CS 433 : Computer Networks Project

Mini Tweet

Overview :

- Client and server connect through TCP connection.
- Server creates separate threads for every user.
- Server provides various options to client to choose from.
- Client responds with the option corresponding to action he wants to perform.
- Server performs the actions for client using the database and gives him next set of option.
- This goes on until the client exits.

A Bird's view:

- Client has two states:
  - For entering Passwords
  - For non password entries

- Server has many states.

- Server is made concurrent through threading.
  - Multiple client can connect to the server at the same time.

- All the functions are executed at the server side.

- Client has two functions:
  - Sends inputs from the user to the server
  - Receive response from the server and show it to the user.

Features:

1. Any client/user can register and set up an account with Mini-Tweet.
2. Client can login, get the feeds and logout.
3. Client can search registered users, follow/unfollow any users and control add/delete followers
4. Supports user to post tweets with hashtags.

Features:

5. Allows user to search and display tweets under specific hashtags. Can show the Top 5 trending hashtags.

6. Client can see the list of active/online followers.

7. Allows user to use other users’ tweets and post the retweets.

8. Concurrent server that can handle several client requests.

9. Users are authenticated with the server before trying to access any of the features.

10. When a user is prompted for a login password, the user input for the password is obscured/masked.

Database :

We have used five different collections:

- **User collection**

```
{

        <user_id>:  {
                              “user_name”: <user_name> ,
                               “password”: <encrypted_password>,
                               “followers” :  [<list_of_user_ids_of_followers> ],
                               “following” : [<list_of_user_ids_of_following >]
                          }

}
```

Databases :

- User name to Id:

```
{
        "<user_name>" : <user_id>
}
```

- Tweet Table :

```
tweet_id : <int>,
user_id_created: <int>,
content  : <string>,
date_time_created : <date time>,
last_update_time : <date time>
```

Databases :

- Hashtag collection:

```
{
        “<hash_tag>” :
        {
          “tweet_count”: <count>,
          “tweet_ids” : [ <tweet_ids> ]

         }
```

- Active users :

```
user_id : <int>,
active  : <int>
```

Thank You !