Posts and Telecommunications Institute of Technology

Assignment

Network Programing

Subject: Online Quiz Game

Group: 8

Members

Nguyễn Minh Đức

Trương Xuân Dũng

Nguyễn Thanh Lâm

Content

A) Task Table

Member name	Task
Trương Xuân Dũng	Implement the game’s login function, show online players, multi-client thread handling and design the game’s database
Nguyễn Thanh Lâm	Designing, document writing, Implement the User management system web pages using Jsp Servlet
Nguyễn Minh Đức	Implement the match UI and logic, design the question table in the database, multi-client thread handling

B) Game Description

Online quiz game:

· The system has one server and many clients. Server stores all information and data.

· In order to play, the player must login to his account from a client machine. After successful login, the interface displays a list of players who are online, each player has the following information: name, total available points of the player, status (or busy if playing with others). , or idle if not playing with anyone).

· If the players want to invite (challenge) someone, they need to click on the opponent's name in the online list.

· When challenged, the player can accept (OK), or decline (Reject).

· When accepting, 2 players will play against each other, and the server will act as the referee.

The game interface includes a list of N questions in the form of multiple choice, a time box and an exit button.

· The server will automatically generate the same question form and send it back to both opponents. Each player must click on the answers to the questions. When finished, click the submit button.

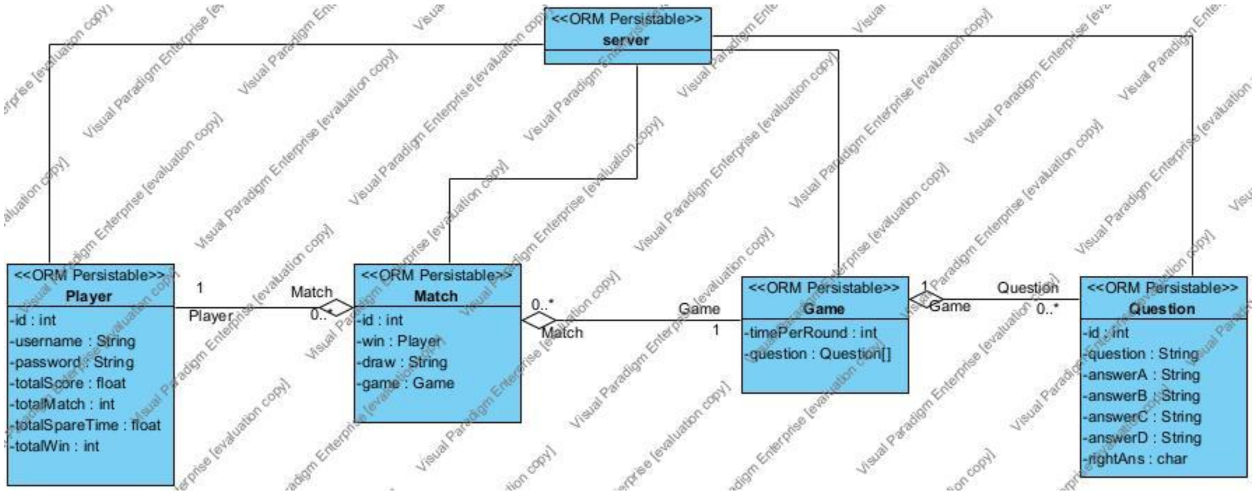
· After each match, the server will check who won and send the results to both opponents: win 1 point, draw 0.5 point, lose 0 point.

· After each game, there is a dialog asking each player if they want to continue. If both continue, continue playing, if one of the opponents stops playing, exit and the server informs the other player.

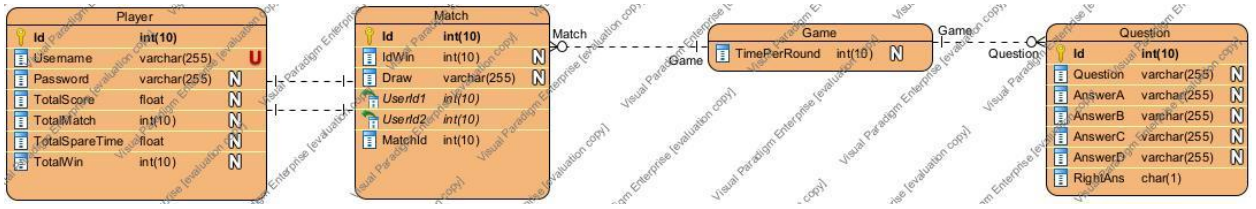
· Match results are saved to the server. Each player can view the rankings of players in the entire system, according to the following criteria: total score (descending), average score of the opponents met (descending), average finish time in wins (increasing).

C) Design

1. Class Diagram

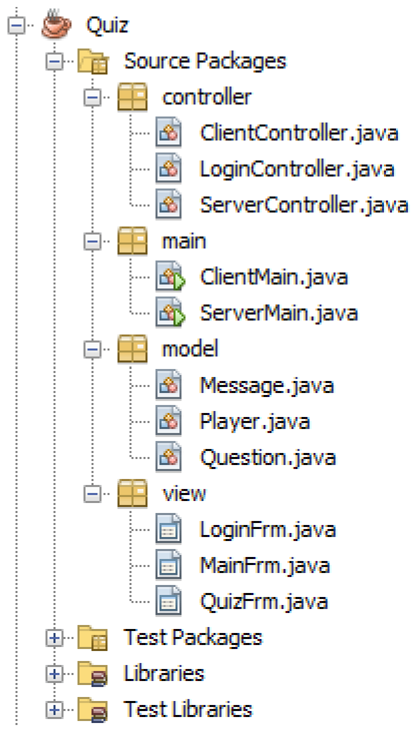


2. ERD Diagram



C) Implementation

1. MVC architecture



```
Quiz
  Source Packages
    controller
      ClientController.java
      LoginController.java
      ServerController.java
    main
      ClientMain.java
      ServerMain.java
    model
      Message.java
      Player.java
      Question.java
    view
      LoginFrm.java
      MainFrm.java
      QuizFrm.java
  Test Packages
  Libraries
  Test Libraries
```

a. Controller

i. ClientController.java

- Acts like an user, it sends message to the server and receive server's message to process
- Contains the user, frames, opponent, and all the user's logic

ii. LoginController.java

- Controls the login action, find the user in the database, if found a match then returns the desired user
- Contains the loginFrm frame

iii. ScoreboardController.java

- Controls the check score board action, find the users' achievement in the database, return a list in ScoreboardFrm frame

iv. *ServerController.java*

- Acts like the server, it sends messages to the specific user and receive each user's messages to process
- Contains a ClientHandler class - a class which handles a specific user in the system - handles all the server side's logic, receives messages and sends messages from/to that specific client
- Uses tcp/ip protocols

b. Main

i. *ClientMain.java*

- Main method for a user

ii. *ServerMain.java*

- Main method for the server

c. Model

i. *Message.java*

- Represents a message sent back and forth in the system, with attributes:
    - private String message
    - private Object data
- Types of messages here are:
    - Client -> Server(ClientHandler)
        - 1."RequestOnlineClient" //request the online client(s) list
        - 2."RequestInvite" //invite the selected player
        - 3."RespondRefuse" //refuse the invitation (on click "REFUSE" in dialog or the player is playing)
        - 4."RespondAccept" //accept the invitation
        - 5."RequestQuestion" //request the question(s) list

        - 6."RequestStartGame" //request starting the game, by click "READY" on the QuizFrm frame -> set player state to “playing”
        - 7."RequestExitGame" //if user click “exit” when game is not finished, send to the opponent to close both QuizFrm frames
        - 8."RespondGameFinish" //game is finished, send result to server
    - Server(ClientHandler) -> Client
        - 1."OnlineClient" //return the online client(s) list
        - 2."Invite" //send invitation to player
        - 3."Refuse" //send refuse invitation to player who invites
        - 4."Accept" // send accept invitation to player who invites
        - 5."Question" //send the question(s) list
        - 6."StartGame" //allow both clients to start game
        - 7."ExitGame" //when game is not finished but client exited

ii. Player.java

- Represents an user in this system, with attributes:
    - private int id
    - private String username
    - private String password
    - private String state
    - private float totalScore
    - private int totalMatch
    - private float totalSpareTime
    - private int totalWin
    - private boolean playing

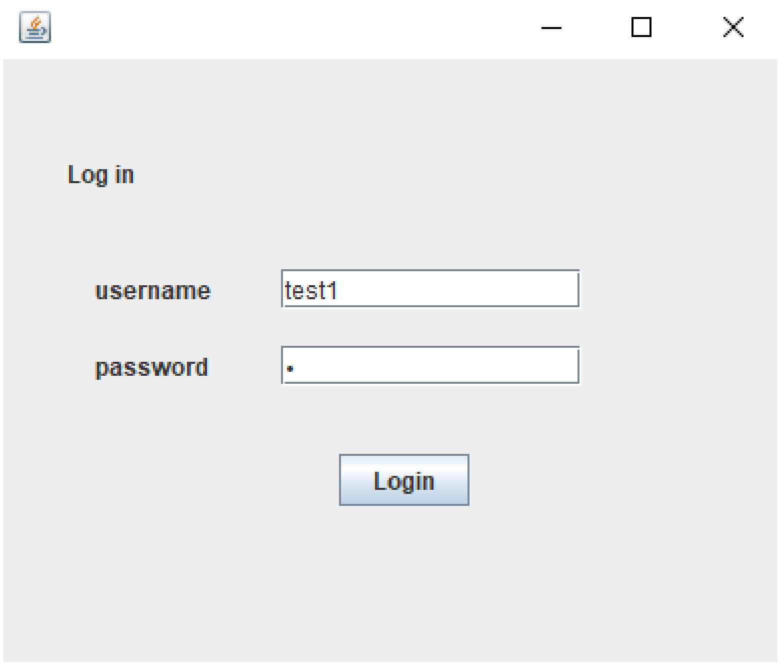
iii. Question.java

- Represents a question from the quiz, with attributes:
    - private String question
    - private String[] options
    - private String answer

d. View

i. LoginFrm.java

- The login frame, which looks like this:

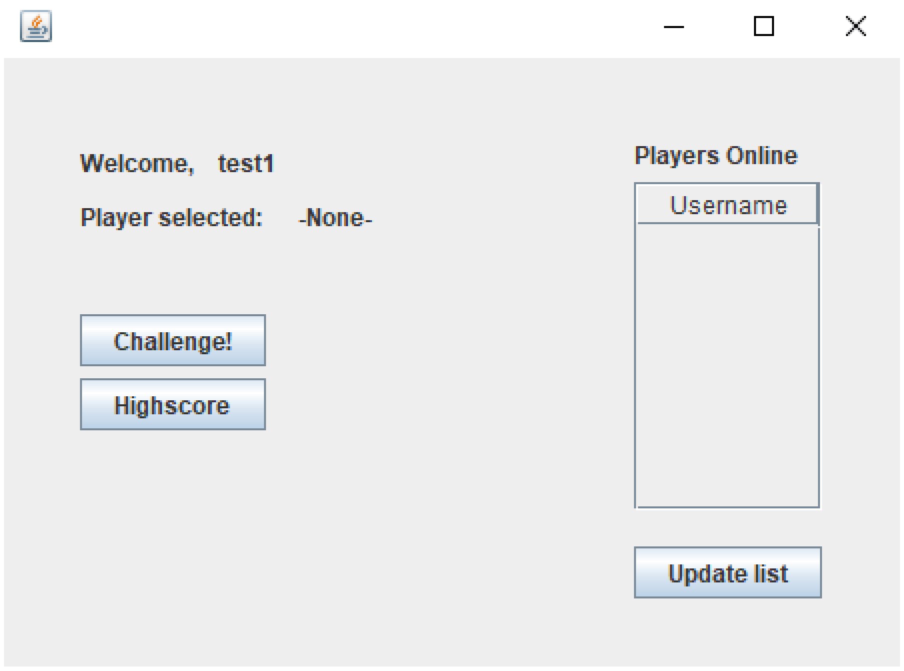

Log in

username test1

password •

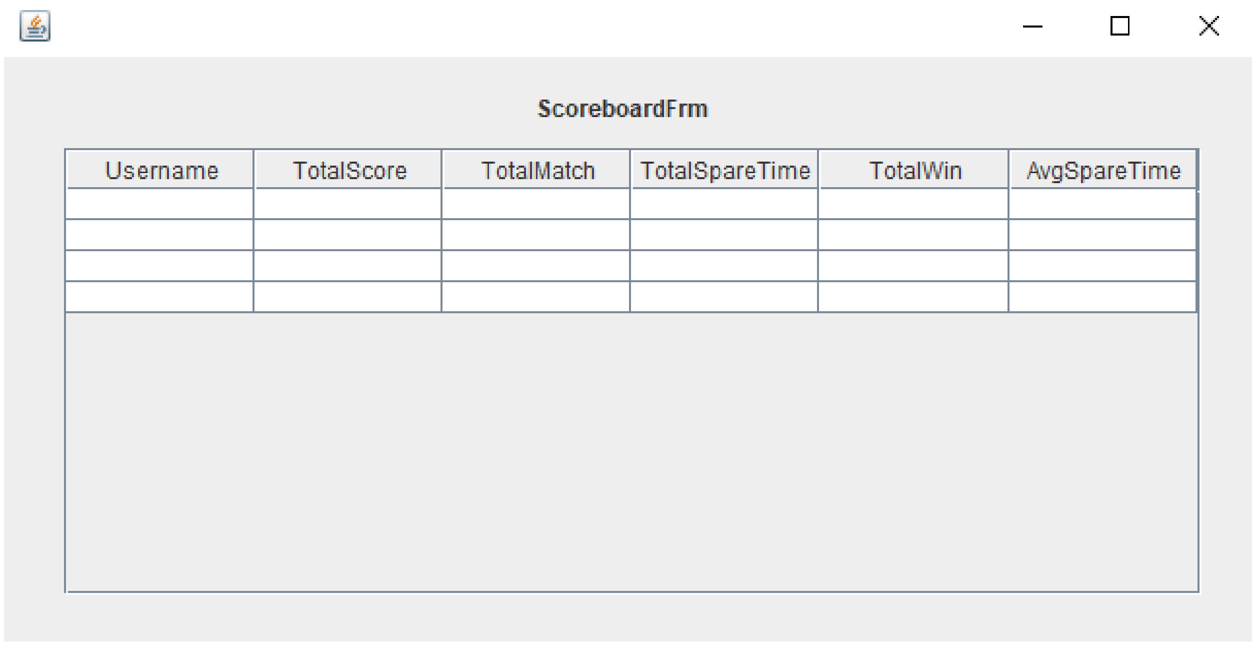
ii. MainFrm.java

- The main frame, which looks like this:



iii. ScoreboardFrm.java

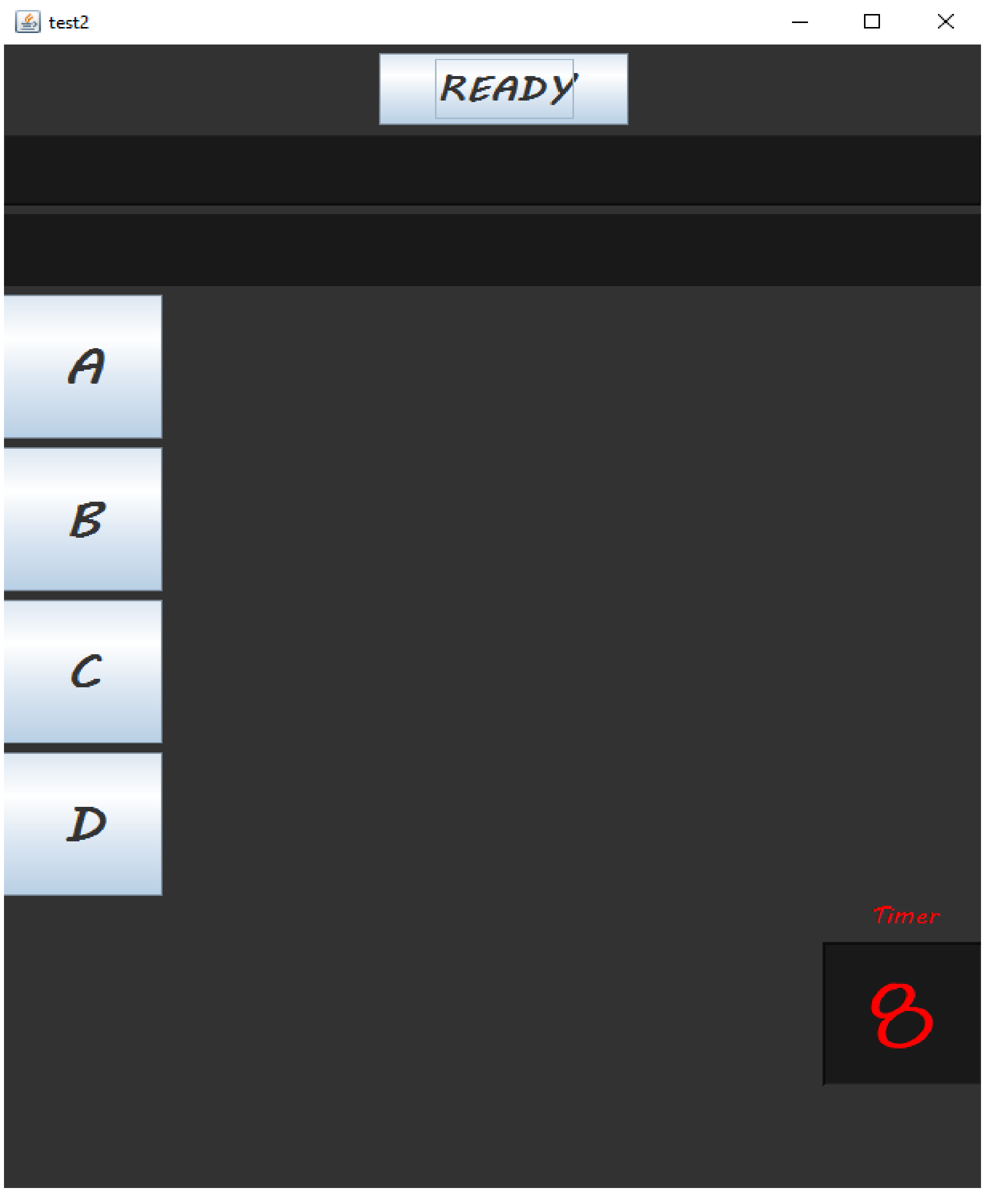
- The score board frame, which looks like this:



Username	TotalScore	TotalMatch	TotalSpareTime	TotalWin	AvgSpareTime

iv. QuizFrm.java

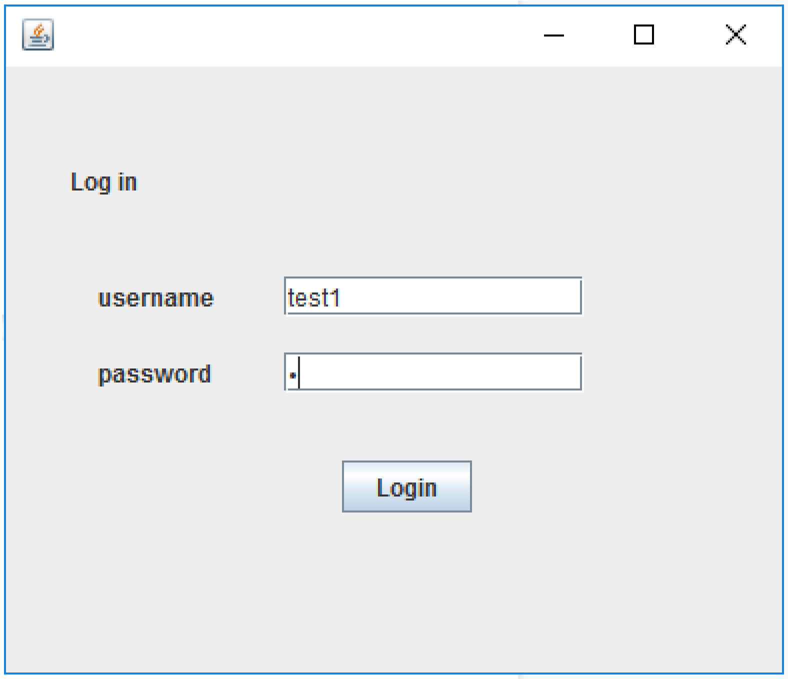
- The quiz frame, which looks like this:

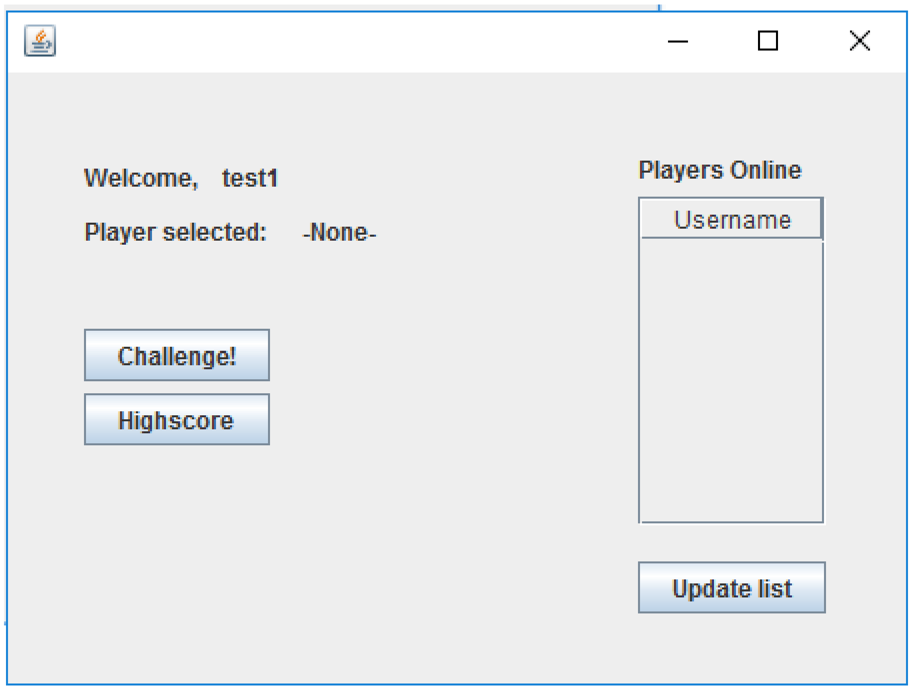


2. Run the game

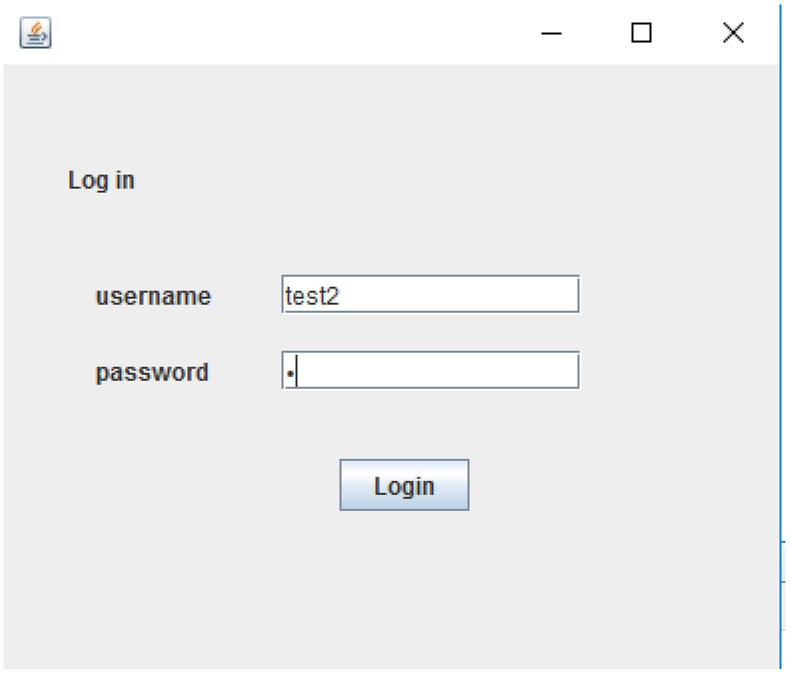
a. Main flow

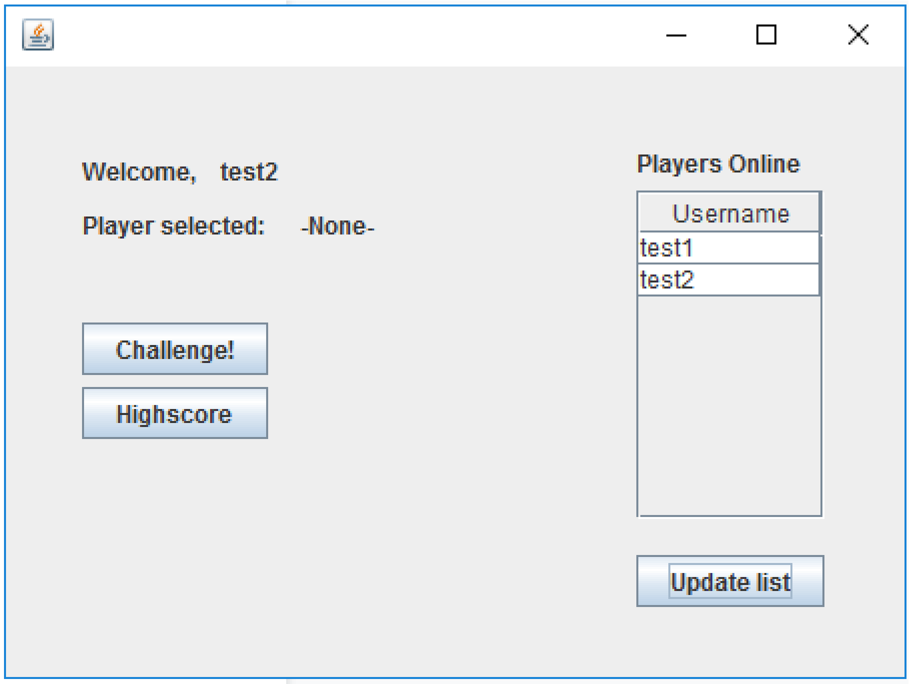
1. User1 login





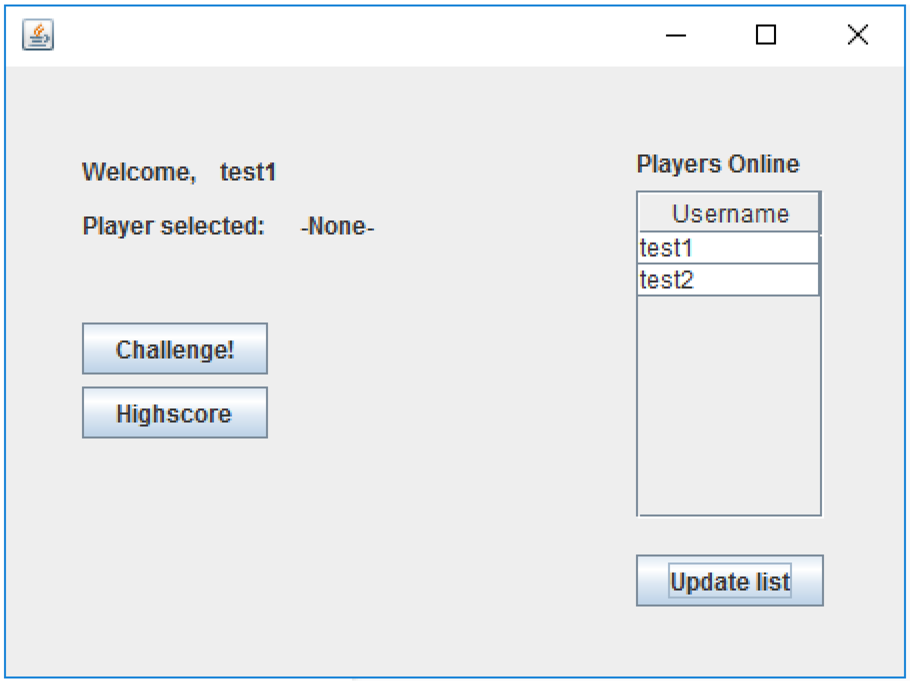
2. User2 login





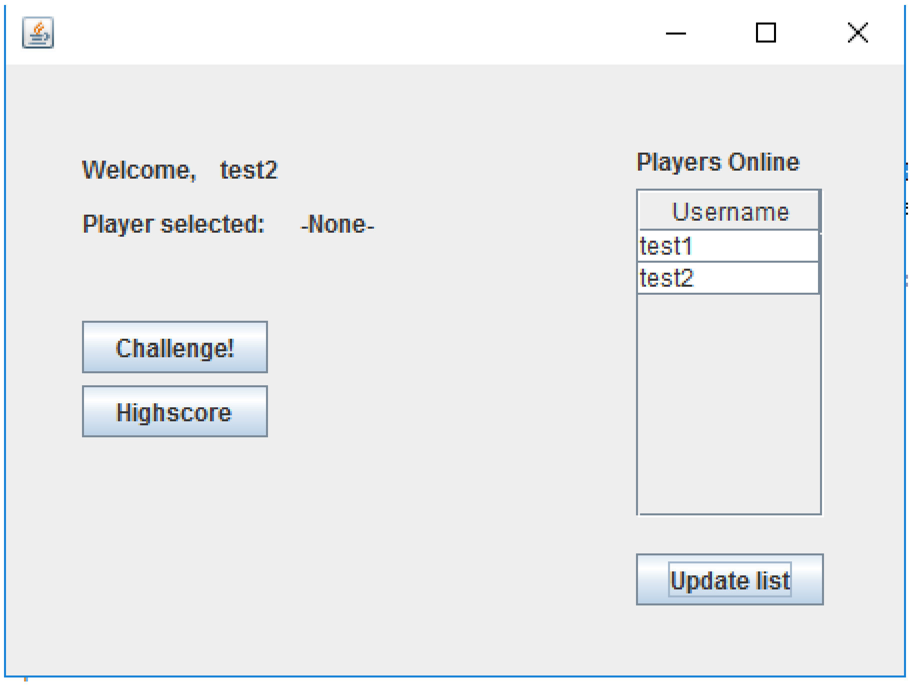
Username
test1
test2

3. User1 request online client(s)


Welcome, test1

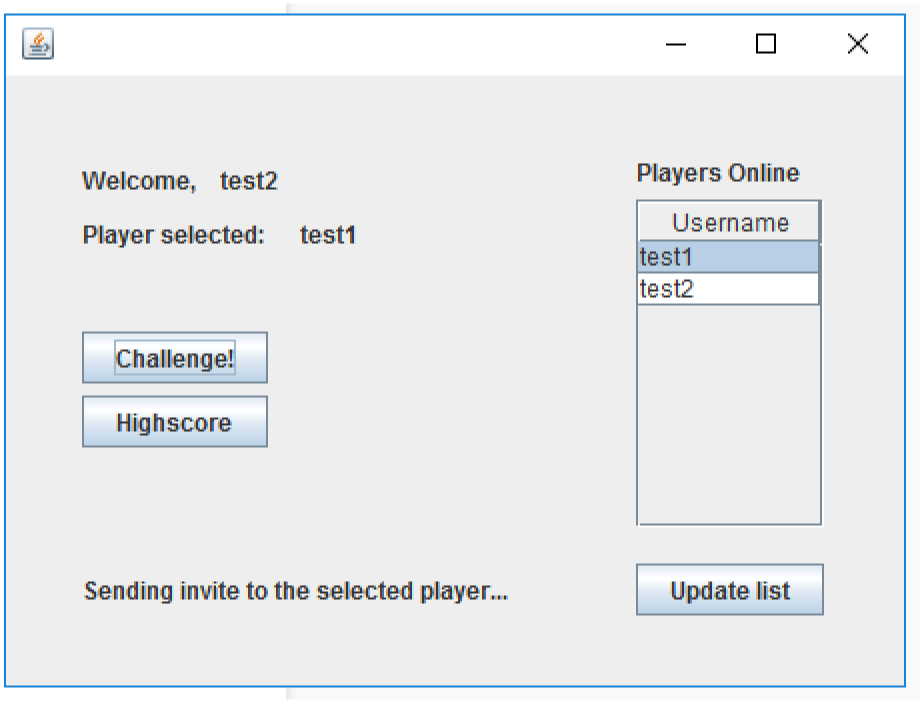
Player selected: -None-

4. User2 request online client(s)


Welcome, test2

Player selected: -None-

5. An user challenges the other user


Welcome, test2

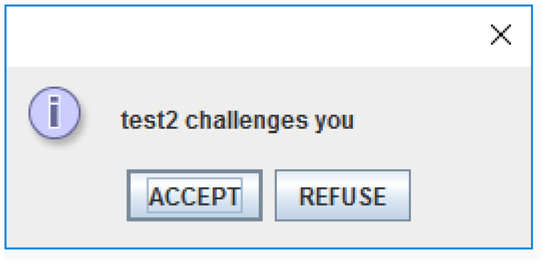
Player selected: test1

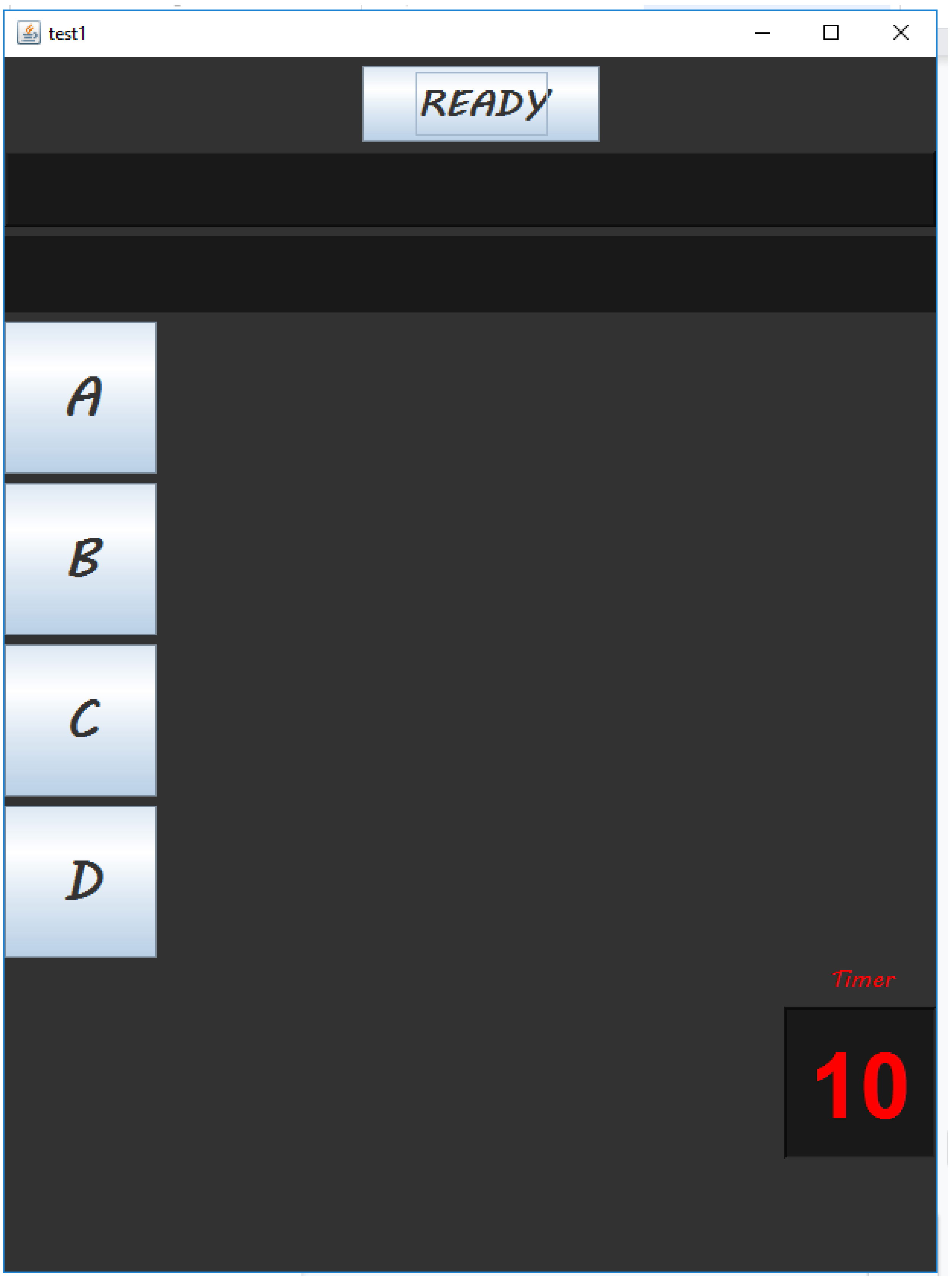
Players Online

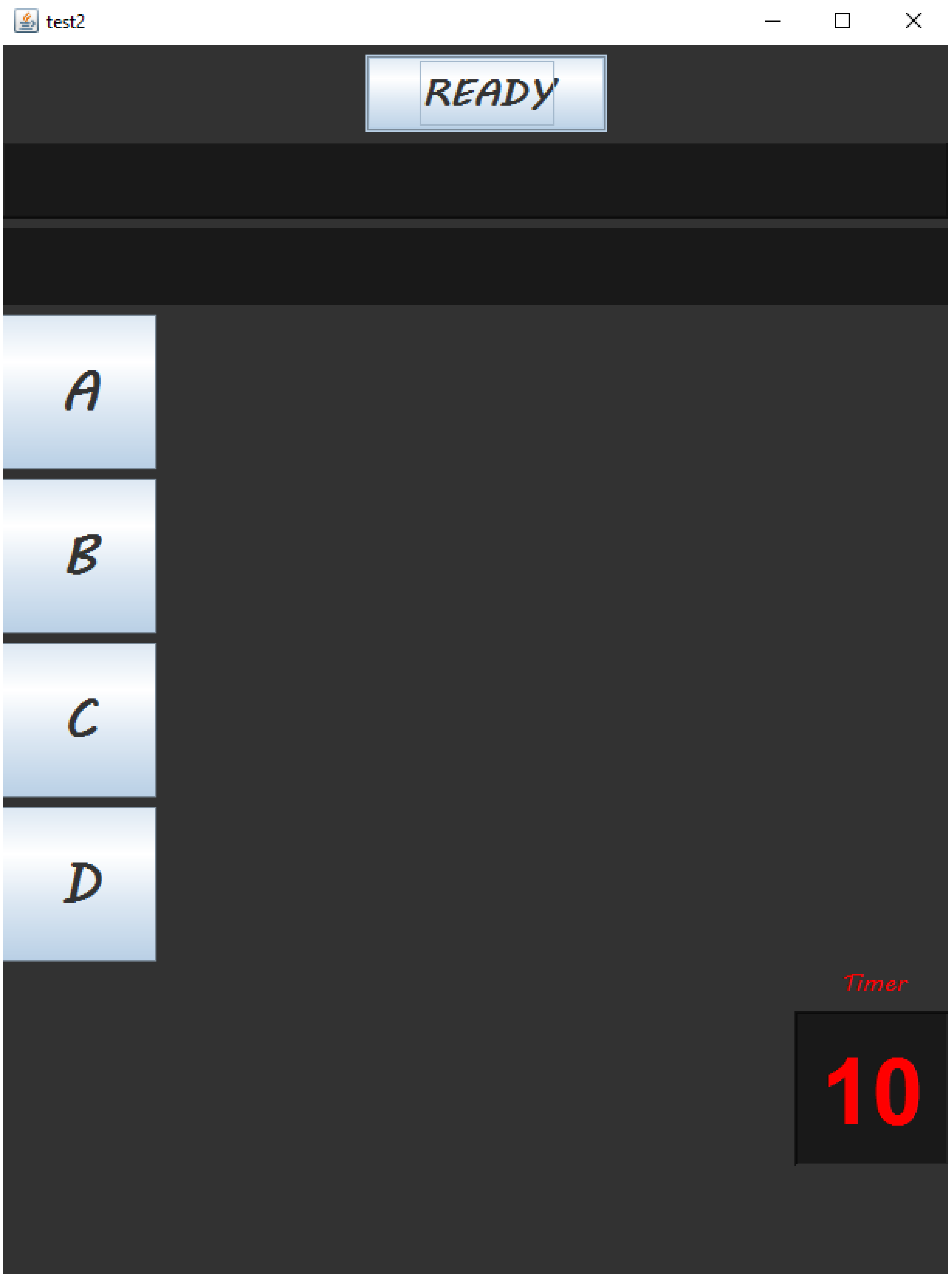
Username
test1
test2

Sending invite to the selected player...

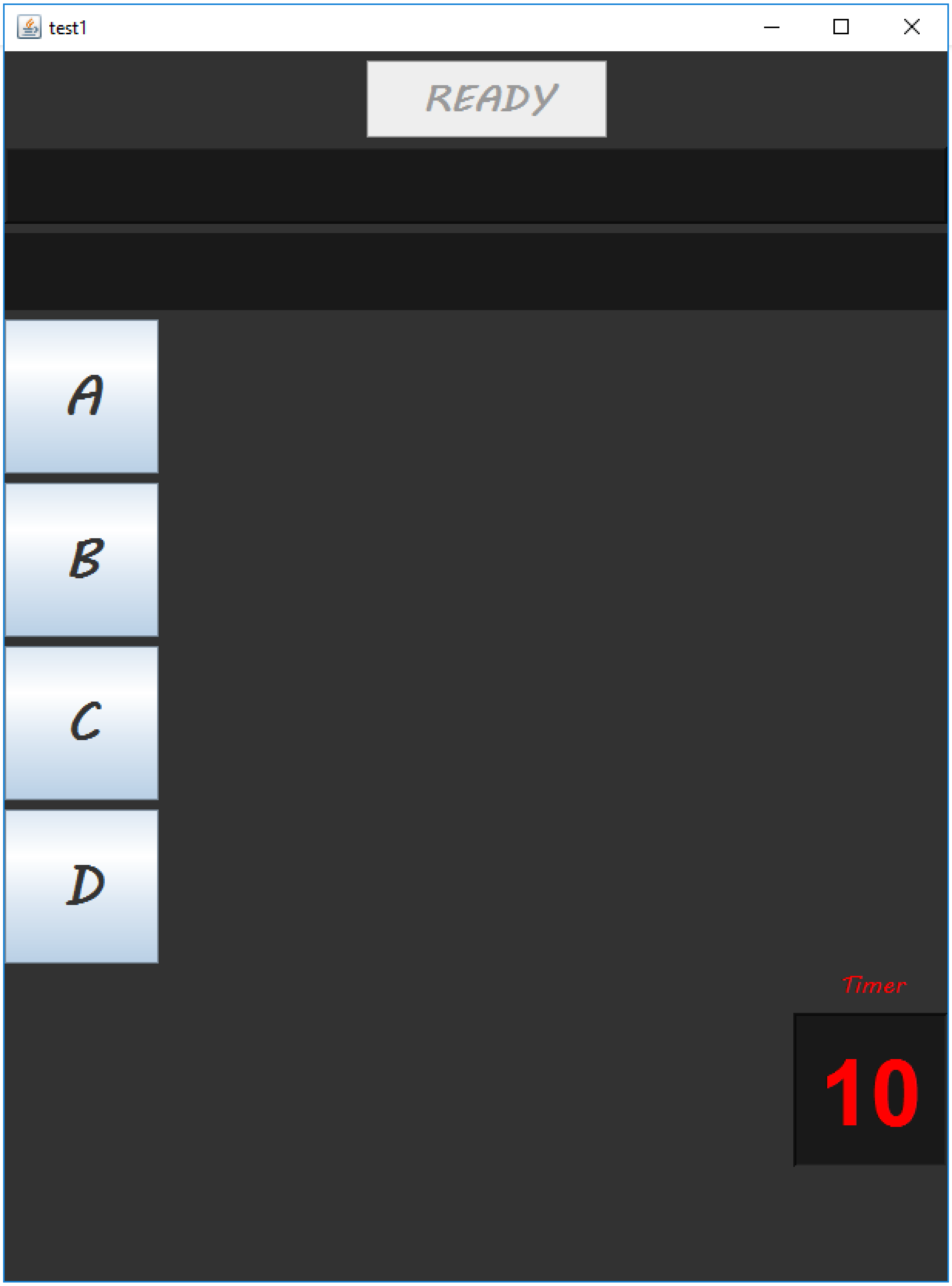
6. The challenged user accepts the challenge

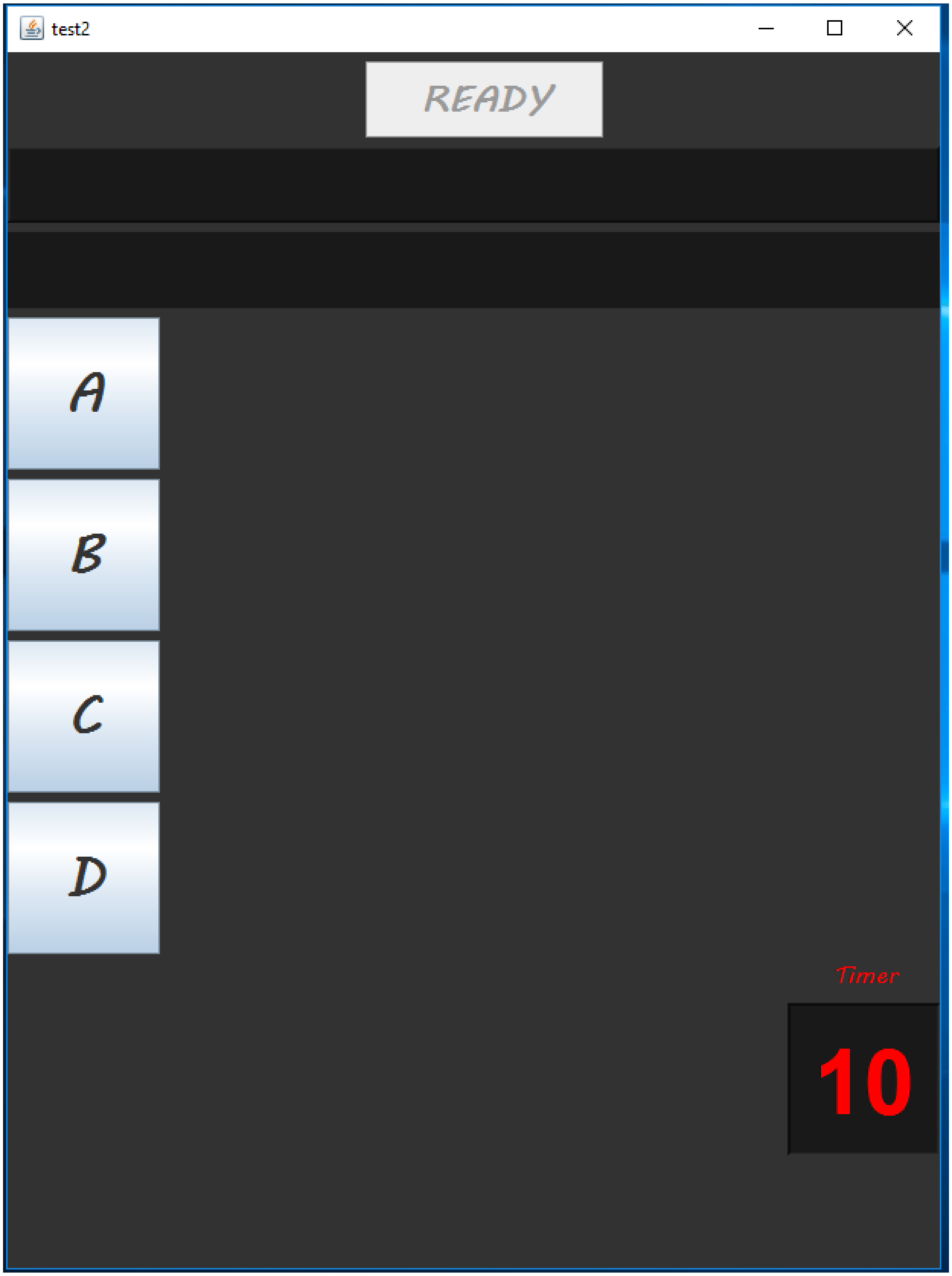

test2 challenges you


test1
READY
A
B
C
D
Timer
10

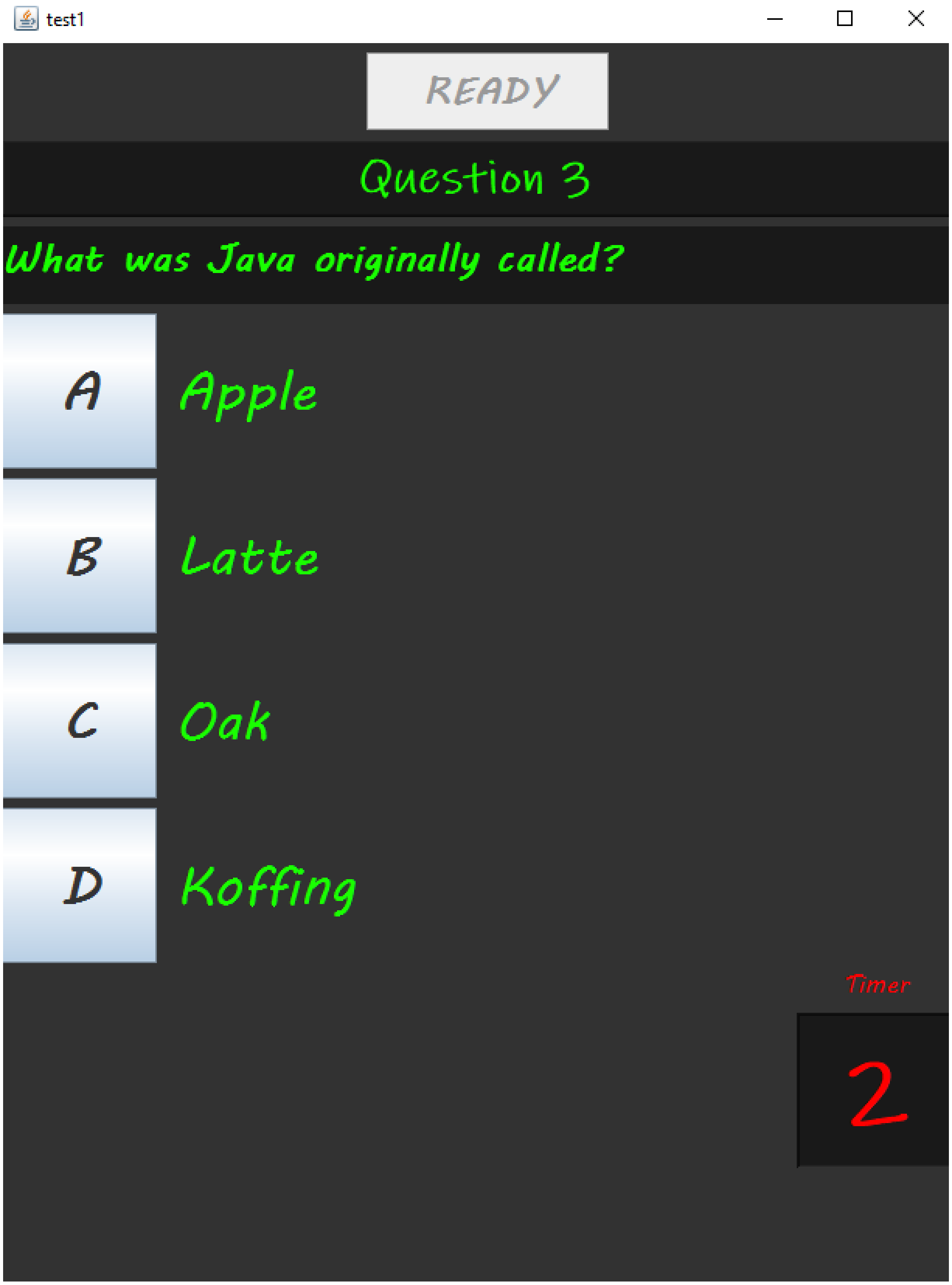

test2
READY
A
B
C
D
Timer
10

7. Both user sequentially click “Ready”


test1
READY
A
B
C
D
Timer
10


test2
READY
A
B
C
D
Timer
10

8. Both quiz frames begin



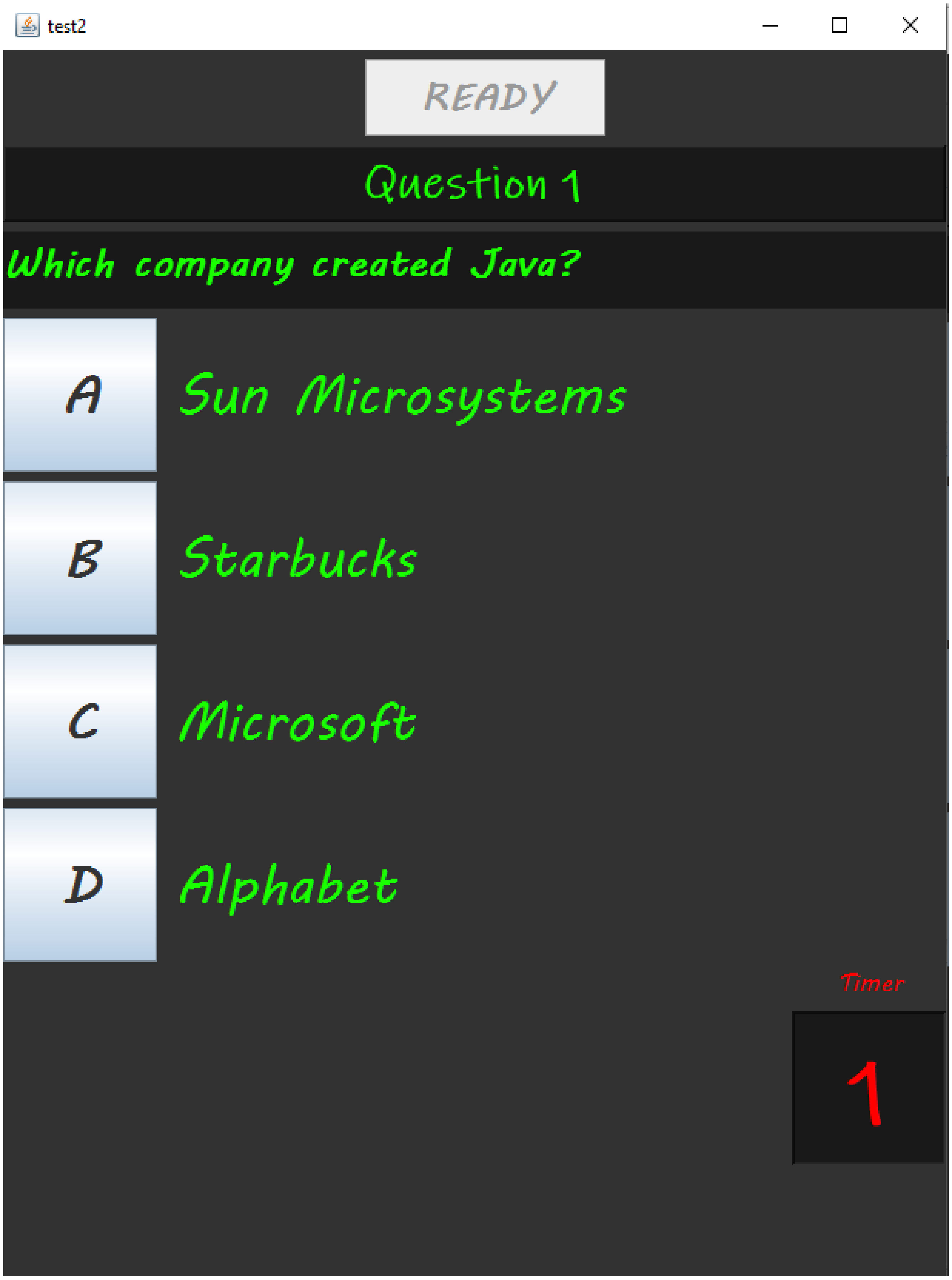
What was Java originally called?

A Apple

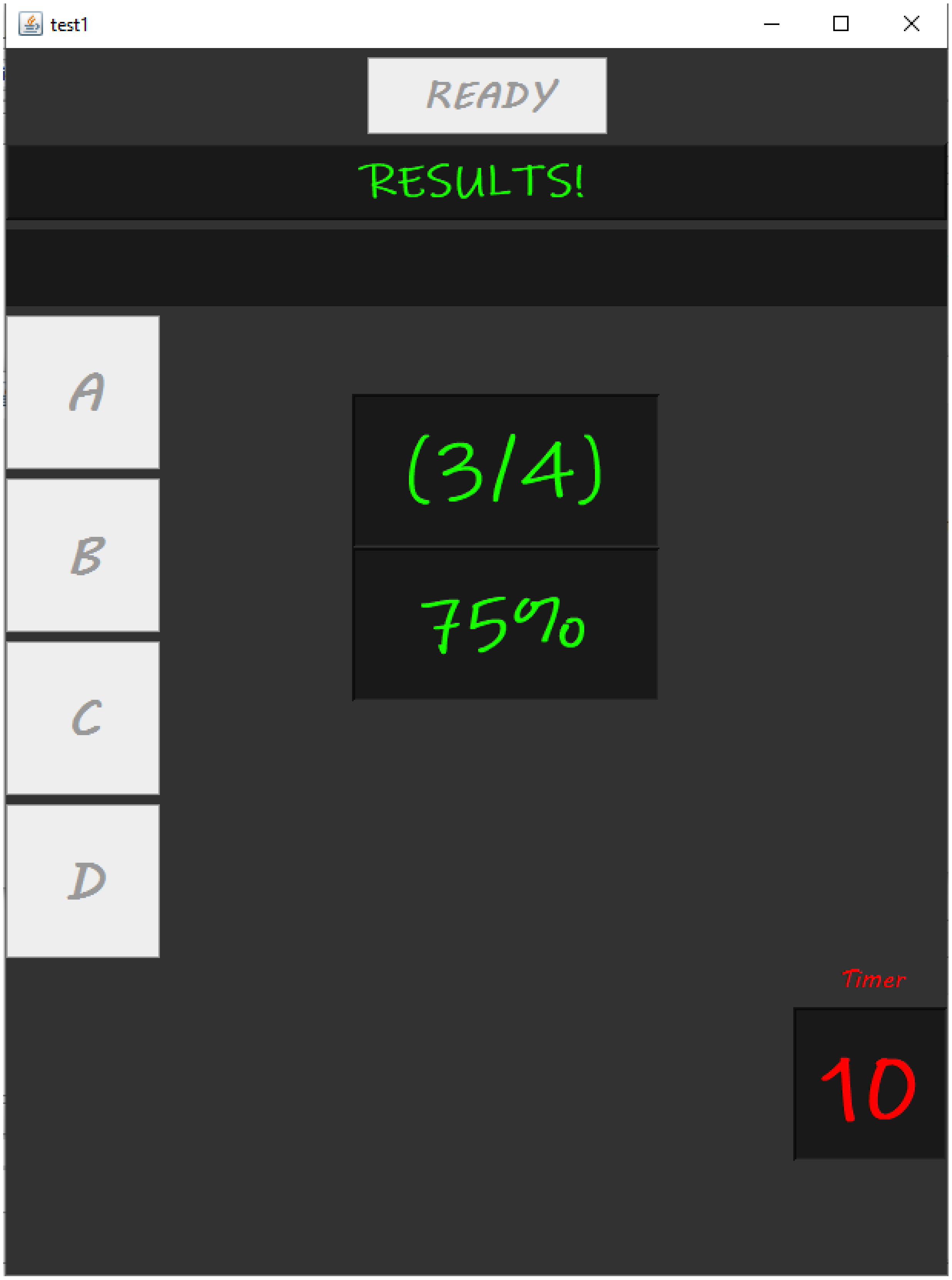
B Latte

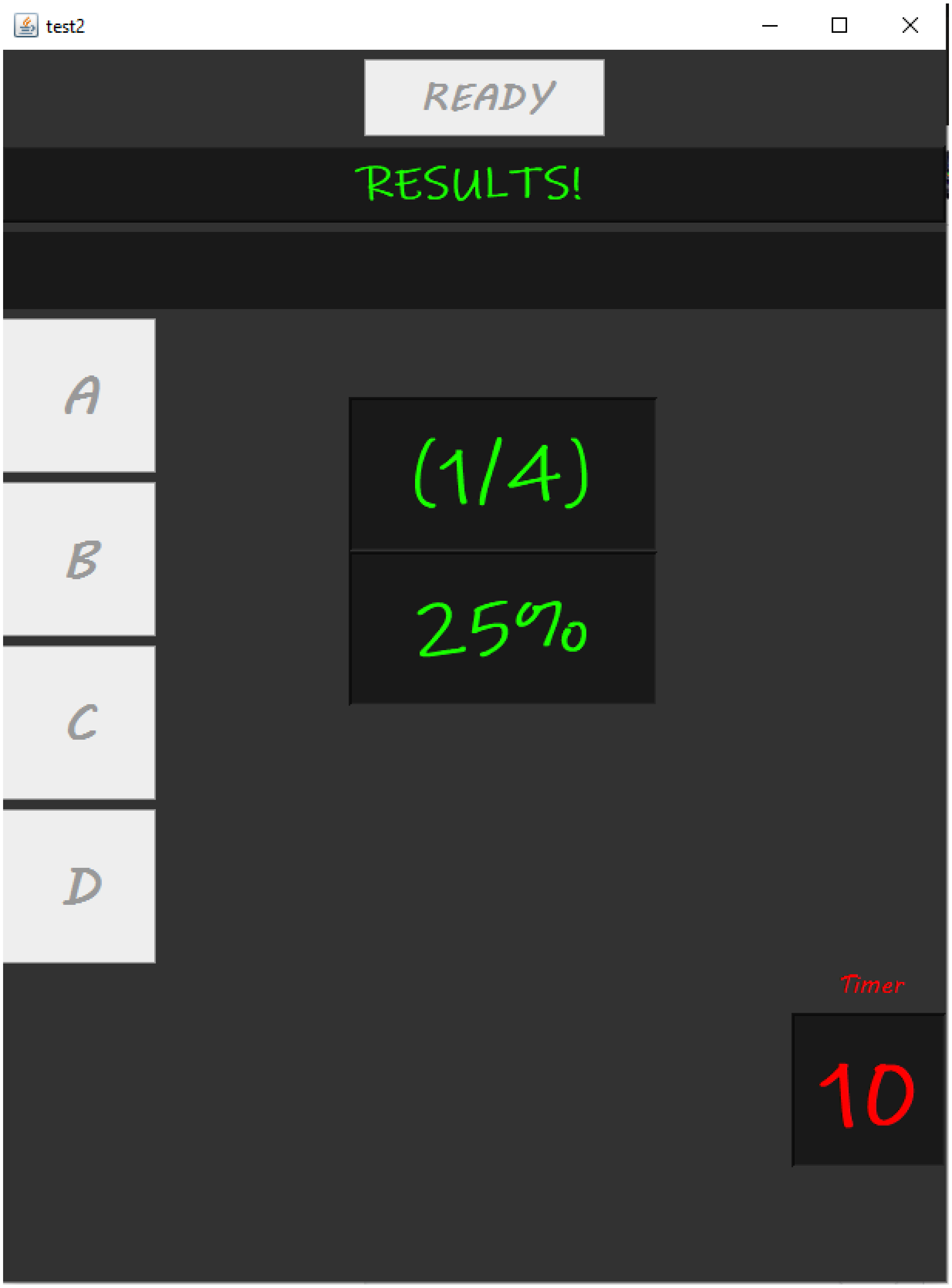
C Oak

D Koffing



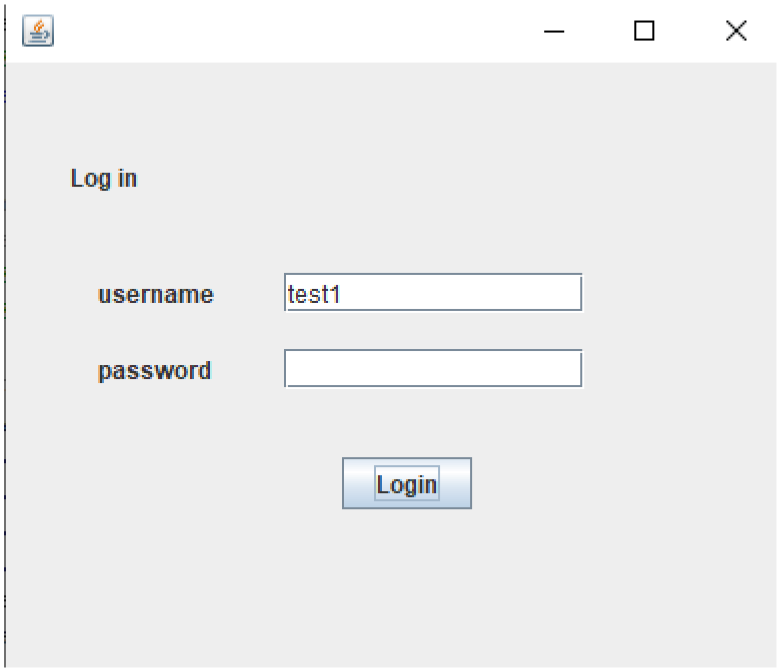
9. Both quiz frames return each of theirs' result


test1
READY
RESULTS!
A
B
C
D
(3/4)
75%
Timer
10


test2
READY
RESULTS!
A
B
C
D
(1/4)
25%
Timer
10

b. Exception flow

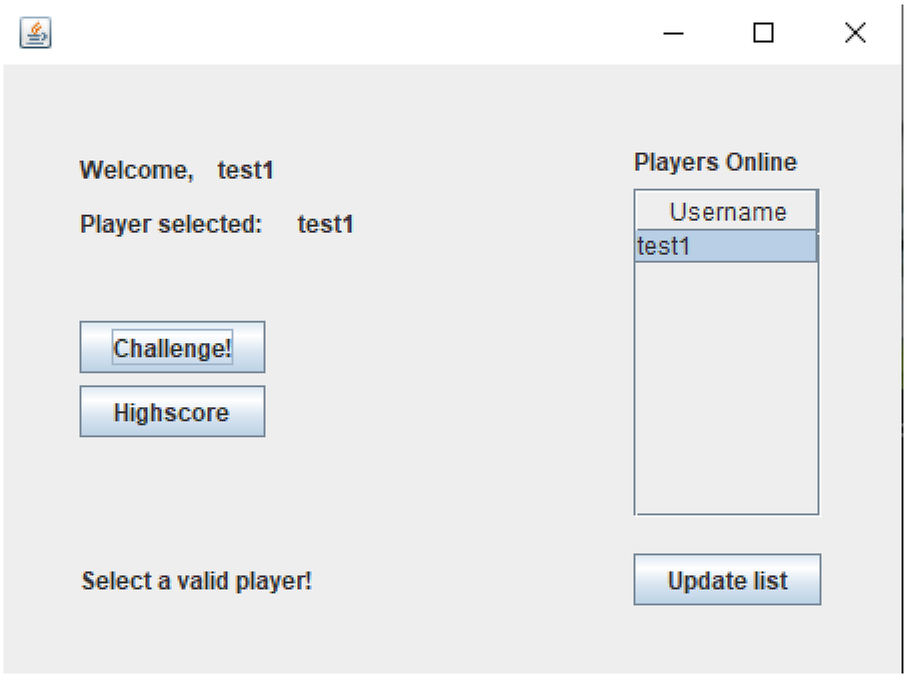
1. User fails to login (nothing happened)


Log in

username test1

password

2. User challenges his/herself


Welcome, test1

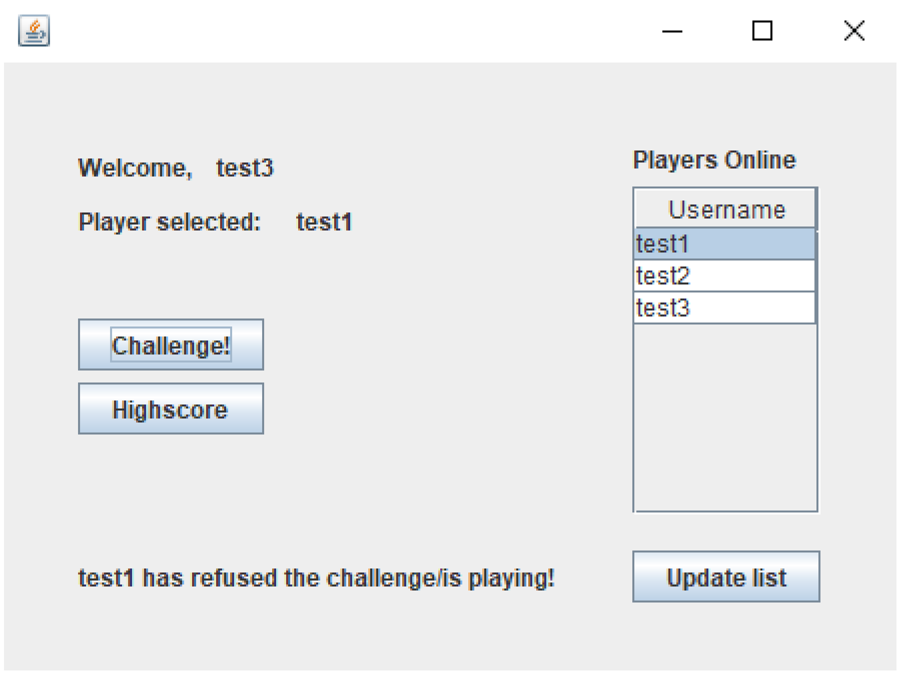
Player selected: test1

Players Online

Username
test1

Select a valid player!

3. User challenges a busy player


Welcome, test3

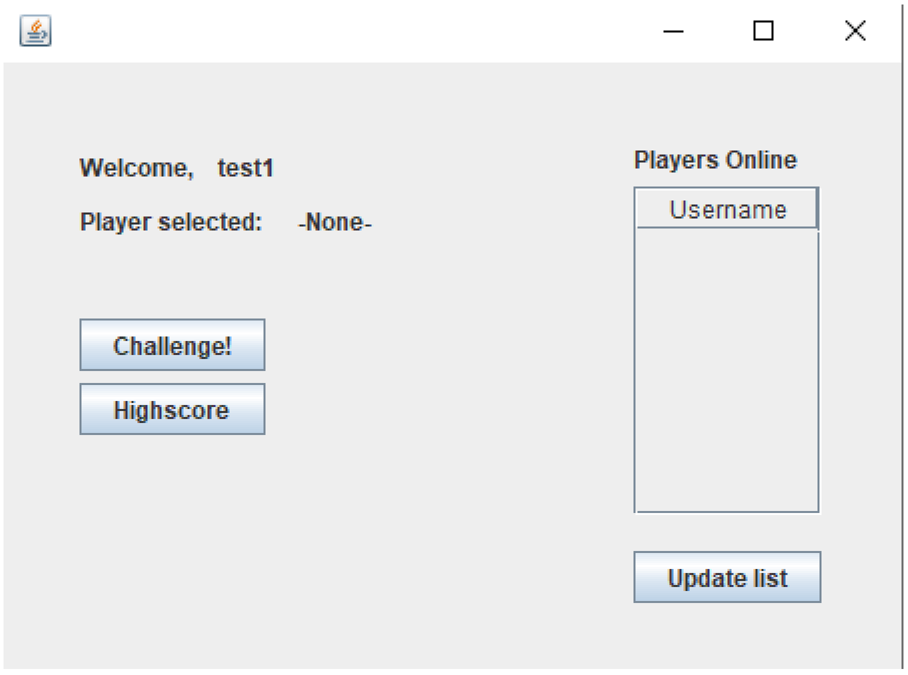
Player selected: test1

test1 has refused the challenge/is playing!

3. Check scoreboard

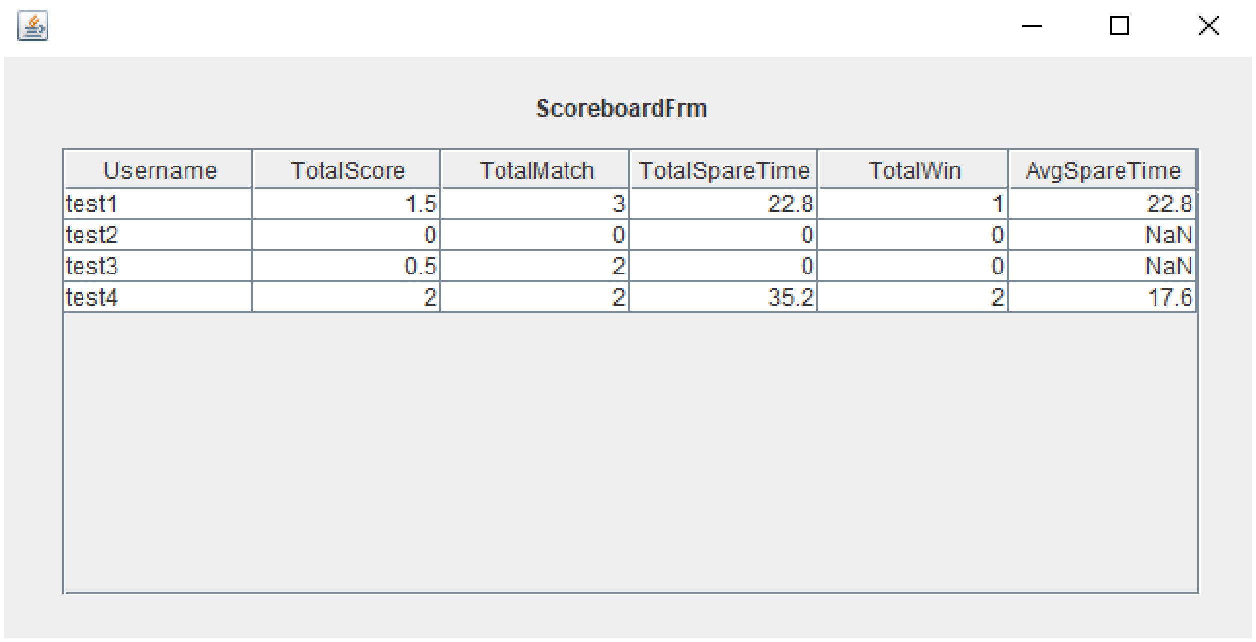
Main flow

1. User click “Highscore”


Welcome, test1

Player selected: -None-

2. The scoreboard appears (click each column’s header to sort)



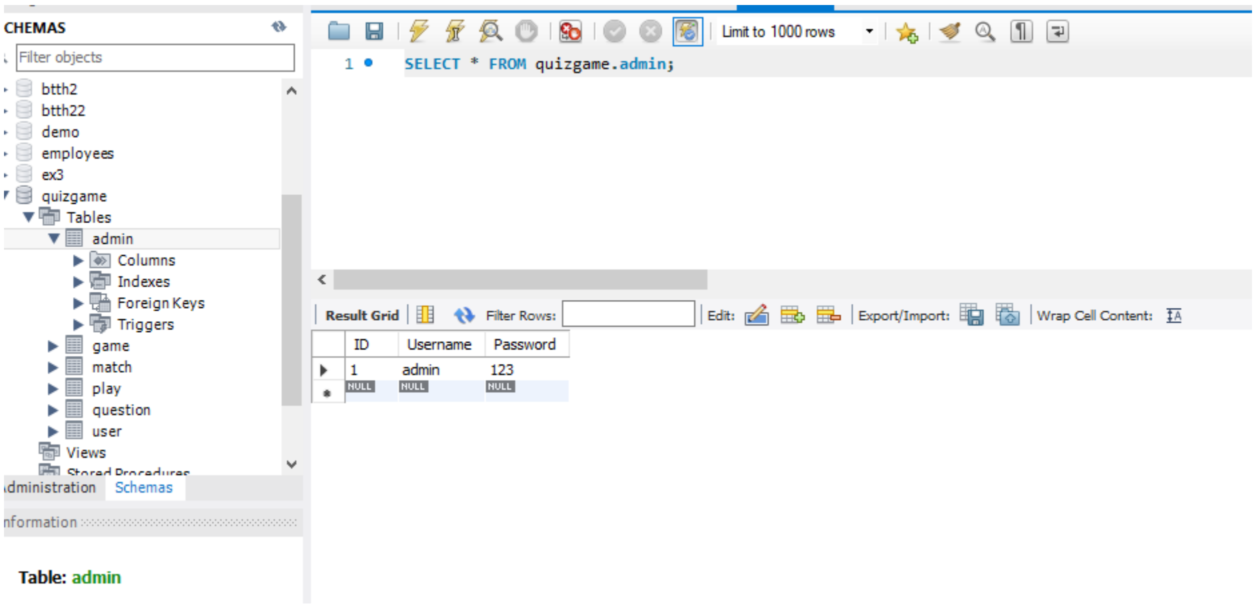
ScoreboardFrm

Username	TotalScore	TotalMatch	TotalSpareTime	TotalWin	AvgSpareTime
test1	1.5	3	22.8	1	22.8
test2	0	0	0	0	NaN
test3	0.5	2	0	0	NaN
test4	2	2	35.2	2	17.6

D) User Management Page (using PHP/JSPServlet)

Prepare the database

Create the admin table for our db:



```
SELECT * FROM quizgame.admin;
```

ID	Username	Password
1	admin	123
NULL	NULL	NULL

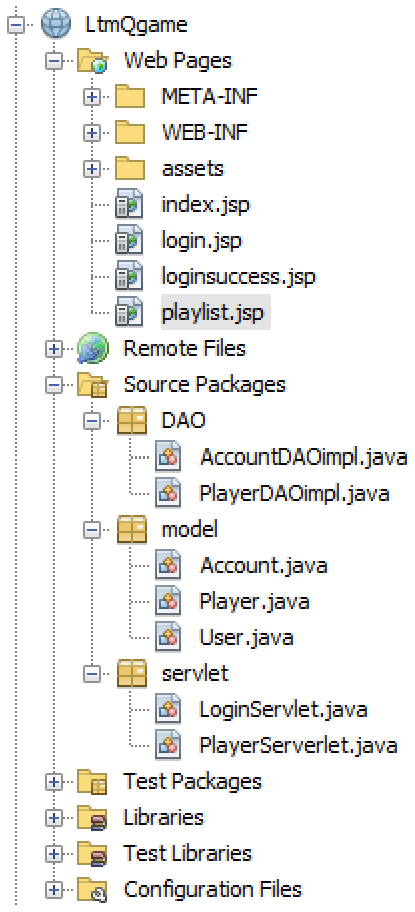
Table: admin

The query for creating the table:

```
CREATE TABLE admin (
  ID int NOT NULL AUTO_INCREMENT,
  Username varchar(255) DEFAULT NULL,
  Password varchar(255) DEFAULT NULL,
  PRIMARY KEY (ID)
)
```

Implementation using JSPServlet

Servlet architecture



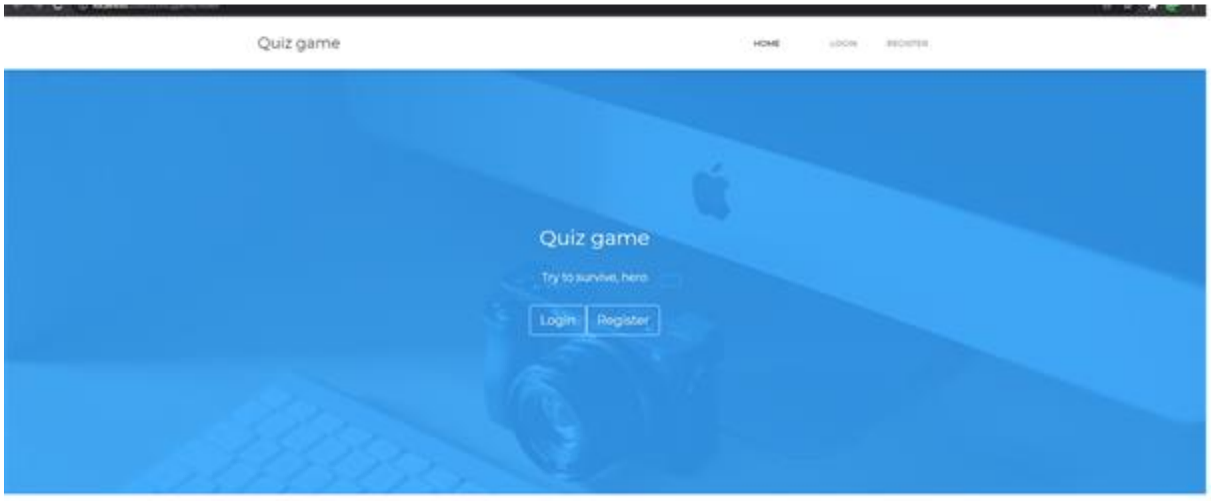
```
LtmQgame
  Web Pages
    META-INF
    WEB-INF
    assets
    index.jsp
    login.jsp
    loginsuccess.jsp
    playlist.jsp
  Remote Files
  Source Packages
    DAO
      AccountDAOimpl.java
      PlayerDAOimpl.java
    model
      Account.java
      Player.java
      User.java
    servlet
      LoginServlet.java
      PlayerServerlet.java
  Test Packages
  Libraries
  Test Libraries
  Configuration Files
```

1. *Package model*
    a. Player
        - private int id
        - private String username
        - private String password
        - private String state
        - private float totalScore
        - private int totalMatch;
        - private float totalSpareTime
        - private int totalWin
        - private boolean playing
    b. Account
        - private int id
        - private String name
        - private String password
    c. User
        - private int id
        - private String name
        - private int age
        - private Account account
2. *Package servlet*
    a. LoginServlet
        - Get data by AccountDAO, if the login credentials are valid, redirect to player.jsp
    b. PlayerServlet
        - Get data by PlayerDAO, Displays a list of players' information

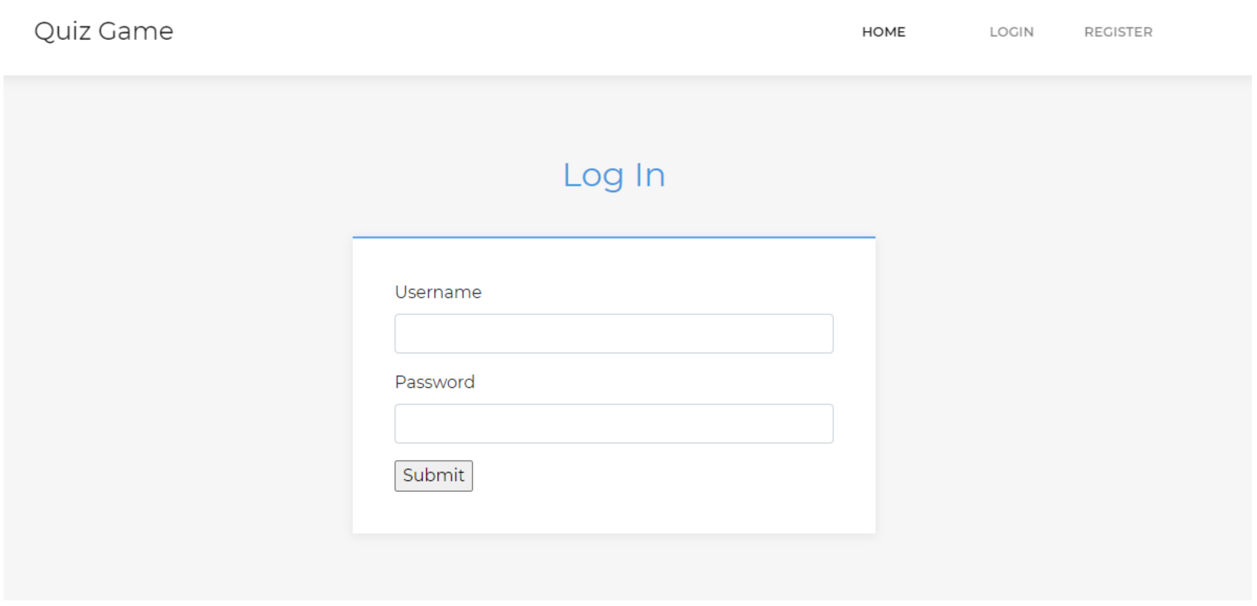
3. *Package DAO*
    a. AccountDAO
        - Connect to database, get information for LoginServlet
    b. PlayerDAO
        - Connect to database, get information for PlayerServlet
4. *Package JSP*
    a. Index.jsp
        - The index page contains link to login form
    b. Login.jsp
        - The login form with username and password fields for the user to log into the system
    c. Playerlist.jsp
        - The Players' statistics page, contains players' usernames, scores,...

Run the web

1. System appears the index page



2. User clicks login, system appears the login form


Quiz Game

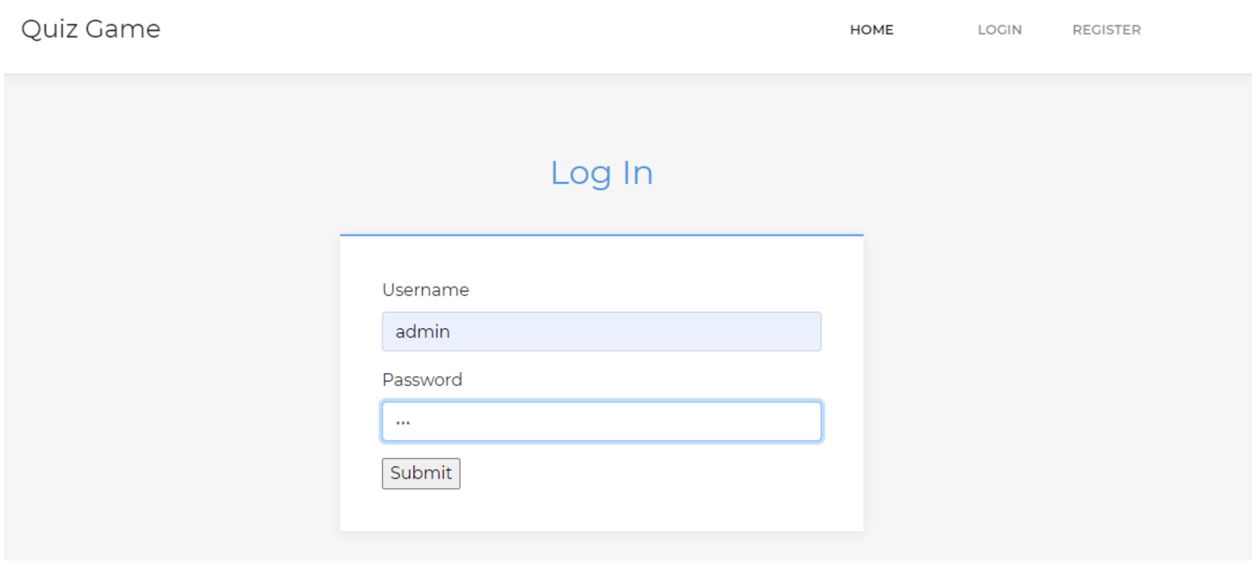
HOME LOGIN REGISTER

Log In

Username

Password

3. User enters username and password and clicks login, system appears the Playerlist page

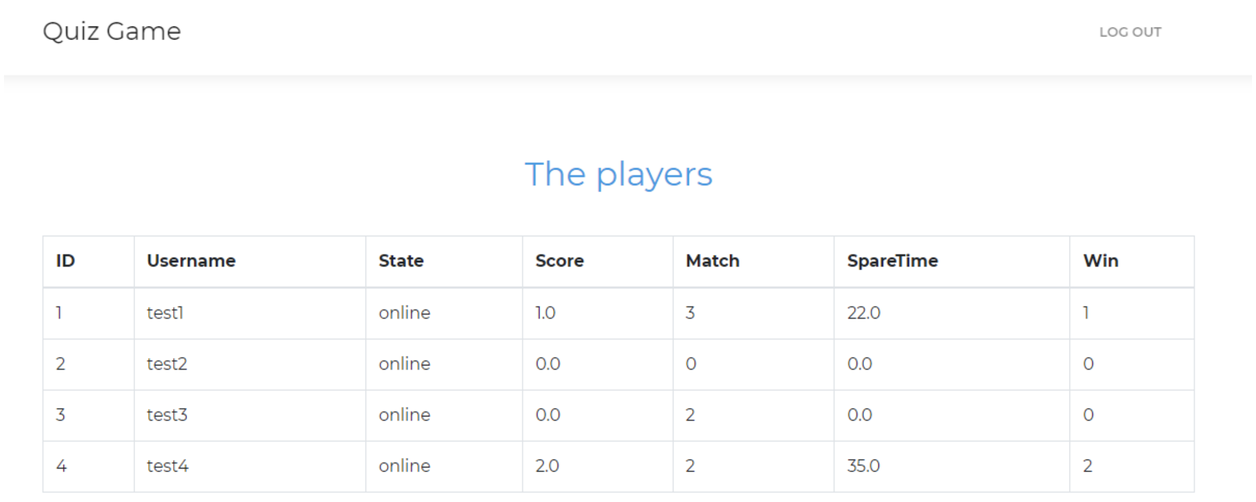

Quiz Game

HOME LOGIN REGISTER

Log In

Username: admin

Password: •••



Quiz Game

LOG OUT

The players

ID	Username	State	Score	Match	SpareTime	Win
1	test1	online	1.0	3	22.0	1
2	test2	online	0.0	0	0.0	0
3	test3	online	0.0	2	0.0	0
4	test4	online	2.0	2	35.0	2