

# **COP5615\_Twitter\_Project**

## **Team Members**

- Rishabh Srivastav: UFID 7659-9488
- Ashish Sunny Abraham: UFID 6388-7782

## **Outline of the project**

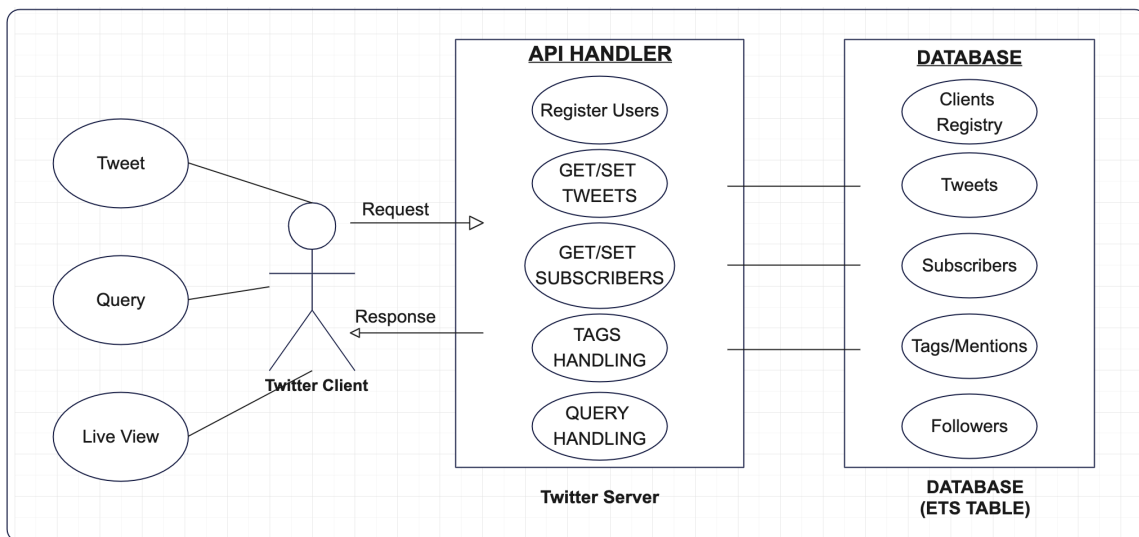
This project aims to design a Twitter Clone which performs all the functionalities of the twitter application and a client tester/simulator to test the above used functionalities. The user should be able to add/register himself on the network, be able to tweet, subscribe and follow people so that the user is able to view tweets of other users as well, and also add the features for querying. Querying is handled in ways like extracting tweets which contain a specific hashtag or tweets that contain a specific @ mention to some user.

## **Commands to run the program (Simulation and Test Purposes)**

- Compile all the .erl files in the zip folder (engine, user, test, server).
- Open one erl shell and run the engine by typing “test:delegate({})” and then enter your IP address.
- Open a second erl shell and run the client simulation by typing “test:delegate({50,49,20})” where the first parameter represents the number of users you want to simulate for and the second parameter represents the maximum number of followers a user can have, which will be 1 less than total users at max. The third parameter represents the percentage of users you want to participate in the disconnect/reconnect functionality.
- Based on the simulation results, we’ve noted down observations in the table and constructed graphs for the same.
- In order to test each functionality manually, the user can refer to the screenshots attached in the features section to see how to test for each feature manually apart from the simulation.

## Architecture

The Twitter Engine acts as an interface between client and database which retrieves information on demand as specified by the user. The engine handles multiple API calls to perform tasks and queries entered by the user and interacts with the database (ETS tables) to return the expected values. The engine is responsible for registering a user, processing tweets, and query handling. The database stores client/user information, list of followers and subscribers of a client and their tweets and mentions. The client module is responsible for sending requests to the server to process tasks based on features demanded by the client. The client then in-turn receives a response back from the engine after every request.



*Architecture Overview*

## Features implemented

- ❖ Twitter Engine start: Starts the server so that the application can become on-line
- ❖ Register user: Allows user to register himself onto the social media clone without which the user can't access any of the functionalities of the application.

```
3> server:register_user("Rishabh").  
Creating account with ID: "Rishabh"  
Account created successfully...  
ok  
4> server:register_user("Rishabh").  
Account already exists, proceed to tweeting!
```

*Creating an account: "Rishabh" and verifying that you can't re-register if you have already registered*

- ❖ Send tweet: Allows a user to express his/her thoughts and send tweets. Checks have been added to ensure a user who hasn't registered himself cannot tweet unless he creates an account.

```
5> server:send_tweet("Rishabh", "Hi, this is my first tweet!").
User: "Rishabh" is tweeting: "Hi, this is my first tweet!"!!
```

```
6> server:send_tweet("Yash", "Hi, trying to tweet without registering!").
Kindly register first before tweeting
```

*Sending tweets with account: "Rishabh", also making sure if a user is not registered, he should register first before tweeting.*

- ❖ Subscribe to users: Provides users the ability to connect with other users whose tweets they find interesting and would want to follow them. By subscribing to a user, one can view all their tweets as part of their own tweet list.

```
8> server:send_tweet("Rishabh", "Hello, I'm new to twitter").
User: "Rishabh" is tweeting: "Hello, I'm new to twitter"!!
User: "Rishabh" follower list
ok
9> server:send_tweet("Ashish", "Good morning everyone").
User: "Ashish" is tweeting: "Good morning everyone"!!
User: "Ashish" follower list
ok
10> server:subscribe("Ashish","Rishabh").
User: "Ashish" has subscribed to user: "Rishabh"!!
Adding user: "Ashish" to user: "Rishabh" follower list
```

```
12> server:get_my_subscriptions("Ashish").
User: "Ashish" subscription list
["Rishabh"]
13> server:get_my_tweets("Ashish").
All tweets by User: "Ashish" , tweets list:
["Good morning everyone","Hi, this is my first tweet!",
"Hello, I'm new to twitter"]
```

*User "Ashish" is subscribing to User "Rishabh" and is now able to view all tweets of the user that he has subscribed to*

- ❖ Retweet: Allows users to view tweets of specific interest to them. This has been incorporated with the subscriber model itself.
- ❖ Querying:
  - Hashtags: Provides a list of tweets which contain a specific hashtag as entered by the user. Returns nothing if no such match is found.

```
14> server:send_tweet("Ashish", "The subject #Dosp is great").
User: "Ashish" is tweeting: "The subject #Dosp is great"!!
```

```
15> server:send_tweet("Rishabh", "I am working on #Dosp project").
User: "Rishabh" is tweeting: "I am working on #Dosp project"!!
```

```
16> server:get_tweets_with_tag("#Dosp").
["The subject #Dosp is great",
"I am working on #Dosp project"]
```

*Tweets containing hashtags "#Dosp" have been filtered out from all the tweets made by all the users*

- Mentions: Provides a list of tweets which contain mentions of other users as requested by some users. Checks have been added to verify if the mentioned user exists in the first place or not. Returns nothing if no such match is found.

```
17> server:send_tweet("Rishabh", "Hi @Ashish , what are you doing? ").
User: "Rishabh" is tweeting: "Hi @Ashish , what are you doing? "!!
```

```
19> server:send_tweet("Rishabh", "Great game today! @Ashish ").
User: "Rishabh" is tweeting: "Great game today! @Ashish "!!
```

```
20> server:get_tweets_with_mention("@Ashish").
["Hi @Ashish , what are you doing? ",
 "Great game today! @Ashish "]
```

*Tweets containing mention of user “@Ashish” have been filtered out from all the tweets made by all the users*

- Subscribe: Allows user to view tweets of another person he/she has subscribed to

```
8> server:send_tweet("Rishabh", "Hello, I'm new to twitter").
User: "Rishabh" is tweeting: "Hello, I'm new to twitter"!!
User: "Rishabh" follower list
ok
9> server:send_tweet("Ashish", "Good morning everyone").
User: "Ashish" is tweeting: "Good morning everyone"!!
User: "Ashish" follower list
ok
10> server:subscribe("Ashish","Rishabh").
User: "Ashish" has subscribed to user: "Rishabh"!!
Adding user: "Ashish" to user: "Rishabh" follower list
```

```
12> server:get_my_subscriptions("Ashish").
User: "Ashish" subscription list
["Rishabh"]
13> server:get_my_tweets("Ashish").
All tweets by User: "Ashish" , tweets list:
["Good morning everyone","Hi, this is my first tweet!",
 "Hello, I'm new to twitter"]
```

*User Ashish is subscribing to User Rishabh and is now able to view all tweets of the user that he has subscribed to*

## **Implementation Details**

**test.erl**: This module contains code for simulation. Steps to execute functions from this file have been provided in steps to run the program.

**engine.erl**: The code for the Twitter engine implementation, which is in charge of handling and implementing tweets, is located in this file. To manage subscriptions, tweets, searches, etc., the engine directly interfaces with the database (built using ETS tables). In order to convey the search queries and distribute the tweets to subscribers, it also directly interacts with clients via the API handler. The clients registry ETS table, which tracks numerous actors inside the server, maintains process states.

**user.erl**: This file contains details about a particular client participant, which corresponds to a single twitter user. This includes its user Id data, which is saved as a string supplied to it upon program startup. The primary registry ETS table, which maintains track of different actors, performs the core logic of preserving process state.

**server.erl**: To manually test for functions and features of the application, use this file and refer to features implemented section to see how to check for each feature. Start server by typing “server:start()” and then manually test the features implemented.

**Zipf distribution** - In accordance with the specifications, we were required to model a Zipf distribution based on the subscriber count. By requesting an extra variable from the client, maxSubscribers, which specifies the most subscribers a client may have throughout the trial, the Zipf distribution was calculated. The user with the second-highest subscriber count had a total subscriber count of maxSubscribers/2, the user with the next-highest subscriber count had a total subscriber count of maxSubscribers/3, and so on. This was done by employing the equation  $\text{noToSubscribe} = \text{round}(\text{Float.floor}(\text{totalSubscribers}/(\text{noOfClients}-\text{count}+1))) - 1$ , where 'count' is the userId of a client, to determine the amount of peers a person should register to in order to ensure zipf distribution.

**Periods of live connection and disconnection for users** – Disconnect clients, the third parameter used by the program to simulate times of live connection and disengagement, collects the proportion of clients who disconnect. The client's simulation console outputs the performance metrics at the conclusion if the disconnectClients> parameter is set to 0. If not, it outputs the statistics and keeps simulating recurrent live connection and disconnection times.

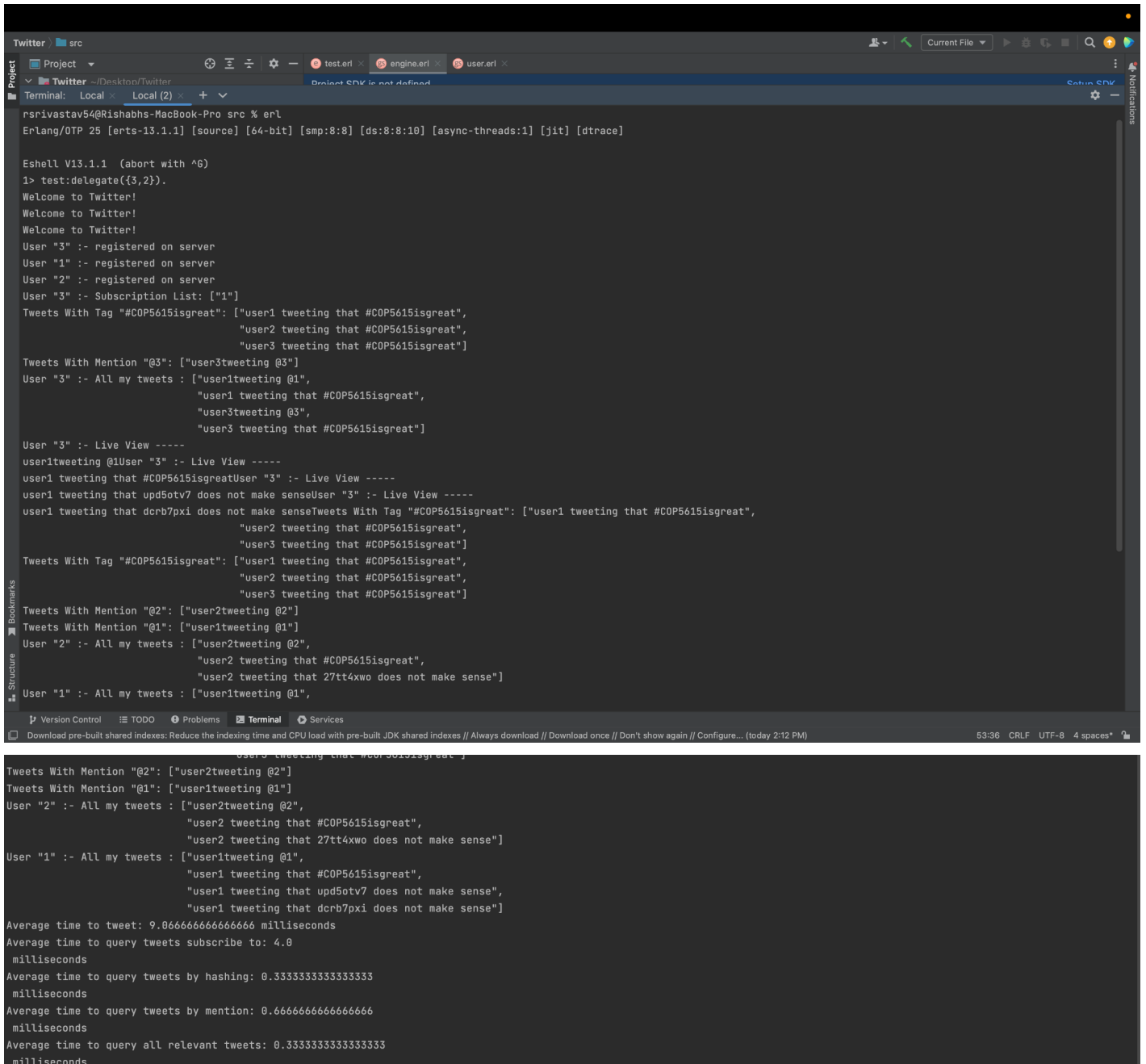
Every client is required to choose a random tweet from one of its subscribers and rebroadcast it to its own subscribers in order to manage retweets. As with the original Twitter, a "-RT" is added to the end of a retweet to distinguish it from regular tweets.

By data transmission to the server and showing the list of tweets acquired on the user end, all queries relating to tweets that were linked to, tweets with particular hashtags, and tweets in which the user is referenced (my mentions) were effectively handled. A person who is online will receive the tweets via live view in real time. In order to determine which user's live view is being updated, User ID is prefixed to this output.

## Observations and graphs for simulation

Testing the simulation for 3 users with maximum of 2 followers each and disconnect and reconnect users parameter set to 0 to obtain statistics

### User Live View:



```
Twitter / src
Project
  Twitter ~/Desktop/Twitter
Terminal: Local x Local (2) x + v
rsrivastav54@Rishabhs-MacBook-Pro src % erl
Erlang/OTP 25 [erts-13.1.1] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [jit] [dtrace]

Eshell V13.1.1 (abort with ^G)
1> test:delegate({3,2}).
Welcome to Twitter!
Welcome to Twitter!
Welcome to Twitter!
User "3" :- registered on server
User "1" :- registered on server
User "2" :- registered on server
User "3" :- Subscription List: ["1"]
Tweets With Tag "#COP5615isgreat": ["user1 tweeting that #COP5615isgreat",
                                     "user2 tweeting that #COP5615isgreat",
                                     "user3 tweeting that #COP5615isgreat"]
Tweets With Mention "@3": ["user3tweeting @3"]
User "3" :- All my tweets : ["user1tweeting @1",
                             "user1 tweeting that #COP5615isgreat",
                             "user3tweeting @3",
                             "user3 tweeting that #COP5615isgreat"]

User "3" :- Live View -----
user1tweeting @1User "3" :- Live View -----
user1 tweeting that #COP5615isgreatUser "3" :- Live View -----
user1 tweeting that upd5otv7 does not make senseUser "3" :- Live View -----
user1 tweeting that dcrb7pxi does not make senseTweets With Tag "#COP5615isgreat": ["user1 tweeting that #COP5615isgreat",
                                     "user2 tweeting that #COP5615isgreat",
                                     "user3 tweeting that #COP5615isgreat"]
Tweets With Tag "#COP5615isgreat": ["user1 tweeting that #COP5615isgreat",
                                     "user2 tweeting that #COP5615isgreat",
                                     "user3 tweeting that #COP5615isgreat"]
Tweets With Mention "@2": ["user2tweeting @2"]
Tweets With Mention "@1": ["user1tweeting @1"]
User "2" :- All my tweets : ["user2tweeting @2",
                             "user2 tweeting that #COP5615isgreat",
                             "user2 tweeting that 27tt4xwo does not make sense"]
User "1" :- All my tweets : ["user1tweeting @1",
                             "user1 tweeting that #COP5615isgreat",
                             "user1 tweeting that upd5otv7 does not make sense",
                             "user1 tweeting that dcrb7pxi does not make sense"]

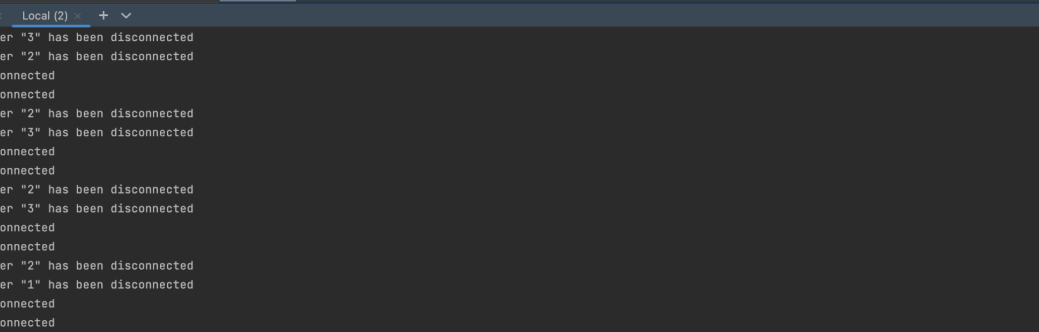
Average time to tweet: 9.066666666666666 milliseconds
Average time to query tweets subscribe to: 4.0
milliseconds
Average time to query tweets by hashing: 0.3333333333333333
milliseconds
Average time to query tweets by mention: 0.6666666666666666
milliseconds
Average time to query all relevant tweets: 0.3333333333333333
milliseconds
```

## Engine view:

```
Twitter / src
Project
  Twitter ~/Desktop/Twitter
Terminal: Local x Local (2) x + v
  test.erl x engine.erl x user.erl x
Project SDK is not defined
Status: SDK
4> test:delegate({}).
Enter your Ip:10.20.170.74
Twitter Engine Start
Server Started
Creating account with ID: "3"
Account created successfully...
Creating account with ID: "1"
Account created successfully...
Creating account with ID: "2"
Account created successfully...
User: "3" has subscribed to user: "1"!
In get tweets - [{"3",[]}]
In get tweets - [{"1",[]}]
Adding user: "3" to user: "1" follower list.
User: "1" is tweeting: "user1tweeting @1"
User: "1" follower list:!!
"3" []
User: "1" is tweeting: "user1 tweeting that #COP5615isgreat"
User: "1" follower list:!!
"3" ["user1tweeting @1"]
User: "2" is tweeting: "user2tweeting @2"
User: "2" follower list:!!
User: "2" is tweeting: "user2 tweeting that #COP5615isgreat"
User: "2" follower list:!!
User: "3" is tweeting: "user3tweeting @3"
User: "3" follower list:!!
User: "3" is tweeting: "user3 tweeting that #COP5615isgreat"
User: "3" follower list:!!
User: "3" subscription list:["1"]
Tweets with HashTag - "#COP5615isgreat" : ["user1 tweeting that #COP5615isgreat",
                                           "user2 tweeting that #COP5615isgreat",
                                           "user3 tweeting that #COP5615isgreat"]
Tweets with Mention: "@3" : ["user3tweeting @3"]
All tweets by User: "3" , tweets list: ["user1tweeting @1",
                                           "user1 tweeting that #COP5615isgreat",
                                           "user3tweeting @3",
                                           "user3 tweeting that #COP5615isgreat"]

User: "2" is tweeting: "user2 tweeting that 27tt4xwo does not make sense"
User: "2" follower list:!!
User: "1" is tweeting: "user1 tweeting that upd5otv7 does not make sense"
User: "1" follower list:!!
"3" ["user1tweeting @1","user1 tweeting that #COP5615isgreat",
     "user3tweeting @3","user3 tweeting that #COP5615isgreat"]
User: "1" is tweeting: "user1 tweeting that dcrb7pxi does not make sense"
User: "1" follower list:!!
"3" ["user1tweeting @1","user1 tweeting that #COP5615isgreat",
     "user3tweeting @3","user3 tweeting that #COP5615isgreat",
     "user1 tweeting that upd5otv7 does not make sense"]
User: "2" subscription list:[]
User: "1" subscription list:[]
Tweets with HashTag - "#COP5615isgreat" : ["user1 tweeting that #COP5615isgreat",
                                           "user2 tweeting that #COP5615isgreat",
                                           "user3 tweeting that #COP5615isgreat"]
Tweets with HashTag - "#COP5615isgreat" : ["user1 tweeting that #COP5615isgreat",
                                           "user2 tweeting that #COP5615isgreat",
                                           "user3 tweeting that #COP5615isgreat"]
Tweets with Mention: "@2" : ["user2tweeting @2"]
Tweets with Mention: "@1" : ["user1tweeting @1"]
All tweets by User: "2" , tweets list: ["user2tweeting @2",
                                           "user2 tweeting that #COP5615isgreat",
                                           "user2 tweeting that 27tt4xwo does not make sense"]
All tweets by User: "1" , tweets list: ["user1tweeting @1",
                                           "user1 tweeting that #COP5615isgreat",
                                           "user1 tweeting that upd5otv7 does not make sense",
                                           "user1 tweeting that dcrb7pxi does not make sense"]
```

**User Live View for testing reconnect and disconnect of user nodes: Testing with 3 users with maximum of 2 followers each and having 50% disconnects and reconnects in between**



```
DoSP Project4 Twitter Clone / src / test.erl
Project
Terminal: Local x Local (2) + v
Simulator :- User "3" has been disconnected
Simulator :- User "2" has been disconnected
User "2" :- reconnected
User "3" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "2" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "2" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "2" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "2" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "2" :- reconnected
Simulator :- User "3" has been disconnected
Simulator :- User "1" has been disconnected
User "1" :- reconnected
User "2" :- reconnected
Simulator :- User "2" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
Simulator :- User "3" has been disconnected
Simulator :- User "1" has been disconnected
User "1" :- reconnected
User "3" :- reconnected
Simulator :- User "1" has been disconnected
Simulator :- User "3" has been disconnected
User "3" :- reconnected
User "1" :- reconnected
Simulator :- User "2" has been disconnected
```

Version Control | TODO | Problems | Terminal | Services

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (19 minutes ago)

4:32 1F UTF-8 2 spaces

```
DoSP Project4 Twitter Clone | src | test.erl
Project
Terminal: Local | Local (2) | + | -
Eshell V13.1.1 (abort with ^G)
i> test:delegate({3,2,50}).
Welcome to Twitter!
Welcome to Twitter!
Welcome to Twitter!
User "1" :- registered on server
User "2" :- registered on server
User "3" :- registered on server
User "3" :- Subscription List: ["1"]
Tweets With Tag "#COP5615": ["user1 tweeting that #COP5615 is great",
                             "user2 tweeting that #COP5615 is great",
                             "user3 tweeting that #COP5615 is great"]
Tweets With Mention "@3": ["user3tweeting @3"]
User "3" :- All my tweets : ["user1tweeting @1",
                             "user1 tweeting that #COP5615 is great",
                             "user3tweeting @3",
                             "user3 tweeting that #COP5615 is great"]
User "3" :- Live View -----
user1tweeting @1User "3" :- Live View -----
user1 tweeting that #COP5615 is greatUser "3" :- Live View -----
user1 tweeting that n-btihn6 for testingUser "3" :- Live View -----
user1 tweeting that fhxf4n8w for testingTweets With Tag "#COP5615": ["user1 tweeting that #COP5615 is great",
                             "user2 tweeting that #COP5615 is great",
                             "user3 tweeting that #COP5615 is great"]
Tweets With Tag "#COP5615": ["user1 tweeting that #COP5615 is great",
                             "user2 tweeting that #COP5615 is great",
                             "user3 tweeting that #COP5615 is great"]
Tweets With Mention "@1": ["user1tweeting @1","user2tweeting @1"]
User "1" :- All my tweets : ["user1tweeting @1",
                             "user1 tweeting that #COP5615 is great",
                             "user1 tweeting that n-btihn6 for testing",
                             "user1 tweeting that fhxf4n8w for testing"]
Simulator :- User "3" has been disconnected
Simulator :- User "2" has been disconnected
User "2" :- reconnected
User "3" :- reconnected
Version Control | TODO | Problems | Terminal | Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (19 minutes ago)
```



## Engine View for testing reconnect and disconnect of user nodes: Testing with 3 users with maximum of 2 followers each and having 50% disconnects and reconnects in between

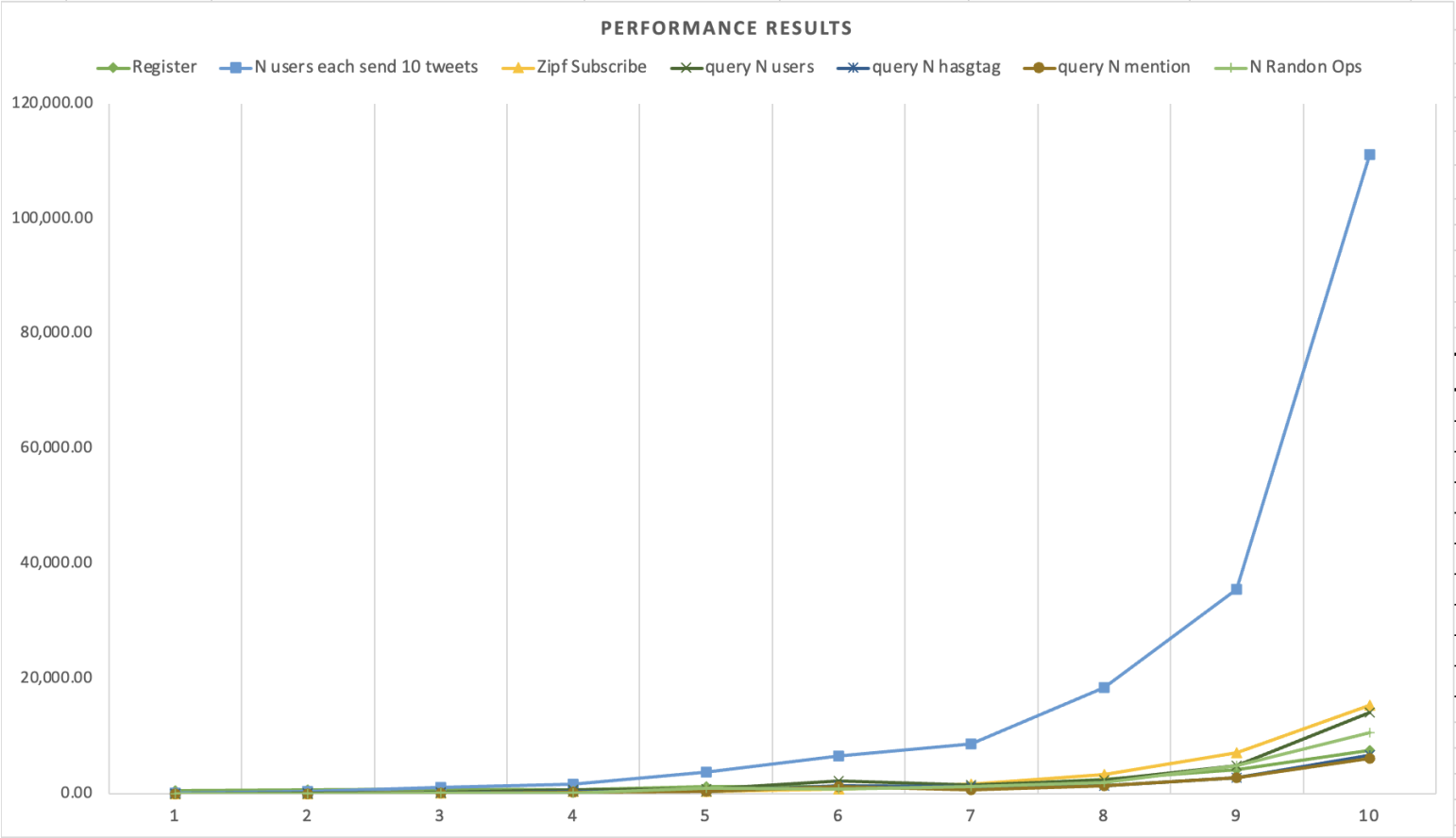
```
DoSP Project4 Twitter Clone / src / test.erl
Terminal: Local < Local (2) < + < v
Eshell V13.1.1 (abort with ^G)
1> test:delegate({}).
Enter your Ip:10.20.170.74
Twitter Engine Start
Server Started
Creating account with ID: "1"
Account created successfully...
Creating account with ID: "3"
Account created successfully...
Creating account with ID: "2"
Account created successfully...
User: "3" has subscribed to user: "1"!
In get tweets - [{ "3", [] }]
In get tweets - [{ "1", [] }]
Adding user: "3" to user: "1" follower list.
User: "2" is tweeting: "user2tweeting @1"
User: "2" follower list:!!
User: "2" is tweeting: "user2 tweeting that #COP5615 is great"
User: "2" follower list:!!
User: "1" is tweeting: "user1tweeting @1"
User: "1" follower list:!!
"3" []
User: "1" is tweeting: "user1 tweeting that #COP5615 is great"
User: "1" follower list:!!
"3" ["user1tweeting @1"]
User: "3" is tweeting: "user3tweeting @3"
User: "3" follower list:!!
User: "3" is tweeting: "user3 tweeting that #COP5615 is great"
User: "3" follower list:!!
User: "3" subscription list:["1"]
Tweets with HashTag - "#COP5615" : ["user2 tweeting that #COP5615 is great",
                                     "user1 tweeting that #COP5615 is great",
                                     "user3 tweeting that #COP5615 is great"]
Tweets with Mention: "@3" : ["user3tweeting @3"]
All tweets by User: "3" , tweets list: ["user1tweeting @1",
                                         "user1 tweeting that #COP5615 is great",
                                         "user3tweeting @3",
                                         "user3 tweeting that #COP5615 is great"]
Version Control  TODO  Problems  Terminal  Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (26 minutes ago)
18:47 LF UTF-8 2 spaces
```

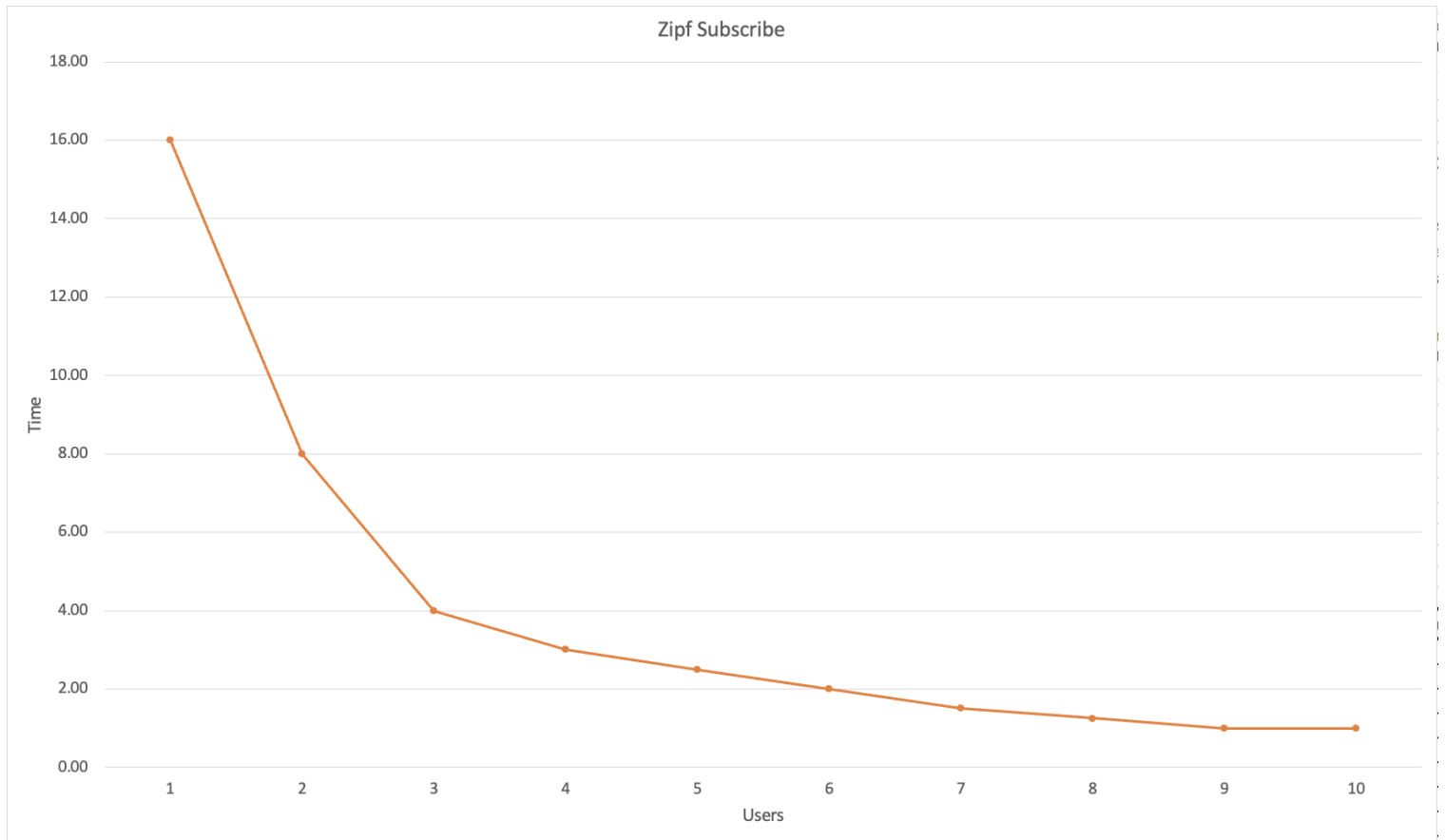
```
DoSP Project4 Twitter Clone / src / test.erl
Terminal: Local < Local (2) < + < v
User: "3" is tweeting: "user3 tweeting that ikf5s/cj for testing"
User: "3" follower list:!!
User: "1" is tweeting: "user1 tweeting that fywtqicx for testing"
User: "1" follower list:!!
"3" ["user1tweeting @1", "user1 tweeting that #COP5615 is great",
     "user3tweeting @3", "user3 tweeting that #COP5615 is great",
     "user1 tweeting that c5/fz3h7 for testing",
     "user1 tweeting that qlbdaffl for testing",
     "user1 tweeting that bnfcfsxtx for testing",
     "user1 tweeting that 3zb5v28f for testing",
     "user1 tweeting that rduntaf3 for testing",
     "user1 tweeting that sc6uhp3f for testing",
     "user1 tweeting that +7wakkax for testing",
     "user3 tweeting that hi/l/hd1 for testing",
     "user3 tweeting that cpugchjg for testing",
     "user3 tweeting that njy7agwc for testing",
     "user3 tweeting that d8oiehrp for testing",
     "user3 tweeting that ikf5s/cj for testing"]
User: "1" is tweeting: "user1 tweeting that 9cygqkj for testing"
User: "1" follower list:!!
"3" ["user1tweeting @1", "user1 tweeting that #COP5615 is great",
     "user3tweeting @3", "user3 tweeting that #COP5615 is great",
     "user1 tweeting that c5/fz3h7 for testing",
     "user1 tweeting that qlbdaffl for testing",
     "user1 tweeting that bnfcfsxtx for testing",
     "user1 tweeting that 3zb5v28f for testing",
     "user1 tweeting that rduntaf3 for testing",
     "user1 tweeting that sc6uhp3f for testing",
     "user1 tweeting that +7wakkax for testing",
     "user3 tweeting that hi/l/hd1 for testing",
     "user3 tweeting that cpugchjg for testing",
     "user3 tweeting that njy7agwc for testing",
     "user3 tweeting that d8oiehrp for testing",
     "user3 tweeting that ikf5s/cj for testing",
     "user1 tweeting that fywtqicx for testing"]
User: "1" is tweeting: "user1 tweeting that n5fdtben for testing"
User: "1" follower list:!!
Version Control  TODO  Problems  Terminal  Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (26 minutes ago)
18:47 LF UTF-8 2 spaces
```

**Tabular Results:**

N	Register	N users : 10 tweets	Zipf Subscribe	N users query	N hashtag query	N mention query	N Randon Ops
5	679.67	264.60	101.87	131.10	109.26	104.12	131.52
10	739.86	467.17	130.54	183.96	127.12	136.18	133.24
25	887.44	1,271.58	208.70	326.32	208.74	203.45	251.58
50	849.08	1,771.38	547.98	844.24	573.44	418.94	299.02
100	1,261.37	3,668.31	366.34	850.03	353.11	369.27	867.73
200	1,142.64	6,475.22	775.96	2,189.67	1,310.85	1,246.93	777.67
400	1,361.06	8,579.69	1,550.86	1,428.99	1,178.31	616.30	1,205.13
900	2,635.78	18,551.73	3,458.86	2,468.88	1,577.32	1,562.44	2,135.49
1800	4,364.85	35,759.25	7,345.58	5,024.35	3,010.32	2,971.00	5,084.73
3600	7,442.84	110,990.34	15,225.70	13,889.39	6,575.74	5,946.99	10,498.63

**Graphs:**





### **Largest Network we managed to test our application with:**

The largest number of users we managed to test with is 3600 and the results for that are:

The time to register 3600 users is 7442.84 ms  
The time to send 10 tweets is 110990.34 ms  
The time to Zipf subscribe 3200 users is 15225.70 ms  
The time to query 3600 users is 13889.39 ms  
The time to query 3600 hashtags is 6575.74 ms  
The time to query 3600 mentions is 5946.99 ms  
The time to 3600 random operations is 10498.63 ms

### **Conclusion:**

The twitter clone application was simulated successfully using the erlang actor model. Simulation results were tabulated and plotted graphically to understand trends and patterns between various features and querying procedures. We plan to further scale this backend model by implementing the UI Design for it in the next project part.