

# PROJECT REPORT

## • GROUP MEMBERS:

1: Name - Aishwarya Sharma

UFID - 74444009

2: Name - Fugnag Deng

UFID – 91203337

## • HOW TO RUN:

The server can be run as below:

```
PS C:\Users\HP\source\repos\TwitterTry> fsi .\TwitterEngine.fsx
```

And the Client can be run as below:

```
PS C:\Users\HP\source\repos\TwitterTry> fsi .\TwitterClient.fsx
```

TwitterClientSimulate.fsx and TwitterEngineSimulate.fsx are the separate files used for simulation.

## Problem Statement and Implementation

Implement a Twitter-like engine with the following functionality:

- Register account

Designed Message format - Register,username,password

Implemented Validation : if user is already registered, it is not registered again

```
[INFO][11/30/2021 7:28:17 AM][Thread 0001]
Client is up!
Type Your Message :
Register,aishwarya,pqr
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Register,Fugnag,lmn
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Register,Fugnag,xyz
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
[INFO][11/30/2021 7:28:18 AM][Thread 0001][remoting (akka://RemoteFSharp)]
Server is up!
Register,aishwarya,pqr
Sent to register actor
In Register Actor
  User registered
map [{"aishwarya", FSI_0002+User}]Server is up!
Register,Fugnag,lmn
Sent to register actor
In Register Actor
  User registered
map [{"Fugnag", FSI_0002+User}; {"aishwarya", FSI_0002+User}]Server is up!
Register,Fugnag,xyz
Sent to register actor
In Register Actor
  User already exists
map [{"Fugnag", FSI_0002+User}; {"aishwarya", FSI_0002+User}]
```

- Send tweet. Tweets can have hashtags (e.g. #COP5615isgreat) and mentions (@bestuser)

Designed Message format - Tweet,username,content

```
Do You wish to continue (Yes/No):Yes
Type Your Message :
Tweet,aishwarya,myFirstTweet
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Tweet,aishwarya,mySecondTweet,#hash1#hash2
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Tweet,aishwarya,myThirdTweet,#hash2#hash3,@Fugnag
Msg Sent to server
Do You wish to continue (Yes/No):Yes
```

```
Tweet,aishwarya,myFirstTweet
Sent to the Tweet actor
map [{"0", FSI_0002+Tweet}]Server is up!
Tweet,aishwarya,mySecondTweet,#hash1#hash2
Sent to the Tweet actor
map [{"hash1", ["1"]}; {"hash2", ["1"]}]]
map [{"0", FSI_0002+Tweet}; {"1", FSI_0002+Tweet}]Server is up!
Tweet,aishwarya,myThirdTweet,#hash2#hash3,@Fugnag
Sent to the Tweet actor
map [{"0", FSI_0002+Tweet}; {"1", FSI_0002+Tweet}; {"2", FSI_0002+Tweet}]
map [{"hash1", ["1"]}; {"hash2", ["1"; "2"]}; {"hash3", ["2"]}]]
map [{"Fugnag", ["2"]}]]
```

- Subscribe to user's tweets

Designed Message format - Subscribe,subscribingUser,subscribedUser

Validation: If any of the users do not exist in the registered users list, message as below is displayed

```
Type Your Message :
Subscribe,aishwarya,Fugnag
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Subscribe,aishwarya,RamdomUser
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
Subscribe,aishwarya,Fugnag
Sent to subscribe actor
Server is up!
Subscribe,aishwarya,RamdomUser
Sent to subscribe actor
The user you want to subscribe does not
```

- Re-tweets (so that your subscribers get an interesting tweet you got by other means)

Designed Message format - Retweet,username,tweetId

```
Type Your Message :
Register,Guest,xyz
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Tweet,Guest,FourthTweet
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Retweet,aishwarya,4
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
Register,Guest,xyz
Sent to register actor
In Register Actor
User registered
map [{"Fugnag", FSI_0002+User}; {"Guest", FSI_0002+User}; {"aishwarya", FSI_0002+User}]Server is up!
Tweet,Guest,FourthTweet
Sent to the Tweet actor
map [{"0", FSI_0002+Tweet}; {"1", FSI_0002+Tweet}; {"2", FSI_0002+Tweet}; {"3", FSI_0002+Tweet}; {"4", FSI_0002+Tweet}]Server is up!
Retweet,aishwarya,4
Sent to the Retweet actor
map [{"0", FSI_0002+Tweet}; {"1", FSI_0002+Tweet}; {"2", FSI_0002+Tweet}; {"3", FSI_0002+Tweet}; {"4", FSI_0002+Tweet}; {"5", FSI_0002+Tweet}]
```

- Allow querying tweets subscribed to, tweets with specific hashtags, tweets in which the user is mentioned (my mentions)

Designed Message format - Query,subscribed,username

```
Type Your Message :
Tweet,Fugnag,SeventhTweet
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Query,subscribed,aishwarya
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
Server is up!
Tweet,Fugnag,SeventhTweet
Sent to the Tweet actor
map
[("0", FSI_0002+Tweet); ("1", FSI_0002+Tweet); ("2", FSI_0002+Tweet);
 ("3", FSI_0002+Tweet); ("4", FSI_0002+Tweet); ("5", FSI_0002+Tweet);
 ("6", FSI_0002+Tweet); ("7", FSI_0002+Tweet)]Server is up!
Query,subscribed,aishwarya
Sent to the query actor
Tweets Subscribed :
"FifthTweet
SeventhTweet"
```

Designed Message format - Query,hashtag,tagname

```
Type Your Message :
Query,hashtag,hash2
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Query,hashtag,hash1
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
Query,hashtag,hash2
Sent to the query actor
Tweets contains this tag:
"mySecondTweet
myThirdTweet
myThirdTweet"
Server is up!
Query,hashtag,hash1
Sent to the query actor
Tweets contains this tag:
"mySecondTweet"
```

Designed Message format - Query,mention,username

```
Type Your Message :
Query,mention,Fugnag
Msg Sent to server
Do You wish to continue (Yes/No):
```

```
Server is up!
Query,mention,Fugnag
Sent to the query actor
Tweets mention this username:
"myThirdTweet"
```

- If the user is connected, deliver the above types of tweets live (without querying)

Designed Message format - Connect,username

```
Type Your Message :
Connect,aishwarya
Msg Sent to server
User Tweets :
  "0:myFirstTweet
1:mySecondTweet
2:myThirdTweet
3:myThirdTweet
5:FourthTweet"
Tweets Subscribed :
  "6:FifthTweet
7:SeventhTweet"
Do You wish to continue (Yes/No):
```

```
Server is up!
Connect,aishwarya
Sent to the query actor
User Tweets :
  "0:myFirstTweet
1:mySecondTweet
2:myThirdTweet
3:myThirdTweet
5:FourthTweet"
Tweets Subscribed :
  "6:FifthTweet
7:SeventhTweet"
```

## Simulation Results

Implement a tester/simulator to test the above

- Simulate as many users as you can

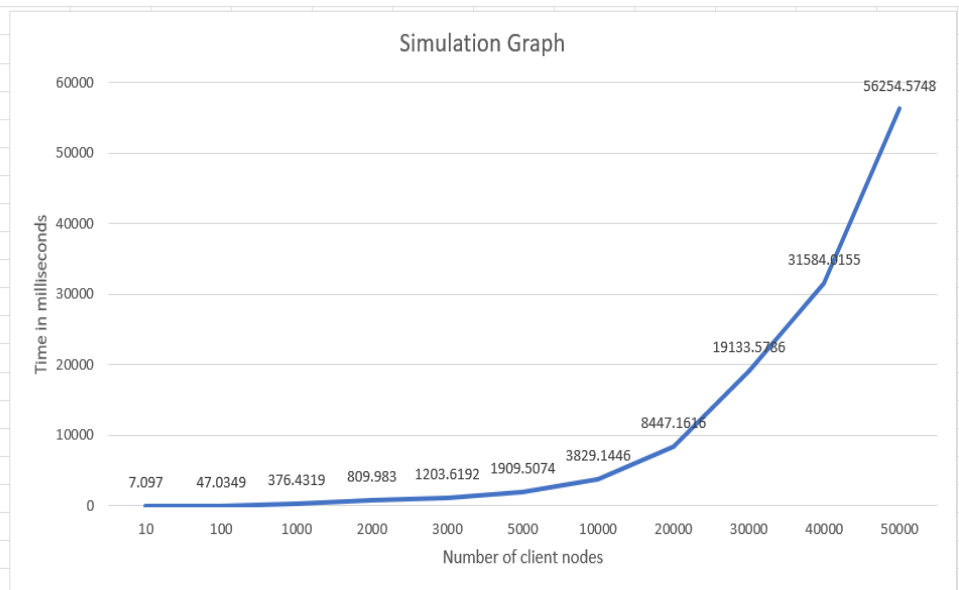
The program was tested for register functionality of up to 50,000 client nodes numbered from 0-49999 as below:

```
Msg of node49992 Sent to server
Msg of node49993 Sent to server
Msg of node49994 Sent to server
Msg of node49995 Sent to server
Msg of node49996 Sent to server
Msg of node49997 Sent to server
Msg of node49998 Sent to server
Msg of node49999 Sent to server
Real: 00:02:00.748, CPU: 00:02:27.796, GC gen0: 333, gen1: 90, gen2: 1
PS C:\Users\HP\source\repos\TwitterTry>
```

```
Register,node49999,node499991
Sent to register actor
In Register Actor
  User registered
Server is up!
Finish
57177.751000
```

Below is the graph for the analysis of running times when client nodes are varied from 10 - 50,000:

Nodes	Time (ms)
10	7.097
100	47.0349
1000	376.4319
2000	809.983
3000	1203.6192
5000	1909.5074
10000	3829.1446
20000	8447.1616
30000	19133.5786
40000	31584.0155
50000	56254.5748



- Simulate periods of live connection and disconnection for users

-When Client connects, performs few operations and disconnects , it resets the underlying tcp connection by `system.Dispose` command

```
Client is up!
Type Your Message :
Register,abc,xyz
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Tweet,abc,MyFirstTweet
Msg Sent to server
Do You wish to continue (Yes/No):No
Real: 00:01:57.806, CPU: 00:00:00.671, GC gen0: 2, gen1: 0, gen2: 0
PS C:\Users\HP\source\repos\TwitterTry> fsi .\TwitterClientSimulate.fsx
```

-When client reconnects, it re-establishes the connection and sends a connect msg to the server

```
Client is up!
Type Your Message :
Tweet,abc,MySecondTweet
Msg Sent to server
Do You wish to continue (Yes/No):Yes
Type Your Message :
Connect,abc
Msg Sent to server
User Tweets :
"0:MyFirstTweet
1:MySecondTweet"
Tweets Subscribed :
""
```

-Server side Response for the entire process is as below

```
[INFO][11/29/2021 2:57:58 AM][Thread 0001][remoting (akka://RemoteFSharp)] Starting remoting
[INFO][11/29/2021 2:57:58 AM][Thread 0001][remoting (akka://RemoteFSharp)] Remoting started; listening on addresses : [akka.tcp://RemoteFSharp@localhost:9007]
[INFO][11/29/2021 2:57:58 AM][Thread 0001][remoting (akka://RemoteFSharp)] Remoting now listens on addresses: [akka.tcp://RemoteFSharp@localhost:9007]
Server is up!
Register,abc,xyz
Sent to register actor
In Register Actor
User registered
map [("abc", FSI_0002+User)]Server is up!
Tweet,abc,MyFirstTweet
Sent to the Tweet actor
map [("0", FSI_0002+Tweet)][INFO][11/29/2021 2:59:33 AM][Thread 0025][TcpServerHandler (akka://RemoteFSharp)] Connection was reset by the remote peer. Channel [::ffff:127.0.0.1]:9007->[::ffff:127.0.0.1]:51327(Id=7377863a)
[INFO][11/29/2021 2:59:33 AM][Thread 0022][akka.tcp://RemoteFSharp@localhost:9007/system/endpointManager/reliableEndpointWriter-akka.tcp%3A%2Fserver-system%40localhost%3A9003-1] Removing receive buffers for [akka.tcp://RemoteFSharp@localhost:9007]->[akka.tcp://server-system@localhost:9003]
Server is up!
Tweet,abc,MySecondTweet
Sent to the Tweet actor
map [("0", FSI_0002+Tweet); ("1", FSI_0002+Tweet)]Server is up!
Connect,abc
Sent to the query actor
User Tweets :
"0:MyFirstTweet
1:MySecondTweet"
Tweets Subscribed :
""
```

- Simulate a Zipf distribution on the number of subscribers. For accounts with a lot of subscribers, increase the number of tweets. Make some of these messages re-tweets

Below is a graph was obtained for registering specified number of users (ie 300-50 ), then making a dummy user named 'node' subscribe to them all and performing tweet and re-tweet actions after that.

