# COMP6231- Distributed System Design

# Assigment-3

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

**Date:**              12 July 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 as a web service which manages player's status across multiple game servers using Java Webservices. In addition to this we have to make our application properly 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.

## ➢ Web Service:

Web Service is a collection of standards and open protocols that are used for communication between devices or applications. Applications written in different programming language and running on different OS can use web services to communicate and exchange information with each other over the network. Web service provides server interface to client which enables the client to communicate with sever. Client uses this interface operations and send request to server and get response for that particular request through standardized XML messaging system which is usually transmitted over HTTP. All the standard web services are worked using these common components: SOAP (Simple Object Access Protocol) , UDDI (Universal Description, Discovery and Integration) & WSDL (Web Services Description Language).

Types of web services:
1. SOAP Web Services
2. RESTful (Representational State Transfer) Web Services

In this assignment I have implemented first type of web service by using JAX-WS.

## ➢ Design Architecture:
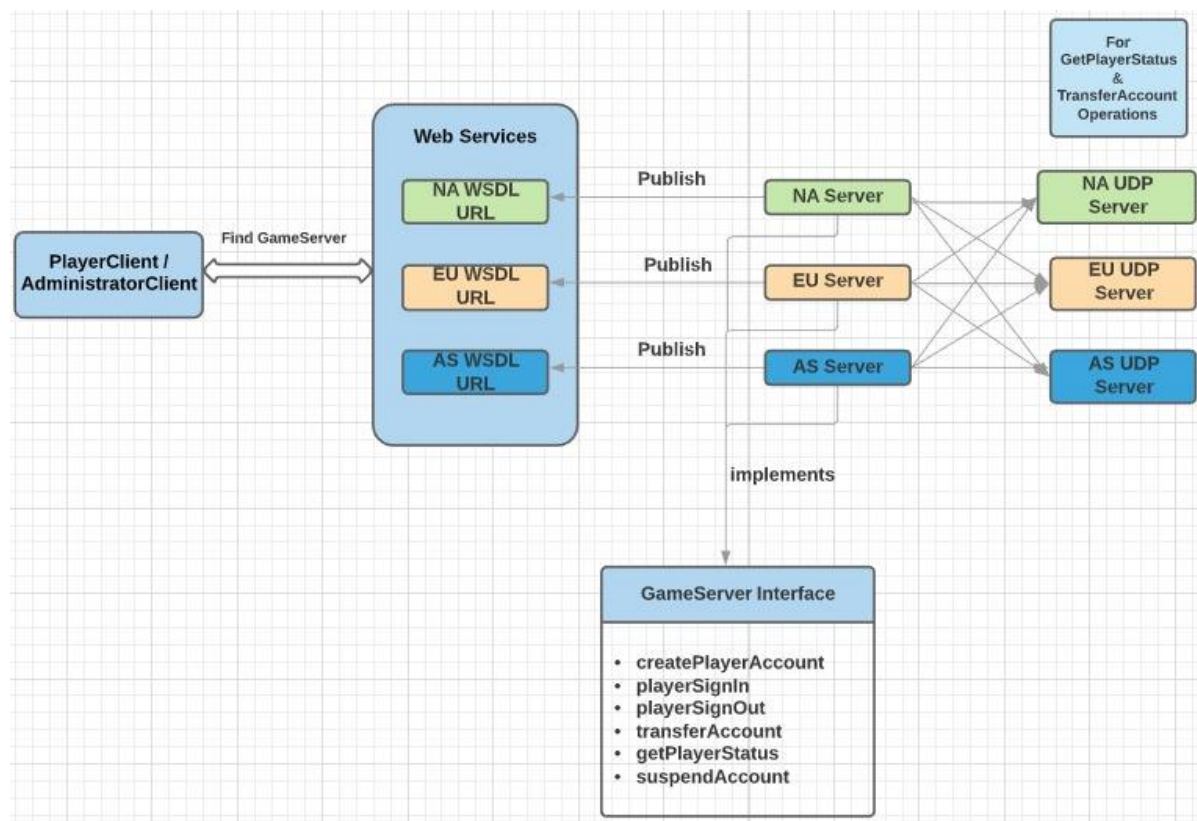
➢ **UDP:**

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.

o   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.

o   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)
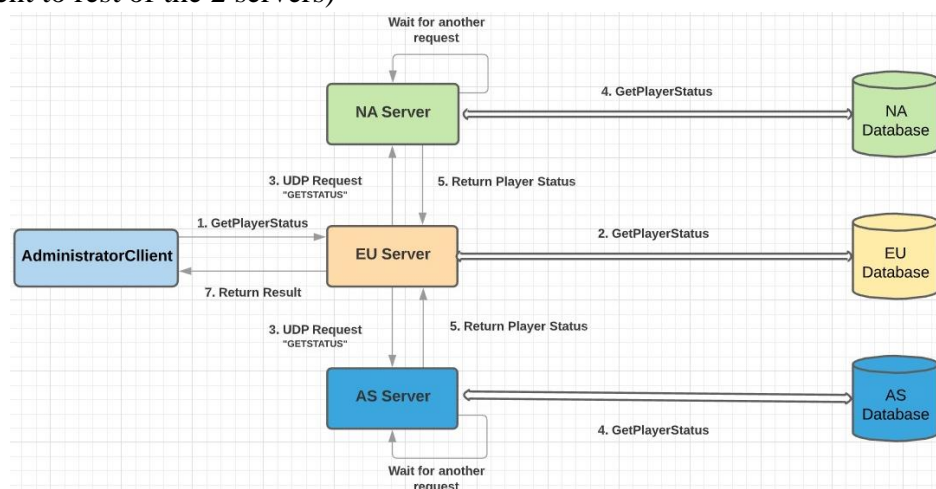


o   In transferAccount() checks the HashMap for the required player account. If found, the server transfers the account to specified remote server by sending UDP request "TRANSFERACCOUNT" and after successfully transferring the record to remote server, it will remove the record from its storage and then 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])
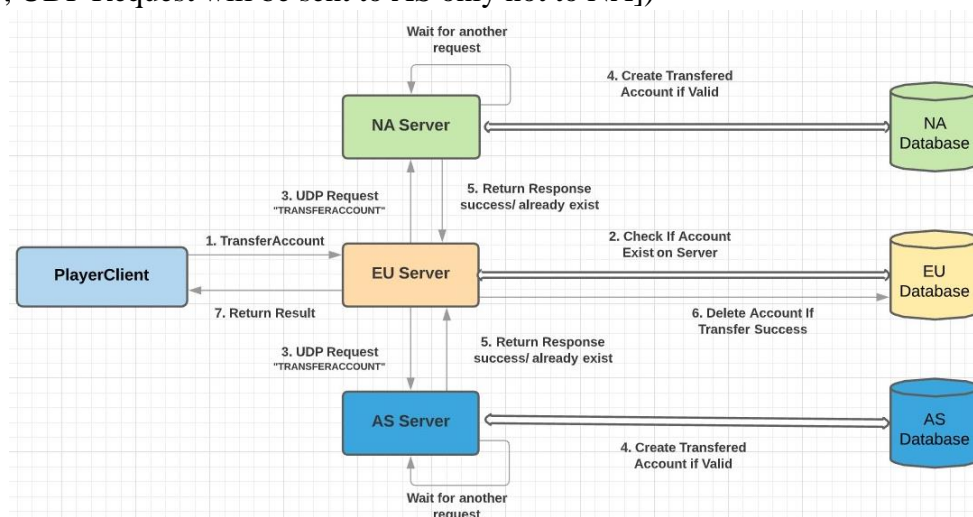
## ➢ Multithreading and Synchronization:

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.

## ➢ Operations:

### 1. Create Player Account.

⇨ While creating player account first all the attributes will be taken as input from user. During the input it will accept only valid inputs for particular field and if invalid input is given it will keep asking for valid input. The validations (All attributes value must be entered. It can't be empty & without any whitespaces) for input are:
1) FirstName & LastName must contain only alphabets.
2) Age must be positive integer.
3) Username must start with alphabet & its length must be between 6 characters to 15 characters.
4) Password length must be minimum 6 characters.
5) IpAddress must be from valid ipV4 format & it should be from one of these range: 132.xxx.xxx.xxx, 182.xxx.xxx.xxx & 93.xxx.xxx.xxx.

⇨ Once all the valid inputs are entered & Player Account with username does not already exist then it will be added to HashMap with Capital-letter of username's first character as Key.

⇨ Message will be return to user as well as it will be written in Log Files.
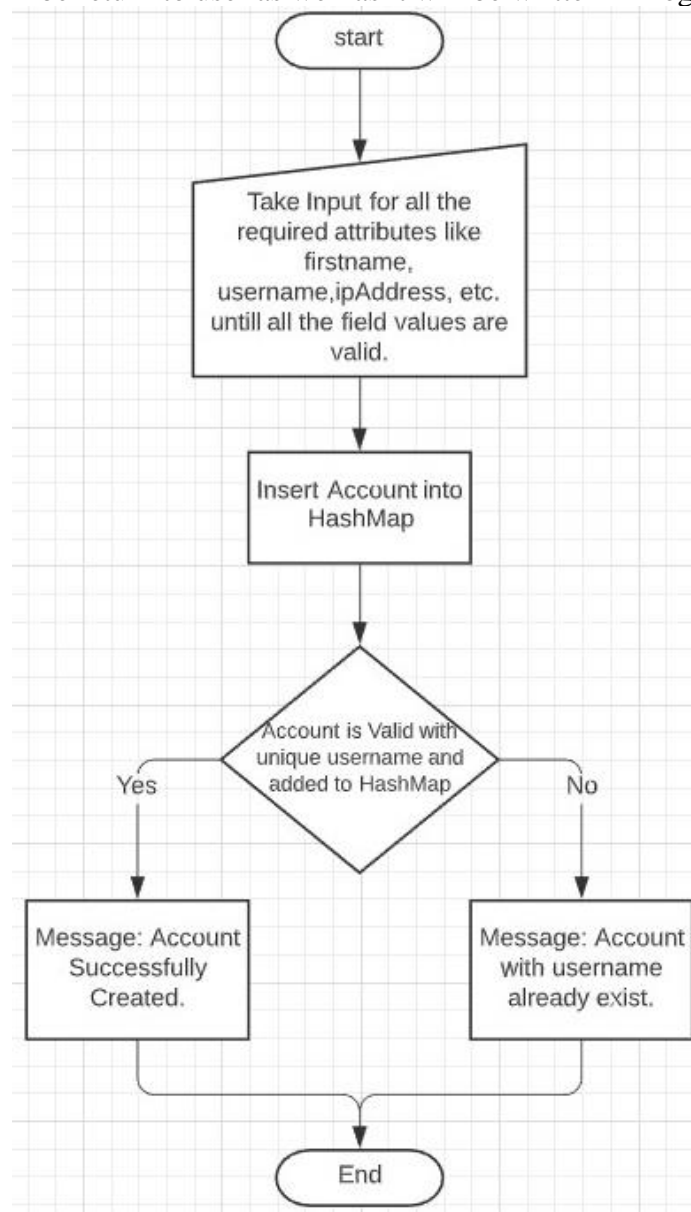
### 2. Player Sign In.

⇨ While client want to sign in into player account program will take username, password and valid IpAddress as input. Then on the server first it will check if there exist a player account with entered username or not. If there is no such a player account with that username it will return message that player account not exist. Otherwise if player account exist it will verify password and if password is correct it will check status and set it online if it's not already online and based on actions it will return success or error message. Message will be return to user as well as it will be written in Log Files.
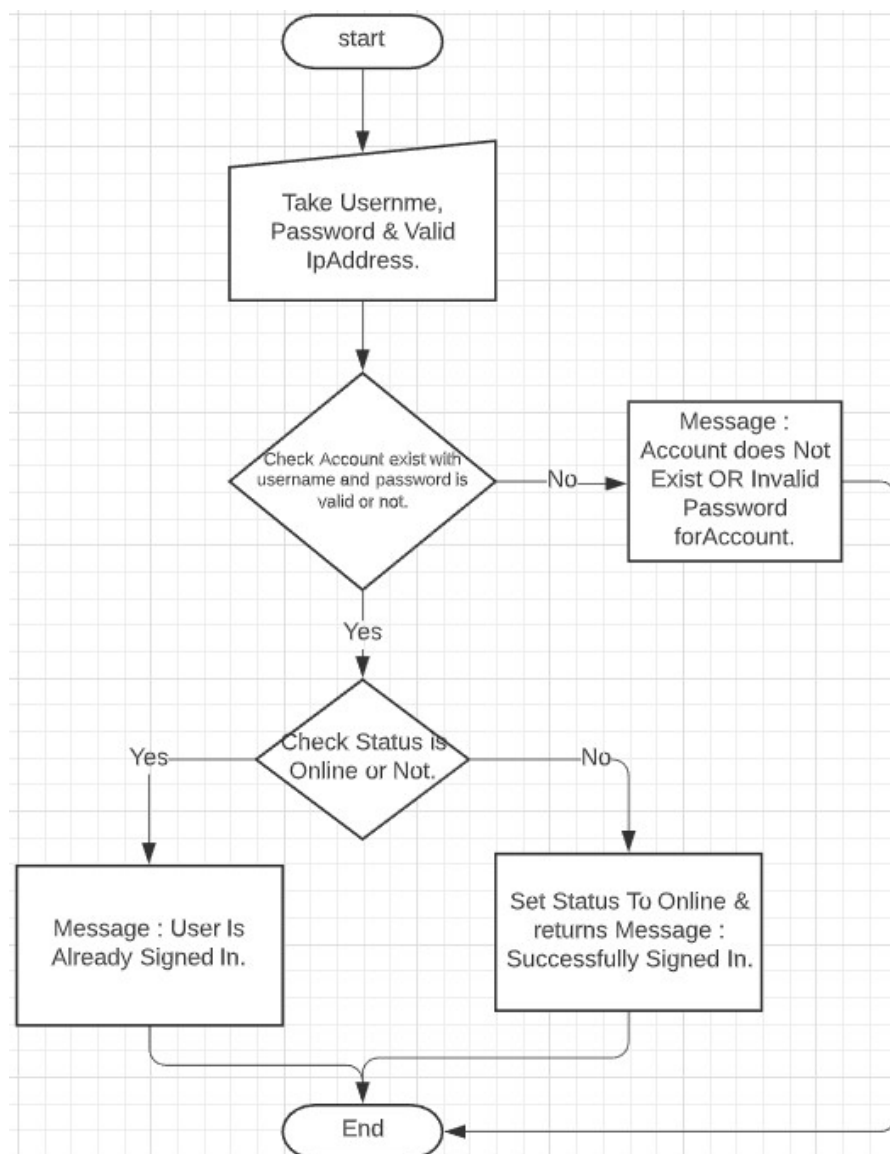


### 3. Player Sign Out.

⇨ While client want to sign out from player account program will take username and valid IpAddress as input. Then on the server first it will check if there exist a player account with entered username or not. If there is no such a player account with that username it will return message that player account not exist. Otherwise if player account exist it will check status and set it offline if it's not already online and based on the actions it will return success message.

⇨ Message will be return to user as well as it will be written in Log Files.
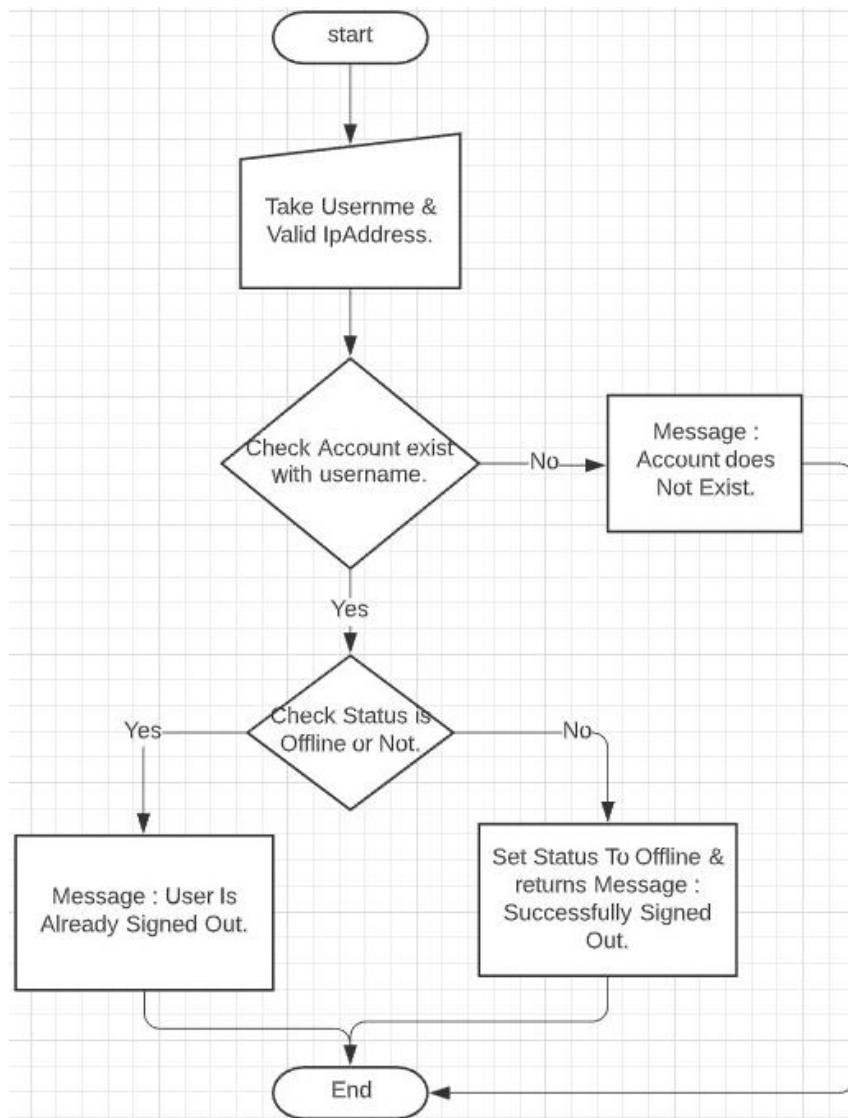
(**Note:** For Transfer Account and Get player status operations the way it works is described in UDP description above where it takes required inputs from user and verifies them and if all valid it will send particular UDP request to other server(s) based on operation. Then server will perform operation based on received request and returns result to server who sent request and then these results are returned to client and also written into log files.)

4. **Suspend Player Account.**
⇨ While Administrator want to suspend player account program will verify Admin account credentials and if they are valid it will take username of player account which he wants to suspend and valid IpAddress as input. Then on the server first it will check if there exist a player account with entered username or not. If there is no such a player account with that username it will return message that player account not exist. Otherwise if player account exist it will remove it from HashMap and return success message. Message will be return to user as well as it will be written in Log Files.
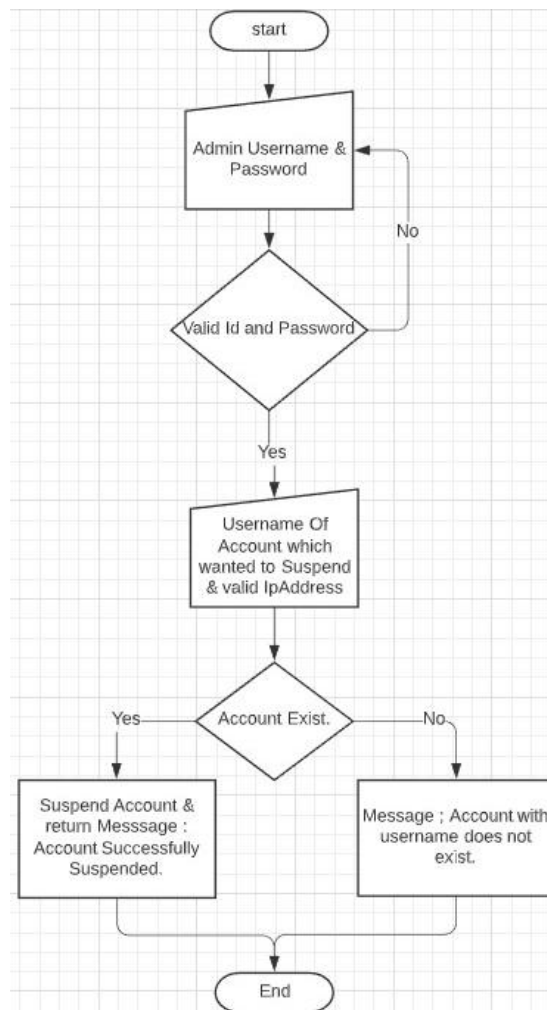
- ➤ **Test Cases:**

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
============================================================================
```

➢ **New IP Address for Transfer Account** must be valid Ip Address as above and it must be different from old Ip Address as Account cannot be transfer to same server.

```
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!!
============================================================================
```

➢ **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: Shivam
Enter Last Name: Hadvani
Enter Age: 24
Enter Username: shivamhadvani
Enter Password: 147258369
Enter IpAddress: 182.2.55.44

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

Player Account "shivamhadvani" Successfully Created!!!
```

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

```
Enter Your Choice: 1
Enter First Name: Paul
Enter Last Name: Heyman
Enter Age: 29
Enter Username: paulheymanguy
Enter Password: P@heyman
Enter IpAddress: 93.95.88.6

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

Player Account With Username "paulheymanguy" 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: rockymcman
Enter Password To SignIn: 159357rocky
Enter IpAddress To SignIn: 132.16.84.55

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

Player Acount with Username "rockymcman" 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: rajmistry2298
Enter Password To SignIn: RWrongPass
Enter IpAddress To SignIn: 93.44.55.7

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

You have Entered Invalid Password For Sigin In into Player Account "rajmistry2298" !!!
```

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

```
Enter Your Choice: 2
Enter Username To SignIn: lanabelitchka
Enter Password To SignIn: L@belitchka
Enter IpAddress To SignIn: 132.1.2.2

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

Player Account "lanabelitchka" 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: lanabelitchka
Enter Password To SignIn: L@belitchka
Enter IpAddress To SignIn: 132.2.5.4

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

Player Account "lanabelitchka" 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: UserNotExist
Enter IpAddress To SignOut: 182.22.66.55

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

Player Acount with Username "UserNotExist" does not exist!!!
```

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

```
Enter Your Choice: 3
Enter Username To SignOut: lanabelitchka
Enter IpAddress To SignOut: 132.22.22.22

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

Player Account "lanabelitchka" 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: username
Enter IpAddress To SignOut: 93.4.55.66

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

Player Account "username" 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: rockNOTEXIST
Enter Password: ROCKPASS
Enter Old IpAddress: 132.44.55.3
Enter New IpAddress: 182.5.99.3

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

Player Acount With Username "rockNOTEXIST" 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: johncena54
Enter Password: JJJWRONGGGGG
Enter Old IpAddress: 93.55.4.2
Enter New IpAddress: 182.4.6.5

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

You have Entered Invalid Password For Transfer Player Account "johncena54" !!!
```

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: paulheymanguy
Enter Password: P@heyman
Enter Old IpAddress: 132.22.55.44
Enter New IpAddress: 182.4.5.6

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

Transfere Unsuccessful Because Player Account With Username "paulheymanguy" Is Alredy Exist On AS!!!
```

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: shivamhadvani
Enter Password: 147258369
Enter Old IpAddress: 182.2.2.5
Enter New IpAddress: 132.5.4.6

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

Player Account shivamhadvani Transfered 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: 132.4.5.8

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

NA: ONLINE : 0, OFFLINE : 5. , EU: ONLINE : 0, OFFLINE : 5., AS: 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: 182.7.88.9

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

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

### After One Player Account Signed In at North-America Server by playerClient:

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

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

NA: ONLINE : 1, OFFLINE : 4. , EU: 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.**

```
Enter Your Choice: 2
Enter Username: Admin
Enter Password: Admin
Enter Username Of Player You Want TO Suspend: notvaliduser
Enter IpAddress: 182.4.5.6

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

Sorry!! Account notvaliduser 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: rajmistry2298
Enter IpAddress: 93.66.33.55

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

Player Account rajmistry2298 Successfully Suspended!!!
```

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

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IpAddress: 93.55.4.8

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

EU: ONLINE : 0, OFFLINE : 4. , NA: ONLINE : 0, OFFLINE : 6., 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: rajmistry2298
Enter Password To SignIn: R@mistry
Enter IpAddress To SignIn: 93.66.55.88

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

Player Acount with Username "rajmistry2298" does not exist!!!
```

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

```
<terminated> MultipleThreadClientTest (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe  (11 Jul, 2020 10:43:52 PM – 10:43:55 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 Exisit 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
Player 't1' Thread Started
Thread :12 : Player Account "newusername" Successfully Created!!!
Thread :13 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :12 : Player Account "newusername" Successfully Signed In!!!
Player 't2' Thread Started
Thread :13 : Player Account "newusername" Already Signed In!!!
Thread :14 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :14 : Player Account "newusername" Already Signed In!!!
Thread :12 : NA: ONLINE : 1, OFFLINE : 5. , EU: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 5.
Thread :13 : NA: ONLINE : 1, OFFLINE : 5. , EU: ONLINE : 0, OFFLINE : 5., AS: ONLINE : 0, OFFLINE : 5.
Thread :12 : Player Account "newusername" Successfully Signed Out!!!
Thread :14 : NA: ONLINE : 1, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
Thread :13 : Player Account "newusername" Already Signed Out!!!
Thread :12 : Player Account newusername Transfered Succesfully To AS !!!
Thread :13 : Player Acount With Username "newusername" does not exist!!!
Thread :14 : Player Acount with Username "newusername" does not exist!!!
Thread :12 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :14 : Player Acount With Username "newusername" does not exist!!!
Thread :13 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :12 : Player Account rajmistry2298 Successfully Suspended!!!
Thread :14 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :13 : Player Account rajmistry2298 Successfully Suspended!!!
Thread :14 : Sorry!! Account rajmistry2298 You Want To Suspend Does Not Exist!!
Thread :12 : Player Acount With Username "rajmistry2298" does not exist!!!
Thread :13 : Player Acount With Username "rajmistry2298" does not exist!!!
Thread :14 : Player Acount 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) Web service implementation using JAX-WS, wsgen & wsimport.
   2) Implementation of Threads for UDP.
   3) Implementing synchronization among shared data and avoiding deadlocks.
   4) UDP implementation of getPlayerStatus() and transferAccount().

➢ **Class Diagram:**



⇨ **NAServer, EUServer & ASServer:** These classes implement GameServer interface, start UDP server and create web services for North-America, Europe and Asia location respectively and publish the web services on below WSDL URLs,
http://localhost:8080/NASever/NA?wsdl
http://localhost:8081/EUServer/EU?wsdl
http://localhost:8082/ASServer/AS?wsdl
(We can see this in browser while servers are running)

⇨ **GameServer.java (Service Endpoint Interface):** This is an interface which declares below methods that client can invoke on the service
  o  String createPlayerAccount (String firstName, String lastName, String age, String username, String password, String ipAddress);
  o  String playerSignIn (String username, String password);
  o  String playerSignOut (String username);
  o  String transferAccount (String username, String password, String newIpAddress);
  o  String getPlayerStatus ();
  o  String suspendAccount (String usernameToSuspend);

⇨ **Player.java (Model):** This is a java bean class to hold following data of players,
First Name, Last Name, Age, Username, Password, Status and Ip Address.

⇨ **NAServerService, EUServerService, ASServerService, GameServer, ObjectFactory, CreatePlayerAccount, CreatePlayerAccountResponse, PlayerSignIn, PlayerSignInResponse, PlayerSignOut, PlayerSignOutResponse, TransferAccount, TransferAccountResponse, GetPlayerStatus, GetPlayerStatusResponse, SuspendAccount & SuspendAccountResponse :**
This All Java files are generated by wsgen and wsimport commands which are used in order to support webservices published by servers.

➢ **References:**

✓ https://www.tutorialspoint.com/webservices/what_are_web_services.htm
✓ https://docs.oracle.com/cd/E19798-01/821-1841/gijti/index.html
✓ http://bedford-computing.co.uk/learning/wp-content/uploads/2016/03/george-coulouris-distributed-systems-concepts-and-design-5th-edition.pdf
✓ https://www.infoworld.com/article/3215966/web-services-in-java-se-part-2-creating-soap-web-services.html
✓ https://www.infoworld.com/article/3215966/web-services-in-java-se-part-2-creating-soap-web-services.html?page=2
✓ https://www.youtube.com/watch?v=od6fNiegu-Q
✓ https://www.youtube.com/watch?v=-3w6LBl8E-8