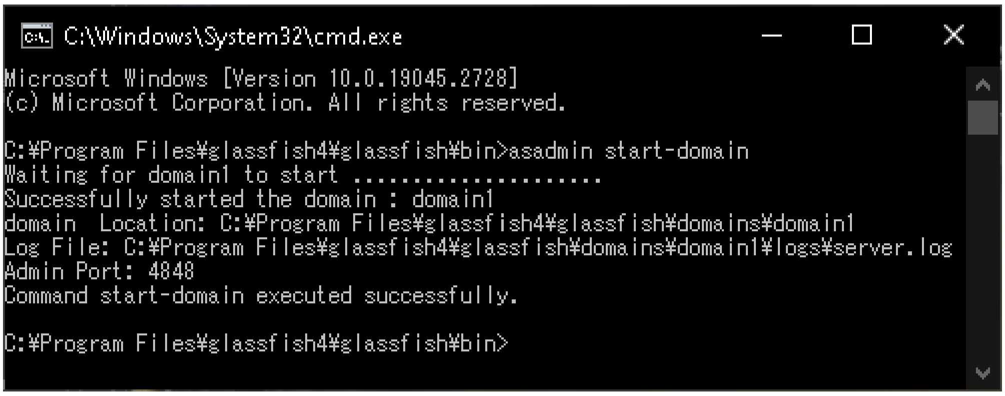
Deployment requirements and steps

*** The RMI & JDBC configurations should be followed first as in Assignment 1 & 2.**

- 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
- Glassfish 4.1.2 Full Platform

1. Start the Glassfish domain with "asadmin start-domain" if it has not be started: **(Note: If using HKUVPN, the domain should be started after the VPN connection has been established. If not, the domain should be stopped first with "asadmin stop-domain", and then restart it with "asadmin start-domain" when using the VPN.)**

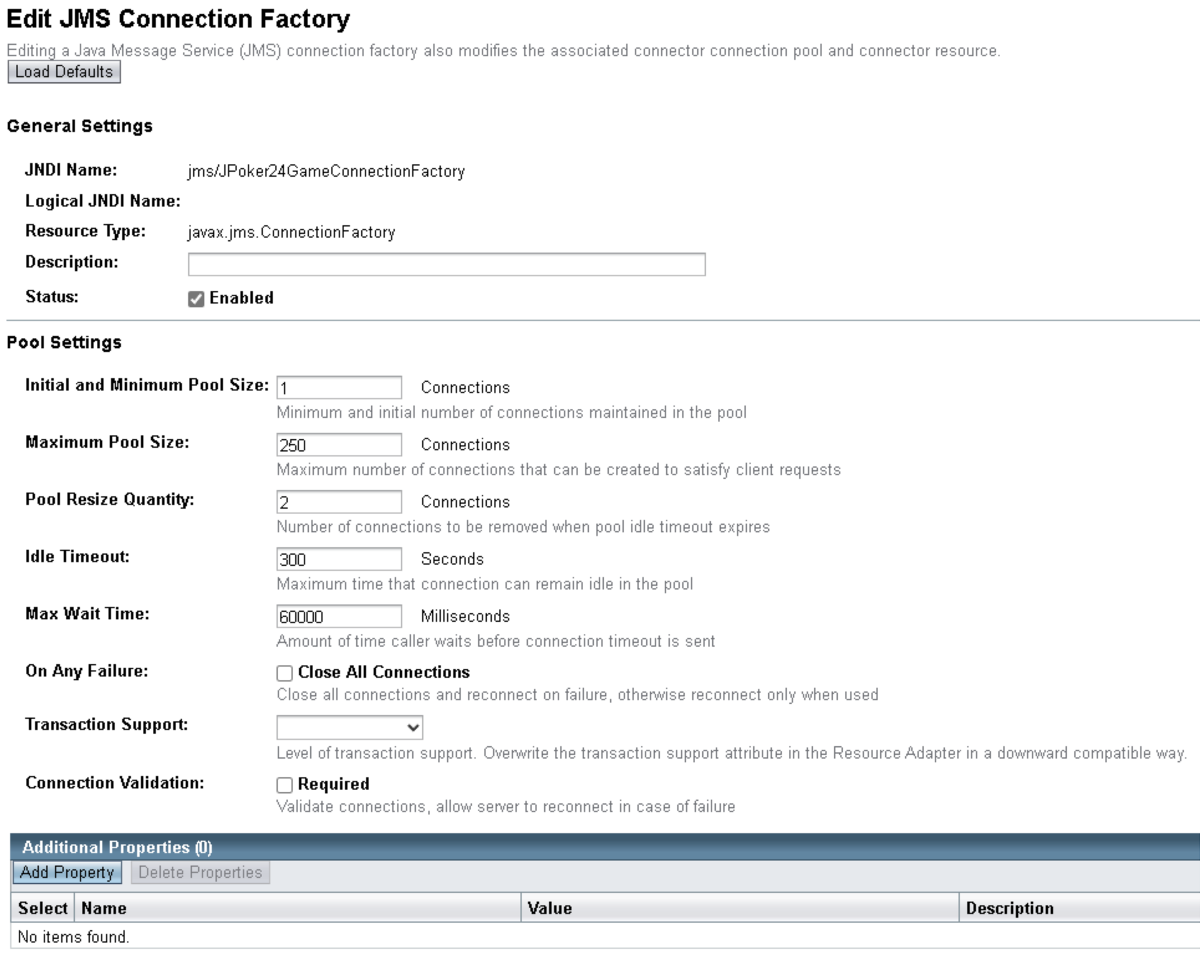


```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:¥Program Files¥glassfish4¥glassfish¥bin>asadmin start-domain
Waiting for domain1 to start ....................
Successfully started the domain : domain1
domain  Location: C:¥Program Files¥glassfish4¥glassfish¥domains¥domain1
Log File: C:¥Program Files¥glassfish4¥glassfish¥domains¥domain1¥logs¥server.log
Admin Port: 4848
Command start-domain executed successfully.

C:¥Program Files¥glassfish4¥glassfish¥bin>
```

2. Create a JMS Connection Factory with JNDI name "jms/JPoker24GameConnectionFactory" and Resource Type "javax.jms.ConnectionFactory" as follows.



Edit JMS Connection Factory

Editing a Java Message Service (JMS) connection factory also modifies the associated connector connection pool and connector resource.

General Settings

JNDI Name: jms/JPoker24GameConnectionFactory

Logical JNDI Name:

Resource Type: javax.jms.ConnectionFactory

Description:

Status: ☑ Enabled

Pool Settings

Initial and Minimum Pool Size: 1 Connections
Minimum and initial number of connections maintained in the pool

Maximum Pool Size: 250 Connections
Maximum number of connections that can be created to satisfy client requests

Pool Resize Quantity: 2 Connections
Number of connections to be removed when pool idle timeout expires

Idle Timeout: 300 Seconds
Maximum time that connection can remain idle in the pool

Max Wait Time: 60000 Milliseconds
Amount of time caller waits before connection timeout is sent

On Any Failure: □ Close All Connections
Close all connections and reconnect on failure, otherwise reconnect only when used

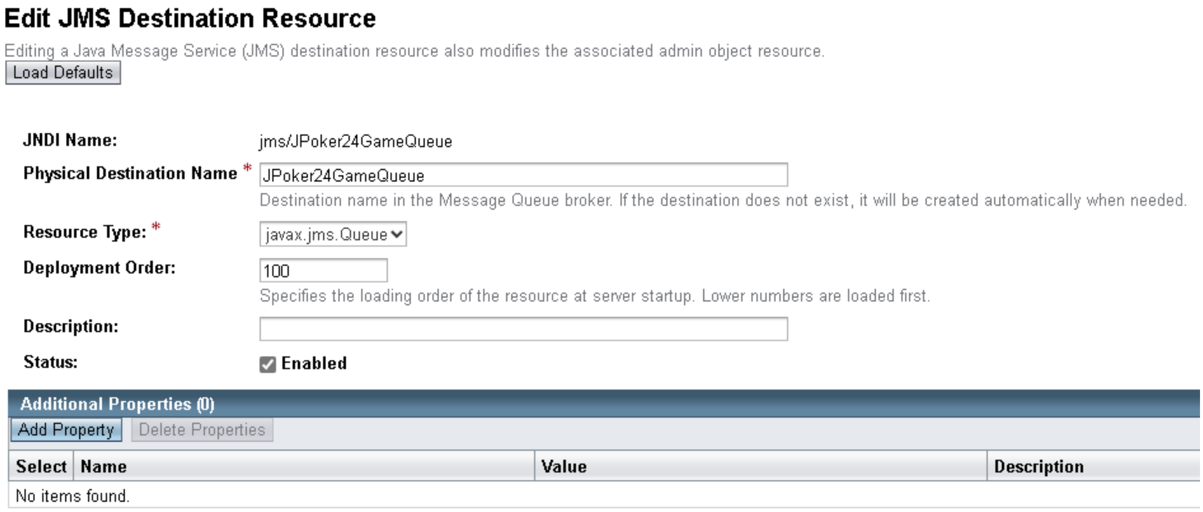
Transaction Support:
Level of transaction support. Overwrite the transaction support attribute in the Resource Adapter in a downward compatible way.

Connection Validation: □ Required
Validate connections, allow server to reconnect in case of failure

Additional Properties (0)

Select	Name	Value	Description
No items found.			

3. Create JPoker24GameQueue and JPoker24GameTopic.



Edit JMS Destination Resource

Editing a Java Message Service (JMS) destination resource also modifies the associated admin object resource.

JNDI Name: jms/JPoker24GameQueue

Physical Destination Name * JPoker24GameQueue
Destination name in the Message Queue broker. If the destination does not exist, it will be created automatically when needed.

Resource Type: * javax.jms.Queue

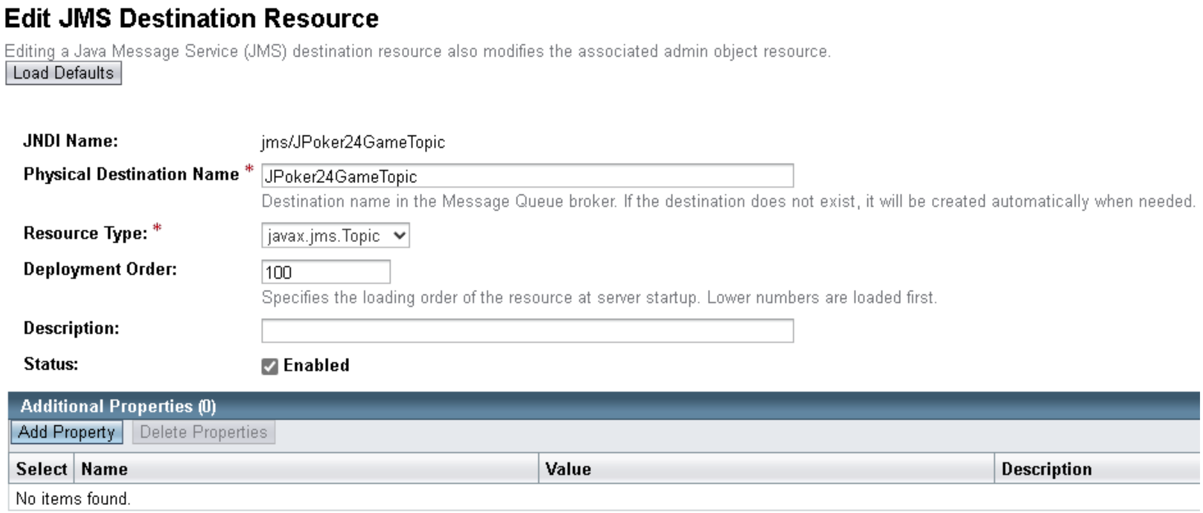
Deployment Order: 100
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Status: ☑ Enabled

Additional Properties (0)

Select	Name	Value	Description
No items found.			



Edit JMS Destination Resource

Editing a Java Message Service (JMS) destination resource also modifies the associated admin object resource.

Load Defaults

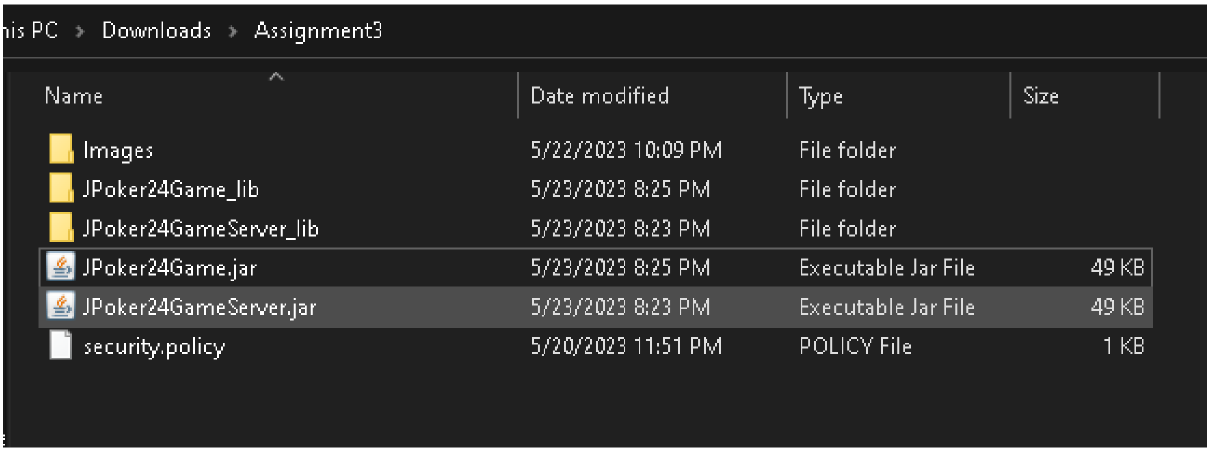
JNDI Name:	jms/JPoker24GameTopic
Physical Destination Name *	JPoker24GameTopic Destination name in the Message Queue broker. If the destination does not exist, it will be created automatically when needed.
Resource Type: *	javax.jms.Topic
Deployment Order:	100 Specifies the loading order of the resource at server startup. Lower numbers are loaded first.
Description:	
Status:	☑ Enabled

Additional Properties (0)

Add Property · Delete Properties

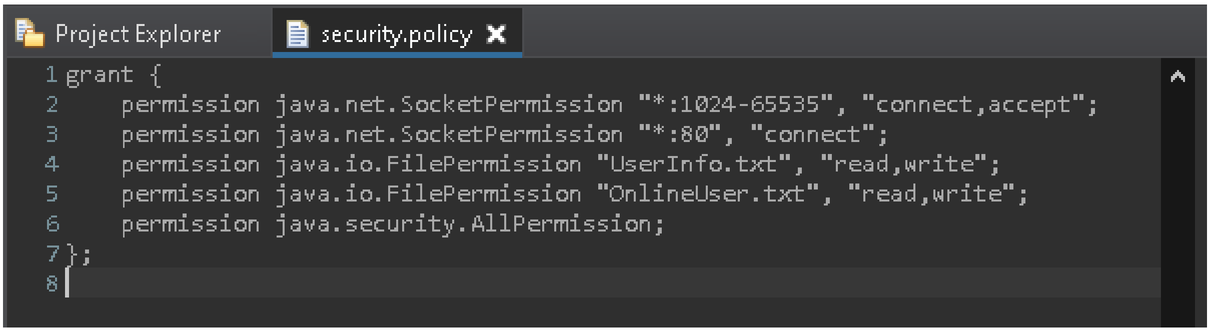
Select	Name	Value	Description
No items found.			

4. Extract the provided files. Make sure that JPoker24Game_lib, JPoker24GameServer_lib, Images folders and security.policy are kept in the same folder as Poker24Game.jar and Poker24GameServer.jar. The directory should look like this.



Name	Date modified	Type	Size
Images	5/22/2023 10:09 PM	File folder	
JPoker24Game_lib	5/23/2023 8:25 PM	File folder	
JPoker24GameServer_lib	5/23/2023 8:23 PM	File folder	
JPoker24Game.jar	5/23/2023 8:25 PM	Executable Jar File	49 KB
JPoker24GameServer.jar	5/23/2023 8:23 PM	Executable Jar File	49 KB
security.policy	5/20/2023 11:51 PM	POLICY File	1 KB

(Note that the security.policy is modified to grant all permissions).



```
grant {
    permission java.net.SocketPermission "*:1024-65535", "connect,accept";
    permission java.net.SocketPermission "*:80", "connect";
    permission java.io.FilePermission "UserInfo.txt", "read,write";
    permission java.io.FilePermission "OnlineUser.txt", "read,write";
    permission java.security.AllPermission;
};

```

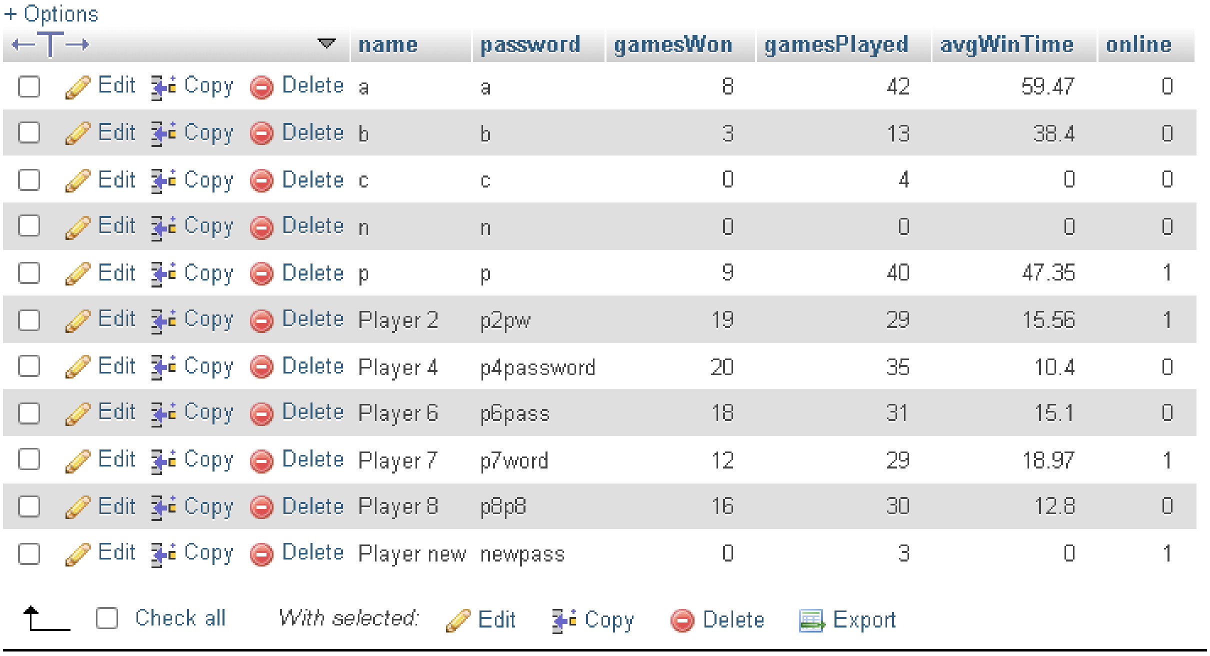
5. After running the RMI registry, Run “java -Djava.security.policy=security.policy -jar JPoker24GameServer.jar” to start the server. For clients, run “java -Djava.security.policy=security.policy -jar JPoker24Game.jar <IP address>”.

*** A zipped Eclipse export is also provided as source codes.**

***Keep the Images folder and security.policy in the same folder as Poker24Game.jar and Poker24GameServer.jar**

Runtime screenshots

<u>Initial Data</u>

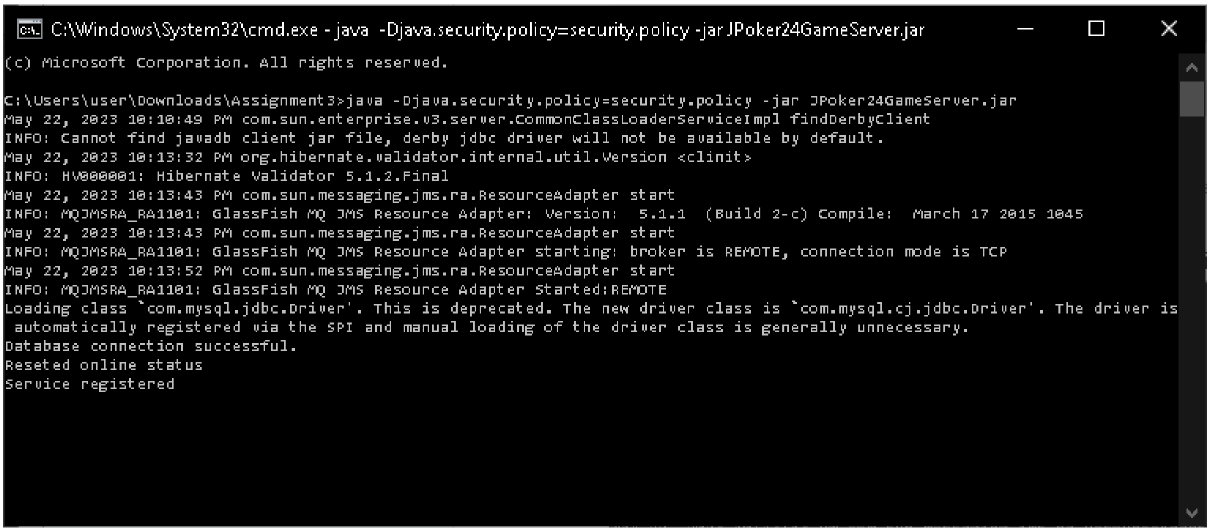


+ Options

				name	password	gamesWon	gamesPlayed	avgWinTime	online
☐	Edit	Copy	Delete	a	a	8	42	59.47	0
☐	Edit	Copy	Delete	b	b	3	13	38.4	0
☐	Edit	Copy	Delete	c	c	0	4	0	0
☐	Edit	Copy	Delete	n	n	0	0	0	0
☐	Edit	Copy	Delete	p	p	9	40	47.35	1
☐	Edit	Copy	Delete	Player 2	p2pw	19	29	15.56	1
☐	Edit	Copy	Delete	Player 4	p4password	20	35	10.4	0
☐	Edit	Copy	Delete	Player 6	p6pass	18	31	15.1	0
☐	Edit	Copy	Delete	Player 7	p7word	12	29	18.97	1
☐	Edit	Copy	Delete	Player 8	p8p8	16	30	12.8	0
☐	Edit	Copy	Delete	Player new	newpass	0	3	0	1

☐ Check all *With selected:* Edit Copy Delete Export

First the server is started. The “Service registered” line indicates the server has successfully started.

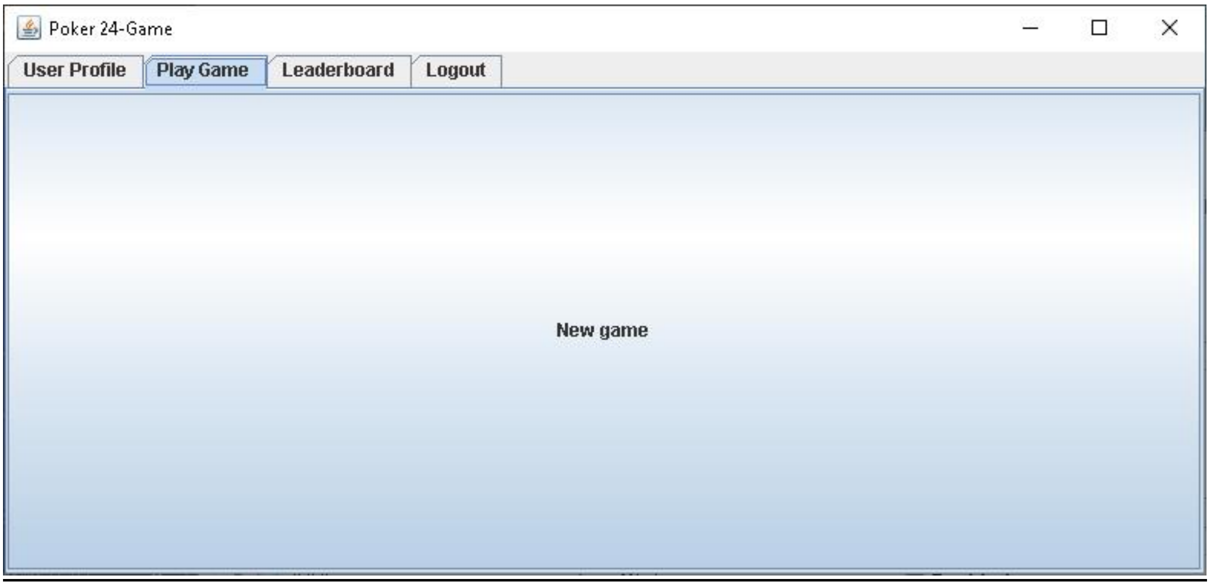


C:\Windows\System32\cmd.exe - java -Djava.security.policy=security.policy -jar JPoker24GameServer.jar

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Downloads\Assignment3>java -Djava.security.policy=security.policy -jar JPoker24GameServer.jar
May 22, 2023 10:10:49 PM com.sun.enterprise.v3.server.CommonClassLoaderServiceImpl findDerbyClient
INFO: Cannot find javadb client jar file, derby jdbc driver will not be available by default.
May 22, 2023 10:13:32 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.1.2.Final
May 22, 2023 10:13:43 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version:  5.1.1  (Build 2-c) Compile:  March 17 2015 1045
May 22, 2023 10:13:43 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
May 22, 2023 10:13:52 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is
 automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
Database connection successful.
Reseted online status
Service registered
```

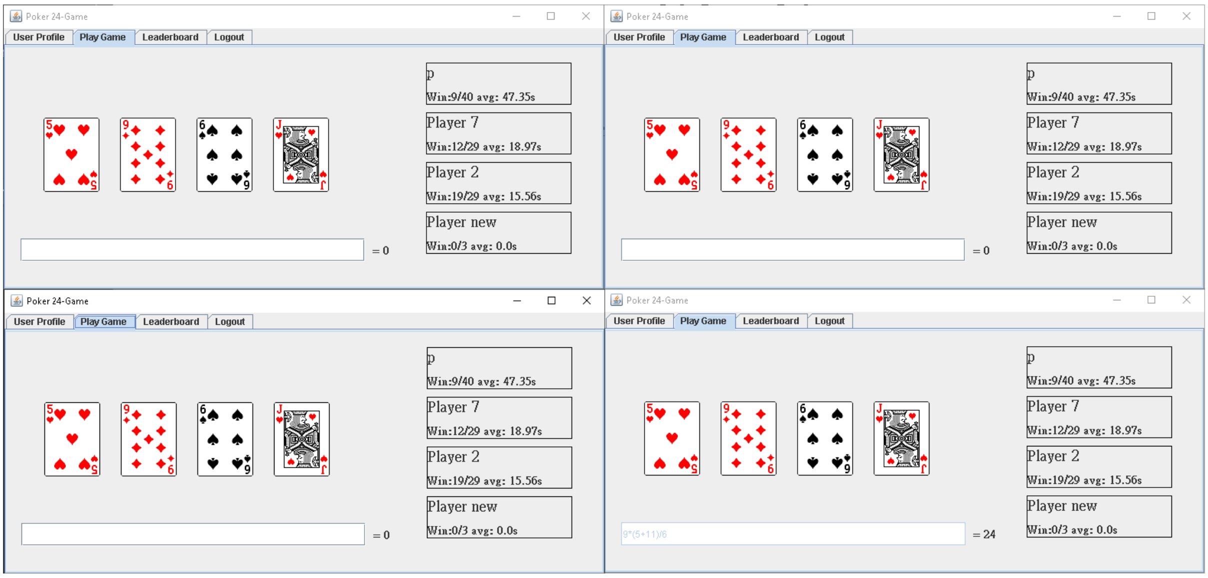
Players, after logging in, can press the new game button in the play game tab.



.

After pressing the button, it sends a message to the server and waits for other players. The server starts a game and informs the players of the players in their room and their hands. Here a 4-player game is started.

The players can see the cards and their opponents and their game statistics. They can then type the answer and submit by pressing enter. The calculated result is shown on the right.



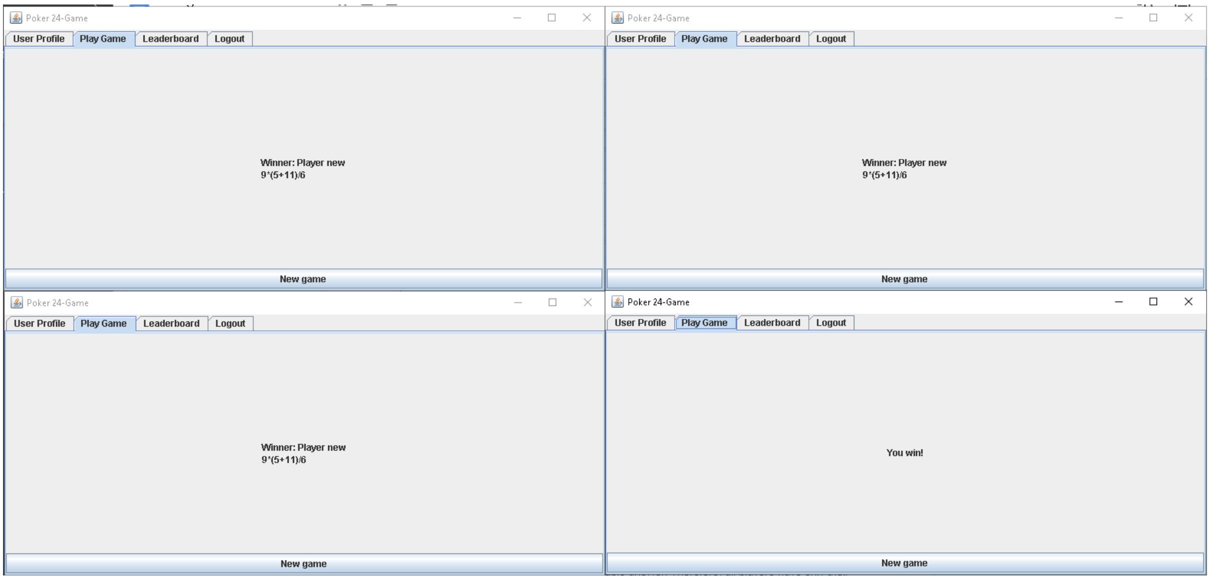
p
Win:9/40 avg: 47.35s
Player 7
Win:12/29 avg: 18.97s
Player 2
Win:19/29 avg: 15.56s
Player new
Win:0/3 avg: 0.0s
= 0

p
Win:9/40 avg: 47.35s
Player 7
Win:12/29 avg: 18.97s
Player 2
Win:19/29 avg: 15.56s
Player new
Win:0/3 avg: 0.0s
= 0

p
Win:9/40 avg: 47.35s
Player 7
Win:12/29 avg: 18.97s
Player 2
Win:19/29 avg: 15.56s
Player new
Win:0/3 avg: 0.0s
= 0

p
Win:9/40 avg: 47.35s
Player 7
Win:12/29 avg: 18.97s
Player 2
Win:19/29 avg: 15.56s
Player new
Win:0/3 avg: 0.0s
9*(5+11)/6 = 24

The winner receives a winning message and others receive the winner’s answer.



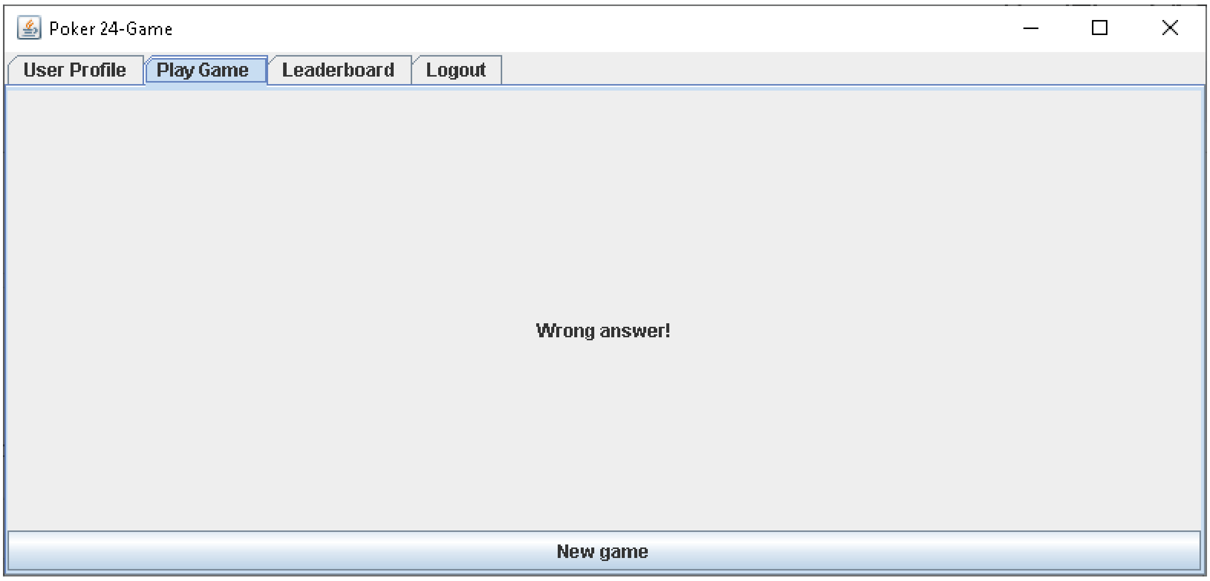
Winner: Player new
9*(5+11)/6

Winner: Player new
9*(5+11)/6

Winner: Player new
9*(5+11)/6

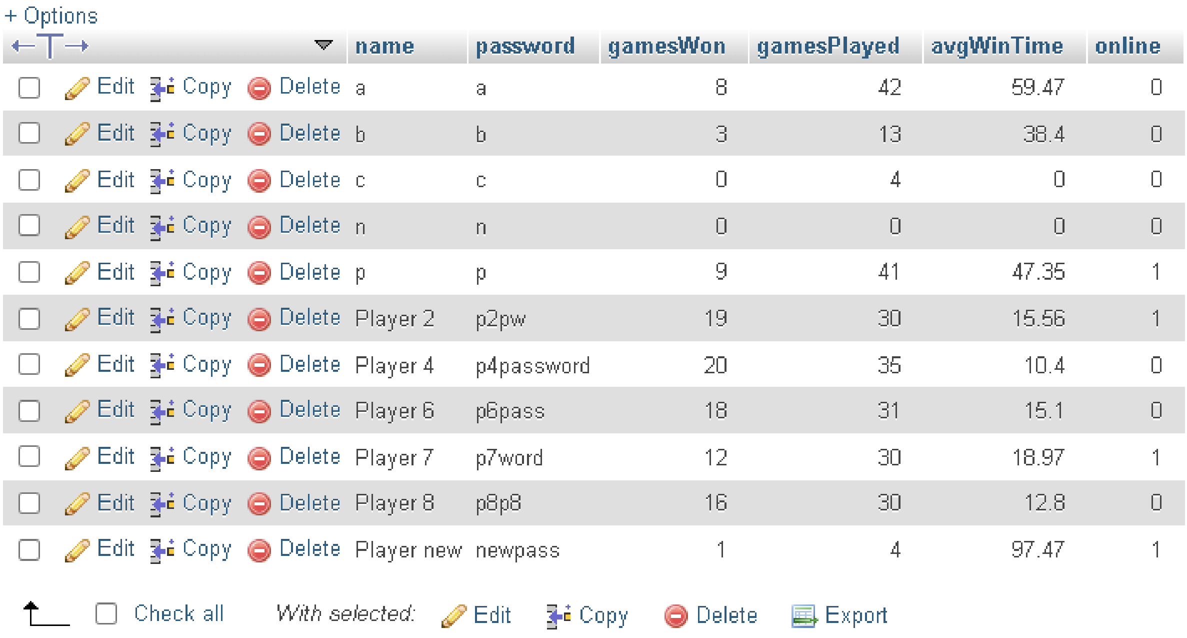
You win!

(It shows the wrong answer if the answer is not valid. Note that the player always leaves the room after submitting an answer. This implementation is a resolution to players getting stuck when there is no possible valid answer.)



Wrong answer!

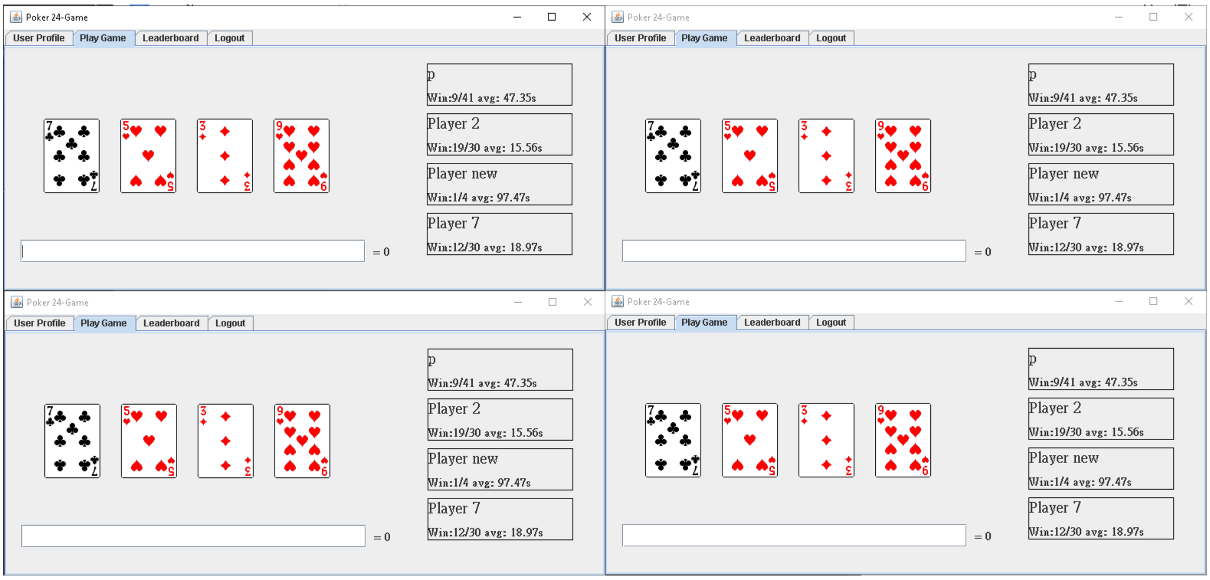
The table is then updated and players can see their updated statistics in the app.



+ Options

	name	password	gamesWon	gamesPlayed	avgWinTime	online
□ Edit Copy Delete	a	a	8	42	59.47	0
□ Edit Copy Delete	b	b	3	13	38.4	0
□ Edit Copy Delete	c	c	0	4	0	0
□ Edit Copy Delete	n	n	0	0	0	0
□ Edit Copy Delete	p	p	9	41	47.35	1
□ Edit Copy Delete	Player 2	p2pw	19	30	15.56	1
□ Edit Copy Delete	Player 4	p4password	20	35	10.4	0
□ Edit Copy Delete	Player 6	p6pass	18	31	15.1	0
□ Edit Copy Delete	Player 7	p7word	12	30	18.97	1
□ Edit Copy Delete	Player 8	p8p8	16	30	12.8	0
□ Edit Copy Delete	Player new	newpass	1	4	97.47	1

The players can then start another game by pressing the new game button. The new game shows updated player statistics.



p
Win:9/41 avg: 47.35s

Player 2
Win:19/30 avg: 15.56s

Player new
Win:1/4 avg: 97.47s

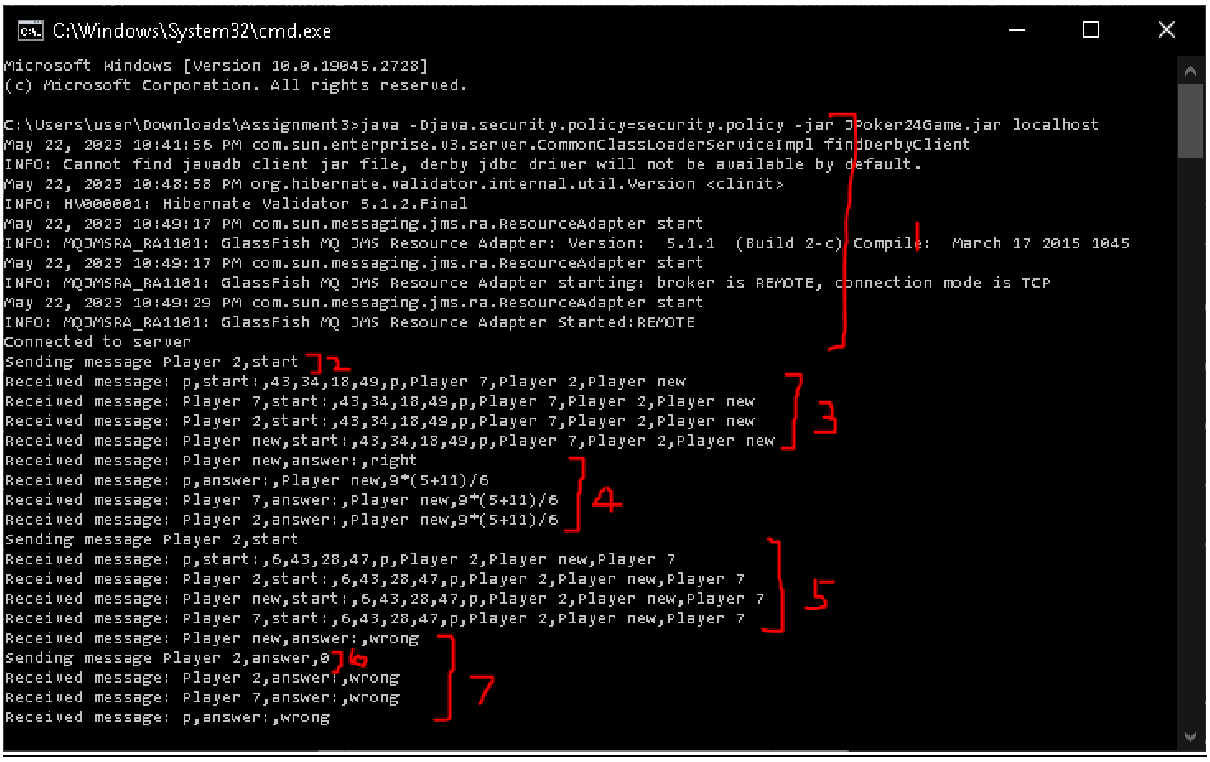
Player 7
Win:12/30 avg: 18.97s

p
Win:9/41 avg: 47.35s
Player 2
Win:19/30 avg: 15.56s
Player new
Win:1/4 avg: 97.47s
Player 7
Win:12/30 avg: 18.97s

p
Win:9/41 avg: 47.35s
Player 2
Win:19/30 avg: 15.56s
Player new
Win:1/4 avg: 97.47s
Player 7
Win:12/30 avg: 18.97s

p
Win:9/41 avg: 47.35s
Player 2
Win:19/30 avg: 15.56s
Player new
Win:1/4 avg: 97.47s
Player 7
Win:12/30 avg: 18.97s

Console view of clients (Player 2)

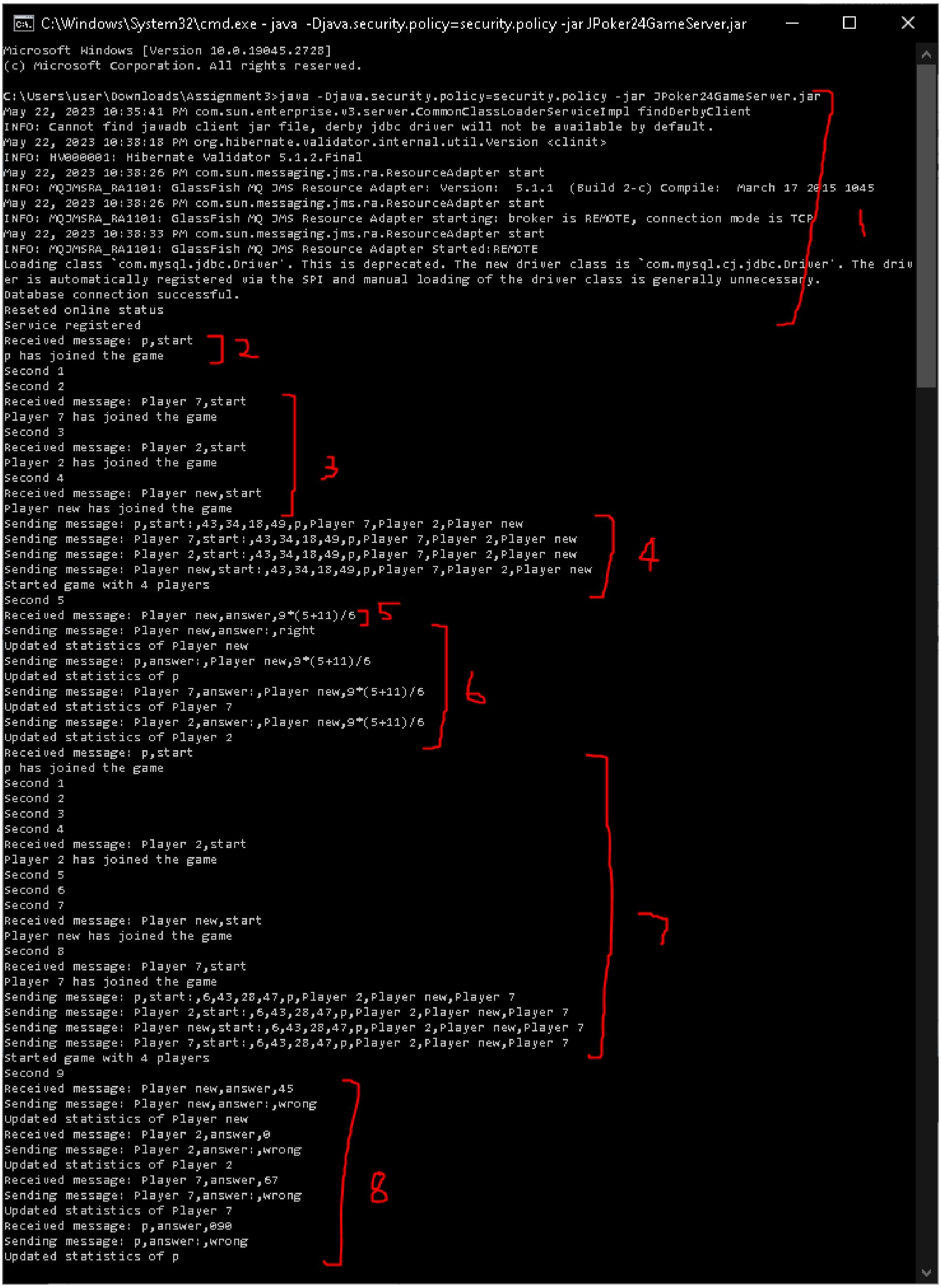


```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Downloads\Assignment3>java -Djava.security.policy=security.policy -jar JPoker24Game.jar localhost
May 22, 2023 10:41:56 PM com.sun.enterprise.v3.server.CommonClassLoaderServiceImpl findDerbyClient
INFO: Cannot find javadb client jar file, derby jdbc driver will not be available by default.
May 22, 2023 10:48:58 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.1.2.Final
May 22, 2023 10:49:17 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version:  5.1.1  (Build 2-c) Compile:  March 17 2015 1045
May 22, 2023 10:49:17 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
May 22, 2023 10:49:29 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Connected to server
Sending message Player 2,start
Received message: p,start:,43,34,18,49,p,Player 7,Player 2,Player new
Received message: Player 7,start:,43,34,18,49,p,Player 7,Player 2,Player new
Received message: Player 2,start:,43,34,18,49,p,Player 7,Player 2,Player new
Received message: Player new,start:,43,34,18,49,p,Player 7,Player 2,Player new
Received message: Player new,answer:,right
Received message: p,answer:,Player new,9*(5+11)/6
Received message: Player 7,answer:,Player new,9*(5+11)/6
Received message: Player 2,answer:,Player new,9*(5+11)/6
Sending message Player 2,start
Received message: p,start:,6,43,28,47,p,Player 2,Player new,Player 7
Received message: Player 2,start:,6,43,28,47,p,Player 2,Player new,Player 7
Received message: Player new,start:,6,43,28,47,p,Player 2,Player new,Player 7
Received message: Player 7,start:,6,43,28,47,p,Player 2,Player new,Player 7
Received message: Player new,answer:,wrong
Sending message Player 2,answer,0
Received message: Player 2,answer:,wrong
Received message: Player 7,answer:,wrong
Received message: p,answer:,wrong
```

1. Client app started and logged in.
2. Pressed new game button and sent a message to notify the server.
3. Received a game message from the server, including card hand and other players' name in the same game.
4. p submitted the correct answer and the game has ended.
5. A new game is started.
6. Send answer message to server.
7. All players (including itself) submitted wrong answers.

Console view of server



C:\Windows\System32\cmd.exe - java -Djava.security.policy=security.policy -jar JPoker24GameServer.jar
1
2
3
4
5
6
7
8

```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Downloads\Assignment3>java -Djava.security.policy=security.policy -jar JPoker24GameServer.jar
May 22, 2023 10:35:41 PM com.sun.enterprise.v3.server.CommonClassLoaderServiceImpl findDerbyClient
INFO: Cannot find javadb client jar file, derby jdbc driver will not be available by default.
May 22, 2023 10:38:18 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.1.2.Final
May 22, 2023 10:38:26 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version:  5.1.1  (Build 2-c) Compile:  March 17 2015 1045
May 22, 2023 10:38:26 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
May 22, 2023 10:38:33 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driv
er is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
Database connection successful.
Reseted online status
Service registered
Received message: p,start
p has joined the game
Second 1
Second 2
Received message: Player 7,start
Player 7 has joined the game
Second 3
Received message: Player 2,start
Player 2 has joined the game
Second 4
Received message: Player new,start
Player new has joined the game
Sending message: p,start:,43,34,18,49,p,Player 7,Player 2,Player new
Sending message: Player 7,start:,43,34,18,49,p,Player 7,Player 2,Player new
Sending message: Player 2,start:,43,34,18,49,p,Player 7,Player 2,Player new
Sending message: Player new,start:,43,34,18,49,p,Player 7,Player 2,Player new
Started game with 4 players
Second 5
Received message: Player new,answer,9*(5+11)/6
Sending message: Player new,answer:,right
Updated statistics of Player new
Sending message: p,answer:,Player new,9*(5+11)/6
Updated statistics of p
Sending message: Player 7,answer:,Player new,9*(5+11)/6
Updated statistics of Player 7
Sending message: Player 2,answer:,Player new,9*(5+11)/6
Updated statistics of Player 2
Received message: p,start
p has joined the game
Second 1
Second 2
Second 3
Second 4
Received message: Player 2,start
Player 2 has joined the game
Second 5
Second 6
Second 7
Received message: Player new,start
Player new has joined the game
Second 8
Received message: Player 7,start
Player 7 has joined the game
Sending message: p,start:,6,43,28,47,p,Player 2,Player new,Player 7
Sending message: Player 2,start:,6,43,28,47,p,Player 2,Player new,Player 7
Sending message: Player new,start:,6,43,28,47,p,Player 2,Player new,Player 7
Sending message: Player 7,start:,6,43,28,47,p,Player 2,Player new,Player 7
Started game with 4 players
Second 9
Received message: Player new,answer,45
Sending message: Player new,answer:,wrong
Updated statistics of Player new
Received message: Player 2,answer,0
Sending message: Player 2,answer:,wrong
Updated statistics of Player 2
Received message: Player 7,answer,67
Sending message: Player 7,answer:,wrong
Updated statistics of Player 7
Received message: p,answer,090
Sending message: p,answer:,wrong
Updated statistics of p
```

1. Server started.
2. p pressed the new game button to send a message to server. The server then start a timer.
3. 3 other players pressed the new game button to send a message to server so the game immediately started.

4. Broadcasts cards and players involved to the players.
5. p sent an answer.
6. p got the correct answer. Sent a correct message to p and lose message to other players. Also updated their statistics in the table.
7. The players started a new game.
8. All players submitted wrong answers. Sent wrong messages to them and updated their statistics in the table.

<u>Message syntax</u>

Client-to-server: (playerName), (command), (additional arguments)

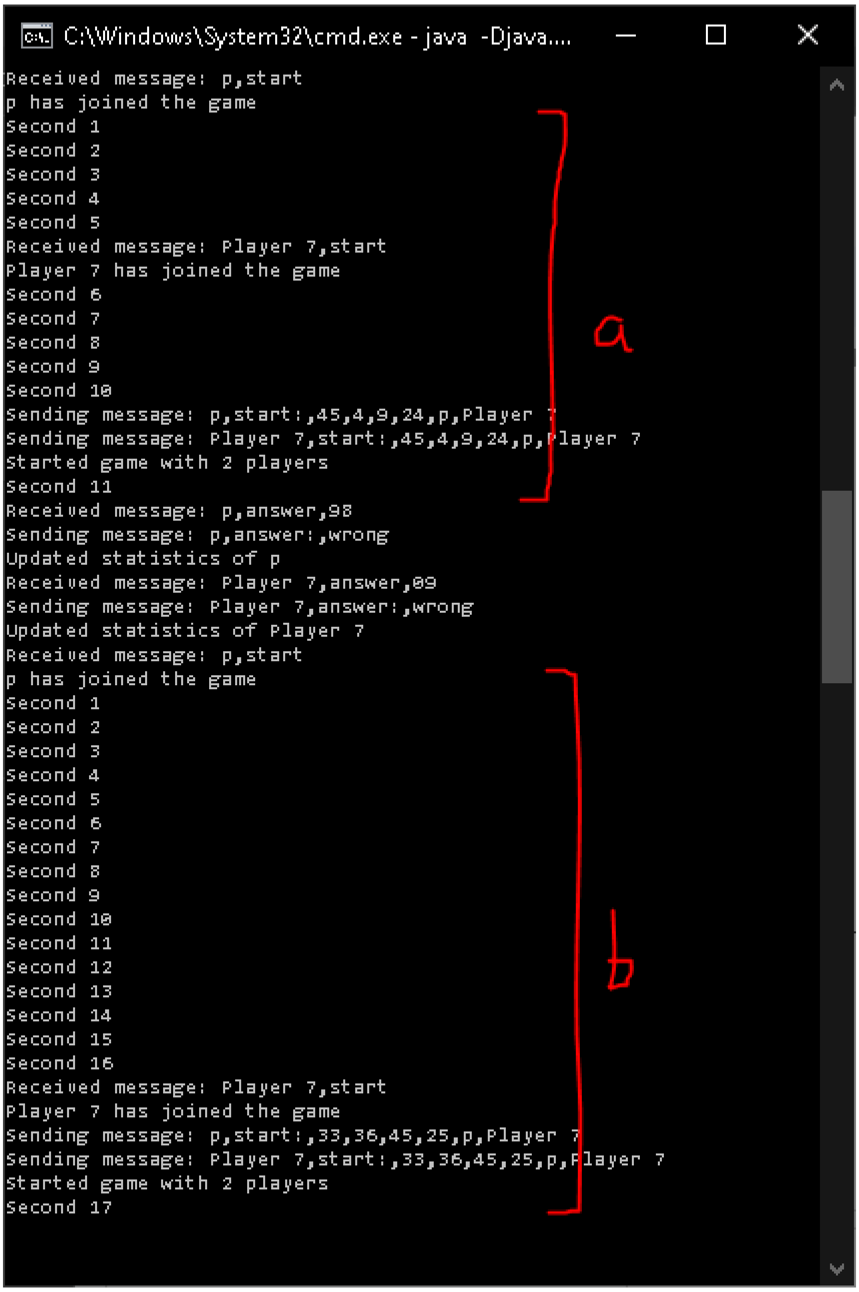
- command = start: notify server the player has joined the game
- command = answer:
    - additional arguments is the answer string in the text field

Server-to-client: (TO: playerName), (command), (additional arguments)

- command = start: inform players a game has started
    - the 4 int entries followed are the cards index, then followed by a list of the players in that game
- command = answer: return when the player answers wrongly, rightly; or another player answered rightly
    - additional arguments = wrong: wrong answer
    - additional arguments = right: right answer
    - additional arguments = (playerName),(itsAnswer): the player with the playerName got the right answer as itsAnswer

<u>Additional notes</u>

1. The above flow shows a game start case by having 4 players waiting. The other two cases are:
    a. 2-3 players joins and 10 seconds have elapsed since the first player joins
    b. 10 seconds have elapsed since the first player joins and another player joins



```
Received message: p,start
p has joined the game
Second 1
Second 2
Second 3
Second 4
Second 5
Received message: Player 7,start
Player 7 has joined the game
Second 6
Second 7
Second 8
Second 9
Second 10
Sending message: p,start:,45,4,9,24,p,Player 7
Sending message: Player 7,start:,45,4,9,24,p,Player 7
Started game with 2 players
Second 11
Received message: p,answer,98
Sending message: p,answer:,wrong
Updated statistics of p
Received message: Player 7,answer,09
Sending message: Player 7,answer:,wrong
Updated statistics of Player 7
Received message: p,start
p has joined the game
Second 1
Second 2
Second 3
Second 4
Second 5
Second 6
Second 7
Second 8
Second 9
Second 10
Second 11
Second 12
Second 13
Second 14
Second 15
Second 16
Received message: Player 7,start
Player 7 has joined the game
Sending message: p,start:,33,36,45,25,p,Player 7
Sending message: Player 7,start:,33,36,45,25,p,Player 7
Started game with 2 players
Second 17
```

2. Sometimes a caught "AWT-Queue" exception occurs after a game ends at client side. Under my testing, this exception does not bring any effect to the game and server-client communication. My intuition is that updating tabs may generate such an exception, since the tabs update are done by removing and adding new tabs, but the tabs still function normally afterwards.

Code modification from Assignment 2

Server.java

1. Added server-to-clients topic and client-to-server queue. The codes are mostly copied from the given samples (for the topic, just replace all "queue" with "topic" in the samples), but
    - The JNDI names of that connection factory, queue and topic are changed.

- sendMessages() in server-to-client queues is modified to send a string message, with the message being the argument
- receiveMessages() in client-to-server queue is significantly modified
    - It is now a infinite loop for receiving messages, which is run by a thread.
    - Upon receiving the message “(playerName),(start),(channel no.)”, add the player channel to the waitlist. Start a timer if a timer has not been started.
    - The timer keeps track of the waitlist. If there are more than 4 elements in the waitList, start a game with 4 people, who are popped from the waitlist. If there are more than 2 elements in the waitlist and 10 seconds have elapsed, start a game with all the waitlisted players. Cancel the timer after the above 2 cases.
    - Start a Game object and add it to an ArrayList when a timer is cancelled. The Game object consists of the player channels and cards in the game. Send “(playerName),start:,(card1),(card2),(card3),(card4),(player 1 channel),(player 2 channel)...” to all playerName involved when instantized. The cards must have different values.
    - Upon receiving the message “(playerName),answer,(channel no.),(answer)”, check the answer. The answer is correct if and only if having exactly 4 numbers that match the 4 cards’ value respectively and evaluate to exactly 24 without decimals. The latter evaluation is done by JavaScript engine.
    - When a player has submitted a wrong answer, send “(playerName),answer:,wrong” to that player and remove it from the Game’s playerList. Increment the player’s gamesPlayed in the table via JDBC. Remove this Game from the arrayList of Game if no players are left.
    - When a player has submitted a right answer, send “(playerName),answer:,right” to that player and “(playerName),answer:,(winnerName),(itsAnswer)” to other remaining players. Increment all players’ gamesPlayed in the table and the winner’s gamesWon and avgWinTime via JDBC. The avgWinTime is updated as ROUND((gamesWon*avgWinTime+thisWinTime)/(gamesWon+1),2). Then remove this Game from the arrayList of Game.

2. Two JDBC methods updateLose() and updateWin() are added to perform the aforementioned operations.
3. Game object:
    - Attributes: ArrayList<String> players, int[4] cards, long startTime
    - Methods:
        - Game(String[] players)
            - Store the players in its ArrayList and draw 4 cards with different values to the card array. Then send start messages to the players.
        - findUser(String player)

        - Find if the player is in the game
    - removeUser(String player)
        - Remove the player in the game. If no players are left, remove this Game from the gameList.
    - getPlayers(), getStart(): getter methods
4. Some variables are added:
    - An ArrayList of storing Game objects
    - A boolean flag indicating if a timer is running. This is to prevent multiple timers running at the same time

<u>Client.java</u>

1. Added server-to-clients topic and client-to-server queues. The codes are mostly copied from the given samples (for the topic, just replace all “queue” with “topic” in the samples), but
    - The JNDI names of that connection factory, queue and topic are changed.
    - sendMessages() in client-to-server queue is modified to send a string message, with the message being the argument.
    - receiveMessages() in server-to-client queue is significantly modified
        - It is now an infinite loop for receiving messages, which is run by a thread.
        - Upon pressing the new game button, send a message “(playerName),(start)” to server and wait for players.
        - Upon receiving the message “start:,(card1),(card2),(card3),(card4),(player1),(player2)...” from server, display the 4 card images based on their indexes, and get and show each player info by calling getTable() via RMI. Also shows a JTextField and a JLabel evaluating the math expression in the JTextField with JavaScript engine.
        - Upon pressing enter in the JTextField, send the message “(playerName),answer,(answer)” to server.
        - Upon receiving the message “answer:,(args)”, shows “You win!”, “Wrong answer!” or the winner and its answer based on args. Then remove tabs UserProfile and Leaderboard and re-add them to refresh the tabs.

2. Implemented the Play Game tab. It is initially a big new game button sending start-game-message to the server. Then receiveMessages() takes care of this tab.