



**COMP6231- Distributed System Design**  
**Assignment-2**

---

*Distributed Player Status System (DPSS) using Java IDL*

---

**Date:** 28 June 2020.

**Recipient:** Prof. Mohammed Taleb

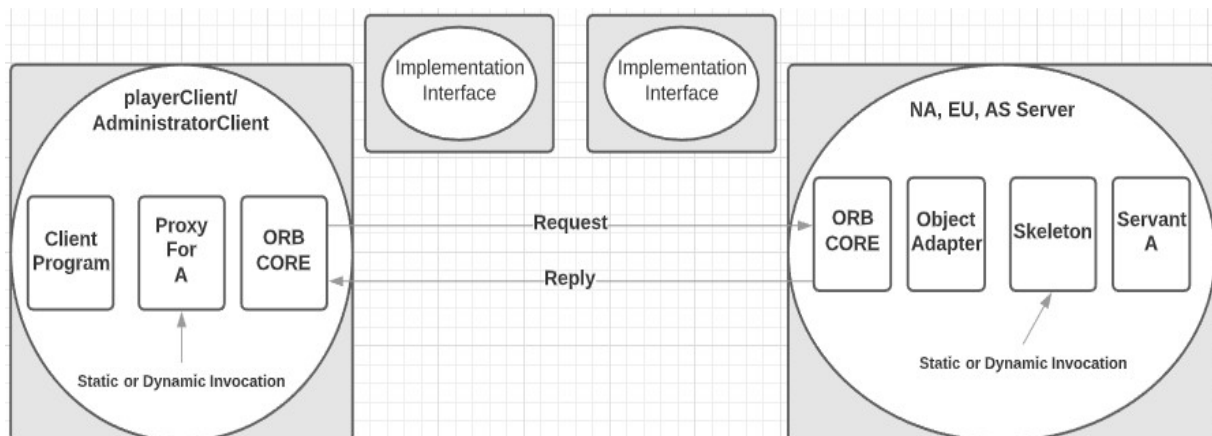
**Student Name:** Raj Mistry (40119206)

## ➤ Introduction:

The main objective of this assignment is to design and implement a Distributed Player Status System which manages player's status across multiple game servers using Java IDL & CORBA. CORBA is one of the middleware solutions based on distributed object that offers simple way to call different methods on server machines to perform various operation and returning objects over network and provide network transparency. In addition to this we have to make our application synchronized with the use of multi-threading in order to handle concurrent requests with ease. User Datagram Protocol is used by each server to communicate with other servers, sending and receiving requests to perform inter-server operations.

## ➤ CORBA Architecture:

In this assignment the architecture used is CORBA – Common Request Broker Architecture. It is not two-tier or three-tier architecture, it is middleware which is designed to allow the communication between various devices that are deployed in different platforms or different OS. As middleware it receives the requests from the client and transfer it to particular server object and then it marshals the exception or response from server and send it back to client. It uses the object-oriented model to provide its benefits like data encapsulation, abstraction, etc to provide transparency. For using object-oriented model, it is not required that the system using CORBA need to be Object-oriented; it enables the communication between systems irrespective of their programming language as well. It allows client to invoke server methods both locally and remotely (statically & dynamically).



**\* CORBA Architecture \***

## ➤ IDL:

IDL stands for Interface Definition Language or Interface Description Language. It's a specification language which is used to describe the program's Interface which is not dependent on any programming language. It defines name and methods that client can request. It enables the communication between client and server programs which may or may not be written in same programming language. IDL commonly used in RPC Software where at the both end systems may run on different OS and use different programming language. It provides bridge between these systems.

```

1 module GameServerApp
2 {
3     interface GameServer
4     {
5         string createPlayerAccount(in string firstName,in string lastName,in string age,in string username,in string password
6                                     ,in string ipAddress);
7         string playerSignIn(in string username,in string password);
8         string playerSignOut(in string username);
9         string transferAccount(in string username,in string password,in string newIpAddress);
10        string getPlayerStatus();
11        string suspendAccount(in string usernameToSuspend);
12    };
13};

```

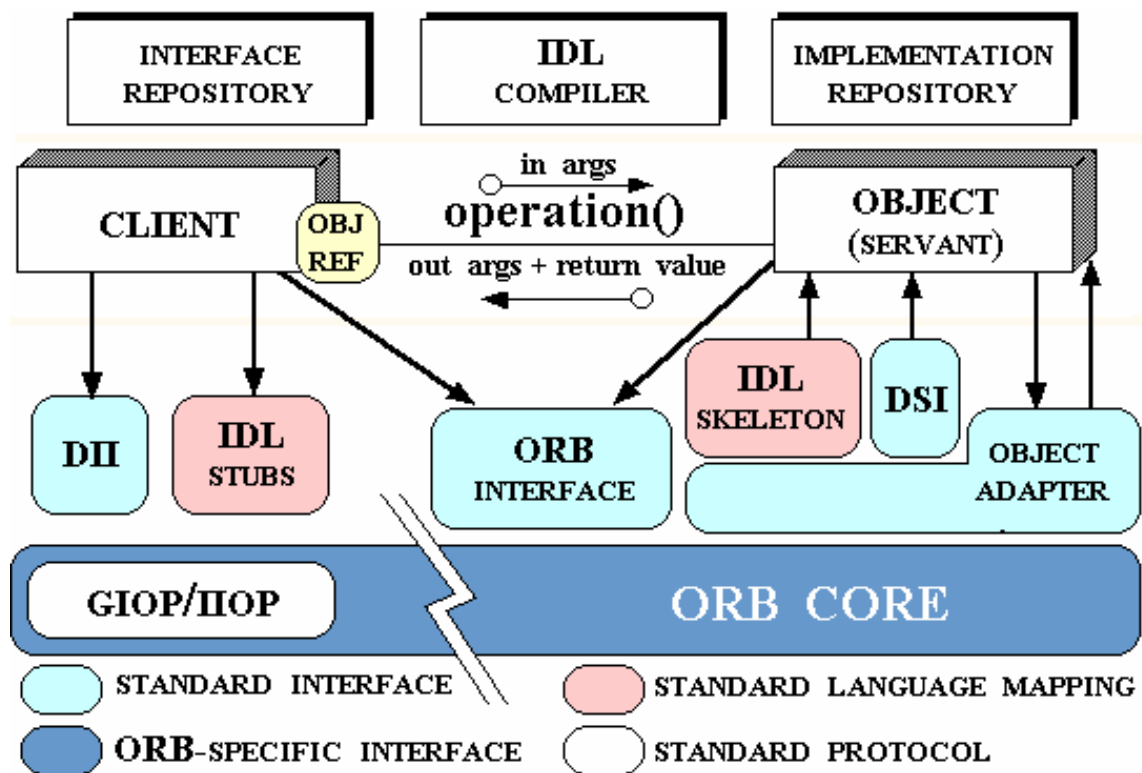
\* IDL Code Snippet \*

### ➤ ORB:

The Object Request Broker is a middleware that uses CORBA specification. ORB manages all the details involved in sending the request from client to object and sending the response to its destination. ORB also manages the transformation of data structure of the process to the raw byte sequence and vice-versa, which is required to transmit it over the network. This is called marshalling or serialization. Other than this ORB also provides features like distributed transactions, directory services or real-time scheduling. The ORB is also responsible to look after the Interface Repository, an OMG (Object Management Group)-standardized distributed database containing OMG IDL interface definitions.

On client side, ORB provides interface definition and performs method invocation using dynamic invocation interface (DII).

On the server side, the ORB de-activates inactive objects, and re-activates them whenever a request comes in. CORBA supports a number of activation patterns, so that different object or component types can activate and de-activate in the way that uses resources best.

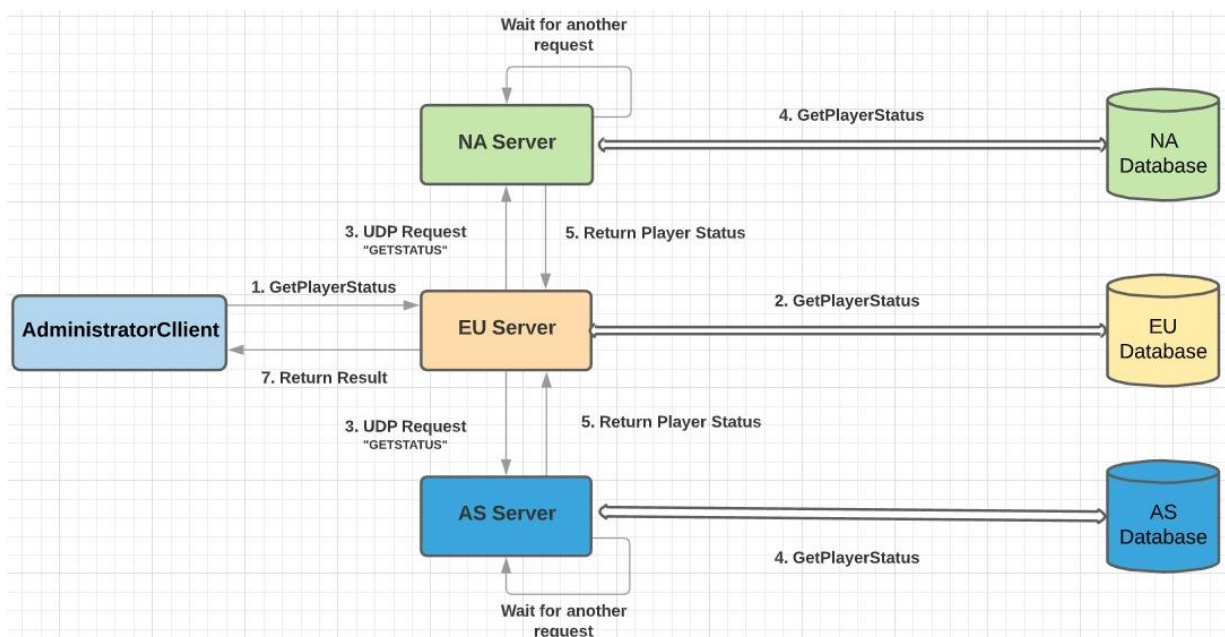


\* ORB Architecture \*

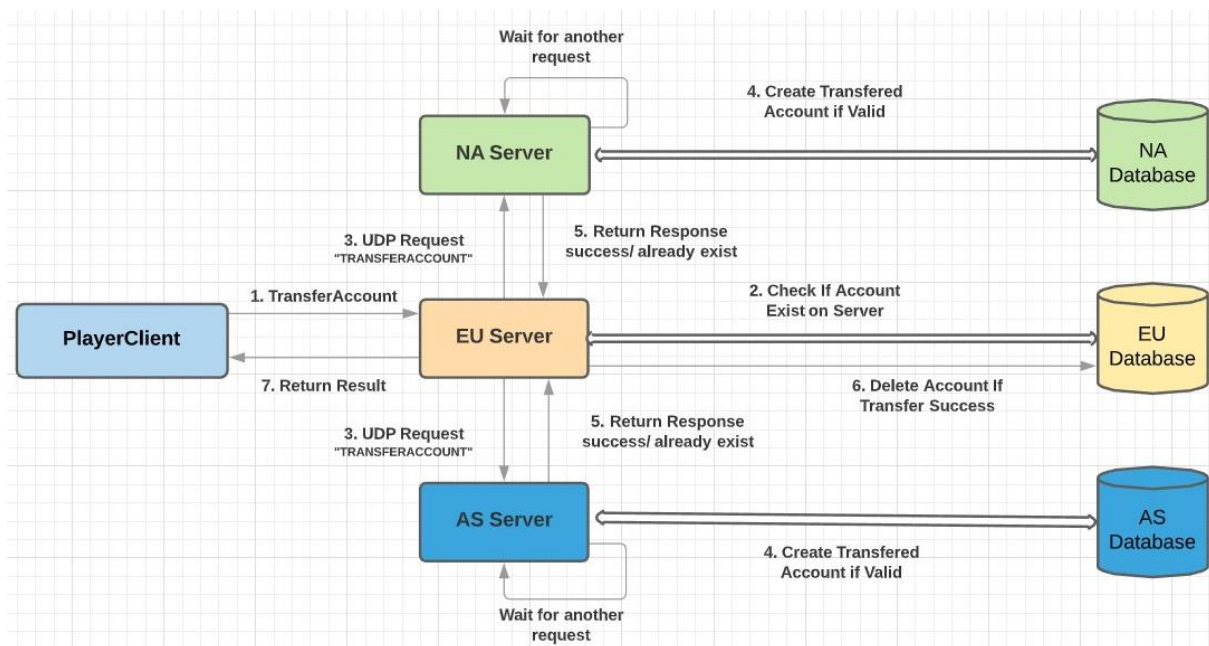
### ➤ UDP:

The User datagram Protocol is a transport layer protocol. It is connectionless and unreliable protocol. To communicate using UDP it is not required to establish connection prior to data transfer. A datagram is an independent, self-contained message sent over the network whose arrival, arrival time, and content are not guaranteed. The Datagram Packet and Datagram Socket classes in the java.net package implement datagram communication using UDP.

- In this assignment I have implemented UDP for communication among three servers (NA, EU and AS) in order to perform operations `getPlayerStatus()` and `transferAccount()`. Each server Creates its own UDP thread when server starts.
- In `getPlayerStatus()` the current server communicates with other servers using UDP request "GETSTATUS" to fetch the player accounts status on corresponding server and returns server specific online and offline player accounts count. (In this UDP request will be sent to rest of the 2 servers)



- In `transferAccount()` checks the `HashMap` for the required player account. If found, the server transfers the account to specified remote server by sending UDP request "TRANSFERACCOUNT" and after successfully transferring the record to remote server, it will remove the record from its storage and then server informs the client that the player account transfer is successful. (In this UDP request will be sent to only 1 server where we want to transfer Player Account [Destination Server] e.g. To transfer Account from EU to AS, UDP Request will be sent to AS only not to NA)



### ➤ Multithreading and Synchronization:

Thread is a lightweight process which has its own stack. When Multiple threads try to access the same resources and produces wrong result because of data corruption. This happens because while one thread is accessing resource same time other thread interferes and access it and now both perform operation on that old value and write their result which may produce wrong result.

Synchronization is a procedure in which is used to avoid interference with the other threads and memory consistent errors. The keyword synchronized is used to implement Competition Synchronization which means while one thread is executing the critical region where shared resource is used all the other threads will wait until its execution of critical region finished.

There are Two types of synchronizations used in java:

- Method Synchronization
- Block Synchronization.

### *Synchronization Used in this Assignment:*

- Block level Synchronization:
  - A specific part or block of code that is needed to be synchronized is put under its own specific scope.
- Method level Synchronization
  - A specific methods or functions in java is synchronized using synchronize keyword before the method name.

### ➤ Data Structure and Concurrency:

Here in this assignment in order to improve concurrency I have used ConcurrentHashMap to as the storage for Player Accounts as it is an enhancement of HashMap, as we know that while dealing with threads in our application simple HashMap is not good choice because performance wise HashMap is not up to the mark. ConcurrentHashMap class is thread safe and

it is good choice when we need high concurrency in our application. It does not throw a “ConcurrentModificationException” if one thread tries to modify it while another is iterating over it.

### Syntax:

```
ConcurrentHashMap<String, List<Object>> accounts = new ConcurrentHashMap<String, List<Player>>();
```

Where Player is the Model class which stores details of player account in its object.

The Fine-grained locking is used to put intrinsic lock while doing parallel access to data resources so that data is accessed as required without infecting the data. This increases the efficiency and maximizes the concurrency to perform concurrent operations.

Concurrency is implemented by using the synchronized(){}-block in java which puts temporary lock on the object or list passed to it as parameter in this implementation.

### ➤ Code Structure:





➤ **Test Cases:****1) Validations (All attributes must be entered & can't be empty)**➤ **First Name and Last Name** Should Contain Only Alphabets without any Whitespaces.

```

Enter First Name:

=====
First Name Should Contain Only Alphabets Without Any Whitespaces!!!
=====

Enter First Name: kjahd kajsdh

=====
First Name Should Contain Only Alphabets Without Any Whitespaces!!!
=====

Enter First Name: sjashd7

=====
First Name Should Contain Only Alphabets Without Any Whitespaces!!!
=====

Enter First Name: akjs888a***

=====
First Name Should Contain Only Alphabets Without Any Whitespaces!!!
=====

Enter First Name: validname
Enter Last Name: validationsameasfirstname

```

➤ **Age** must be Positive Integer without any whitespaces.

```

Enter Age: two

=====
Age Should Contains Only Positvie Integer Without Any Whitespaces!!
=====

Enter Age: 5 4

=====
Age Should Contains Only Positvie Integer Without Any Whitespaces!!
=====

Enter Age: -87

=====
Age Should Contains Only Positvie Integer Without Any Whitespaces!!
=====

Enter Age: 22
Enter Username:

```

- **Username** must be start with alphabet and should not contain any whitespaces and length must be minimum of 6 and maximum of 15 characters.

Enter Username: `raj mistry`

```
=====
Username Should Start With Alphabet Without Any Whitespaces!!
-> It should have a length minimum of 6 characters and a maximum of 15 characters.
=====
```

Enter Username: `5rajmistry%`

```
=====
Username Should Start With Alphabet Without Any Whitespaces!!
-> It should have a length minimum of 6 characters and a maximum of 15 characters.
=====
```

Enter Username: `raj`

```
=====
Username Should Start With Alphabet Without Any Whitespaces!!
-> It should have a length minimum of 6 characters and a maximum of 15 characters.
=====
```

Enter Username: `rajmistryrajmistry`

```
=====
Username Should Start With Alphabet Without Any Whitespaces!!
-> It should have a length minimum of 6 characters and a maximum of 15 characters.
=====
```

Enter Username: `validname`

- **Password** must not contain any whitespaces and should have minimum length of 6 characters.

Enter Password: `pass with whitespace`

```
=====
Password Should Have Minimum 6 Characters And Without Any Whitespaces!!
-> It Can Contain Alphabets, Numbers OR Special Characers[! @ # $ % ^ &].
=====
```

Enter Password: `6Ggj`

```
=====
Password Should Have Minimum 6 Characters And Without Any Whitespaces!!
-> It Can Contain Alphabets, Numbers OR Special Characers[! @ # $ % ^ &].
=====
```

Enter Password: `validpassword$$`

Enter IPAddress:



- **IP Address** must be entered as per valid IPv4 format & must be entered from one of the 3 ranges (132.xxx.xxx.xxx, 93.xxx.xxx.xxx, 182.xxx.xxx.xxx)

Enter IPAddress: 1adjas.aksjdha.asd.asd

```
=====
  IPAddress Should be Valid[0-255] And Without Any Whitespaces & Alphabets!!)
  -> It should be from one of the below Ranges:
  -> 132.xxx.xxx.xxx
  -> 93.xxx.xxx.xxx
  -> 182.xxx.xxx.xxx
=====
```

Enter IPAddress: one.two.three.four

```
=====
  IPAddress Should be Valid[0-255] And Without Any Whitespaces & Alphabets!!)
  -> It should be from one of the below Ranges:
  -> 132.xxx.xxx.xxx
  -> 93.xxx.xxx.xxx
  -> 182.xxx.xxx.xxx
=====
```

Enter IPAddress: 164.2.4.5

```
=====
  IPAddress Should be Valid[0-255] And Without Any Whitespaces & Alphabets!!)
  -> It should be from one of the below Ranges:
  -> 132.xxx.xxx.xxx
  -> 93.xxx.xxx.xxx
  -> 182.xxx.xxx.xxx
=====
```

Enter IPAddress:

- **New IP Address for transfer Account** must be different from old Ip Address as Account cannot be transfer to same server & it must be entered as per valid IPv4 format & must be entered from one of the 3 ranges (132.xxx.xxx.xxx, 93.xxx.xxx.xxx, 182.xxx.xxx.xxx).

Enter Your Choice: 4  
 Enter Username: rajmisrtu  
 Enter Password: 5464646  
 Enter Old IPAddress: 132.2.2.2  
 Enter New IPAddress: 132.5.4.4

```
=====
  New IPAddress Should be Valid[0-255] And Without Any Whitespaces & Alphabets!!
  -> It should be from one of the below Ranges:
  -> 132.xxx.xxx.xxx
  -> 93.xxx.xxx.xxx
  -> 182.xxx.xxx.xxx
=====
```

New IPAddress For Player Account To Transfer Must Not Be Same As Its Old IPAddress!!  
 Because You Can't Transfer Player Account To The Same Server!!

Enter New IPAddress: 164.5.5.4

```
=====
  New IPAddress Should be Valid[0-255] And Without Any Whitespaces & Alphabets!!
  -> It should be from one of the below Ranges:
  -> 132.xxx.xxx.xxx
  -> 93.xxx.xxx.xxx
  -> 182.xxx.xxx.xxx
=====
```

New IPAddress For Player Account To Transfer Must Not Be Same As Its Old IPAddress!!  
 Because You Can't Transfer Player Account To The Same Server!!

Enter New IPAddress: 182.2.2.2

- **Username and Password for Administrator** must be “Admin” without any whitespaces and must be exact same as shown (Case Sensitive).

```

***** Operation Menu *****

1. Get Player Status.
2. Suspend Player Account.
3. Exit.

Enter Your Choice: 1
Enter Username: admin56

=====
Username Is Not Valid!!!
=====

Enter Username: Admin
Enter Password: admin newpass

=====
Password Is Not Valid!!!
=====

Enter Password: Admin
Enter IPAddress:

```

## 2) Create New Player Account (Successfully Created New Player Account)

```

***** Operation Menu *****

1. Create New Player Account.
2. Player Sign In.
3. Player Sign Out.
4. Transfer Player Account.
5. Exit.

Enter Your Choice: 1
Enter First Name: rock
Enter Last Name: mcman
Enter Age: 54
Enter Username: rockmcman
Enter Password: 147258369
Enter IPAddress: 182.2.2.2

***** Welcome to AS *****

Player Account "rockmcman" Successfully Created!!!

```

**3) Create New Player Account (Unsuccessful – Player Account with same username Already Exist)**

```
Enter Your Choice: 1
Enter First Name: raj
Enter Last Name: mistry
Enter Age: 25
Enter Username: johncena54
Enter Password: 123456789
Enter IPAddress: 132.2.2.2

***** Welcome to NA *****

Player Account With Username "johncena54" Is Alredy Exist!!!
```

**4) Player Sign In (Unsuccessful - Player Account in which trying to Sign In Not Exist)**

```
Enter Your Choice: 2
Enter Username To SignIn: rockyjohnson
Enter Password To SignIn: RockyJohnson
Enter IPAddress To SignIn: 182.2.2.2

***** Welcome to AS *****

Player Account with Username "rockyjohnson" does not exist!!!
```

**5) Player Sign In (Unsuccessful - Wrong Password Entered For Player Account to Sign In)**

```
Enter Your Choice: 2
Enter Username To SignIn: johncena54
Enter Password To SignIn: 154275663
Enter IPAddress To SignIn: 93.5.4.8

***** Welcome to EU *****

You have Entered Invalid Password For Sign In into Player Account "johncena54" !!!
```

**6) Player Sign In (Successfully Signed in into Player Account)**

```
Enter Your Choice: 2
Enter Username To SignIn: rajmistry2298
Enter Password To SignIn: R@mistry
Enter IPAddress To SignIn: 93.3.3.3

***** Welcome to EU *****

Player Account "rajmistry2298" Successfully Signed In!!!
```

**7) Player Sign In (Trying to sign in into Player Account which is Already Signed In)**

```
Enter Your Choice: 2
Enter Username To SignIn: rajmistry2298
Enter Password To SignIn: R@mistry
Enter IPAddress To SignIn: 93.5.8.9

***** Welcome to EU *****

Player Account "rajmistry2298" Already Signed In!!!
```

**8) Player Sign Out (Unsuccessful - Player Account from which trying to Sign Out Not Exist)**

```
Enter Your Choice: 3
Enter Username To SignOut: ravimistry27
Enter IPAddress To SignOut: 132.5.4.6

***** Welcome to NA *****

Player Account with Username "ravimistry27" does not exist!!!
```

**9) Player Sign Out (Player Account Successfully Signed Out)**

```
Enter Your Choice: 3
Enter Username To SignOut: rajmistry2298
Enter IPAddress To SignOut: 93.5.8.7

***** Welcome to EU *****

Player Account "rajmistry2298" Successfully Signed Out!!!
```

**10) Player Sign Out (Player Account Trying To Sign Out Which Is Already Signed Out)**

```
Enter Your Choice: 3
Enter Username To SignOut: lanabelitchka
Enter IPAddress To SignOut: 182.5.9.8

***** Welcome to AS *****

Player Account "lanabelitchka" Already Signed Out!!!
```



**11) Transfer Player Account (Unsuccessful – Player Account which client trying to transfer does not exist on server)**

```
Enter Your Choice: 4
Enter Username: ravimistry85
Enter Password: 1593574862
Enter Old IPAddress: 132.5.4.8
Enter New IPAddress: 182.2.4.1

***** Welcome to NA *****

Player Account With Username "ravimistry85" does not exist!!!
```

**12) Transfer Player Account (Unsuccessful – Player Account which client trying to transfer is exist on server but wrong password is entered for Account)**

```
Enter Your Choice: 4
Enter Username: rajmistry2298
Enter Password: 1593574682
Enter Old IPAddress: 132.2.2.2
Enter New IPAddress: 93.3.3.3

***** Welcome to NA *****

You have Entered Invalid Password For Transfer Player Account "rajmistry2298" !!!
```

**13) Transfer Player Account (Unsuccessful – Player Account which client trying to transfer, Player Account with same username already exist on destination server)**

```
Enter Your Choice: 4
Enter Username: rajmistry2298
Enter Password: R@mistry
Enter Old IPAddress: 132.2.2.2
Enter New IPAddress: 93.5.4.6

***** Welcome to NA *****

Transfere Unsuccessful Because Player Account With Username "rajmistry2298" Is Alredy Exist On EU!!!
```

**14) Transfer Player Account (Successful – Player account exist and valid password and account with same username not exist on destination server.)**

```
Enter Your Choice: 4
Enter Username: rockmcman
Enter Password: 147258369
Enter Old IPAddress: 182.4.5.6
Enter New IPAddress: 132.2.2.2

***** Welcome to AS *****

Player Account rockmcman Transferred Succesfully To NA !!!
```

**15) Administrator Get Player Status.****Initially each server has 5 Player Accounts:**

```
***** Operation Menu *****
```

1. Get Player Status.
2. Suspend Player Account.
3. Exit.

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IPAddress: 132.4.5.6
```

```
***** Welcome to NA *****
```

```
NA: ONLINE : 0, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
```

**After Creating One Player Account On Asia Server by playerClient:**

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IPAddress: 182.2.5.4
```

```
***** Welcome to AS *****
```

```
AS: ONLINE : 0, OFFLINE : 6. , NA: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
```

**After One Player Account Signed In at Europe Server by playerClient:**

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IPAddress: 93.5.4.6
```

```
***** Welcome to EU *****
```

```
EU: ONLINE : 1, OFFLINE : 4. , NA: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 6.
```

**After Transferring 1 Player Account from Asia Server to North-America Server.**

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IPAddress: 132.2.2.2
```

```
***** Welcome to NA *****
```

```
NA: ONLINE : 0, OFFLINE : 6. , EU: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 5.
```



**16) Suspend Player Account.****Unsuccessful – Player Account with entered username not exist on server.**

```

***** Operation Menu *****

1. Get Player Status.
2. Suspend Player Account.
3. Exit.

Enter Your Choice: 2
Enter Username: Admin
Enter Password: Admin
Enter Username Of Player You Want TO Suspend: rockmcman
Enter IpAddress: 182.2.2.2

***** Welcome to AS *****

Sorry!! Account rockmcman You Want To Suspend Does Not Exist!!

```

**Suspend Player Account successfully.**

```

Enter Your Choice: 2
Enter Username: Admin
Enter Password: Admin
Enter Username Of Player You Want TO Suspend: johncena54
Enter IpAddress: 132.2.2.2

***** Welcome to NA *****

Player Account johncena54 Successfully Suspended!!!

```

**Player Status after Suspending One Account from NA server.**

```

Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IpAddress: 132.7.55.41

***** Welcome to NA *****

NA: ONLINE : 0, OFFLINE : 5. , EU: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 5.

```

**Try to sign in into player account which was suspended by Admin.**

```

Enter Your Choice: 2
Enter Username To SignIn: johncena54
Enter Password To SignIn: J@cena54
Enter IpAddress To SignIn: 132.2.2.2

***** Welcome to NA *****

Player Account with Username "johncena54" does not exist!!!

```

## 17) Multithreaded Test For Checking Concurrency and Atomicity for Transfer Account And Suspend Account.

```
<terminated> MultipleThreadClientTest [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (28 Jun, 2020 2:55:12 PM -
***** This Test Case Is To Check Concurrency *****
-> 3 Threads Will Be Created For NORTH-AMERICA Server Which Will Perform Following Operations.
1. Create Player Account : USERNAME : 'newusername' & PASSWORD : 'newpassword'
2. SignIn In This New Player Account.
3. Get Player Status.
4. Sigout From This New Account.
5. Transfer This New Player Account From NA Server To ASIA Server.
6. Try To Suspend This New Account On NA Server.
7. Suspend Player Account 'rajmistry2298' From NA Server.
8. Try To Transfer Account 'rajmistry2298' To EUROPE Server.
All The Threads will Perform This all Operations And Order OF Threads Is Not Fixed. So After The
Completion Of All This Operations By All Threads Result Should Be As Below :
=>One Thread Will Be Able To Create New Account. Other 2 Will get Already Exist Message.
=>One Thread Will Be Able To SignIn Successfully. Other 2 Will get Already SignedIn Message.
=>Get Player Status: Either All Servers Have Accounts Offline Or All Are Offline Except 1 On NA
=>One Thread Will Be Able To Sign Out Successfully. Other 2 Will get Already Signed Out Message.
=>One Thread Will Be Able To Transfer Successfully. Other 2 Will get Not Exist Message.
=>For Suspend: All Threads Will get Not Exist Message.
=>One Thread Will Be Able To Suspend Account Successfully. Other 2 Will get Not Exist Message.
=>For Transfer: All Threads Will get Not Exist Message.
*****
Thread 't' Started
Thread :13 : Player Account "newusername" Successfully Created!!!
Thread :13 : Player Account "newusername" Successfully Signed In!!!
Player 't1' Thread Started
Thread :16 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :16 : Player Account "newusername" Already Signed In!!!
Player 't2' Thread Started
Thread :19 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :19 : Player Account "newusername" Already Signed In!!!
Thread :16 : NA: ONLINE : 1, OFFLINE : 5. , EU: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 5.
Thread :16 : Player Account "newusername" Successfully Signed Out!!!
Thread :13 : NA: ONLINE : 1, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
Thread :19 : NA: ONLINE : 1, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
Thread :16 : Player Account newusername Transferred Succesfully To AS !!!
Thread :19 : Player Account with Username "newusername" does not exist!!!
Thread :13 : Player Account with Username "newusername" does not exist!!!
Thread :16 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :19 : Player Account With Username "newusername" does not exist!!!
Thread :16 : Player Account rajmistry2298 Successfully Suspended!!!
Thread :13 : Player Account With Username "newusername" does not exist!!!
Thread :19 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :16 : Player Account With Username "rajmistry2298" does not exist!!!
Thread :13 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :19 : Sorry!! Account rajmistry2298 You Want To Suspend Does Not Exist!!
Thread :13 : Sorry!! Account rajmistry2298 You Want To Suspend Does Not Exist!!
Thread :19 : Player Account With Username "rajmistry2298" does not exist!!!
Thread :13 : Player Account With Username "rajmistry2298" does not exist!!!

After All Operations Performed If This Test Is Run on Initial Servers where each server has 5
Initial Player Accounts Then At the end it should be EXPECTED to have player status:
NA : Online:0 Offline:4., EU : Online:0 Offline:5., AS : Online:0 Offline:6

At Last After All The Threads Finished Their Execution Actual Player Status Is:
EU: ONLINE : 0, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 6., NA: ONLINE : 0, OFFLINE : 4.
```

### ➤ Challenges (Most Important / Difficult Part):

- 1) CORBA implementation.
- 2) Implementation of Threads for UDP.
- 3) Implementing synchronization among shared data and avoiding deadlocks.
- 4) UDP implementation of getPlayerStatus() and transferAccount().

[illegible]

- **References:**

- ✓ <http://www.ejbtutorial.com/corba/tutorial-for-corba-hello-world-using-java>
- ✓ <http://bedford-computing.co.uk/learning/wp-content/uploads/2016/03/george-coulouris-distributed-systems-concepts-and-design-5th-edition.pdf>
- ✓ [https://en.wikipedia.org/wiki/Common\\_Object\\_Request\\_Broker\\_Architecture](https://en.wikipedia.org/wiki/Common_Object_Request_Broker_Architecture)
- ✓ [https://en.wikipedia.org/wiki/Interface\\_description\\_language](https://en.wikipedia.org/wiki/Interface_description_language)
- ✓ [https://en.wikipedia.org/wiki/Object\\_request\\_broker](https://en.wikipedia.org/wiki/Object_request_broker)
- ✓ [https://www.corba.org/orb\\_basics.htm](https://www.corba.org/orb_basics.htm)
- ✓ <https://research.cs.queensu.ca/home/xiao/DS/node5.html>
- ✓ <https://www.geeksforgeeks.org/countdownlatch-in-java/>