Pseudocode

Login\_Server.java

Easy Group

**START**

Create a private integer “clientCount” variable to count number of clients

Create boolean isSignUpSucceeded is false

//Admin email address

Create a String “adminEmail” to hold admin’s email

Create a String adminEmailPassword” to hold admin’s email password

//Connection to the database

Create a private Connection connection;

Create a private ServerSocket serverSocket;

Create a final static integer PORT is 8000 for connection to the port number

Create Statement statement to execute SQL commands

Create a private TextArea “jtasqlCommand” to enter SQL command

Create a private TextArea “jtaSQLResult” to display results from SQL commands

//JDBC info for a database connection

Create a private TextField “jtfUsername” to fill a username for database

Create a private PasswordField “jpfPassword” to fill a password for database

//Buttons

Create a private Button “jbtExecuteSQL” with label “Execute SQL Command"

Create a private Button “jbtClearSQLCommand” with label “Clear"

Create a private Button “jbtConnectDB1” with label "Connect to Database"

Create a private Button “jbtClearSQLResult” with label "Clear Result"

//Create titled

Create a private Label “titledBorder1” with setting text is "Enter an SQL Command"

Create a private Label “titledBorder2” with setting text is “SQL Execution Result"

Create a private Label “titledBorder3” with setting text is “Enter Database Information”

//Labels

Create a private Label “lbUsername” with setting text is “Username:"

Create a private Label “lbPassword” with setting text is "Password: "

Create a private Label “lbConnectionStatus” with setting text is "Database Status:”

Create a private Label “jlblConnectionStatus” ” with setting text is "No connection now"

private Text txDbStatus = new Text("disconnected");

Display primary stage scene(1) with

THE LEFT SIDE THE STAGE to connect with database

Create VBox “paneDBInfo” and add with

Label “titledBorder3”,

label “lbUsername”,

jTextField “jtfUsername”,

label “lbPassword”,

jPasswordField “jpfPassword”,

jButton “jbtConnectDB1”

Label lbConnectionStatus,

and Text “txDbStatus”

THE RIGHT SIDE THE STAGE to input SQL command

Create VBox “paneSQLCommand” and add with

Label “titledBorder1”, and JTextArea “jtasqlCommand”

THE BELOW SQL COMMAND’S TEXT AREA

THE TOP

Create HBox “paneTop” and add with

VBox paneDBInfo and paneSQLCommand

THE CENTER

Create HBox “paneButtons” and add with

jButton jbtExecuteSQL, “jbtClearSQLCommand”, and “jbtClearSQLResult”

THE BOTTOM

Create VBox “paneSQLResult” and add with

Label “titledBorder2” and jTextArea “jtaSQLResult”

Create VBox “paneMain” and add with HBox paneTop,HBox paneButtons,VBox paneSQLResult

Listen for an initial action on the the jbtExecuteSQL button whenever it is fired, call the method executeSQL()

Listen for an initial action on the the jbtConnectDB1 button whenever it is fired, call the method connectToDB()

Listen for an initial action on the the jbtClearSQLCommand button whenever it is fired, set text in TextArea “jtasqlCommand” has no letter

Listen for an initial action on the the jbtClearSQLResult button whenever it is fired, set text in TextArea “jtaSQLResult” has no letter

State server create new thread

TRY

Create server socket with “PORT”

Set text in “jtaSQLResult” is "MultiThreadServer started at” and date&time started

WHILE is TRUE connented with client

Increment the number of client “clientCount”

Display jTextArea “jtaSQLResult” with “Starting thread for client” and counting number of clients

Displasy jTextArea “jtaSQLResult” with client’s host name and client’s IP address

CATCH

Print out the error in command prompt

NEW CLASS “HandleAClient” and implements Runnable for handling new connection

Create connection socket

Construct a thread

METHOD void run

TRY

Create data input “inputFromClient” and output “outputToClient” stream from clients

Create data input object “inputFromClientObj” and output object “outputToClientObj” from clients

WHILE is TRUE (Always serve the client)

SWITCH by input integer from client

CASE is 0 or login

Create a String “uN” is a username received a string from client

Create a String “pW” is a password received a string from client

Create String “sql” to be a sql command to get username and password for database where the username and password in the database are matched with received username and password from client

IF the executeSQLDataBaseCertainData() method was called and executed “sql” and return TRUE

IF the password “pW” is match with the password which gets from database and the username “uN” match with the username which gets from database

Output boolean to client TRUE

Output UserID as an object to client

Display jtaSQLResult “Logged In….”

Output Ranking list as an object to client

ELSE Output boolean to client FALSE

ELSE Output boolean to client FALSE

CASE is 1 or SIGN UP

SET Boolean “isSignUpSucceeded” is false

TRY

Receive UserID as an object from client

Create a String sql command “sql” is insert into an account table with the username, password, email and current time

IF “isSignUpSucceeded” is true

Output TRUE to client

Display in TextArea “jtaSQLResult” is “Signed up…”

ELSE

Output FALSE to client

CATCH

Output FALSE to client

CASE 2 or CHANGE\_PASSWORD

TRY

Receive UserID as an object from client

Create a String sql command update the table

Execute SQL

IF isSignUpSucceeded is true

Output TRUE to client

Display in TextArea “jtaSQLResult” is “Changed password…”

ELSE

Output FALSE to client

CATCH

Output FALSE to client

CASE 3 or FORGET\_PASSWORD

TRY

Create a string the hold the email with get form client

Call getUserByEmail() method to get user information with the same email

Create Mail class

Call function sendEmail() in Mail class

Output TRUE to client

CATCH

Output FALSE to client

CASE 3 or WIN

Get a String username from client

Get a user total score from client

Get UserID from database by calling getUser() method

Average the user score

Set average score in UserID

Set total match in UserID

Set total win in UserID

Create sql state to update database to store user information

Execute SQL

CASE 4 or LOSE

Get a String username from client

Get a user total score from client

Get user’s information “UserID” from database by calling getUser() method

Average the user score

Set average score in UserID

Set total match in UserID

Set total win in UserID

Create sql state to update database to store user information

Execute SQL

**END**