***JULY 1, 2020***

# GROUP CHAT APPLICATION IN PYTHON

## COMPUTER NETWORKS

### PROJECT REPORT

# DESIGNED & DEVELOPED BY


- *180374 – ABDULLAH*
- *180380 – ABDUR RAHMAN*
- *180369 – USAMA AMEER*
- *180392 – RAVEEHA RIZWAN*
- *180397 – MALIK EHTISHAM ALI*


# SUBMITTED TO

- ***EE-Engr. WAJIHA JAVED***
- ***EE-Engr. Dr. AHMED TARIQ SHEIKH***

# *Table of Contents*

# OBJECTIVES

The objective of this complex engineering activity is to carry out research, analysis, design, Investigation, and implementation of a real-world complex software engineering project that has the following attributes:

1. **Depth of Analysis Required**: The activity requires abstract thinking, originality in analysis to formulate suitable software models of the activity.

2. **Innovation:** The activity involves creative use of engineering principles and research based knowledge in novel ways.

3. **Familiarity:** The activity can extend beyond previous experiences by applying principles-based approaches.

# INTRODUCTION

If you're the owner of a business, you probably use consumer messaging apps such as WhatsApp, Facebook Messenger, Snapchat, Viber or even Skype to group chat with Clients, Suppliers, family and friends. Group chat lets you instantly send messages, pictures and videos to multiple people at once while seamlessly keeping a conversation going for days, weeks, or months.

We use them for work, school friends, sport and birthdays. These chats bridge the gap between struggling to maintain connections and avoiding isolation. They make you seem a little less closed off from everything, when your world is finally beginning to open up for the first time.

Business Group Chat is starting to become a popular choice for many company and business owners. It is very essential and looking at the numerous advantages it contributes to the business. It is a means of communication that allows the owners to reach more people at the same time. The group chat can be used for presentations or video calls. This helps especially when you have to communicate with employees who are in different parts of the country as it gets you closer and improves the sense of urgency. It helps users learn and work together to achieve a common goal.

When you have a group of people you want to get together for something you can just add everyone into a group chat. So everyone can discuss where they want to go or what they want to do, instead of separately texting everyone because someone could get left out that way.

Privacy is an essential element for the business. The consumer messaging apps do not guarantee our privacy. The can use our private data into some other work, In fact, against us. Most importantly, if there system went down your business can collapse too. So having your own private messaging app could be a relief.

Various academic announcements that will need to be made, in order to inform students and teachers of the developments, that may be of interest to them. Everything from emergency alerts to relevant discussions can be delivered via group chats, so it isn't exactly something that can be taken lightly. It's quite the handy system.

The primary use of a group chat is to share information via text with a group of other users. Generally speaking, the ability to converse with multiple people in the same conversation differentiates chat rooms from instant messaging programs. The users in a particular chat room are generally connected via a shared internet or other similar connection, and chat rooms exist catering for a wide range of subjects. New technology has enabled the use of file sharing and webcams to be included in some programs. This would be considered a group chat.

# BACKGROUNG

The first Group Chat system was used by the U.S. Government in 1971. It was developed by Murray Turoff, a young PhD graduate from Berkeley, and its first use was during President Nixon's wage-price freeze under Project Delphi. The system was called EMISARI and would allow 10 regional offices to link together in a real-time online chat known as the party line. It was in use up until 1986.

The first public online chat system was called Talkomatic, created by Doug Brown and David R. Woolley in 1973. It offered several channels, each of which could accommodate up to five people, with messages appearing on all users' screens character-by-character as they were typed. Talkomatic was very popular among PLATO users into the mid-1980s.

Now-a-days, the most famous and widely use group chat application is "WhatsApp" owned by "Facebook". It has more than 2 billion users in 180 countries. WhatsApp is also available for businesses called "WhatsApp Business". But having a private chat is still awesome and have several advantages in different ways. However, if someone cannot afford or manage private sever, then they should use these applications.

# WORKING

We started with the very basic code of socket programming in python of server and client. Then implemented Multi-threading on both side. Then we made different functions in client and server, whenever client sends request to server, that request has a specific pattern. Then server interprets the request and send specific request to specific function. Then that specific function will perform specific task.

We also create Repository on Git to collaborate with our friend. (Repository link:: https://github.com/Abdullah296/MessangingAppPython ). Here are some Functions which we have implement in our code. We also did port forwarding on Server's router so that our Server can be access by any place. We also try to set our server on some cloud service.

- **Sign In/Sign Up:** First of all, the client will be prompt by the sign in/sign up menu. If the client is not the register user then the client will sign up first. Client will enter his Name, E-mail(optional) and password. The server will generate a unique ID for the client. The client will use this ID to sign in again. The maximum number of clients at a time in this program is 10. It can be change easily.
- **Menu:** After the sign in, the client will be prompt by another menu where he can see all the available options. He can send message to any register client, create group, add contact and much more.
- **Create New Group:** When a client creates a group and add other register clients. All the other added clients will get a request notification, which they can accept or reject.
- **Change Group Admin:** Here the group admin can remove himself from the administration and transfer the administration of the group to any other client.
- **Add to Group:** If the admin wants to add any other client to the existing group. Then here he can add client. The client will be prompt by the request, which client can accept or reject.
- **Remove from the Group:** If the admin wants to remove any client from the existing group. Then here he can remove any client. The client will be removed directly.
- **Chat:** Here the user client can send any message to the other register client and can chat with that client. Client can also send file in the group.
- **Go Offline:** Client can change his status from online to offline and can exit from the program.
- **My Profile:** Client can see his profile. The profile exists of the Name, Unique ID, E-mail(if added) and Password.
- **Add New Contact:** The client can add register clients as contacts in his profile.
- **Contacts:** The client can see all his added contacts in the profile.
- **View Group Members:** The client can all his groups.
- **View Messages:** The client can see all the received messages from the other clients.
- **View Requests:** The client can see all the received requests from the other clients and groups.
- **Exit:** The client can exit from the program.

# OUTPUT

```
1. Go Online
2. Exit
>>>1
Creating Socket
connecting to server : ('localhost', 8080)
connected to the server

1. Sign UP
2. Sign In
>>>1
Enter your User Name :bce
Enter Your Password :1234

Sign Up Successfully
Your user name : bce
Your ID is : 2
```

*Figure 1: Client Sign-Up*

```
Creating Socket ...
Running Server at address : ('localhost', 8080)
Server Running Successfully
Waiting for connections
got a connection from  ('127.0.0.1', 50367)
Waiting for connections
['r', 'up', '1234']
Generating Random key ...
Adding to Client's list ...
{'2': '1234'}
Added Successfully ...
Sending response to Client ...
Sended Successfully ...
updating Client's status ...
updated successfully ...
{'2': 'offline'}
```

*Figure 2: Server Response to sign up*

```
1. Sign UP
2. Sign In
>>>2
Enter Your Password :1234

Sign in successful


1. Create New Group
2. Change Group Admin
3. Add to Group
4. Remove from Group
5. Chat
6. Go Offline
7. My Profile
8. Add New Contact
9. Contacts
a. View Group Members
b. View Messages
c. View Requests
d. Exit
```

*Figure 3: Client Sign In*

```
updated successfully ...
{'2': 'offline'}
['r', 'in', '2', '1234']
Sign in request form  2
ID found ...
Password matched ...
{'2': '1234'}
Adding Client's Socket to Available Clients Lists ...
added successfully ...
available clients are
{'2': <socket.socket fd=584, family=AddressFamily.AF_INET, type=SocketK:
updating Client's status ...
updated successfully ...
{'2': 'online'}
```

*Figure 4: Server Response to sign in*

```
>>>1
Enter Group Name :cn
Enter 'end' to exit
Enter User Id to Add in this Group:10
Enter User Id to Add in this Group:\end
Enter User Id to Add in this Group:end
```

```
>>>
Group Created
 My Group are:
 {'g7': 'cn'}
```

*Figure 5: First Client Creating Group & inviting the second Client*

```
>>>5
1. In group
2. with client
3. Exit
>>>2
{}
Enter Client's ID :10
1. Send message
2. Send File
>>>1
Enter \end to  End conversation
>>>hello
```

*Figure 6: First Client Sending Message*

```
Got Messages (1)
b
From 2 --> hello
```

*Figure 7: Second Client Receiving Message*

```
c. View Requests
d. Exit
>>>c
0. Group ID: g7 Group Name: cn Present Response: pen
['req<gjr<g7<cn']

1. Change Response
2. Exit
>>>1
Enter Request No: 0
Enter 'yes' (for joining)
Enter 'no' (for rejecting)
Enter 'pending' (for later)
>>> yes
Enter Group ID: g7
```

*Figure 8: Second Client Checking the group invitation request & Accepting it*

```
6. Go Offline
7. My Profile
8. Add New Contact
9. Contacts
a. View Group Members
b. View Messages
c. View Requests
d. Exit
>>>6
1. Go Online
2. Exit
```

*Figure 9: Client Went Offline in the program*

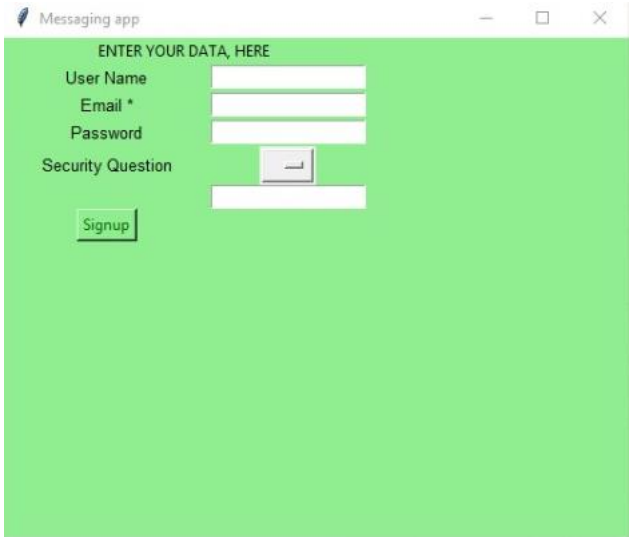


*Figure 10: GUI Sign-Up Page*


*Figure 11: GUI Sign-In Page*

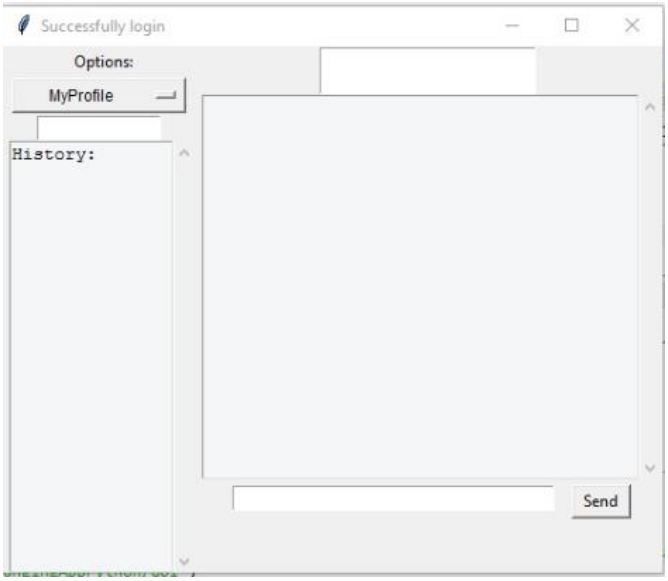


*Figure 12: GUI Page After Sign-In*

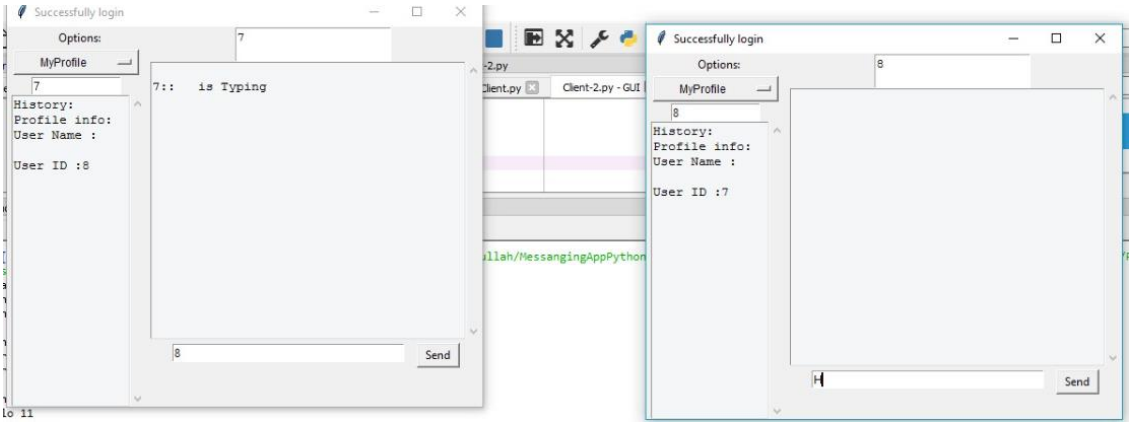


*Figure 13: Client is Typing*

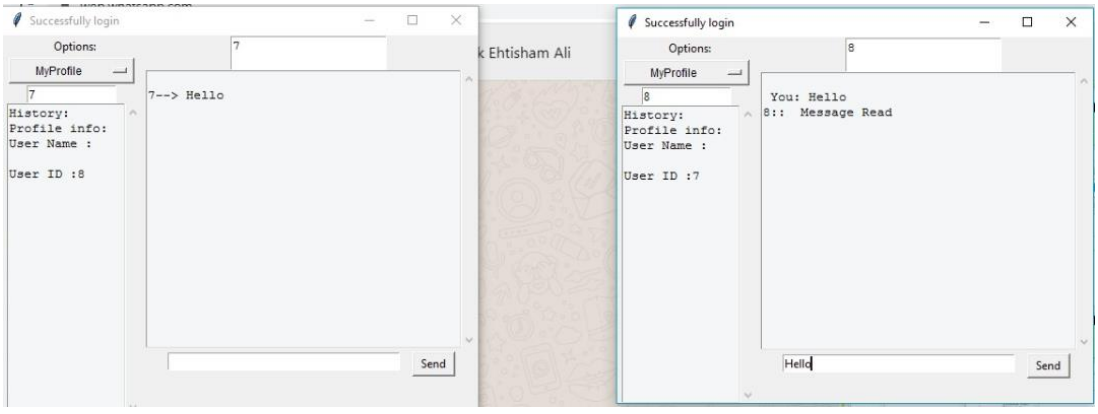


*Figure 14: GUI Message send and received*

# ADDITIONAL WORK

We have also tested our server by running it on a virtual machine on an operating system called “Centos OS” in USA and remotely accessed from Islamabad. We connected the clients from Gujranwala and Bahawalpur. They successfully connected to each other and were able communicate easily.

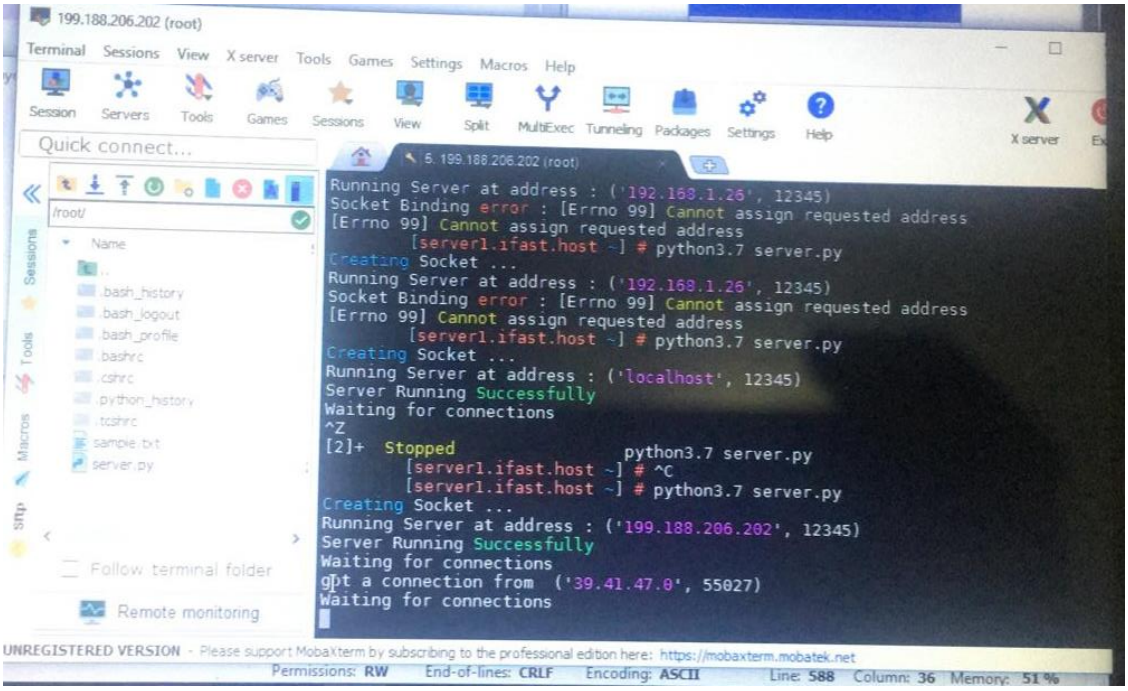


*Figure 15: Server Running On a Virtual Machine*

# CONCLUSION

*We successfully created the project which is Chatting application which enable us to send message to single client or in a group chat and also file sharing between group members. This Project uses Client-Server Model for its working which is connected by TCP protocol. There are unique pattern for every request which are defined, which enable client to communicate with server and vice versa. Every request to the server and response from the server enables communication. Every action which a client can execute has a specific pattern to carry that action. For Example to send a message to a single client or in a group there is a format (m<ID<message) which automatically has to followed to send message. By these formats Server and client both can understand each other what other is trying to do.*

*This Complex Engineering Activity also enable us to search and do more on the python projects. It helped us to understand the networking in python. We learnt a lot of new things in python while doing this group project. We leant to do projects in group in a good manner.*