Distributed Systems

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

Project

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

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

Submitted by:
Vaibhav Malhotra
ID: 40079373

Development tools

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

The project is designed as a system that uses passive replication, reliable UDP FIFO, and a CORBA front end. In this system a leader receives requests from the front end, the leader processes the requests and broadcasts the requests to it's replicas, which in turn process and return a result to the leader, the leader accumulates the results and decides correct result by majority and sends that back to the front end.

Build and Run

IDE (IntelliJ)

- Open the project in folder DPSS
- Setup the SDK
- Add runtime params for all the below classes (`-ORBInitialPort 1050 - ORBInitialHost`
- Run: ReplicaManager (Run all server replicas)
- Run: PlayerClient (to launch a player window)
- Run: AdminClient (to launch a Admin window)
- To run multiple clients change the configuration to "Allow parallel run".

Command Line

1. **Command Line:**
    - o - Move to *DPSS* directoy: **cd DPSS**.
    - o - Create a new folder named *dist* in the current folder : **mkdir dist**
    - o - Compile the code (outputting into dist folder): **javac -d dist src/**/*.java**
    - o - Move to *dist* folder: **cd dist**

• Run all 3 servers using following commands (will have to open different terminals):

Run ORBD through cmd:

```
start orbd -ORBInitialPort 1050
```

Start Servers:

```
start java ReplicaManager.ReplicaManager -ORBInitialPort 1050 - ORBInitialHost
localhost
```

- Run the clients (will have to open in different terminals):

```
java PlayerClient -ORBInitialPort 1050 -ORBInitialHost localhost

java AdminClient -ORBInitialPort 1050 -ORBInitialHost localhost
```

Ports

Replica1 (leader)

- R1_AmericanServer – 8080
- R1_EuropeanServer - 8081
- R1_AsianServer – 8082

Replica2

- R1_AmericanServer – 2001
- R1_EuropeanServer - 2002
- R1_AsianServer – 2003

Replica3

- R1_AmericanServer – 3001
- R1_EuropeanServer - 3002
- R1_AsianServer – 3003

The ports can be changed in Constants file.

Architecture

The below diagram describes the major components of the system. The client(player and admin), the front end, the replicas, the replica manager, and the reliable FIFO UDP.

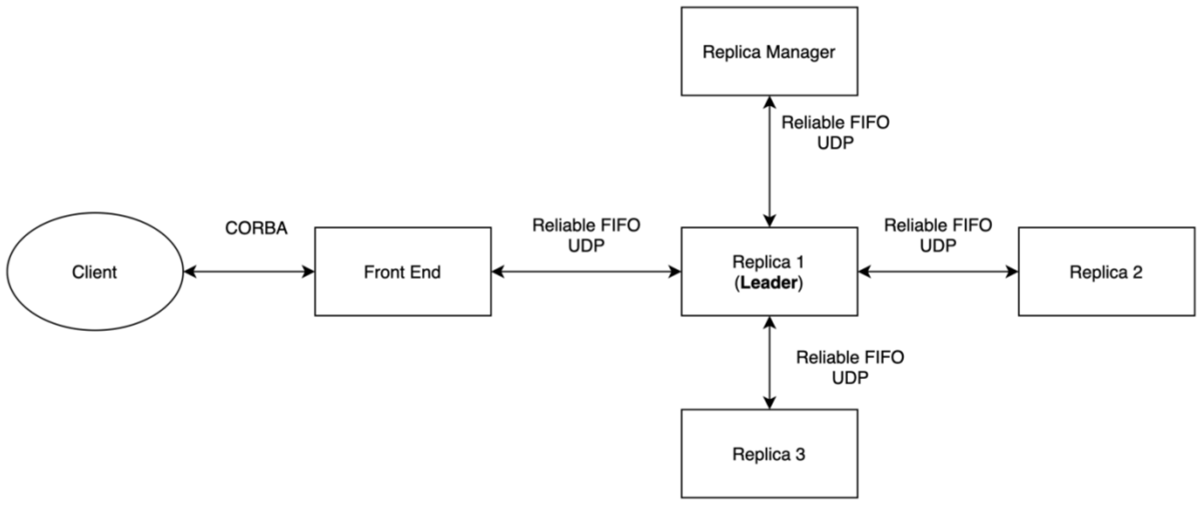


Figure 1. High level architecture

The client communicates with the front-end using CORBA invocation, and all the other communications in the system are done using reliable FIFO UDP.
The Front-end is connected with the leader replica with UDP which will be used by Front-end to send client's requests to the leader replica.
The leader replica sends the client's request to other follower replicas via reliable UDP FIFO queue. The leaders also communicates with the Replica Manager to update the reposes of other replicas.
A more detailed architecture is available on the next page.

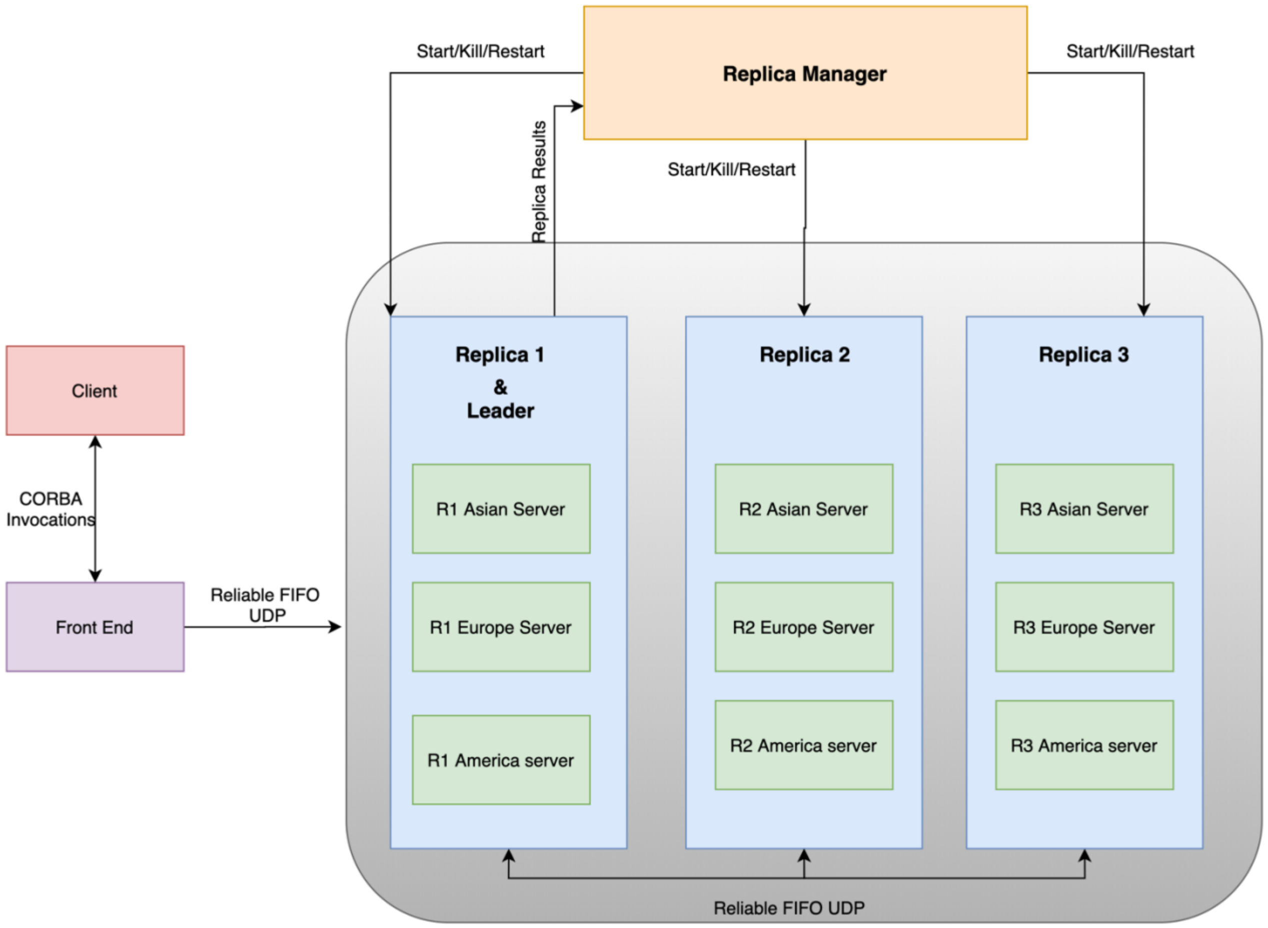


Figure 2. Detailed architecture of the distributed system

Below is the description of the components of the system above:

- **The Replica Manager**
  The Replica Manager creates and initializes the actively replicated server subsystem. RM initializes all the replicas and designates one of them as a leader. The RM also manages the replica group information, maintains if a replica has produced incorrect result and restarts a failed replica that produces incorrect result three times successively.
  It has a Reliable FIFO UDP server which awaits requests from the leader, which informs it of the correctness of the results produced by each replica and the leader, if there is one of the instances in the group produces an error, it increments the error count for that instance. Once the error count reaches 3, it stop the particular instance, and requests a start of a new instance of that server. This handles software bugs (i.e. non-malicious Byzantine failures).

- **The Front End**

The front end is a single entry point to the entire system. It is an implementation of the IDL CORBA file for the player and admin. It implements their respective methods and sets up the CORBA server.
It starts a FIFO UDP client for the respective geo-location, sends a request to the leader of that geo-location, waits for the reply, and sends back the result to the client via CORBA.

- **The Game Server**
  The game server runs as a leader or a replica, depending on the RM's specification.

  Regular Server:
  The regular server receives requests from the leader, executes the request and sends the response back to the leader.

  Leader Server:
  The leader receives a request from the CORBA front end, then FIFO broadcasts the request to all the server replicas, receives the responses from the server replicas and sends a single correct response back to the client. It also sends the status of all responses to RM to help keep track of a malicious server

- **The Clients**
  The player and admin clients are the same as from the original implementation in assignment 2. Both the player and admin client uses CORBA invocation to call methods that are implemented in the front end.

- **Reliable FIFO UDP**
  The UDP is used for the communications between FE, leader, RM and other replicas. This communication is made reliable and FIFO in order to avoid message loss and guarantee correctness.

  FIFO ordering algorithm:
  In this project FIFO ordering is implemented over unicast, as we had to run this project on a single system.

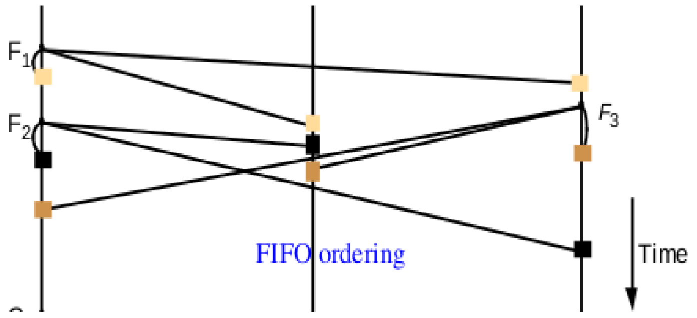


Figure 3. FIFO ordering

A **ConcurrentLinkedQueue** has been implemented on the server to store the incoming me requests. The *head* of the queue is that element that has been on the queue the longest time. The *tail* of the queue is that element that has been on the queue the shortest time.

Reliability:
Reliability is composed of 3 things

- Integrity - Duplicate messages detected and rejected
  This has been achieved by adding a sequence number to every request.

- Validity - Message send is the same one received
  The messages in the system are defined to a particular standard, if anything different than that is received then the message is rejected and a new request is generated.

- Agreement - Processes can detect missing messages
  Sequence number helps achieve this.

<u>Techniques Used</u>

1. CORBA using Java IDL

CORBA is used to a design specification for an Object Request Broker (ORB). This ORB provides the mechanism required for distributed objects to communicate and invoke server methods.

2. Reliable FIFO UDP

The UDP is used for the communications between FE, leader, RM and other replicas. This communication is made reliable and FIFO in order to avoid message loss and guarantee correctness.
Please refer to page above for in-depth explanation.

3. Multi-threading

a. All servers run on their individual thread
b. All UDP requests are sent on a new thread
c. All client requests are sent on a new thread

4. HashTables - Data Structure

a. Player data on server are stored in a Hashtables. Hashtables are thread-safe and promote concurrency.

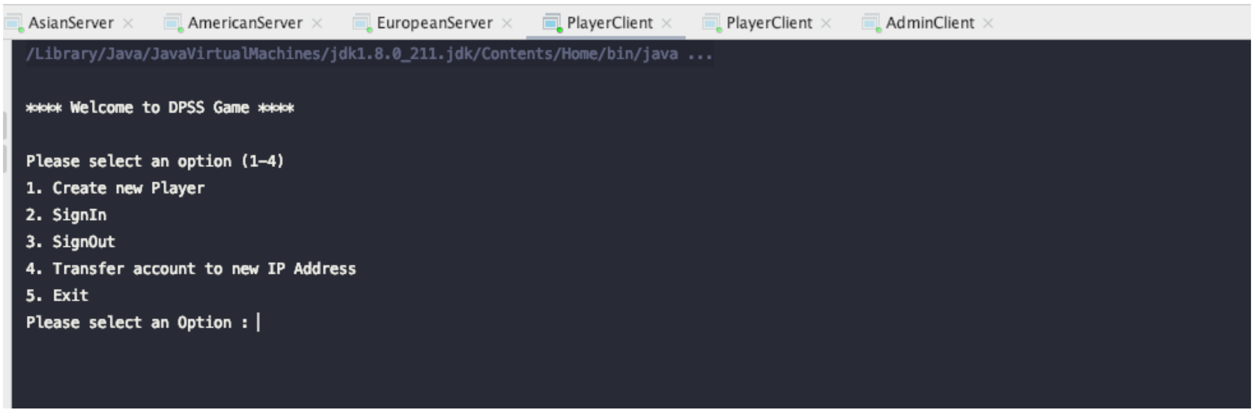
5. Locks

- Lock (ReentrantLock) is used for proper synchronization to allow multiple users to perform operations for the same or different accounts at the same time.

Testing

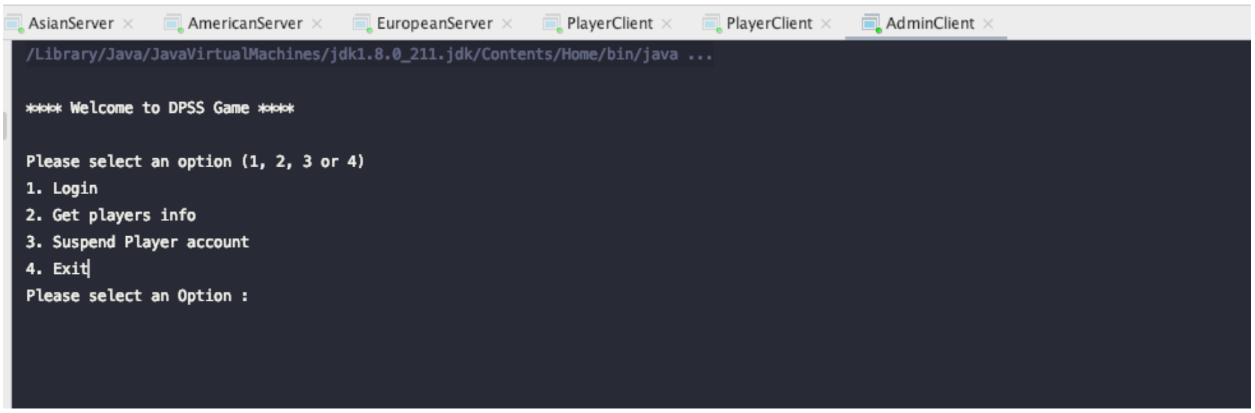
Intro screens:

Player Screen:



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java ...

**** Welcome to DPSS Game ****

Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option :
```

Admin Screen:

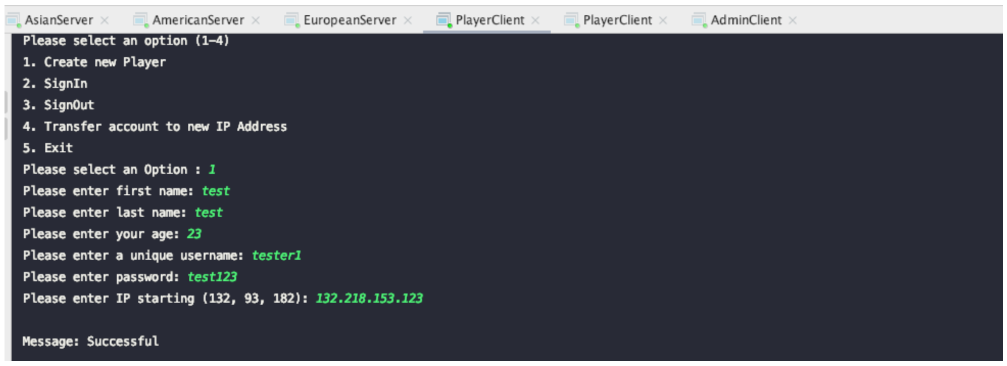


```
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java ...

**** Welcome to DPSS Game ****

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option :
```

Player testing:
In this section we will be testing Player Client functionalities.

Test Number	Scenario	Reason
1	Create player	Major functionality
2	Login player	Major functionality
3	Login player who is already logged in	Major functionality, Concurrent access
4	Logout player	Major functionality
5	Transfer Player to another server	Major functionality
6	Create a user (Already on another server)	Major functionality

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

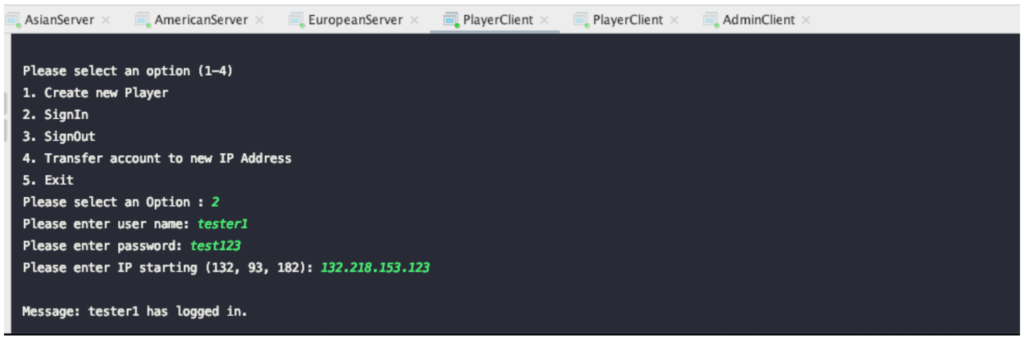


AsianServer × AmericanServer × EuropeanServer × PlayerClient × PlayerClient × AdminClient ×

```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 1
Please enter first name: test
Please enter last name: test
Please enter your age: 23
Please enter a unique username: tester1
Please enter password: test123
Please enter IP starting (132, 93, 182): 132.218.153.123

Message: Successful
```

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

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

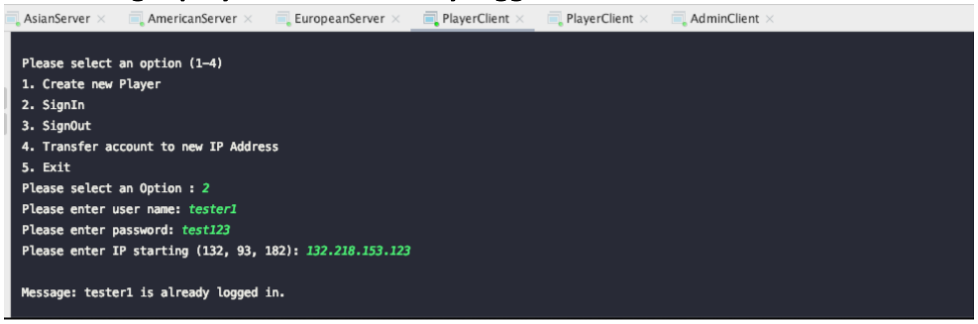


AsianServer × AmericanServer × EuropeanServer × PlayerClient × PlayerClient × AdminClient ×

```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 2
Please enter user name: tester1
Please enter password: test123
Please enter IP starting (132, 93, 182): 132.218.153.123

Message: tester1 has logged in.
```

Player data is validated. Player logged in successfully and message is displayed.

3. **Login player who is already logged in.**

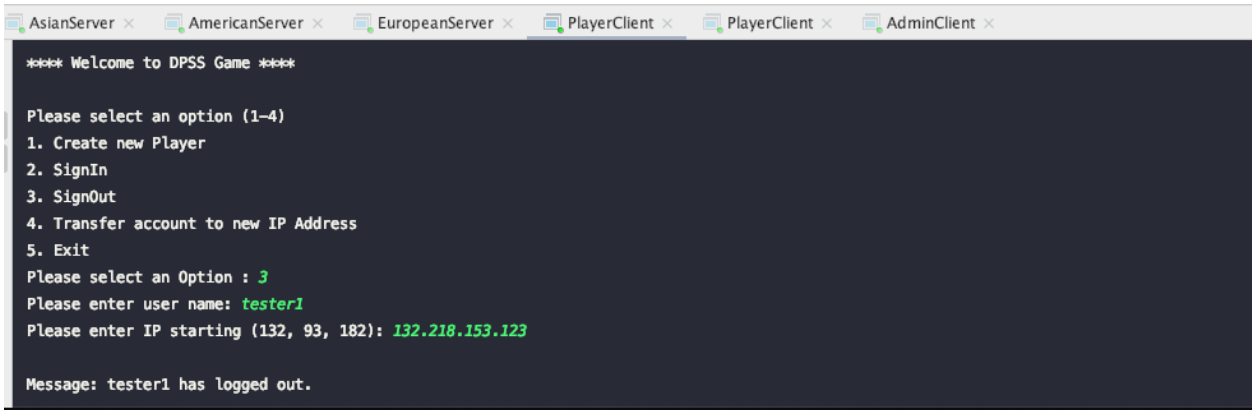


AsianServer × AmericanServer × EuropeanServer × PlayerClient × PlayerClient × AdminClient ×

```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 2
Please enter user name: tester1
Please enter password: test123
Please enter IP starting (132, 93, 182): 132.218.153.123

Message: tester1 is already logged in.
```

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

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

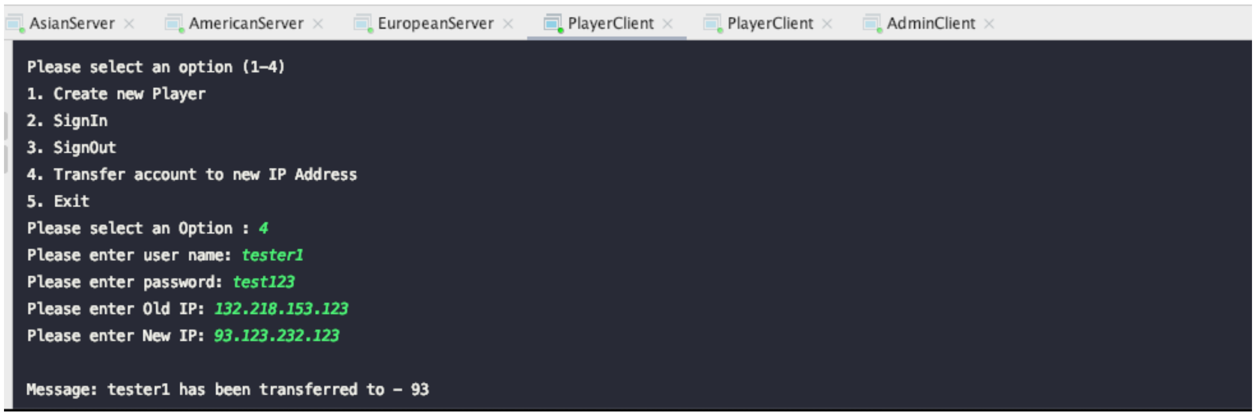


AsianServer | AmericanServer | EuropeanServer | PlayerClient | PlayerClient | AdminClient

```
**** Welcome to DPSS Game ****

Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 3
Please enter user name: tester1
Please enter IP starting (132, 93, 182): 132.218.153.123

Message: tester1 has logged out.
```

Player data is validated.
Player logged out successfully and message is displayed.

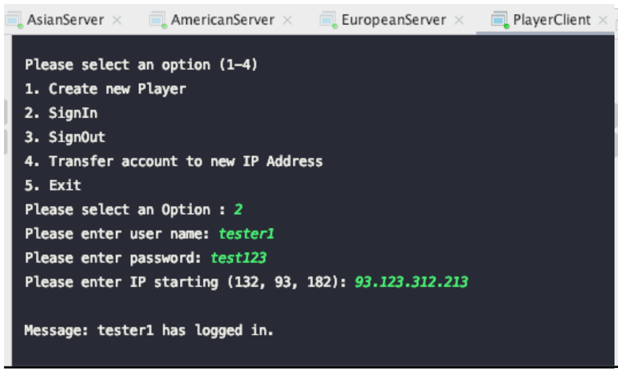
5. Transfer Player to another server - *playerSignOut* (*Username, IPAddress*)



AsianServer | AmericanServer | EuropeanServer | PlayerClient | PlayerClient | AdminClient

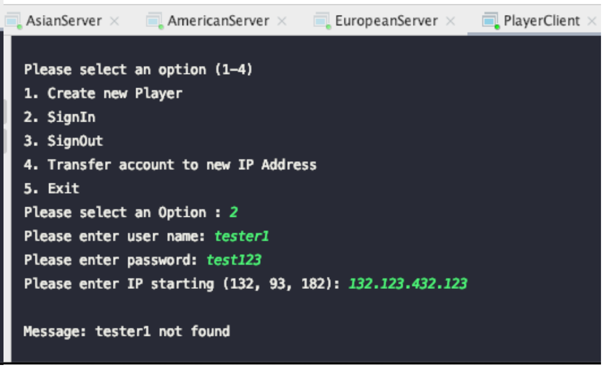
```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 4
Please enter user name: tester1
Please enter password: test123
Please enter Old IP: 132.218.153.123
Please enter New IP: 93.123.232.123

Message: tester1 has been transferred to - 93
```

The player is transferred to it's new server and then removed from old server.



AsianServer | AmericanServer | EuropeanServer | PlayerClient

```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 2
Please enter user name: tester1
Please enter password: test123
Please enter IP starting (132, 93, 182): 93.123.312.213

Message: tester1 has logged in.
```

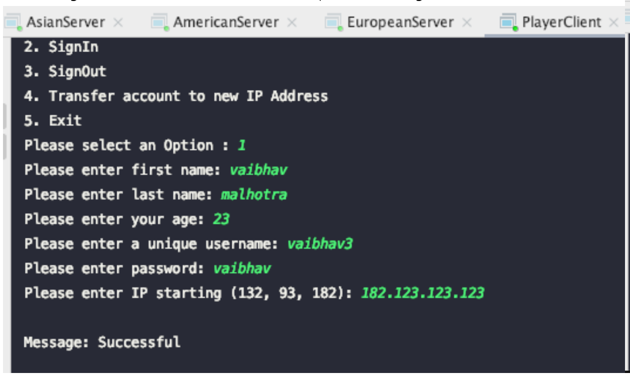


AsianServer | AmericanServer | EuropeanServer | PlayerClient

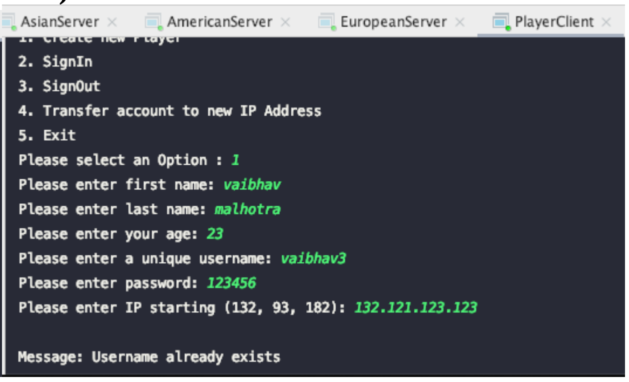
```
Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 2
Please enter user name: tester1
Please enter password: test123
Please enter IP starting (132, 93, 182): 132.123.432.123

Message: tester1 not found
```

The player is now on server IP 93(New Server). Player not available on server IP 132(Old Server)

6. Try to create a user (Already on another server)



```
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 1
Please enter first name: vaibhav
Please enter last name: malhotra
Please enter your age: 23
Please enter a unique username: vaibhav3
Please enter password: vaibhav
Please enter IP starting (132, 93, 182): 182.123.123.123

Message: Successful
```



```
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 1
Please enter first name: vaibhav
Please enter last name: malhotra
Please enter your age: 23
Please enter a unique username: vaibhav3
Please enter password: 123456
Please enter IP starting (132, 93, 182): 132.121.123.123

Message: Username already exists
```

Player creation unsuccessful on server 132, Because player already existed on server 182.

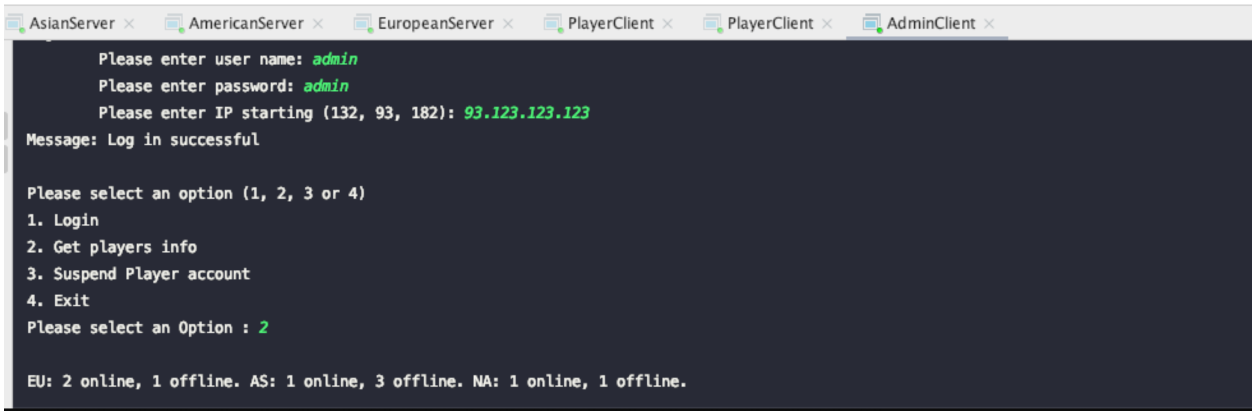
Admin testing:

In this section we will be testing Admin Client functionalities.

Test Number	Scenario	Reason
7	Get player status from different servers	Major functionality, Concurrent access
8	Suspend player account	Major functionality

7. **Get Player status from different servers - *getPlayerStatus* (*AdminUsername, AdminPassword, IPAddress*)**

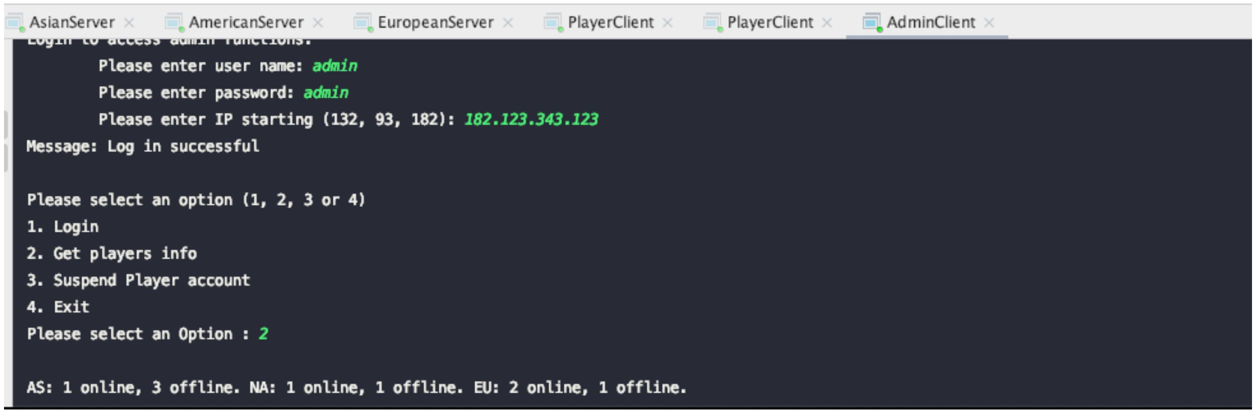
   a. On server IP 93



```
        Please enter user name: admin
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 93.123.123.123
Message: Log in successful

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 2

EU: 2 online, 1 offline. AS: 1 online, 3 offline. NA: 1 online, 1 offline.
```

Admin has logged in successfully on server IP 93 i.e server Europe.
Europe server sends request to Asia and America Server to gets their player status, appends it's result and displays.

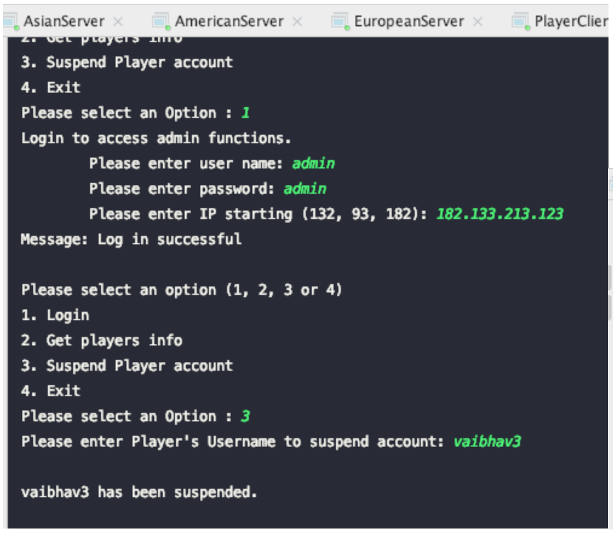
   b. On server IP 182



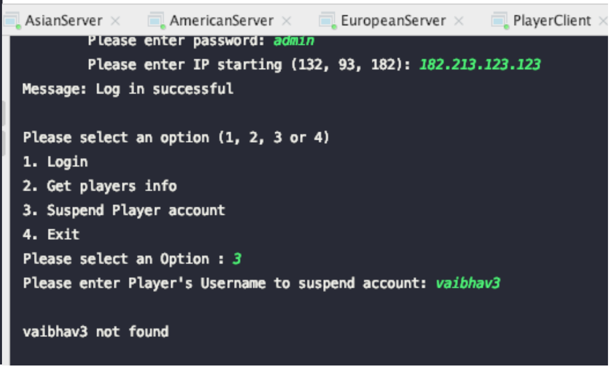
```
Login to access admin functions.
        Please enter user name: admin
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 182.123.343.123
Message: Log in successful

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 2

AS: 1 online, 3 offline. NA: 1 online, 1 offline. EU: 2 online, 1 offline.
```

Admin has logged in successfully on IP 182 i.e on server Asia.
Asia server sends request to Europe and America Server to get player status, appends result and displays.

8. **Suspend Player account (AdminUsername, AdminPassword, AdminIP, UsernameToSuspend)**



```
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 1
Login to access admin functions.
        Please enter user name: admin
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 182.133.213.123
Message: Log in successful

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 3
Please enter Player's Username to suspend account: vaibhav3

vaibhav3 has been suspended.
```



```
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 182.213.123.123
Message: Log in successful

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 3
Please enter Player's Username to suspend account: vaibhav3

vaibhav3 not found
```

Admin suspended player with username – *vaibhav3* on server IP *182*.
Player could not login with username – *vaibhav3* on server IP *182* because it has been suspended.

Multithreading and atomicity testing:

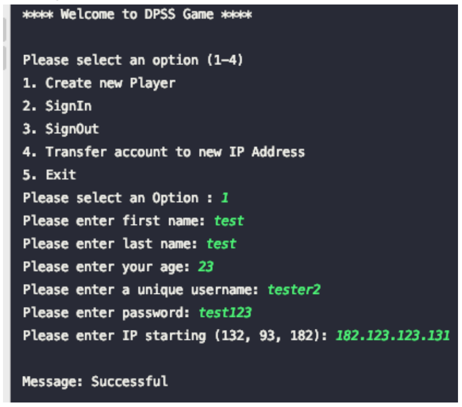
Test Number	Scenario	Reason
9	Multiple player account creation	Test Concurrency
10	Suspend player account	Test Atomicity

9. Multiple account creation

Scenario: 3 PlayerClients are trying to create new accounts simultaneously
Player1 – Username – tester2, server – 182 (Asia)
Player2 – Username – tester2, server – 93 (Europe)

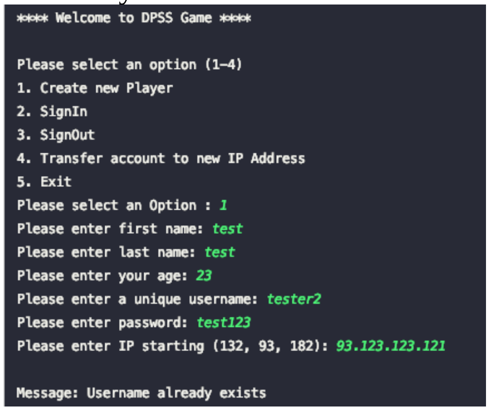
As we can see that Player1 was successful but as Username – tester2 was already taken at Asian server, Player2 was unable to create an account. The time difference between 2 calls is of microseconds. This tests the concurrent access to the data-structures in the project.

PLayerClient1



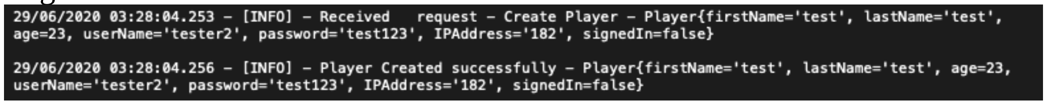
```
**** Welcome to DPSS Game ****

Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 1
Please enter first name: test
Please enter last name: test
Please enter your age: 23
Please enter a unique username: tester2
Please enter password: test123
Please enter IP starting (132, 93, 182): 182.123.123.131

Message: Successful
```

PlayerClient2



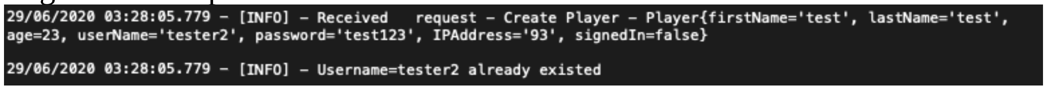
```
**** Welcome to DPSS Game ****

Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 1
Please enter first name: test
Please enter last name: test
Please enter your age: 23
Please enter a unique username: tester2
Please enter password: test123
Please enter IP starting (132, 93, 182): 93.123.123.121

Message: Username already exists
```

Logs from Asia



```
29/06/2020 03:28:04.253 - [INFO] - Received   request - Create Player - Player{firstName='test', lastName='test',
age=23, userName='tester2', password='test123', IPAddress='182', signedIn=false}

29/06/2020 03:28:04.256 - [INFO] - Player Created successfully - Player{firstName='test', lastName='test', age=23,
userName='tester2', password='test123', IPAddress='182', signedIn=false}
```

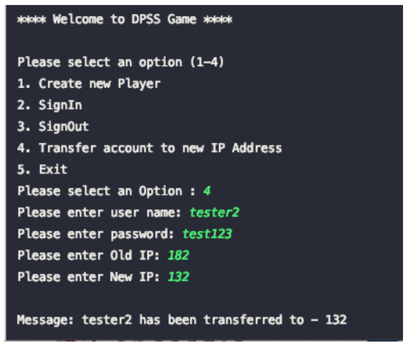
Logs from Europe



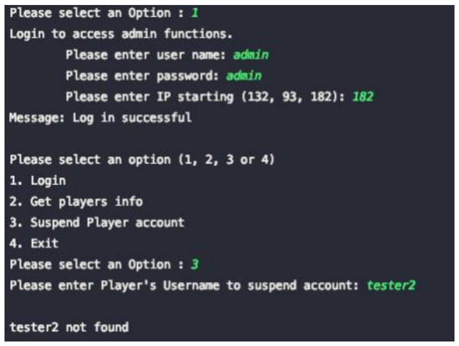
```
29/06/2020 03:28:05.779 - [INFO] - Received   request - Create Player - Player{firstName='test', lastName='test',
age=23, userName='tester2', password='test123', IPAddress='93', signedIn=false}

29/06/2020 03:28:05.779 - [INFO] - Username=tester2 already existed
```

10. Suspending and transferring account at the same time

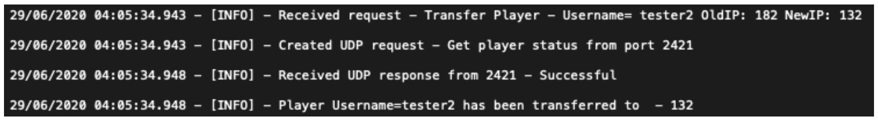
Scenario: Transfer request and suspend request are generated at the same time. For atomicity only one of the requests should be successful. In our case transfer was successful.



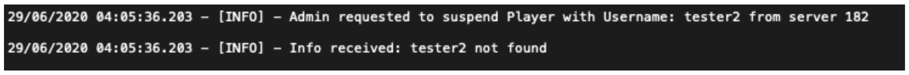
```
**** Welcome to DPSS Game ****

Please select an option (1-4)
1. Create new Player
2. SignIn
3. SignOut
4. Transfer account to new IP Address
5. Exit
Please select an Option : 4
Please enter user name: tester2
Please enter password: test123
Please enter Old IP: 182
Please enter New IP: 132

Message: tester2 has been transferred to - 132
```



```
Please select an Option : 1
Login to access admin functions.
        Please enter user name: admin
        Please enter password: admin
        Please enter IP starting (132, 93, 182): 182
Message: Log in successful

Please select an option (1, 2, 3 or 4)
1. Login
2. Get players info
3. Suspend Player account
4. Exit
Please select an Option : 3
Please enter Player's Username to suspend account: tester2

tester2 not found
```



```
29/06/2020 04:05:34.943 - [INFO] - Received request - Transfer Player - Username= tester2 OldIP: 182 NewIP: 132
29/06/2020 04:05:34.943 - [INFO] - Created UDP request - Get player status from port 2421
29/06/2020 04:05:34.948 - [INFO] - Received UDP response from 2421 - Successful
29/06/2020 04:05:34.948 - [INFO] - Player Username=tester2 has been transferred to  - 132
```

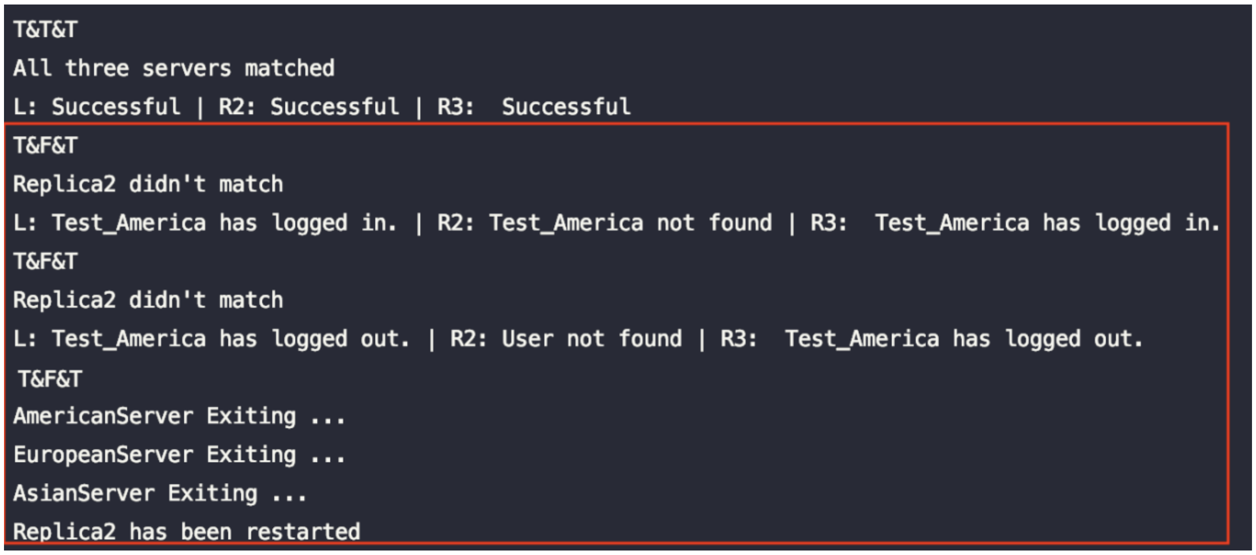


```
29/06/2020 04:05:36.203 - [INFO] - Admin requested to suspend Player with Username: tester2 from server 182
29/06/2020 04:05:36.203 - [INFO] - Info received: tester2 not found
```

11. Non-Malicious Byzantine Failure

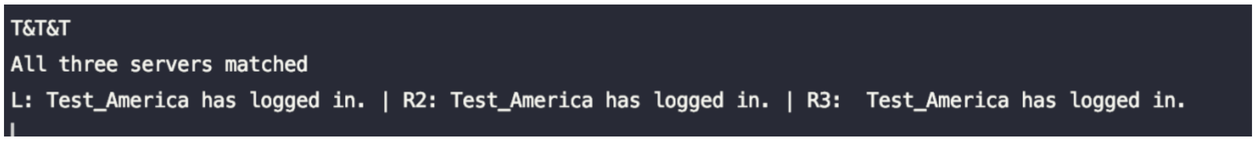
Scenario: I introduced a software bug in the system by changing user details in replica 2 i.e changed the username for Test_America. This replica is added as replicaFaulty.

As we can see, In the below screenshot that the servers were working fine before. But as soon as we try to login Test_America replica2 sends different response. We repeat the step 2 more time and the replica manager detects and restarts replica2.



```
T&T&T
All three servers matched
L: Successful | R2: Successful | R3:  Successful
T&F&T
Replica2 didn't match
L: Test_America has logged in. | R2: Test_America not found | R3:  Test_America has logged in.
T&F&T
Replica2 didn't match
L: Test_America has logged out. | R2: User not found | R3:  Test_America has logged out.
 T&F&T
AmericanServer Exiting ...
EuropeanServer Exiting ...
AsianServer Exiting ...
Replica2 has been restarted
```

Now on restarting the RM has launched the right replica2 which is bug free. Screenshot below depicts the scenario.



```
T&T&T
All three servers matched
L: Test_America has logged in. | R2: Test_America has logged in. | R3:  Test_America has logged in.
```

Conclusion

The implementation accomplishes the goals of the project, that is to build a fault tolerant, passively replicated system that uses reliable FIFO UDP for communication.
The front end receives CORBA invocations and forwards it to the leaders while also adding a sequence number. The replica manager restarts replicas if 3 successive errors occur on a replica.
All internal communications are done using the reliable FIFO UDP subsystem.
The system can recover from software bugs i.e. Non-Malicious Byzantine Failure