Distributed-System-Assignment1

Distributed Player Status System (DPSS) using Java RMI

Concordia University
Engineering and Computer Science

Assignment #1
Distributed Player Status System (DPSS) using Java RMI

Submitted to:
Dr. Mohamed Taleb
Email: mohamed.taleb@concordia.ca
Office: S-EV 3233

Submitted by:
Vaibhav Malhotra
ID: 40079373

Development tools

All code is written in IntelliJ IDE, Java JDK version 8.

Build and Run

IntelliJ:
- Open the project in folder DPSS
- Setup the SDK
- Run: AmericaGameServer, EuropeGameServer, AsiaGameServer.
- Run: PlayerClient (to launch a player window)
- Run: AdminClient (to launch a Admin window)
- To run multiple clients change the configuration to "Allow parallel run".

Command Line:

- Move to *DPSS* directoy: **cd DPSS**.
- Create a new folder named *dist* in the current folder : **mkdir dist**
- Compile the code (outputting into dist folder): **javac -d dist src/**/*.java**
- Move to *dist* folder: **cd dist**
- Run all 3 servers using following commands (will have to open different terminals):
  - java -cp . GameServers.AmericaServer.AmericaGameServer
  - java -cp . GameServers.EuropeServer.EuropeGameServer
  - java -cp . GameServers.AsiaServer.AsiaGameServer

- Run the clients (will have to open in different terminals):
  - java -cp . Client.PlayerClient
  - java -cp . Client.AdminClient

Vaibhav Malhotra – 40079373

Concepts implemented

1. RMI

Remote Method Invocation has been used to invoke client requests on the servers based on user's IP address.

2. UDP

For below communication between server UDP is used:

- getPlayerStatus() – When admin requests this method on a server, that server sends UDP request to other two servers to get the player info.
- createPlayer() – When a user tries to create a new player on a server, that server sends a UDP request to other servers to check if Username already exists.

3. Multi-threading

- All servers run on their individual thread
- All UDP requests are sent on a new thread
- All client requests are sent on a new thread
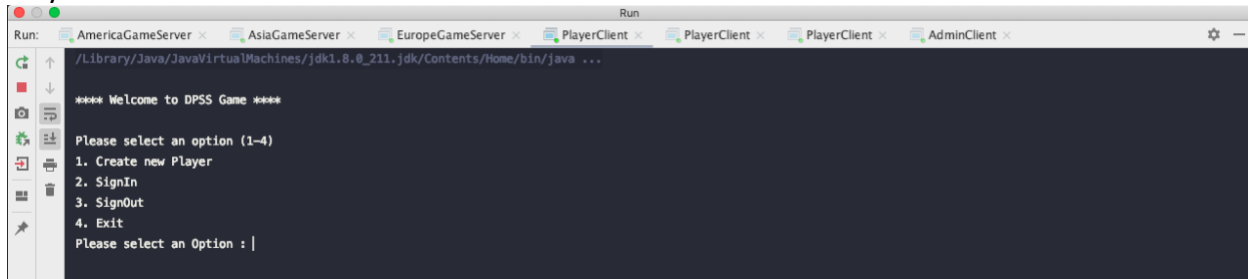
4. Locks and HashTable

- Player data on server are stored in a Hashtables. Hashtables are thread-safe and promote concurrency.
- Lock (ReentrantLock) is used for proper synchronization to allow multiple users to perform operations for the same or different accounts at the same time.

Testing

Intro screens:

Player Screen:
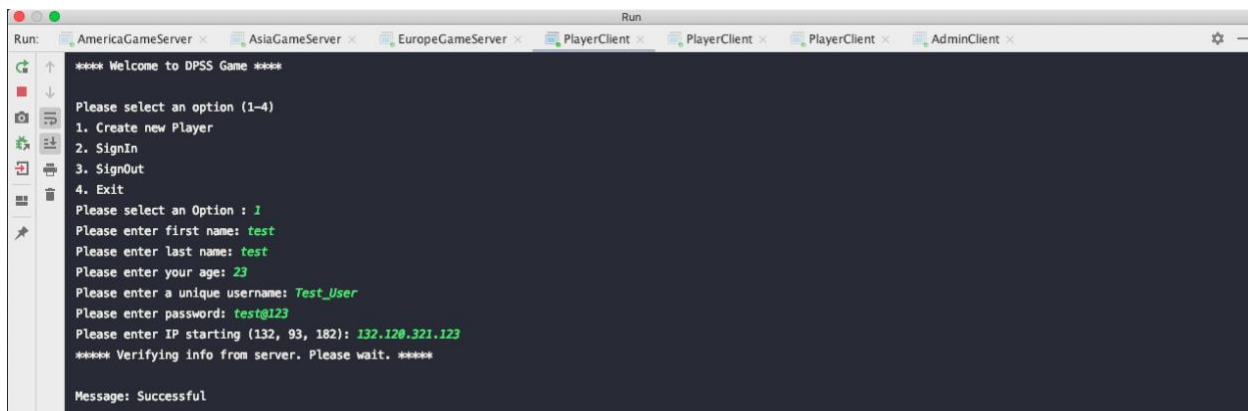
Admin Screen:

Player testing:

1. Create player - *createPlayerAccount* (*FirstName, LastName, Age, Username, Password, IP Address*)

Input data is validated.
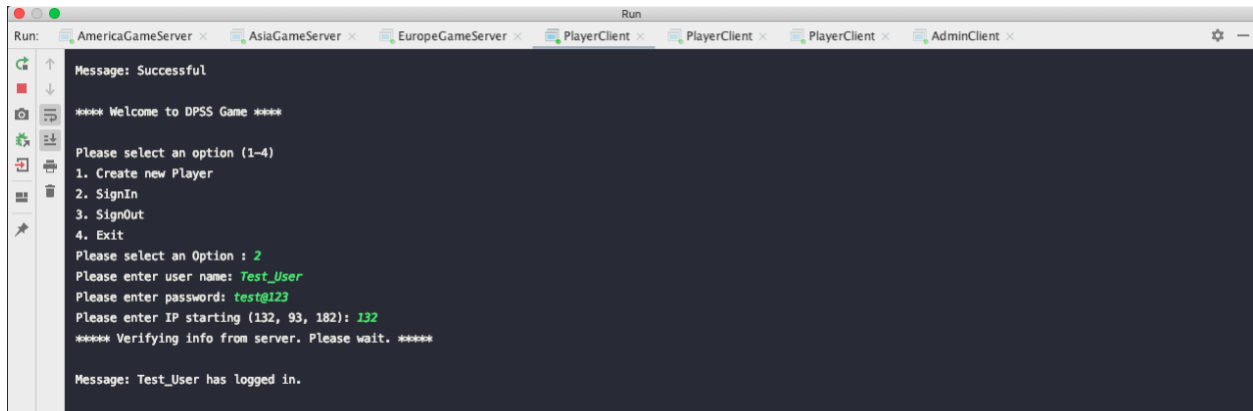Player created successfully and message is displayed.

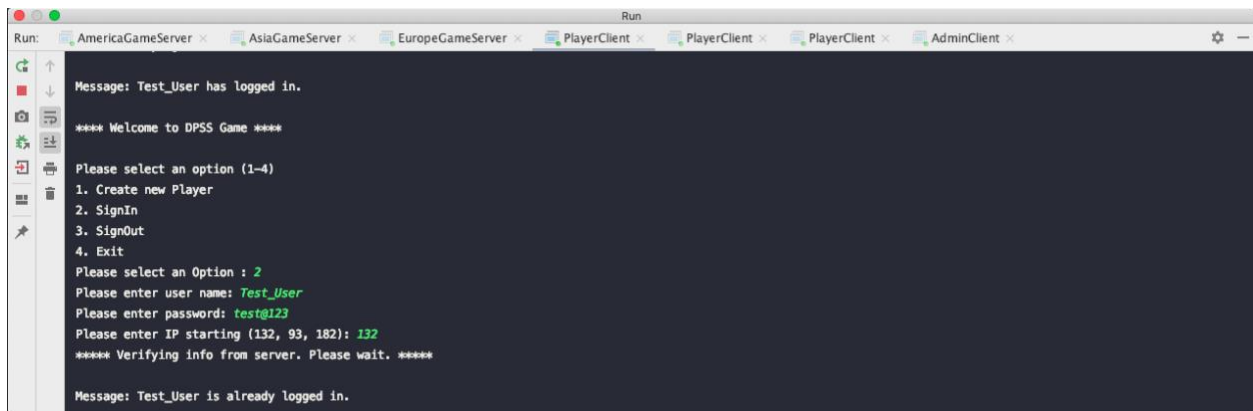2. Login player - *playerSignIn* (*Username, Password, IPAddress*)

Player data is validated.
Player logged in successfully and message is displayed.
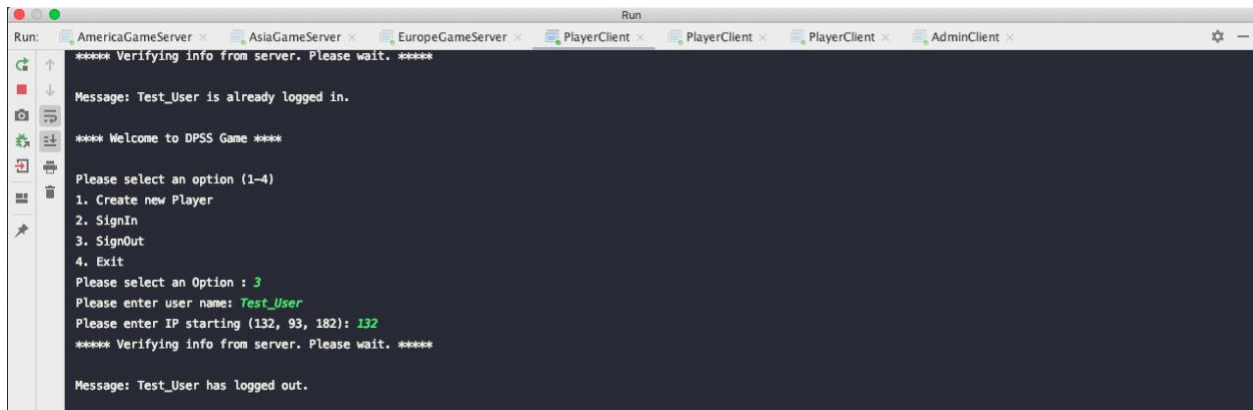
3. Login player who is already logged in.

Player data is validated.
Player log in unsuccessfully because player is already logged in and message is displayed.

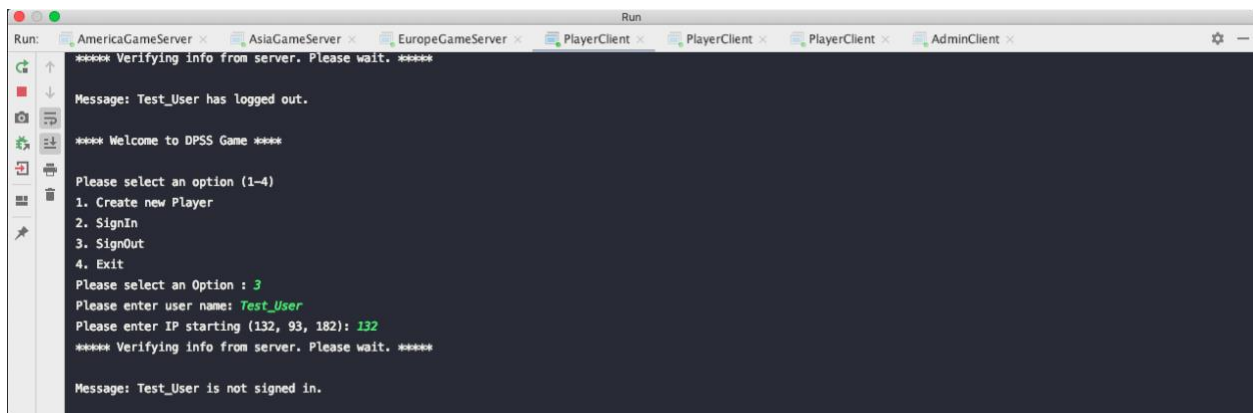4. Logout player - *playerSignOut* (*Username, IPAddress*)

Player data is validated.
Player logged out successfully and message is displayed.
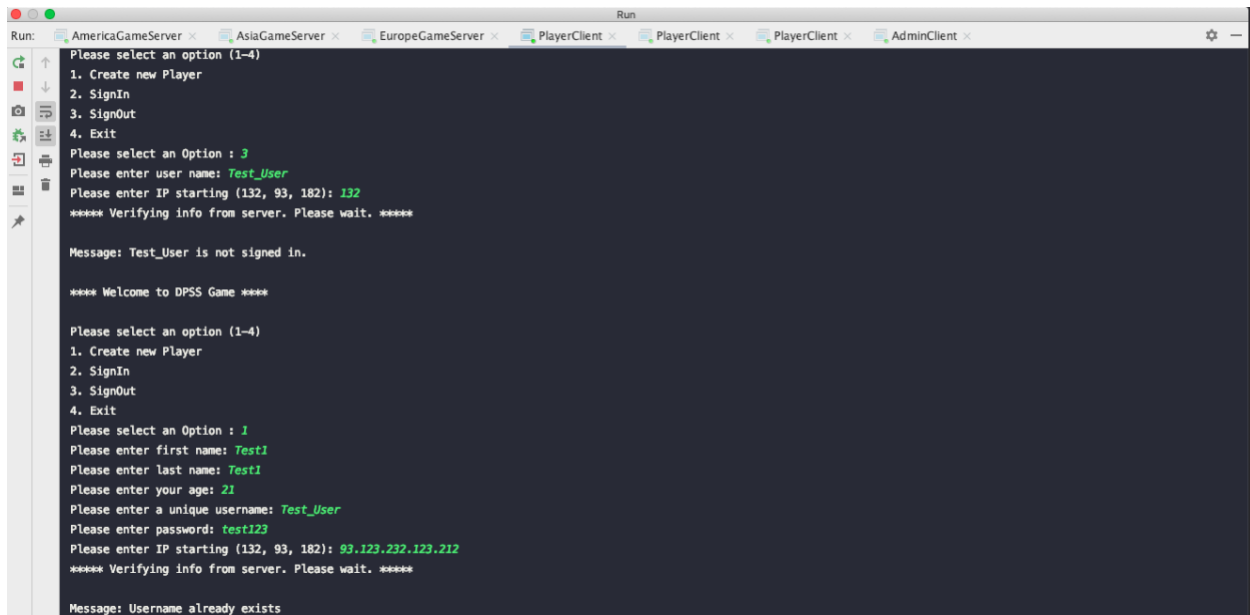
5. Logout player who is not logged in - *playerSignOut* (*Username, IPAddress*)

Player logout unsuccessfully because player is not logged in and message is displayed.
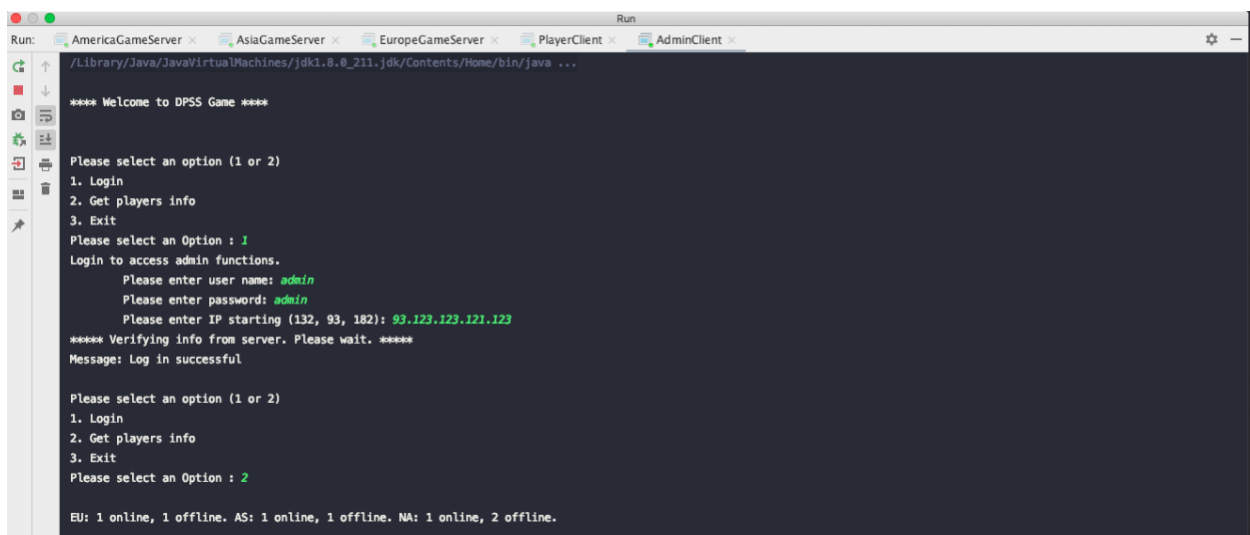
6. Try to create a user (Already on another server)

Player creation unsuccessful on server 93.123.123.123.121, Because player already existed on server 132.

Admin:
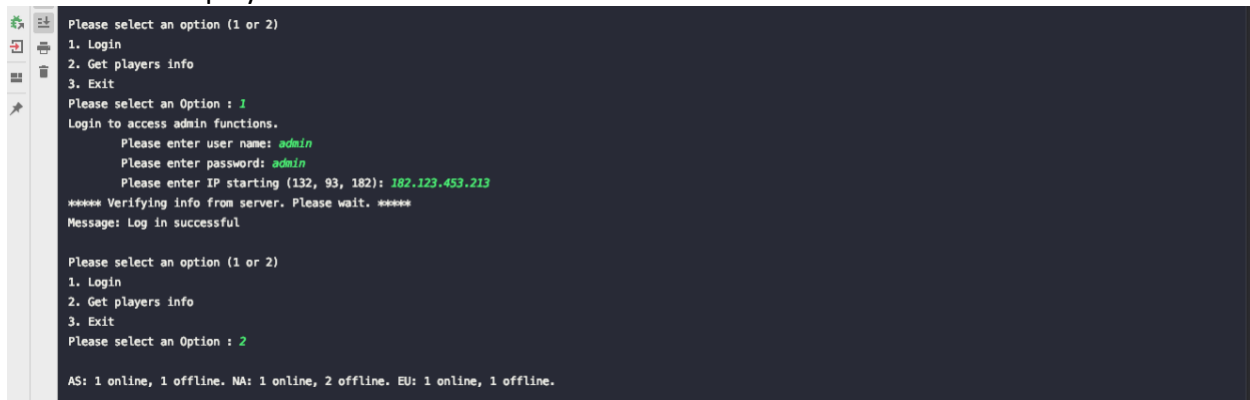
1. On server IP 93 - *getPlayerStatus* (*AdminUsername, AdminPassword, IPAddress*)

Admin has logged in successfully from IP 93.123.123.121.123 on server Europe.
Europe server sends request to Asia and America Server to gets their player status, appends it's result and displays.

2. Get player status on server IP 182

```
Please select an option (1 or 2)
1. Login
2. Get players info
3. Exit
Please select an Option : 1
Login to access admin functions.
        Please enter user name: admin
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 182.123.453.213
***** Verifying info from server. Please wait. *****
Message: Log in successful

Please select an option (1 or 2)
1. Login
2. Get players info
3. Exit
Please select an Option : 2

AS: 1 online, 1 offline. NA: 1 online, 2 offline. EU: 1 online, 1 offline.
```

Admin has logged in successfully from IP 182.123.453.213 on server Asia.
Asia server sends request to Europe and America Server to gets their player status, appends it's result and displays.

3. Invalid IP or credentials

```
Login to access admin functions.
        Please enter user name: admin123
        Please enter password: admin123
        Please enter IP starting (132, 93, 182): 32.122.123.322
        Please enter IP starting (132, 93, 182): 132.122.123.322
***** Verifying info from server. Please wait. *****
Error: Credentials invalid. Please try again.
```

Admin enters invalid IP, so it's asked to enter IP again.
Admin enters invalid credentials so it's asked to enter credentials again.