Mini Project Specification

A mini project (software development to be demonstrated by week 24 latest, and report writing – week 24)

Develop a java networked database project called myTicTacToe to have a server based database accessible by multiple users. The database should hold information on users (names, usernames, and passwords). Anyone should be able to add him/herself as a user. Further discussions will take part in a lecture.

The given TicTacToe program is a java application. You are expected to do the following:

1. Import and test the given project before you attempt to improve it.
2. Convert the program to a simple client-server program to enable two users on separate client machines play the game against each other. Listing 1 shows a sample code snippet for the server side. You do not need a GUI for the server side but you may choose to have one if you wish to. *Hints: Use the source code given, add a client class, and add a server class. The source code given will be part of server side implementation. Use it as a separate class at server end but NOT as the main class since the server class should be the main class.*

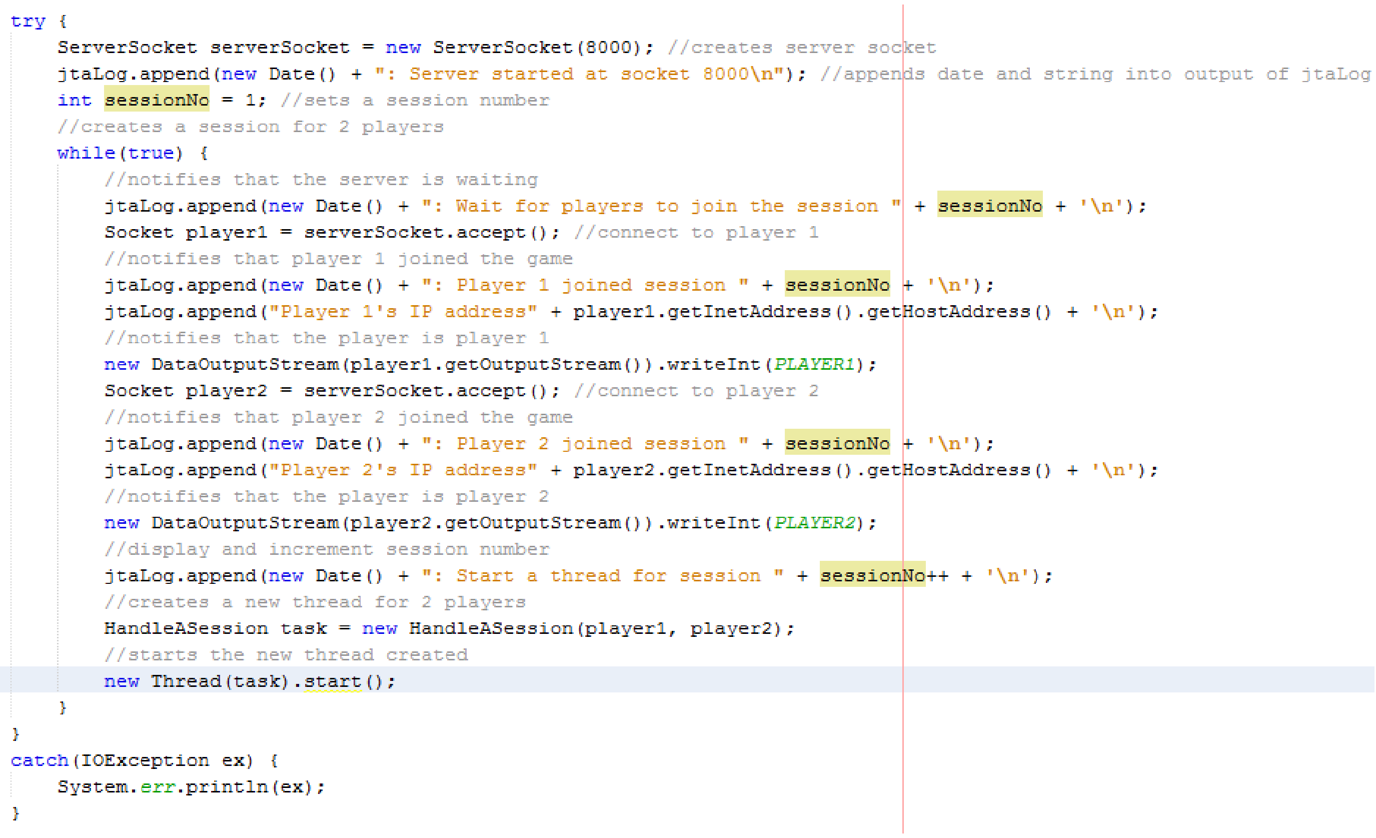
   [0-40 Marks][1]
3. Add a database to the project. A server side database should hold user information as shown in the tables below. The specifications given are bare minimum. Enhance tables adding one or more columns of your choice (e.g. date enrolled, date played etc.)
   a. Create a table called PLAYERS. This will be used to keep a record of players. Although no authentication is expected at this stage, this can later be added.
   b. Create a table called RESULTS. This should hold results of games between any two players played at any time. The column Winner should take values 1, 2, or 0. The first two identifies the winner while 0 indicates a draw.
   c. Note that at this stage only one game can be played (no multi-threading yet).
4. The game should start as soon as a second player becomes active.
5. Write SQL scripts as part of methods to do the following (you will need new GUI components to be added to your application):
   a. Add new user.
   b. Get a person's results (show the person's name, opponent's name, won/lost/drew as well as total number of points obtained). Each wins worth 1 and each draw worth 0.5. Zero for losses.
   c. Produce the cumulative results (i.e. as in 5b. but for all players.)

   [0-29 Marks] [2]
6. Add multi-threading to your program to enable multiple sessions of the game run simultaneously.
7. Active and free users should be flagged so that a new player can find an opponent. A free user is one who started the application and waiting for an opponent.

   [0-31 Marks] [3]

[1] Please see the marking scheme for details
[2] Please see the marking scheme for details
[3] Please see the marking scheme for details

Listing 1: Sample code



```
try {
    ServerSocket serverSocket = new ServerSocket(8000); //creates server socket
    jtaLog.append(new Date() + ": Server started at socket 8000\n"); //appends date and string into output of jtaLog
    int sessionNo = 1; //sets a session number
    //creates a session for 2 players
    while(true) {
        //notifies that the server is waiting
        jtaLog.append(new Date() + ": Wait for players to join the session " + sessionNo + '\n');
        Socket player1 = serverSocket.accept(); //connect to player 1
        //notifies that player 1 joined the game
        jtaLog.append(new Date() + ": Player 1 joined session " + sessionNo + '\n');
        jtaLog.append("Player 1's IP address" + player1.getInetAddress().getHostAddress() + '\n');
        //notifies that the player is player 1
        new DataOutputStream(player1.getOutputStream()).writeInt(PLAYER1);
        Socket player2 = serverSocket.accept(); //connect to player 2
        //notifies that player 2 joined the game
        jtaLog.append(new Date() + ": Player 2 joined session " + sessionNo + '\n');
        jtaLog.append("Player 2's IP address" + player2.getInetAddress().getHostAddress() + '\n');
        //notifies that the player is player 2
        new DataOutputStream(player2.getOutputStream()).writeInt(PLAYER2);
        //display and increment session number
        jtaLog.append(new Date() + ": Start a thread for session " + sessionNo++ + '\n');
        //creates a new thread for 2 players
        HandleASession task = new HandleASession(player1, player2);
        //starts the new thread created
        new Thread(task).start();
    }
}
catch(IOException ex) {
    System.err.println(ex);
}
```

Table 1: **PLAYERS**:

	ID	Name	Surname	Username	Password	email
Type	Integer	Varchar	Varchar	Varchar	Varchar	Varchar
Size	N/A	30	30	30	30	30
Primary key?	Yes	No	No	No	No	No
Unique?	Yes	No	No	Yes	No	Yes
Null?	No	No	No	No	Yes	Yes
Foreign key?						

Table 2: **RESULTS**

	Game_No	Name1	Surname1	Name2	Surname2	Winner
Type	Integer	Varchar	Varchar	Varchar	Varchar	Integer
Size	N/A	30	30	30	30	1
Primary key?	Yes	No	No	No	No	No
Unique?	Yes	No	No	No	No	No
Null?	No	No	No	No	No	No
Foreign key?						

The coursework report should contain the following sections:

- **Title page** - giving title, module, date and authors' details
- **Introduction** - a short description of the work done
- **List of contents** - with page numbers
- **Design and development** – The design and development of your program.
- **Testing, results, and discussions** – supported by screen shots.
- **Conclusion** - A short summary of the work done and your conclusions, possible improvements and enhancements

- **References used** - a full list of all sources used; books, journals/magazines, electronic sources

Bonus: Implement authentication so that 5 points is added to your average laboratory grade (only if functional). This may be as significant as moving you up by a class.

Alternative table

Table 2: **RESULTS**

	Game_No	Date Played	Player1_ID	Player2_ID	Winner
Type	Integer	Date	Varchar	Varchar	Integer
Size	N/A		30	30	1
Primary key?	Yes	No	No	No	No
Unique?	Yes	No	No	No	No
Null?	No	No	No	No	No
Foreign key?					