COMP6231- Distributed System Design

Project

Software Failure Tolerant/Highly Available Distributed Player Status System (DPSS)

Date: 12 August 2020.

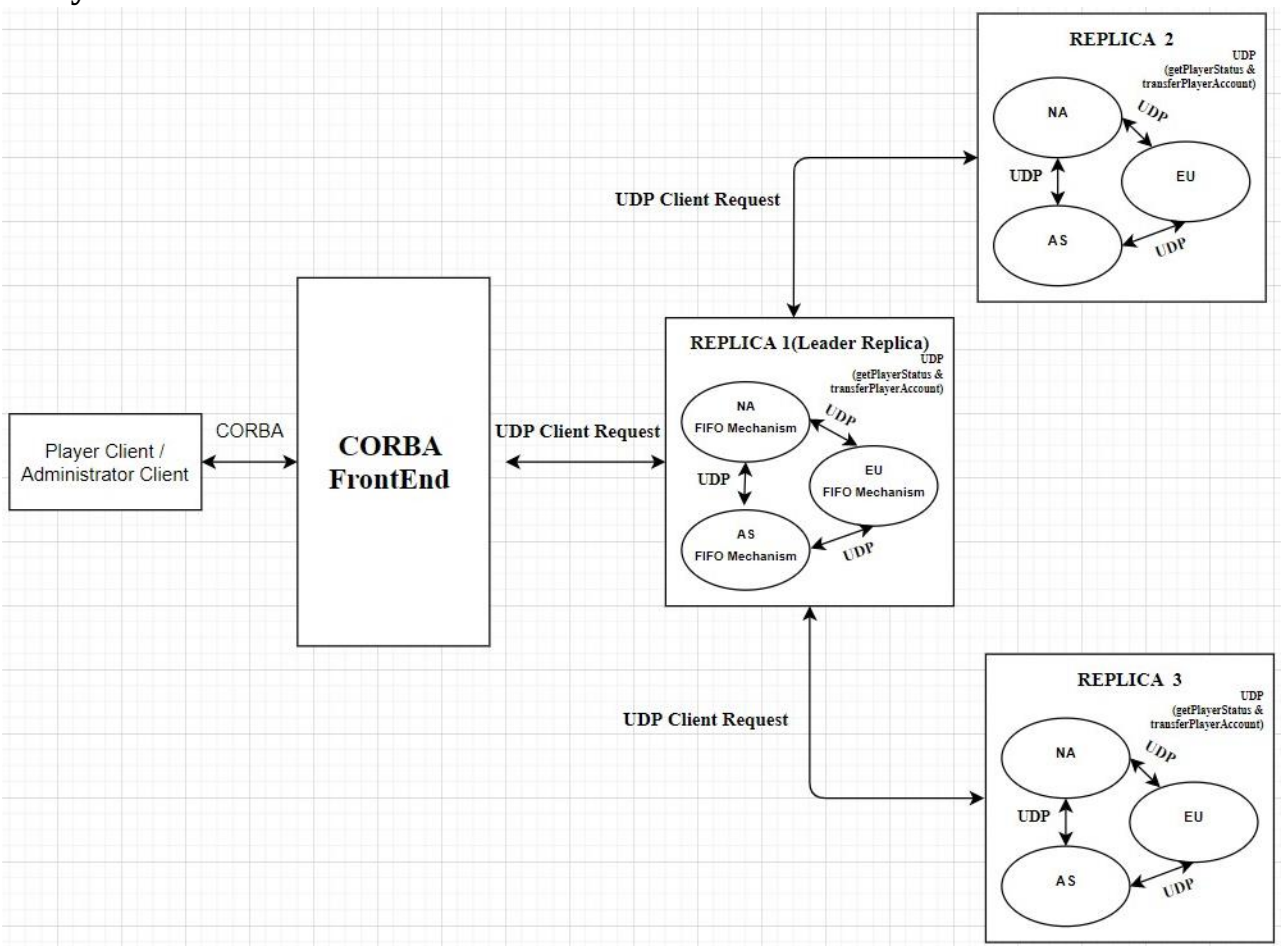
Recipient: Prof. Mohammed Taleb

Student Name: Raj Mistry (40119206)

➢ Introduction:

To implement a highly available CORBA Distributed Player Status System, which tolerates software failure (non-malicious Byzantine) using active replication. There are three replicas which contains a copy of each 3 servers. One of the replicas works as a leader, which takes all the request from client and will send it to all other replicas using FIFO broadcast which sends client requests in same sequence order of received operation, then it receives response from them and sends single correct response to client. Also, leader informs Replica Manager about software failure. The servers and replicas communicate via unreliable UDP protocol.

➢ System Architecture:

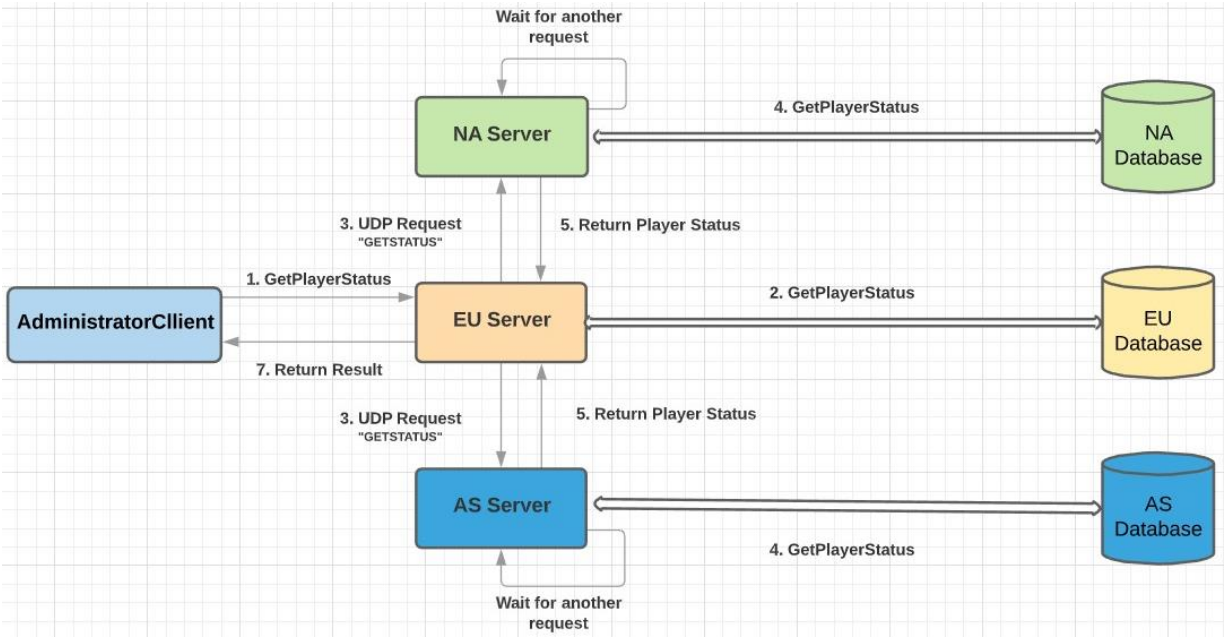


➢ UDP (Inter Server Communication):

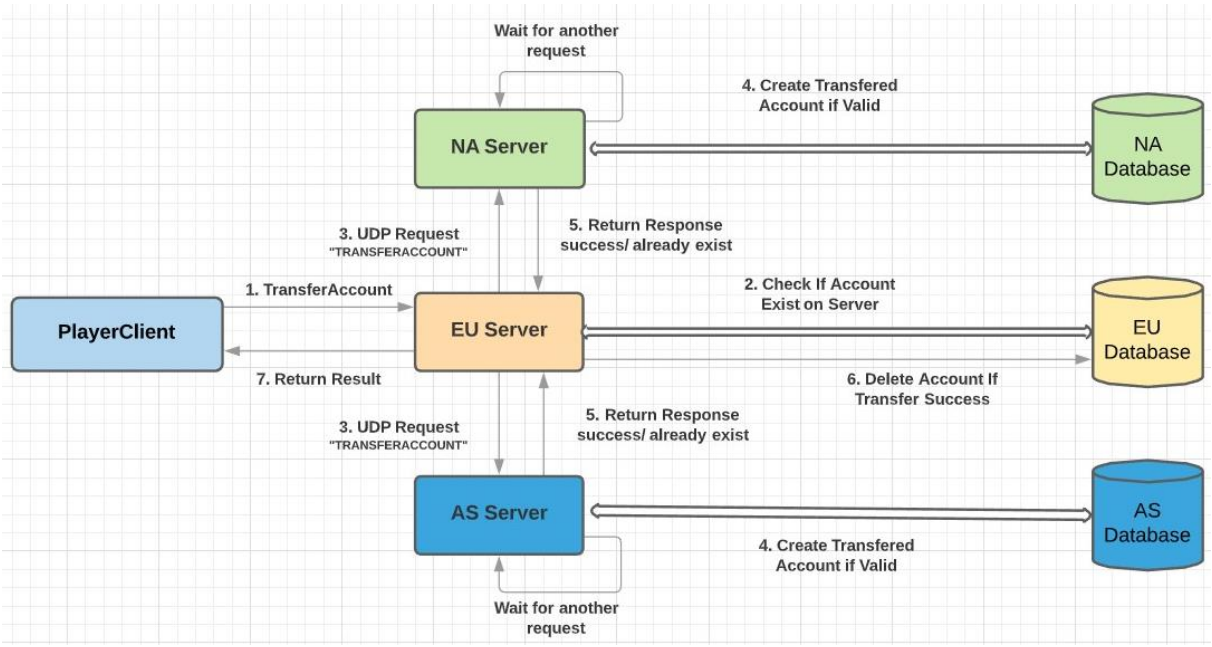
The User datagram Protocol is a transport layer protocol. It is connectionless and unreliable protocol. To communicate using UDP it is not required to establish connection prior to data transfer. A datagram is an independent, self-contained message sent over the network whose arrival, arrival time, and 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.

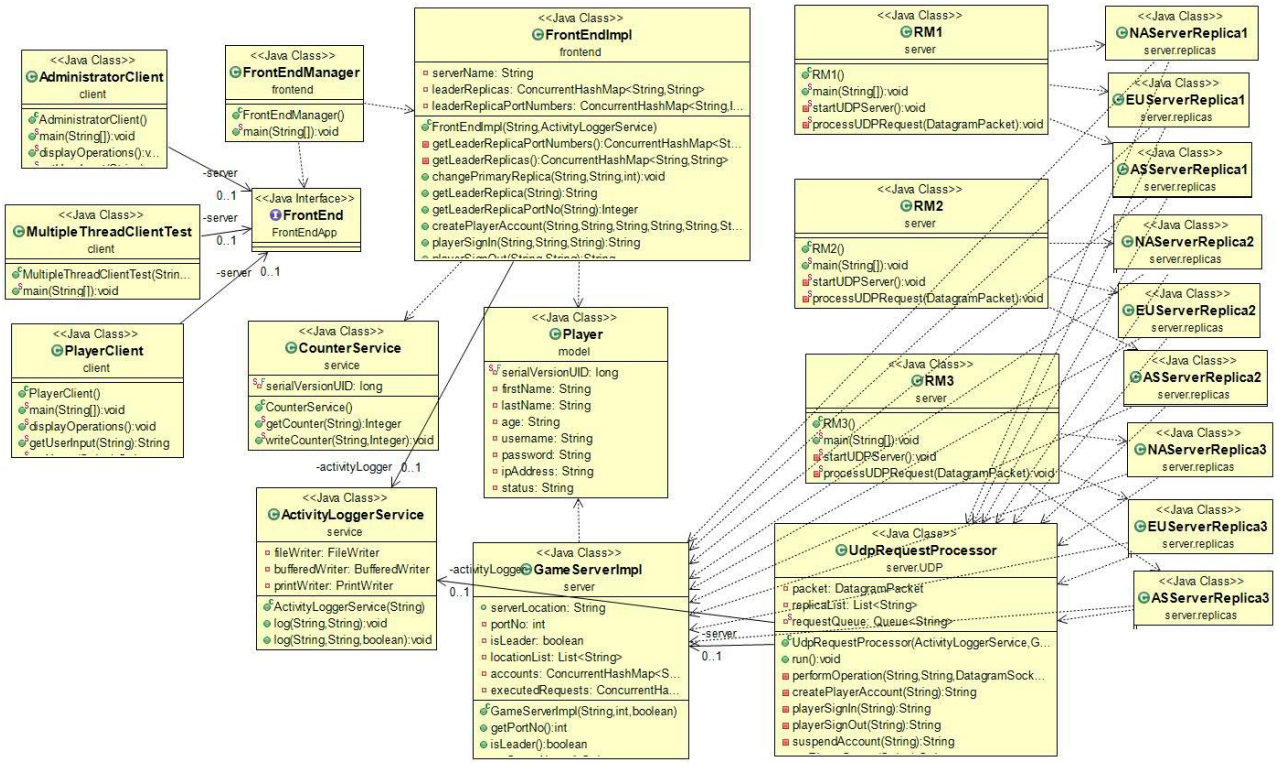
- In getPlayerStatus () the current server communicates with other servers using UDP request "GETSTATUS" to fetch the player accounts status on corresponding server and returns server specific online and offline player accounts count. (In this UDP request will be sent to rest of the 2 servers)



- 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])



➢ Class Diagram:



➢ Classes Description:

- **PlayerClient.java , AdministratorClient.java & MultipleThreadClient.java:**
  These are the client classes which communicates with FrontEnd through CORBA. Player Client can perform 4 operations and Admin client can perform 2 operations which are: createPlayerAccount, playerSignIn, playerSignOut, transferPlayerAccount, getPlayerStatus and suspendPlayerAccount. From the client's end, the whole system's functionalities are encapsulated while the complexities are hidden from the end users.

- **FrontEnd.java:**
  This is an interface which declares below methods that client can invoke
  - String createPlayerAccount (String firstName, String lastName, String age, String username, String password, String ipAddress);
  - String playerSignIn (String username, String password);
  - String playerSignOut (String username);
  - String transferAccount (String username, String password, String newIpAddress);
  - String getPlayerStatus ();
  - String suspendAccount (String usernameToSuspend);

- **FrontEndImpl.java:**
  This Class implements all the operations of FrontEnd interface. When client wants to perform any operation, it will send the request to FE through CORBA. FE will generate request id and append it to request and send it to appropriate server in Leader Replica.

- **NAServerReplica1/2/3 .java, EUServerReplica1/2/3 .java, ASServerReplica1/2/3 .java:**
  These are the server classes which can work leader as well as normal server. The only difference between leader and normal server is one Boolean Value "isLeader" set true or false. Each server replica will start its own UDP server. In order to start all these server replicas, we need to run their respective Replica Manager (RM) java files.
  For starting Replica 1 servers run: **RM1.java**
  For starting Replica 2 servers run: **RM2.java**
  For starting Replica 3 servers run: **RM3.java**

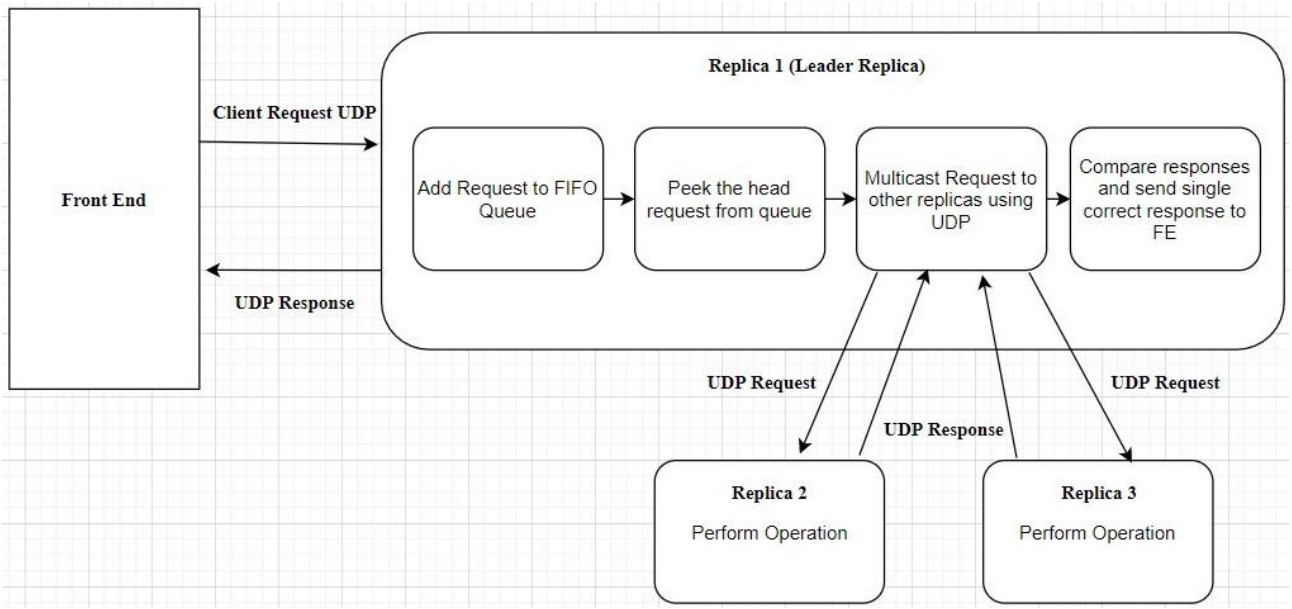
- **GameServeImpl.java:**
  This class is used to perform all the operations at server side. It contains implementation of all the client operations.

- **CounterService.java**
  This class is serializable class used to generate the request id for client requests coming at FE and it keeps id unique among all servers. This class will read latest counter from *RequestCounter.txt*

- **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.

➢ FIFO Mechanism

Whenever FE sends client request to Leader replica, it will add that request into FIFO queue. Here I have used *ConcurrentLinkedQueue* data structure to create FIFO queue as it is more thread safe in multithreading environment. Leader replica will peek head request from the queue and perform the operation and simultaneously it will multicast that request to other 2 replicas through UDP, after receiving response from other replicas leader replica will compare all the responses including its response and will send correct single response to FrontEnd.



➢ UDP Reliability

In this project I have used socket timeout and maximum 3 attempt to perform any operation to make the UDP reliable. While sending UDP request I have set socket timeout of 5 seconds, so that sender server will wait for 5 seconds for getting response and after that it will throw timeout exception. In socket timeout exception code block I have decreased the attempt count and resend that request. This process will repeat 3 time as maximum attempts are set to 3. This will help UDP server to maintain its reliability and avoid one UDP call of sending acknowledgement from destination server to source server which in turn reduce the delay of UDP calls.

➢ Data Structures:

- **Account Storage:**
  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>>();

- **FIFO Mechanism**
  I have used *ConcurrentLinkedQueue* in order to create FIFO queue as it is more thread safe in multithreading and concurrent process environment.
  Syntax:
  Queue<String> requestQueue = new *ConcurrentLinkedQueue<String>();*

- **Response History**
  I am maintaining response history at each replica to avoid duplicity of accounts. To achieve this, I have used ConcurrentHashMap data structure.
  Syntax:
  ConcuurentHashMap<String, String> executedRequests = new ConcuurentHashMap <String, String>();

➢ 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.

➢ Operations:

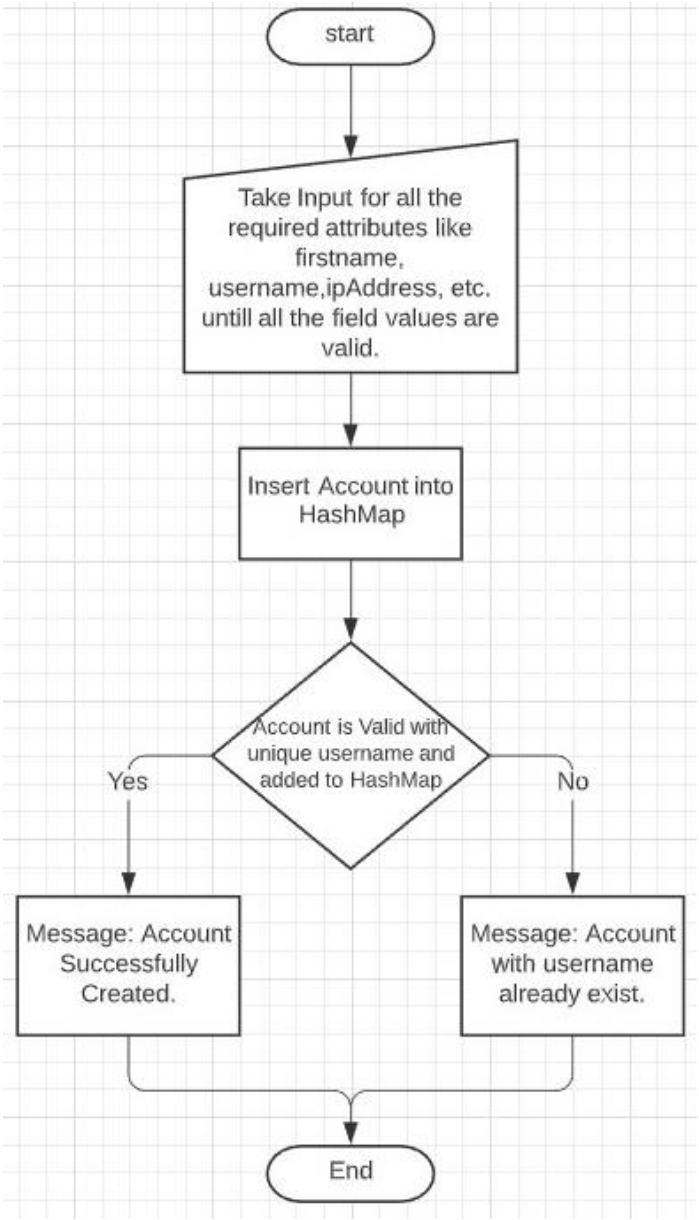
1. Create Player Account.

⇨ 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.

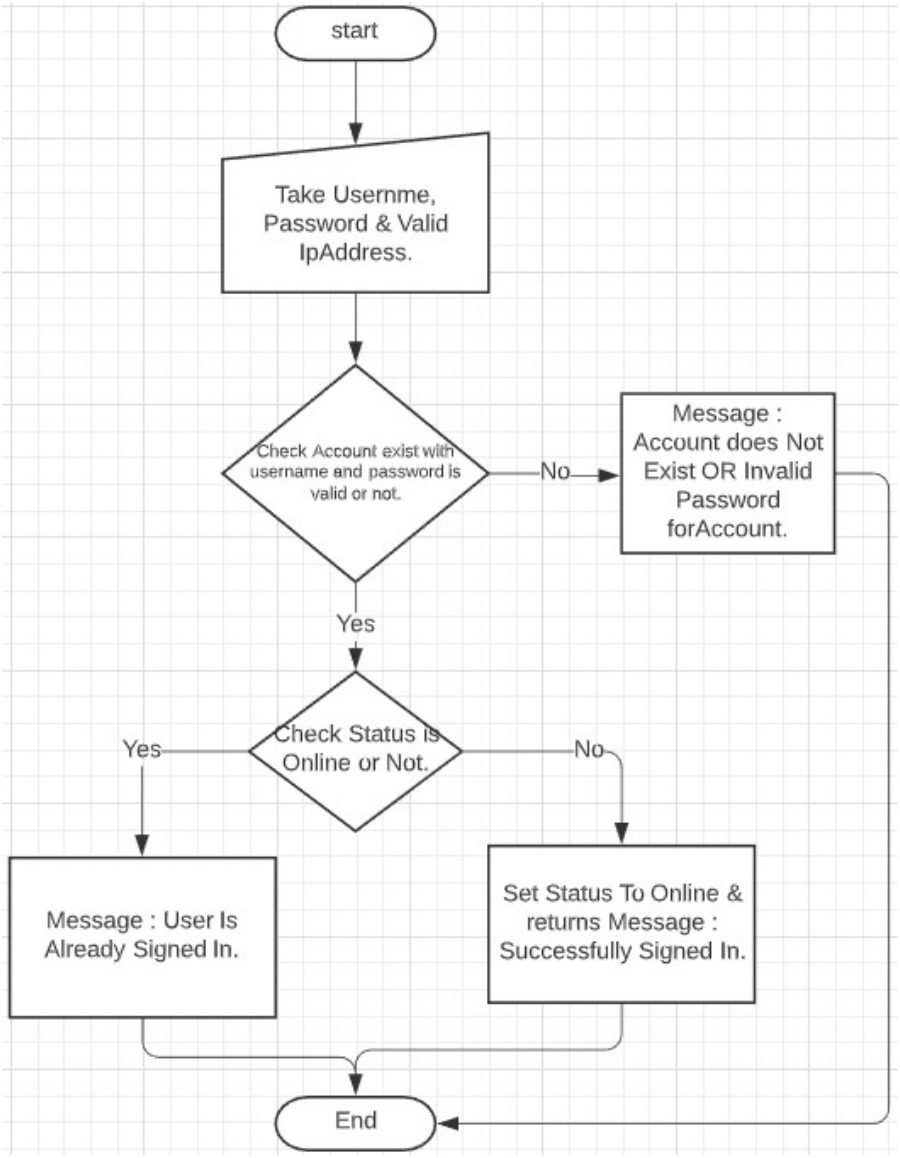
⇨ Once all the valid inputs are entered & Player Account with username does not already exist then it will be added to HashMap with Capital-letter of username's first character as Key.

⇨ Message will be return to user as well as it will be written in Log Files.



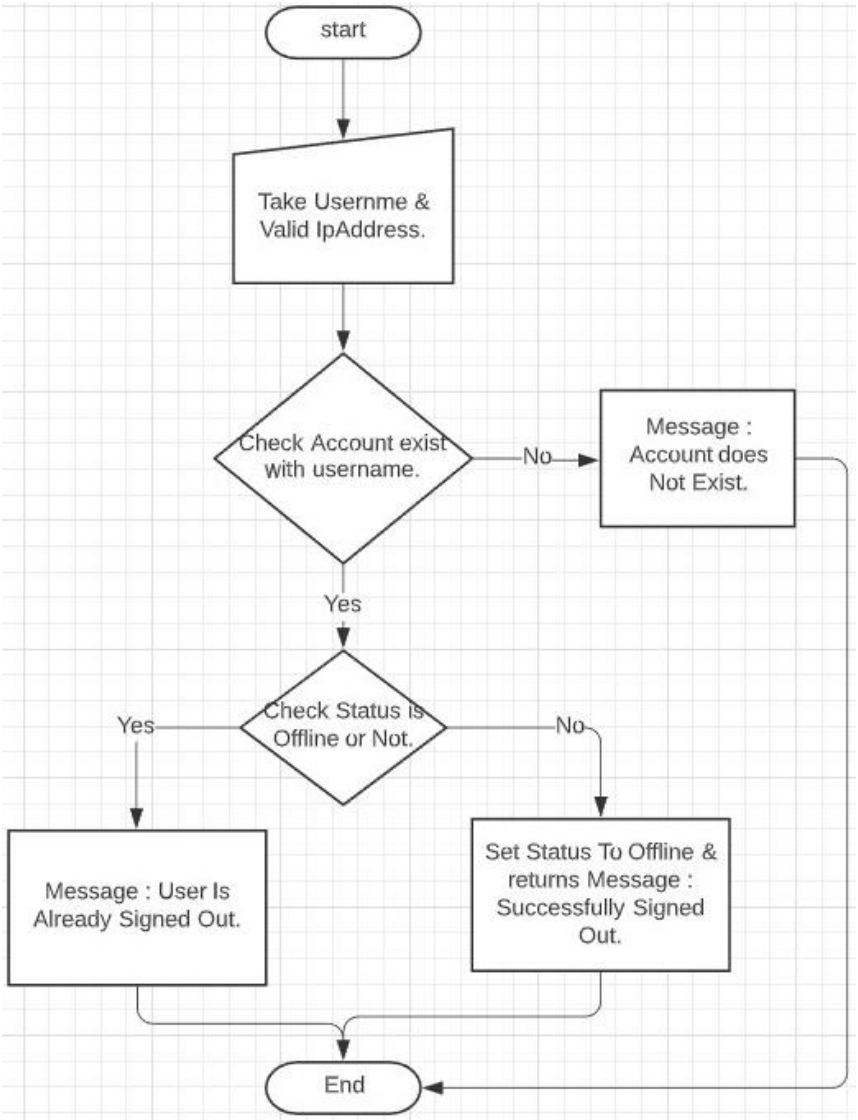
2. Player Sign In.

⇨ 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.

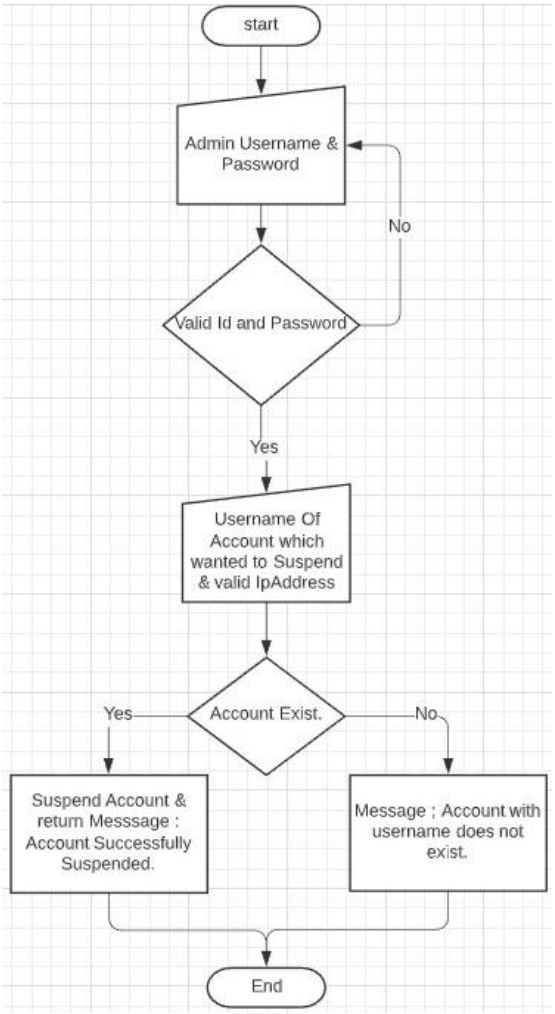
- ⇨ While client want to sign out from player account program will take username and valid IpAddress as input. Then on the server first it will check if there exist a player account with entered username or not. If there is no such a player account with that username it will return message that player 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
================================================================================

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: Ravi
Enter Last Name: Mistry
Enter Age: 24
Enter Username: ravimistry
Enter Password: R@mistry
Enter IpAddress: 182.2.2.2

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

12 Aug, 2020 8:07:44 PM => INFO : Player Account "ravimistry" 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: 22
Enter Username: rajmistry2298
Enter Password: R@mistry
Enter IpAddress: 93.3.5.8

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

12 Aug, 2020 8:10:48 PM => INFO : Player Account With Username "rajmistry2298" 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: notvaliduser
Enter Password To SignIn: RockMcman
Enter IpAddress To SignIn: 132.5.4.6

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

12 Aug, 2020 8:12:49 PM => INFO : Player Acount with Username "notvaliduser" 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: WrongPass
Enter IpAddress To SignIn: 182.2.2.2

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

12 Aug, 2020 8:14:15 PM => INFO : 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: rajmistry2298
Enter Password To SignIn: R@mistry
Enter IpAddress To SignIn: 93.6.5.9

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

12 Aug, 2020 8:15:48 PM => INFO : 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: 93.3.3.3

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

12 Aug, 2020 8:18:07 PM => INFO : 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: usernoteist
Enter IpAddress To SignOut: 132.5.8.9

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

12 Aug, 2020 8:19:27 PM => INFO : Player Acount with Username "usernoteist" 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: 93.93.6.5

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

12 Aug, 2020 8:20:19 PM => INFO : 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: 182.5.6.9

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

12 Aug, 2020 8:21:18 PM => INFO : 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: usernotonNA
Enter Password: 145987632
Enter Old IpAddress: 132.2.2.2
Enter New IpAddress: 182.5.5.5

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

12 Aug, 2020 8:22:35 PM => INFO : Player Acount With Username "usernotonNA" 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: 1574289366
Enter Old IpAddress: 182.5.6.6
Enter New IpAddress: 93.6.6.6

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

12 Aug, 2020 8:24:10 PM => INFO : 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: 93.6.6.5
Enter New IpAddress: 132.2.2.2

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

12 Aug, 2020 8:25:21 PM => INFO : Transfere Unsuccessful Because Player Account With Username "rajmistry2298" Is Alredy Exist On NA!!!
```

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: ravimistry
Enter Password: R@mistry
Enter Old IpAddress: 182.2.5.4
Enter New IpAddress: 132.2.2.2

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

12 Aug, 2020 8:26:47 PM => INFO : Player Account ravimistry 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: 132.4.5.6

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

12 Aug, 2020 8:05:28 PM => INFO : 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: 182.2.2.5

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

12 Aug, 2020 8:09:13 PM => INFO : AS: ONLINE : 0, OFFLINE : 6. , EU: ONLINE : 0, OFFLINE : 5., NA: 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: 93.5.6.5

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

12 Aug, 2020 8:16:48 PM => INFO : EU: ONLINE : 1, OFFLINE : 4. , AS: ONLINE : 0, OFFLINE : 6., NA: ONLINE : 0, OFFLINE : 5.
```

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

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IpAddress: 132.2.2.5

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

12 Aug, 2020 8:27:50 PM => INFO : 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: WrongUser
Enter IpAddress: 182.6.9.5

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

12 Aug, 2020 8:30:00 PM => INFO : Sorry!! Account WrongUser 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: ravimistry
Enter IpAddress: 132.2.2.5

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

12 Aug, 2020 8:31:06 PM => INFO : Player Account ravimistry Successfully Suspended!!!
```

Player Status after Suspending One Account from NA server.

```
Enter Your Choice: 1
Enter Username: Admin
Enter Password: Admin
Enter IpAddress: 132.8.7.9

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

12 Aug, 2020 8:31:55 PM => INFO : NA: ONLINE : 0, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
```

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

```
Enter Your Choice: 2
Enter Username To SignIn: ravimistry
Enter Password To SignIn: R@mistry
Enter IpAddress To SignIn: 132.2.2.2

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

12 Aug, 2020 8:33:07 PM => INFO : Player Acount with Username "ravimistry" does not exist!!!
```

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

```
************************** 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
Player 't2' Thread Started
Thread :13 : Player Account "newusername" Successfully Created!!!
Thread :16 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :19 : Player Account With Username "newusername" Is Alredy Exist!!!
Thread :13 : Player Account "newusername" Successfully Signed In!!!
Thread :16 : Player Account "newusername" Already Signed In!!!
Thread :19 : Player Account "newusername" Already Signed In!!!
Thread :13 : NA: ONLINE : 1, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
Thread :16 : NA: ONLINE : 1, OFFLINE : 5. , AS: ONLINE : 0, OFFLINE : 5., EU: ONLINE : 0, OFFLINE : 5.
Thread :13 : Player Account "newusername" Successfully Signed Out!!!
Thread :19 : Player Account "newusername" Successfully Signed Out!!!
Thread :16 : Player Account "newusername" Already Signed Out!!!
Thread :13 : Player Account newusername Transfered Succesfully To AS !!!
Thread :19 : Player Acount with Username "newusername" does not exist!!!
Thread :16 : Player Acount With Username "newusername" does not exist!!!
Thread :13 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :16 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :19 : Player Acount With Username "newusername" does not exist!!!
Thread :13 : Player Account rajmistry2298 Successfully Suspended!!!
Thread :16 : Sorry!! Account rajmistry2298 You Want To Suspend Does Not Exist!!
Thread :19 : Sorry!! Account newusername You Want To Suspend Does Not Exist!!
Thread :13 : Player Acount With Username "rajmistry2298" does not exist!!!
Thread :16 : Player Acount With Username "rajmistry2298" does not exist!!!
Thread :19 : Sorry!! Account rajmistry2298 You Want To Suspend Does Not Exist!!
Thread :19 : 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. , NA: ONLINE : 0, OFFLINE : 4., AS: ONLINE : 0, OFFLINE : 6.
```

18) **Byzantine Error:** Here I have Implemented Byzantine error Test scenario on replica 3 on request number 4, 5 & 6 (It will be printed by Leader but client will get correct result). At Request number 6 it will be 3rd successive byzantine error. So Leader will inform Replica Manager 3. After that Replica Manager 3 will send message to Replica Manager 2 to give correct Data. So, RM 2 will inform its 3 servers to send correct data to RM 3, then RM 3 will send this to its particular server.

```
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from FrontEnd Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT request has been sent to EU_Replica_3 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT request has been sent to EU_Replica_2 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT response has been received from EU_Replica_2 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT response has been received from EU_Replica_3 Server.
12 Aug, 2020 8:10:48 PM => ERROR : ===== BYZANTINE ERROR OCCURED AT REPLICA 3 =====
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT response has been sent to FrontEnd Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN request has been received from FrontEnd Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN request has been sent to NA_Replica_2 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN request has been sent to NA_Replica_3 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN response has been received from NA_Replica_3 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN response has been received from NA_Replica_2 Server.
12 Aug, 2020 8:12:49 PM => ERROR : ===== BYZANTINE ERROR OCCURED AT REPLICA 3 =====
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN response has been sent to FrontEnd Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN request has been received from FrontEnd Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN request has been sent to AS_Replica_2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN request has been sent to AS_Replica_3 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN response has been received from AS_Replica_2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN response has been received from AS_Replica_3 Server.
12 Aug, 2020 8:14:15 PM => ERROR : ===== BYZANTINE ERROR OCCURED AT REPLICA 3 =====
12 Aug, 2020 8:14:15 PM => INFO : 3 SUCCESSIVE BYZANTINE ERRORS AT Replica 3
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN response has been sent to FrontEnd Server.
12 Aug, 2020 8:14:15 PM => INFO : ====== Message Sent To Replica Manager 3 To Inform About Byzantine Error ======
```

Leader Got 3 Successive Byzantine Errors From Replica 3 & Informed To RM3

```
12 Aug, 2020 8:04:46 PM => INFO : EU_Replica_3 - UDP server has been started and running.
12 Aug, 2020 8:04:46 PM => INFO : AS_Replica_3 - UDP server has been started and running.
12 Aug, 2020 8:04:46 PM => INFO : NA_Replica_3 - UDP server has been started and running.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET STATUS request has been received from NA_Replica_1 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS request has been received from NA_Replica_3 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS response has been sent to NA_Replica_3 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS request has been received from NA_Replica_3 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS response has been sent to NA_Replica_3 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET STATUS response has been sent to NA_Replica_1 Server.
12 Aug, 2020 8:07:44 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:07:44 PM => INFO : UDP CREATE PLAYERACCOUNT response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET STATUS request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS request has been received from AS_Replica_3 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS request has been received from AS_Replica_3 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS response has been sent to AS_Replica_3 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET STATUS response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS response has been sent to AS_Replica_3 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from EU_Replica_1 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT response has been sent to EU_Replica_1 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN request has been received from NA_Replica_1 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN response has been sent to NA_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : =====RM3 Recieved Request Regarding solving BYZANTINE Error : Recover Your Data=====
12 Aug, 2020 8:14:15 PM => INFO : =====RM3 Sent Request To RM2 For Getting Correct Data To Resolve BYZANTINE Error.======
12 Aug, 2020 8:14:15 PM => INFO : =====RM3 Recieved Correct Data From EU_Replica_2 & Sent It To EU_Replica_3======
12 Aug, 2020 8:14:15 PM => INFO : =====RM3 Recieved Correct Data From AS_Replica_2 & Sent It To AS_Replica_3=====
12 Aug, 2020 8:14:15 PM => INFO : ======RM3 Recieved Correct Data From NA_Replica_2 & Sent It To NA_Replica_3=====
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data request has been received from ReplicaManager3 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data request has been received from ReplicaManager3 Server.
12 Aug, 2020 8:14:15 PM => INFO : =====AS_Replica_3 Recovered Data Successfully.=====
12 Aug, 2020 8:14:15 PM => INFO : =====EU_Replica_3 Recovered Data Successfully.=====
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data response has been sent to ReplicaManager3 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data response has been sent to ReplicaManager3 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data request has been received from ReplicaManager3 Server.
12 Aug, 2020 8:14:15 PM => INFO : =====NA_Replica_3 Recovered Data Successfully.=====
12 Aug, 2020 8:14:15 PM => INFO : UDP Change Recovered Data response has been sent to ReplicaManager3 Server.
```

RM3 Got Byzantine Error Info & Recovered data from RM2

```
12 Aug, 2020 8:04:37 PM => INFO : EU_Replica_2 - UDP server has been started and running.
12 Aug, 2020 8:04:37 PM => INFO : AS_Replica_2 - UDP server has been started and running.
12 Aug, 2020 8:04:37 PM => INFO : NA_Replica_2 - UDP server has been started and running.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET STATUS request has been received from NA_Replica_1 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS request has been received from NA_Replica_2 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS response has been sent to NA_Replica_2 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS request has been received from NA_Replica_2 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET_STATUS response has been sent to NA_Replica_2 Server.
12 Aug, 2020 8:05:28 PM => INFO : UDP GET STATUS response has been sent to NA_Replica_1 Server.
12 Aug, 2020 8:07:44 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:07:44 PM => INFO : UDP CREATE PLAYERACCOUNT response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET STATUS request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS request has been received from AS_Replica_2 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS request has been received from AS_Replica_2 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS response has been sent to AS_Replica_2 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET STATUS response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:09:13 PM => INFO : UDP GET_STATUS response has been sent to AS_Replica_2 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from EU_Replica_1 Server.
12 Aug, 2020 8:10:48 PM => INFO : UDP CREATE PLAYERACCOUNT response has been sent to EU_Replica_1 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN request has been received from NA_Replica_1 Server.
12 Aug, 2020 8:12:49 PM => INFO : UDP PLAYER SIGNIN response has been sent to NA_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN request has been received from AS_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP PLAYER SIGNIN response has been sent to AS_Replica_1 Server.
12 Aug, 2020 8:14:15 PM => INFO : =====RM2 Recieved Request To Send Correct Data TO RM3=====
12 Aug, 2020 8:14:15 PM => INFO : =====RM2 Sent Request To Its 3 Servers NA, EU & AS To Send Correct Data TO RM3=====
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 request has been received from ReplicaManager2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 request has been received from ReplicaManager2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 request has been received from ReplicaManager2 Server.
12 Aug, 2020 8:14:15 PM => INFO : =====EU_Replica_2 has Sent Correct Data TO RM3=====
12 Aug, 2020 8:14:15 PM => INFO : =====AS_Replica_2 has Sent Correct Data TO RM3=====
12 Aug, 2020 8:14:15 PM => INFO : =====NA_Replica_2 has Sent Correct Data TO RM3=====
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 response has been sent to ReplicaManager2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 response has been sent to ReplicaManager2 Server.
12 Aug, 2020 8:14:15 PM => INFO : UDP Send Recovery Data To RM3 response has been sent to ReplicaManager2 Server.
```

RM2 Got Request To Send Correct Data From RM3

19) **UDP Reliability Test:** Here is test for UDP reliability. As mentioned above that if FrontEnd not get any response from Leader, it will wait for 5 seconds and resend request. It will send Maximum 3 times after that if it not get response then it will give response to client that no response from server so that client not wait forever.

```
#========== !! Front End Started !! ==========#
12 Aug, 2020 7:23:00 PM => INFO : UDP CREATE PLAYERACCOUNT request has been sent to EU_Replica_1 Server.
12 Aug, 2020 7:23:15 PM => ERROR : No response has been sent from EU_Replica_1 Server.
```

FrontEnd

```
12 Aug, 2020 7:21:39 PM => INFO : EU_Replica_1 - UDP server has been started and running.
12 Aug, 2020 7:21:39 PM => INFO : NA_Replica_1 - UDP server has been started and running.
12 Aug, 2020 7:21:39 PM => INFO : AS Replica 1 - UDP server has been started and running.
12 Aug, 2020 7:23:00 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from FrontEnd Server.
12 Aug, 2020 7:23:05 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from FrontEnd Server.
12 Aug, 2020 7:23:10 PM => INFO : UDP CREATE PLAYERACCOUNT request has been received from FrontEnd Server.
```

Leader (Replica 1)

```
Enter Your Choice: 1
Enter First Name: raj
Enter Last Name: mistry
Enter Age: 22
Enter Username: rajmistry123
Enter Password: R@mistry98
Enter IpAddress: 93.3.3.3

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

12 Aug, 2020 7:23:15 PM => INFO : false => No response has been sent from EU_Replica_1 Server.
```

Client

➢ Challenges (Most Important / Difficult Part):

1) Managing concurrency between 3 replicas.
2) FIFO Implementation & make UDP reliable.
3) Implementing synchronization among shared data and avoiding deadlocks.
4) Byzantine Error Implementation.

➢ References:

- ✓ 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/Interface_description_language
- ✓ https://en.wikipedia.org/wiki/Object_request_broker
- ✓ https://www.corba.org/orb_basics.htm
- ✓ https://research.cs.queensu.ca/home/xiao/DS/node5.html
- ✓ https://www.geeksforgeeks.org/countdownlatch-in-java/
- ✓ https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ConcurrentLinkedQueue.html
- ✓ https://stackoverflow.com/questions/9580457/fifo-class-in-java
- ✓ https://stackoverflow.com/questions/24085571/synchronizing-queue-in-java-on-multiple-threads