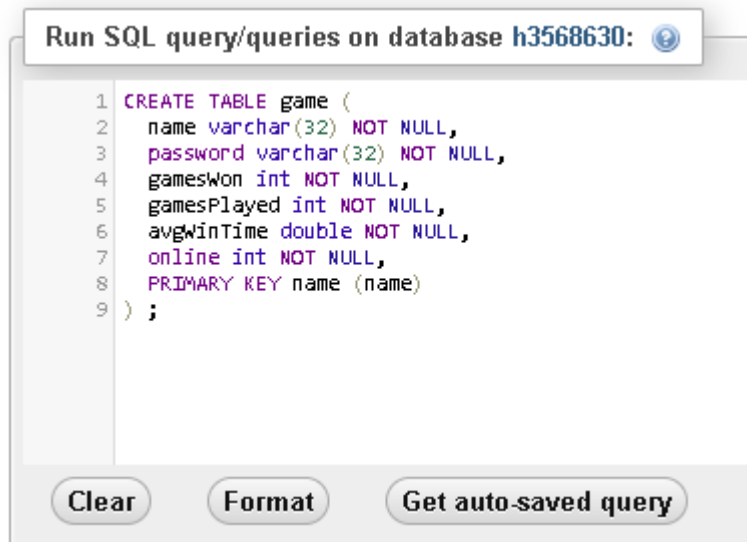


## Deployment requirements and steps

\* The RMI configurations should be followed first as in Assignment 1.

- Windows 10 64-bit platform
- JDK 1.8.0\_361
- MySQL from CS Department
- MySQL Connector J 8.0.33 (added to the Eclipse project's build path)
- HKUVPN for JDBC connection to CS MySQL account

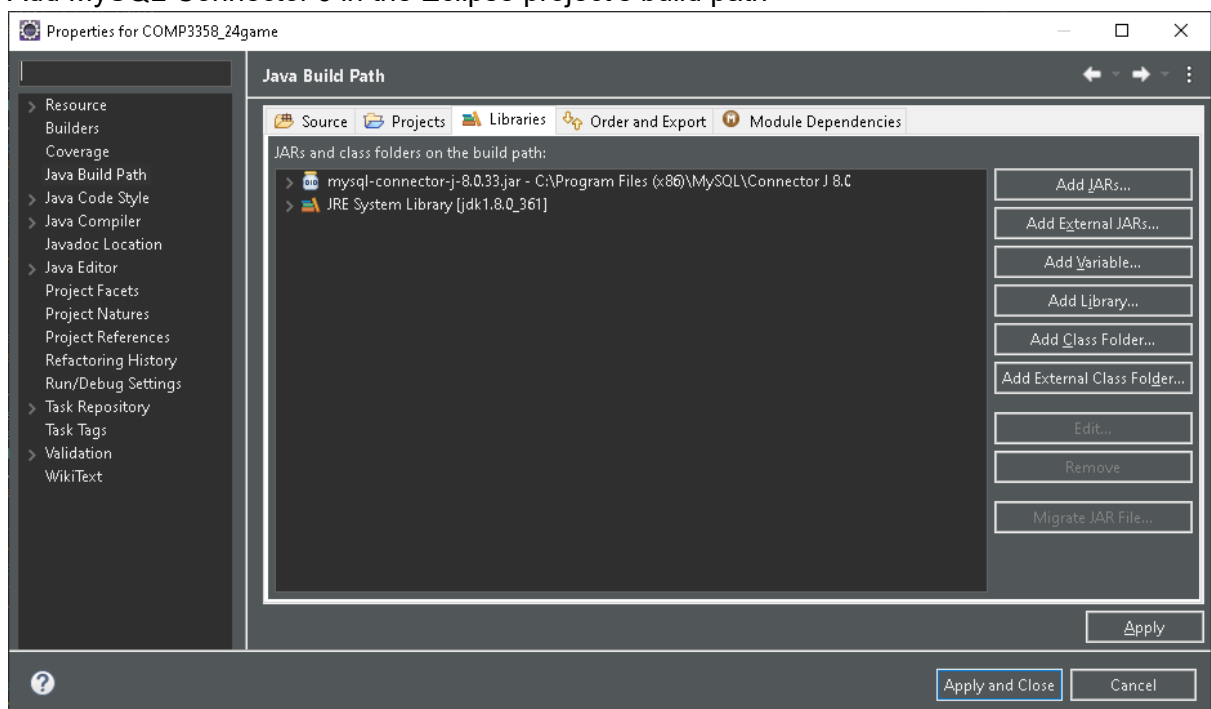
1. Create a table in MySQL via my CS MySQL account.



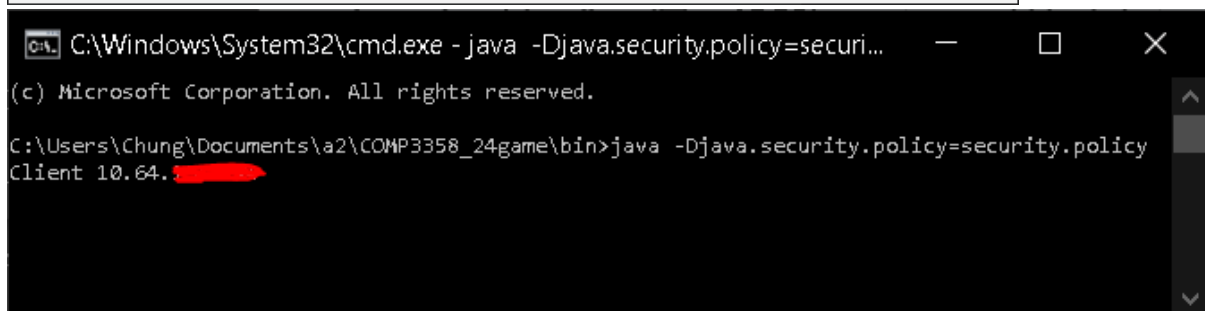
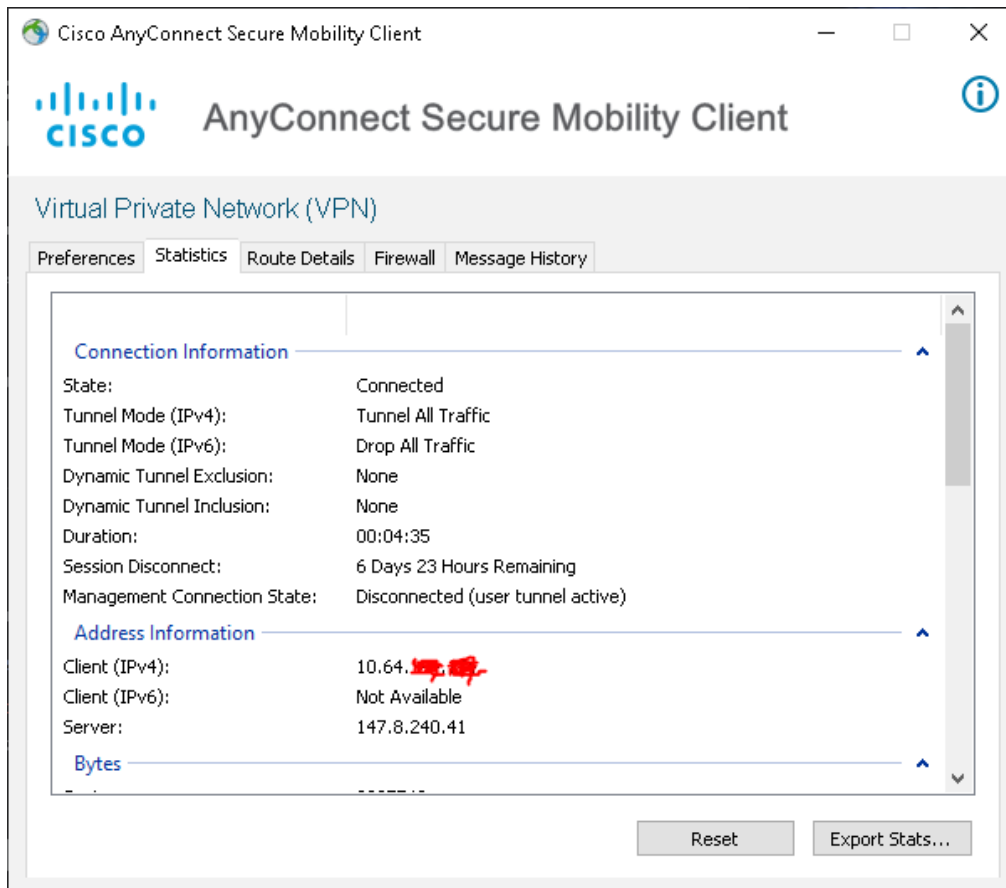
(The generated table)

name	password	gamesWon	gamesPlayed	avgWinTime	online
------	----------	----------	-------------	------------	--------

2. Add MySQL Connector J in the Eclipse project's build path



Then the server can run `rmiregistry.exe` followed by `Server.java`, and the clients can run `Client.java`. If the server is using HKUVPN (for JDBC to MySQL), the clients need to use HKUVPN as well and the server IP address should be its VPN address instead of its local IP address.



## Runtime screenshots

### Initial Table

Instead of using `UserInfo.txt` and `OnlineUser.txt` to keep track of players' data, a single table is used to store such data. This table acts as the base reference to see if there are any changes in the two files upon certain actions.

We start by inputting some sample data into the table. Note that all entries in the "online" column are set to 0 each time the server starts. The `online=1` of Player 7 is because I have logged in with a client already.

+ Options

</

## Login

The client can attempt to login by clicking the “Login” button or go to the register frame by clicking the “Register” button.

After clicking the “Login” button, the client can successfully login if and only if the name-password pair is found and its online flag is 0 in the table. Otherwise, an alert box with the associated error will be displayed. The server only updates the online flag to 1 on successful login, as shown in the below screenshots, where the right shows the content of the table as a result of the client’s login conditions shown in the left.

Login

Login na

Player 9

Password

...

Login

Register

Message

Incorrect login name or password.

OK

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	0
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

ected:

 Edit
  Copy
  Delete
  Export

Login

Login na

Player 7

Password

...

Login

Register

Message

You have already logged in.

OK

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	0
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

ected:

 Edit
  Copy
  Delete
  Export

Login

Login name

Player 2

Password

...

Login

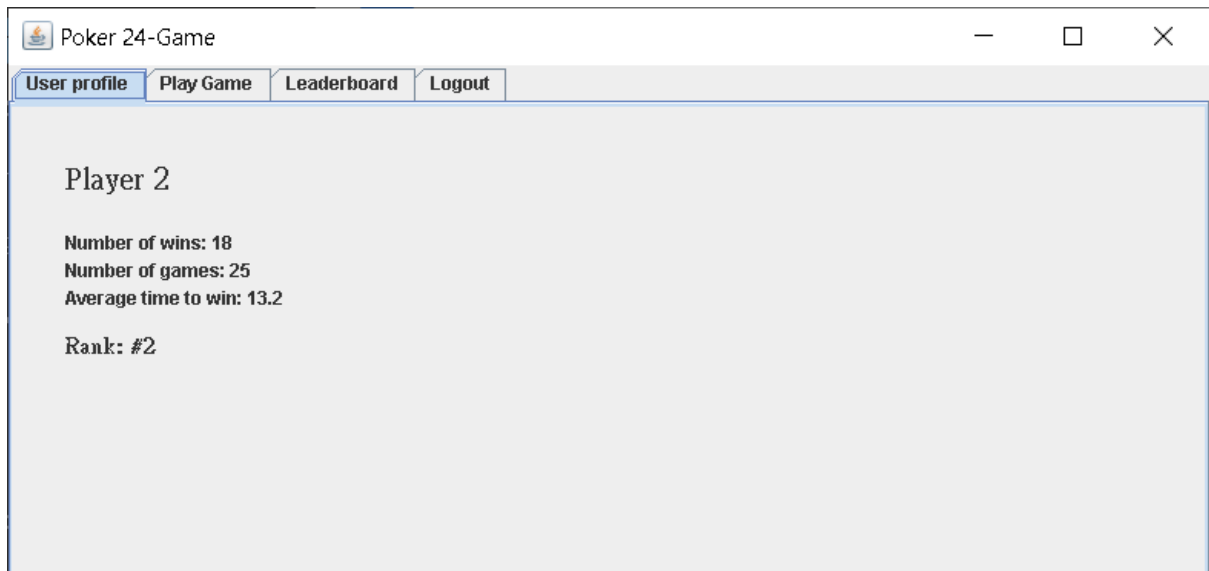
Register

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	1
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

ected:

 Edit
  Copy
  Delete
  Export

Upon successful login, the client can view tabs like user profile and leaderboard.

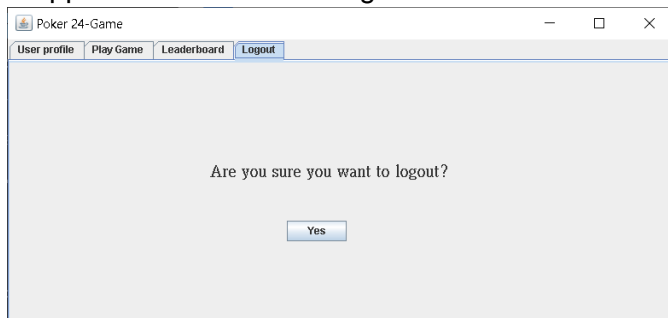


The screenshot shows the same window with the "Leaderboard" tab selected. It displays a table with the following data:

Rank	Player	Games won	Games played	Avg. winning time
1	Player 4	20	35	10.4
2	Player 2	18	25	13.2
3	Player 6	18	31	15.1
4	Player 8	16	30	12.8
5	Player 7	10	25	10.2

### Logout

Once the client clicks the button in the “Logout” tab or closes the window, the Login frame reappears and its online flag is set to 0 in the table.



Login

Login name

Player 2

Password

.....

Login

Register

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	0
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

Selected:

Edit

Copy

Delete

Export

## Registration

The client can attempt to register by clicking the “Register” button or go back to the login frame by clicking the “Login” button.

The server inserts a dataline to the table and only if the username is available, the password is not blank, and the password field matches the confirm password field. Otherwise, an alert box with the associated error will be displayed. The server updates UserInfo.txt on successful registration, where the right shows the content of the server’s UserInfo.txt as a result of the client’s register conditions shown in the left.

Register

Login name

Player 2

Message

The name has already been registered.

OK

Password

...

Confirm

...

Register

Cancel

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	0
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

Selected:

Edit

Copy

Delete

Export

Register

Login name

Player new

Message

Login name or password should not be empty.

OK

Password

...

Confirm

...

Register

Cancel

name	password	gamesWon	gamesPlayed	avgWinTime	online
Player 2	p2pw	18	25	13.2	0
Player 4	p4password	20	35	10.4	0
Player 6	p6pass	18	31	15.1	0
Player 7	p7word	10	25	10.2	1
Player 8	p8p8	16	30	12.8	0

Selected:

Edit

Copy

Delete

Export

Register

Login name

Player new

Message

Password and confirm password do not match.

OK

Password

...

Confirm

.....

Register

Cancel

Selected:

Edit

Copy

Delete

Export

Register

Login name

Player new

Message

Registration success.

OK

Password

...

Confirm

.....

Register

Cancel

Selected:

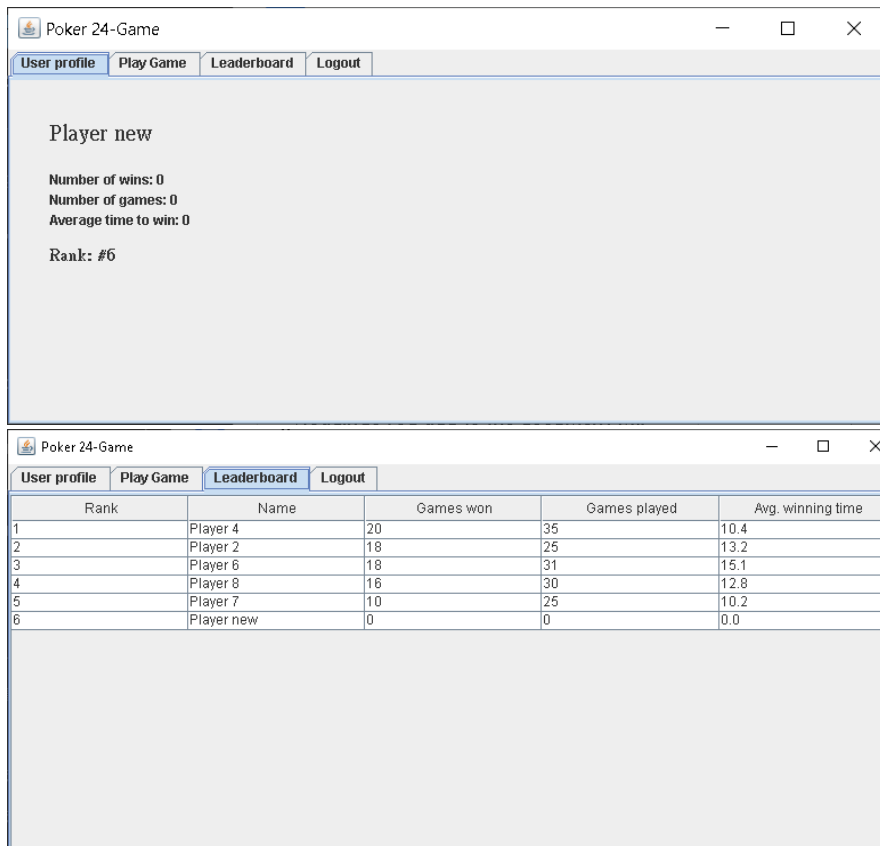
Edit

Copy

Delete

Export

The new registered user's data is appended to the table with its online flag set at 1. Like login, the client can view tabs like user profile and leaderboard upon successful registration. The server also appends to OnlineUser.txt.



## Code modification from Assignment 1

### Server.java

Implements a JDBC class for MySQL connection and replaces all file I/O operations from/to TableModel with JDBC.

Methods in the implemented JDBC class:

1. JDBC()
  - Constructor class called at the beginning of Server's main()
  - Load drivers and connect to MySQL.
2. void insert(String name, String password, int gamesWon, int gamesPlayed, double avgWinTime, int online)
  - Insert a dataline with given attributes to the table
  - Called after successful registration
3. void updateOnline(String name, int online)
  - Update the online flag to 1 for a specified player
  - Called after successful login or registration
4. void resetOnline()
  - Set all players' online flag to 0
  - Called at the beginning of Server's main()
5. boolean checkOnline(String name)
  - Check and return true if the player is online
  - Called during login
6. boolean checkName(String name)
  - Check and return true if the name is in the table
  - Called during registration

7. private boolean checkNamePassword(String name, String password)
  - Check and return true if the name-password pair is in the table
  - Called during login
8. JTable list()
  - Returns a JTable with columns rank, name, gamesWon, gamesPlayed, avgWinTime, with the rank column created within the query
  - Called after login for showing user profile and leaderboard, i.e. called when getTable() is called

#### Client.java

Minor changes of some TableModel.getValueAt() in user profile and leaderboard due to slight changes in the JTable format returned by getTable().