + Users select a quiz category, answer multiple-choice questions, and accumulate a score.
  + Incorrect answers result in a deduction of points.

**Scoreboard:**

* + The top users and their scores are displayed on the scoreboard.
  + Scores are updated based on user performance in quizzes.

1. **3 Used Technology**

* **Programming Language:** JAVA
* **Development Environment:** NetBeans IDE
* **Version Control:** GitHub for version control
* **Dependencies:** No external dependencies

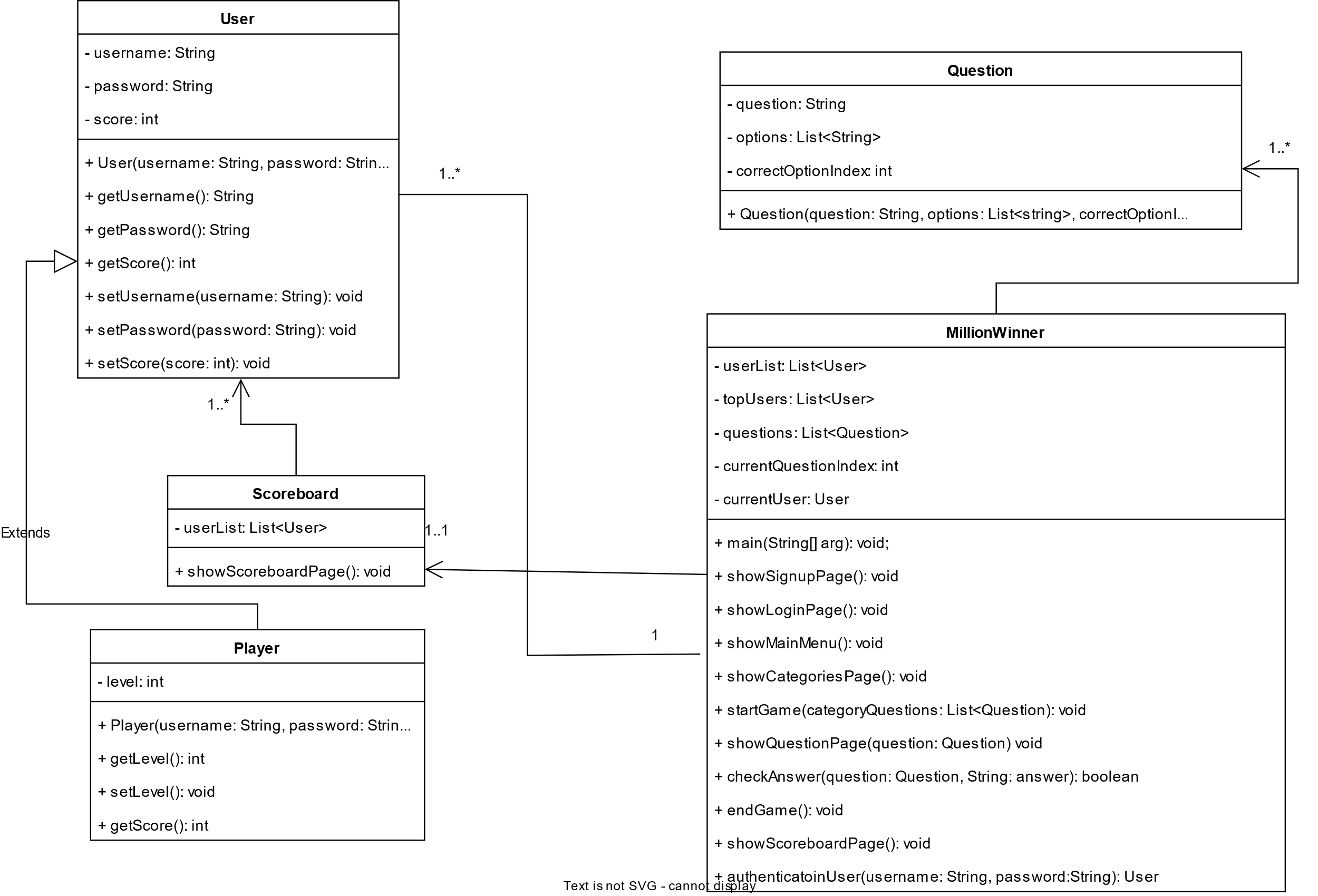
**Chapter 2**

**System Design**

**Chapter 2**

**System Design**

1. **1 Use Case Diagram**
2. **2 Class Diagram**

****

1. **3 Flowchart**

**Chapter 3**

**Implementation**

**Chapter 3**

**Implementation**

1. **1 GUI Implementation**
2. **2 Classes Implementation**
3. **2 Data Store Implementation**

**Chapter 4**

**Conclusion**

**Chapter 4**

**Conclusion**

1. **1 Conclusion**
2. **2 Result’s Screenshot**