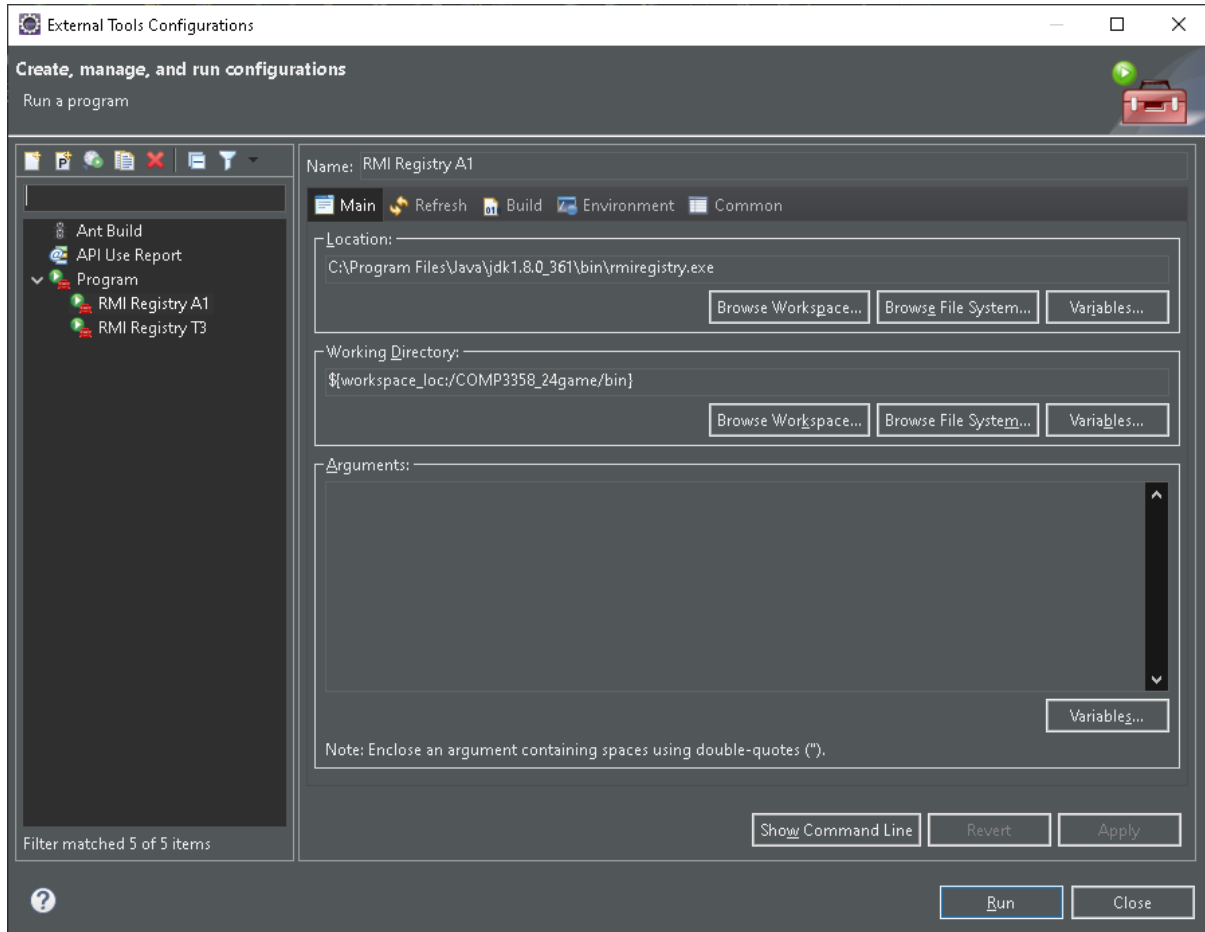
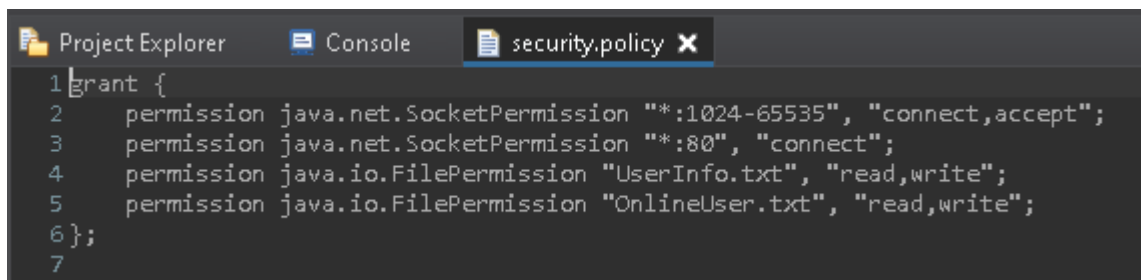


Deployment requirements and steps

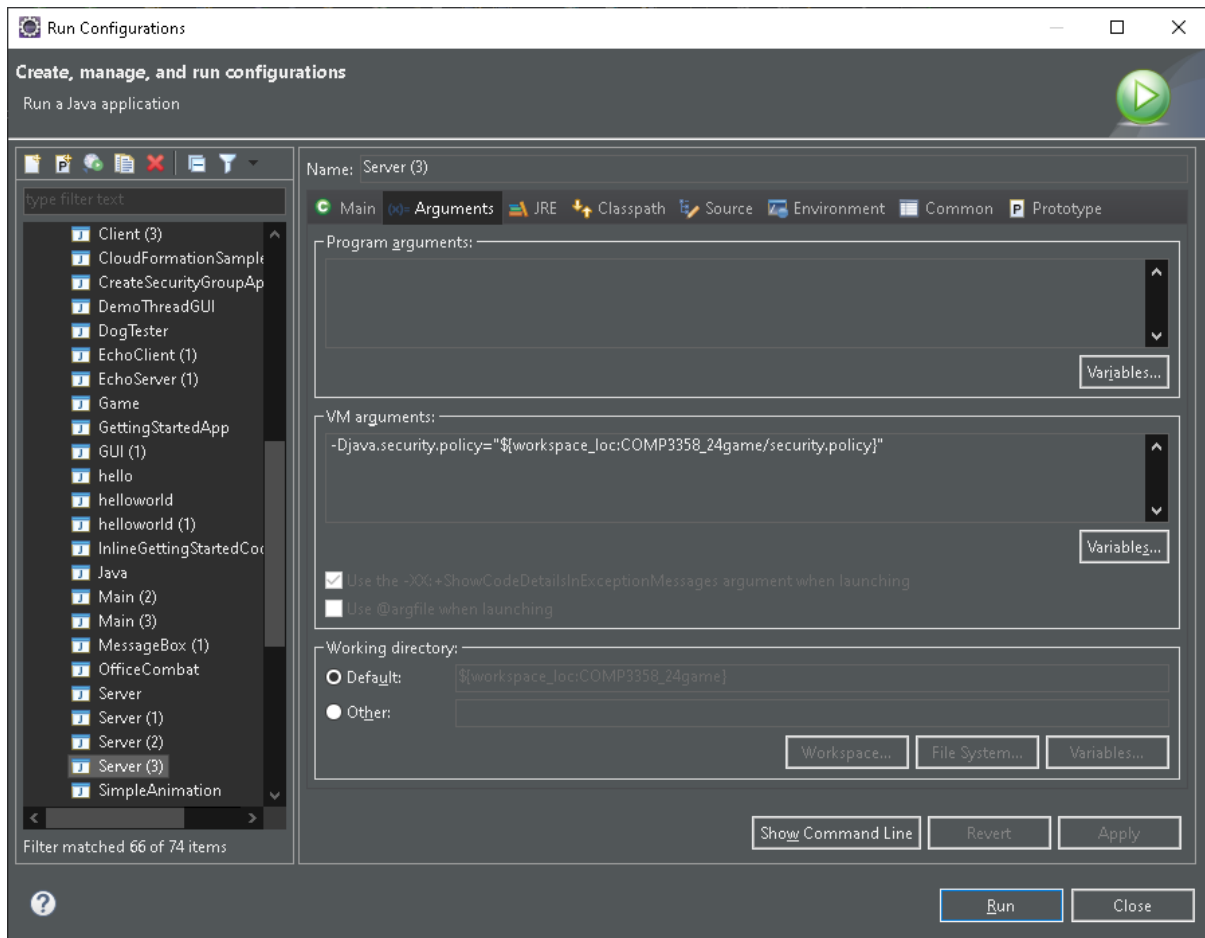
- Windows 10 64-bit platforms
 - JDK 1.8.0_361 on server machine
 - Java 8 on client machine
1. Configure RMI Registry in External Tools Configuration. The location is set to `rmiregistry.exe` in the JDK folder and the working directory is set to the bin folder of the project directory.



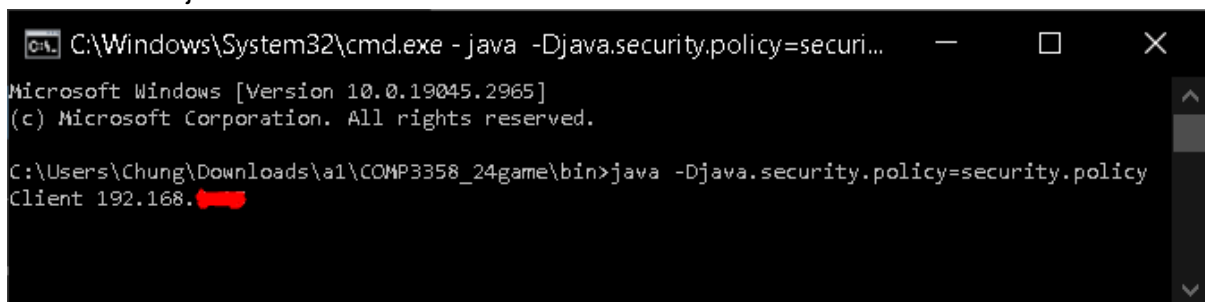
2. Produce a security.policy file to grant permission to connect and perform file I/O via RMI.



3. Add the security.policy to the VM arguments in Server.java.



4. Run `rmiregistry.exe` followed by `Server.java` in the server machine. Then run `Client.java` in the client machine as below.

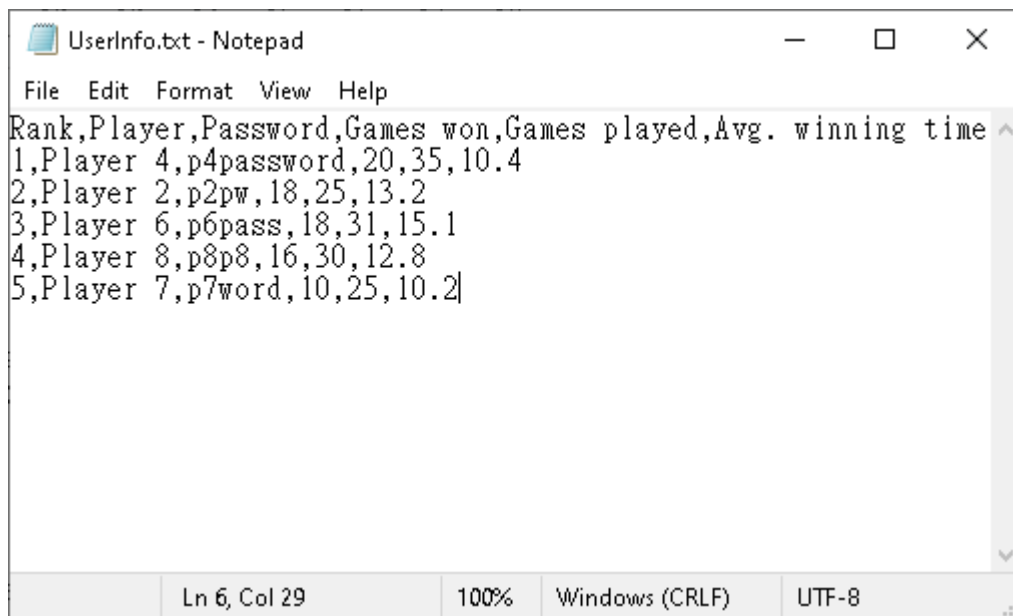


Runtime screenshots

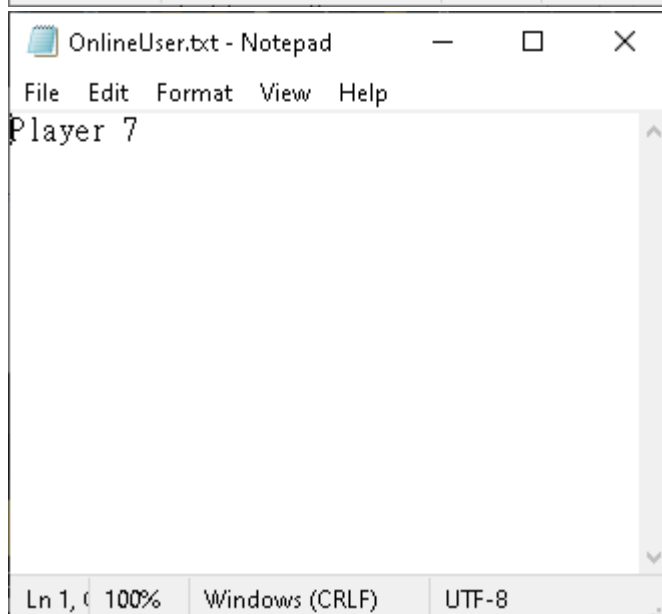
Initial UserInfo.txt and OnlineUser.txt

`UserInfo.txt` is in .csv format and `OnlineUser.txt` keeps track of each user in each line. These act as the base references to see if there are any changes in the two files upon certain actions.

Note that `OnlineUser.txt` should be cleared each time the server starts.



```
UserInfo.txt - Notepad
File Edit Format View Help
Rank,Player,Password,Games won,Games played,Avg. winning time
1,Player 4,p4password,20,35,10.4
2,Player 2,p2pw,18,25,13.2
3,Player 6,p6pass,18,31,15.1
4,Player 8,p8p8,16,30,12.8
5,Player 7,p7word,10,25,10.2
Ln 6, Col 29 100% Windows (CRLF) UTF-8
```

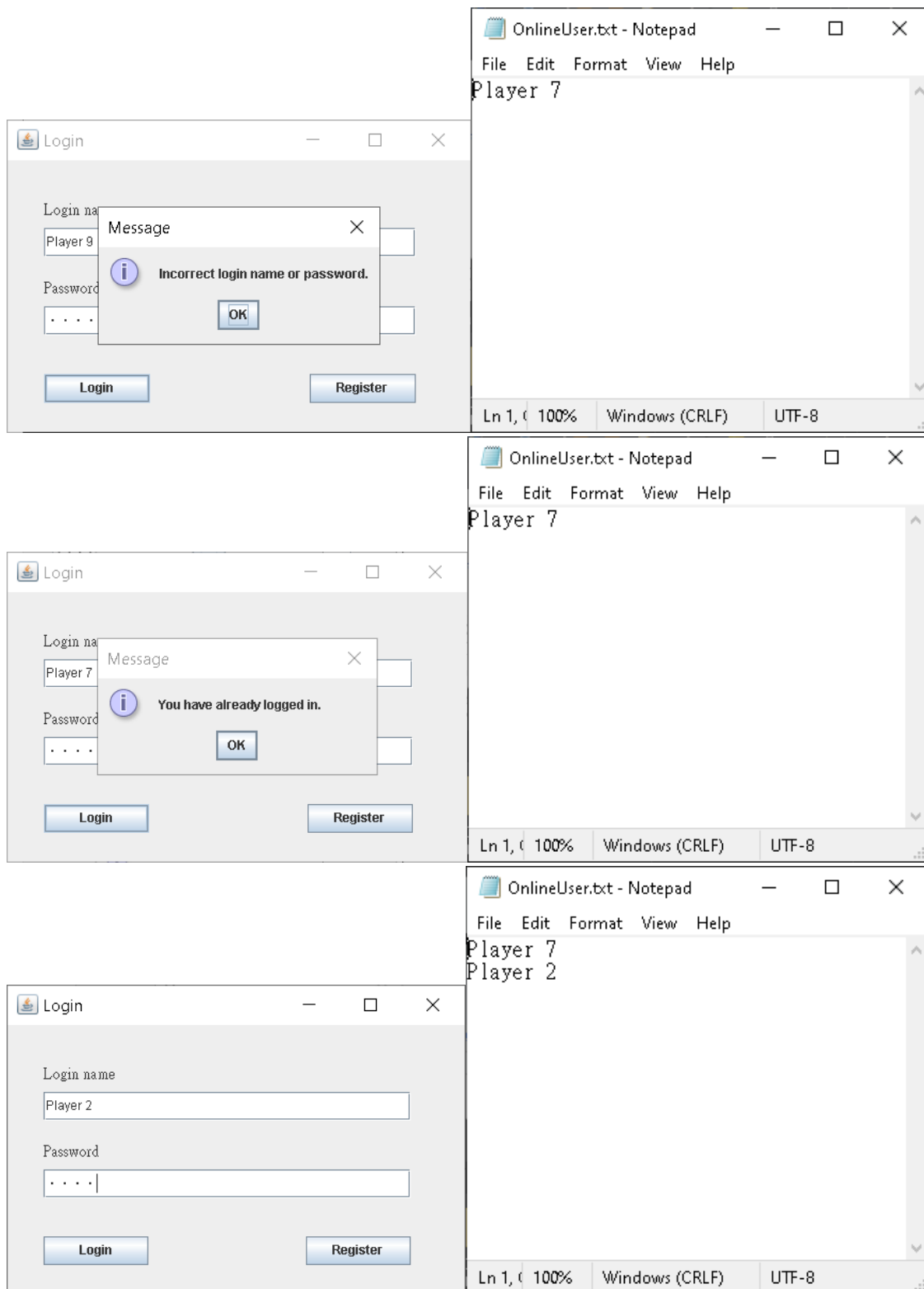


```
OnlineUser.txt - Notepad
File Edit Format View Help
Player 7
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

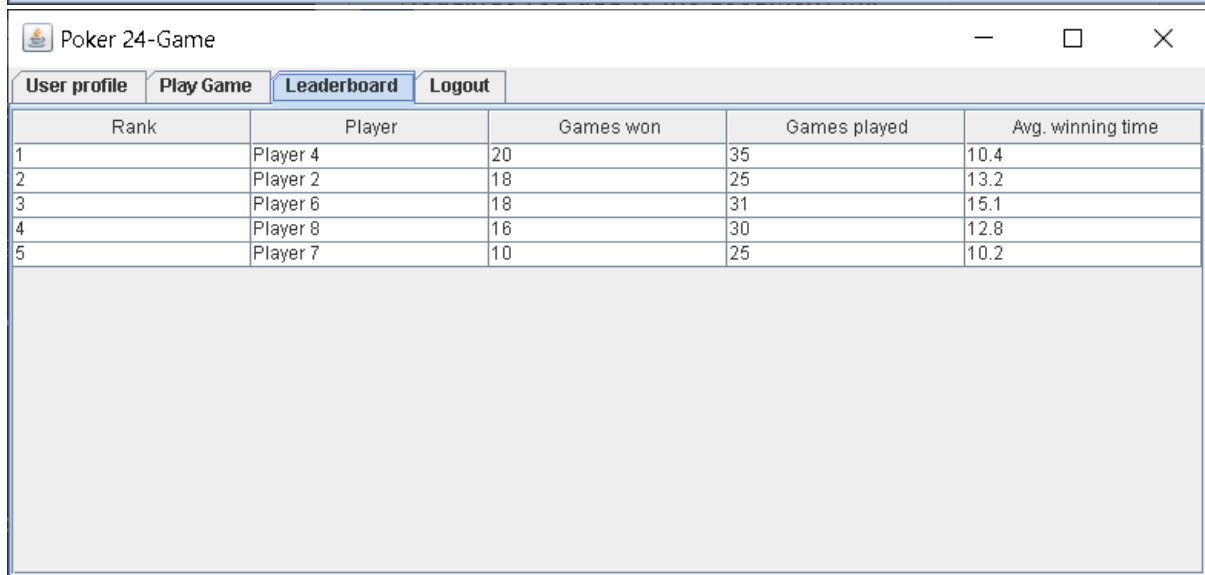
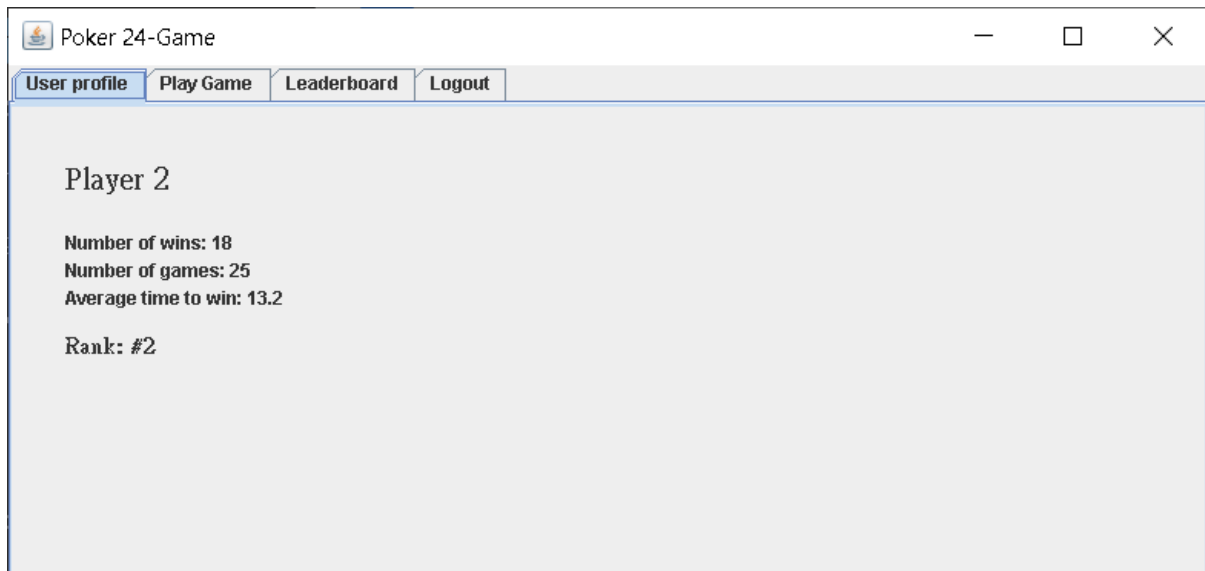
Login

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 in UserInfo.txt and its name is not in OnlineUser.txt. Otherwise, an alert box with the associated error will be displayed. The server only updates OnlineUser.txt on successful login, as shown in the below screenshots, where the right shows the content of the server’s OnlineUser.txt as a result of the client’s login conditions shown in the left.

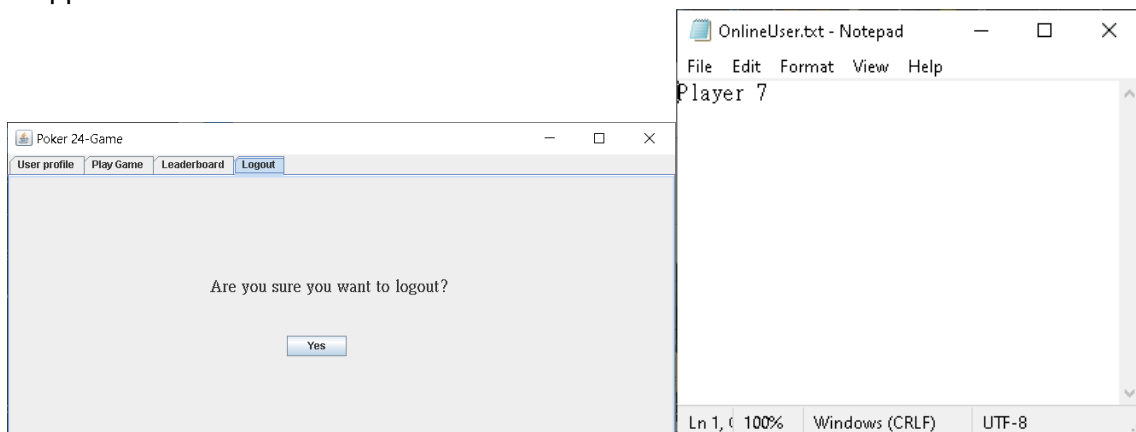


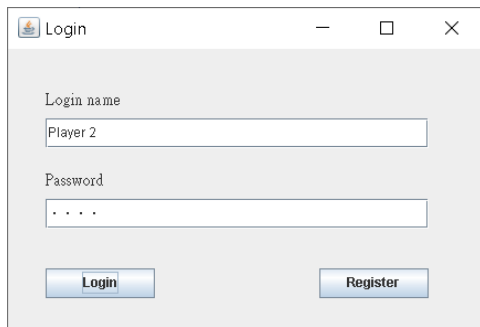
Upon successful login, the client can view tabs like user profile and leaderboard.



Logout

Once the client clicks the button in the "Logout" tab or closes the window, the Login frame reappears and the server will remove its username in OnlineUser.txt.





Login

Login name

Player 2

Password

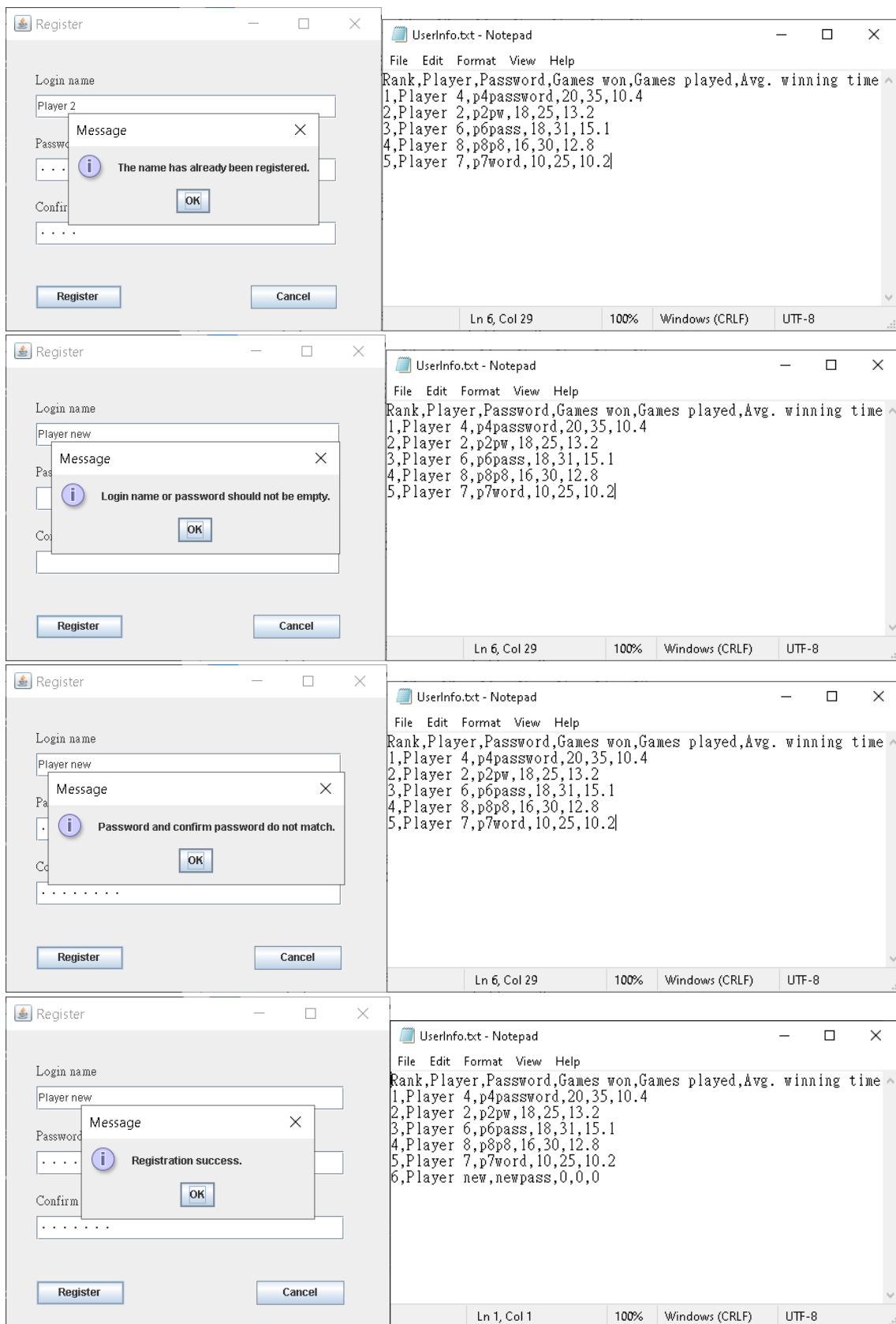
• • • •

Login Register

Registration

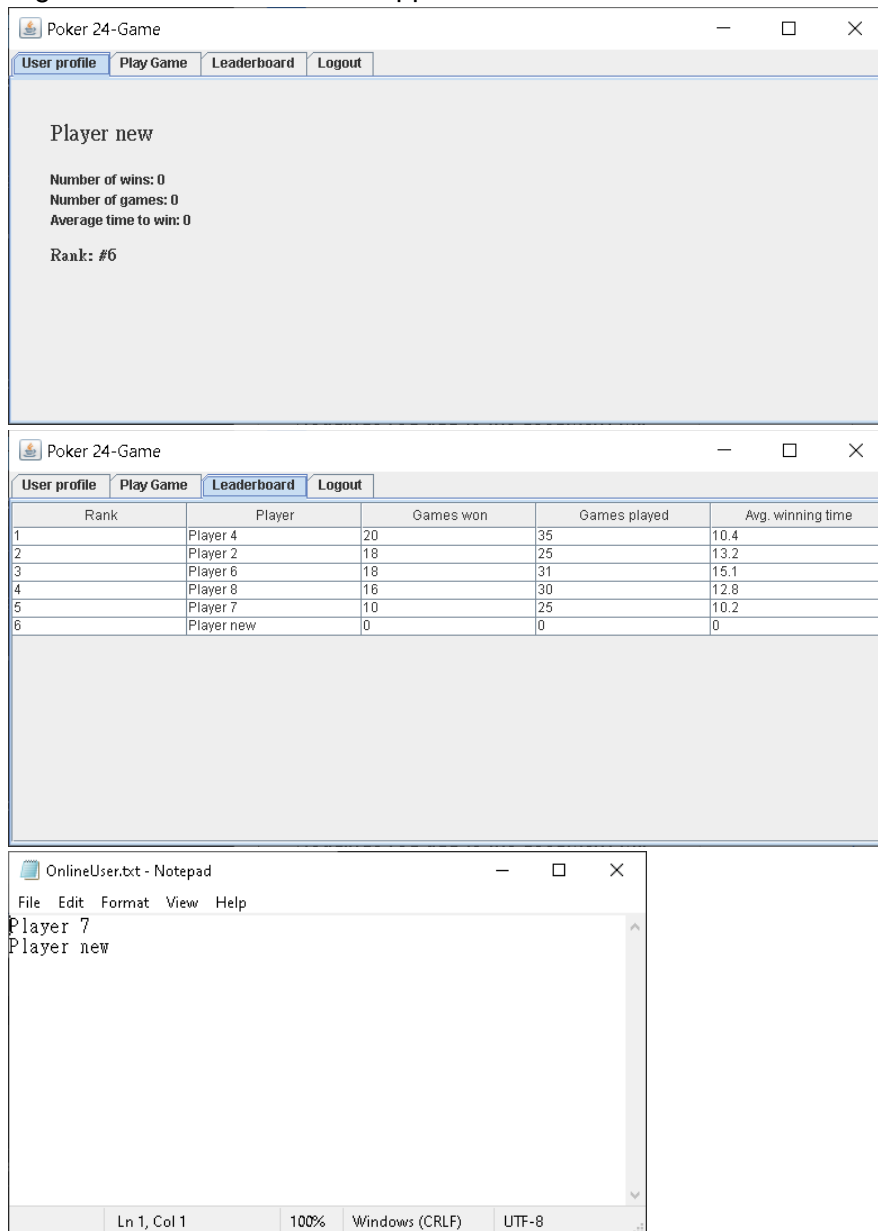
The client can attempt to register by clicking the “Register” button or go back to the login frame by clicking the “Login” button.

The server updates UserInfo.txt if 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.



The new registered user's data is appended to the last line in UserInfo.txt, with its rank being the last and other statistics being 0.

Like login, the client can view tabs like user profile and leaderboard upon successful registration. The server also appends to OnlineUser.txt.



Code description

Functions.java

An interface containing the login(), logout(), register() and getTable() for Server.java to implement.

Server.java

It binds to the RMI registry to let clients access the functions remotely, and implement the methods in Functions.java. It also empties OnlineUser.txt on start.

Implemented functions:

1. `int login(String name, String password)`
 - Reads UserInfo.txt and validates if the name-password pair exists in it.
 - On successful login, appends the name to OnlineUser.txt and returns 0.
 - Returns 1 if the name-password pair is not found in UserInfo.txt.

- Returns 2 if the name is found in OnlineUser.txt.
- Returns -1 upon exception.
- 2. `int register(String name, String password)`
 - Reads UserInfo.txt and checks if the name is already in it.
 - On successful login, appends the data of the new user (last rank, name, password, 0, 0, 0) to UserInfo.txt and returns 0.
 - Returns 1 if the name is on in UserInfo.txt.
 - Returns -1 upon exception.
- 3. `int logout(String name)` throws `RemoteException`;
 - Removes the name in OnlineUser.txt and returns 0
- 4. `JTable getTable()` throws `RemoteException`;
 - Read UserInfo.txt and convert it into a JTable.

Client.java

It lookups the functions from the RMI registry and maintain the GUI of 24-Game.

Frames and its contained elements:

1. Login
 - Name field and password field
 - "Login" to pass `login(name, password)`
 - "Register" button to show the Register frame
2. Register
 - Name field, password field and confirm password field
 - "Register" to pass `register(name, password)`
 - *Note that the empty name/password and password-confirm password match checking is done in the client's program before passing `login(name, password)`.
 - "Cancel" button to show the Login frame
3. Logged
 - Contains 4 tabs - User profile, Play game, Leaderboard, Logout
 - User profile shows the player's name, number of wins, number of games, average time to win and ranks. The data is retrieved by calling `getTable()`.
 - Leaderboard shows every player's rank, name, number of wins, number of games and average time to win. The data is retrieved by calling `getTable()`.
 - Logout shows a confirm button to let clients logout. After clicking it, it calls `logout()` and shows the Login frame.