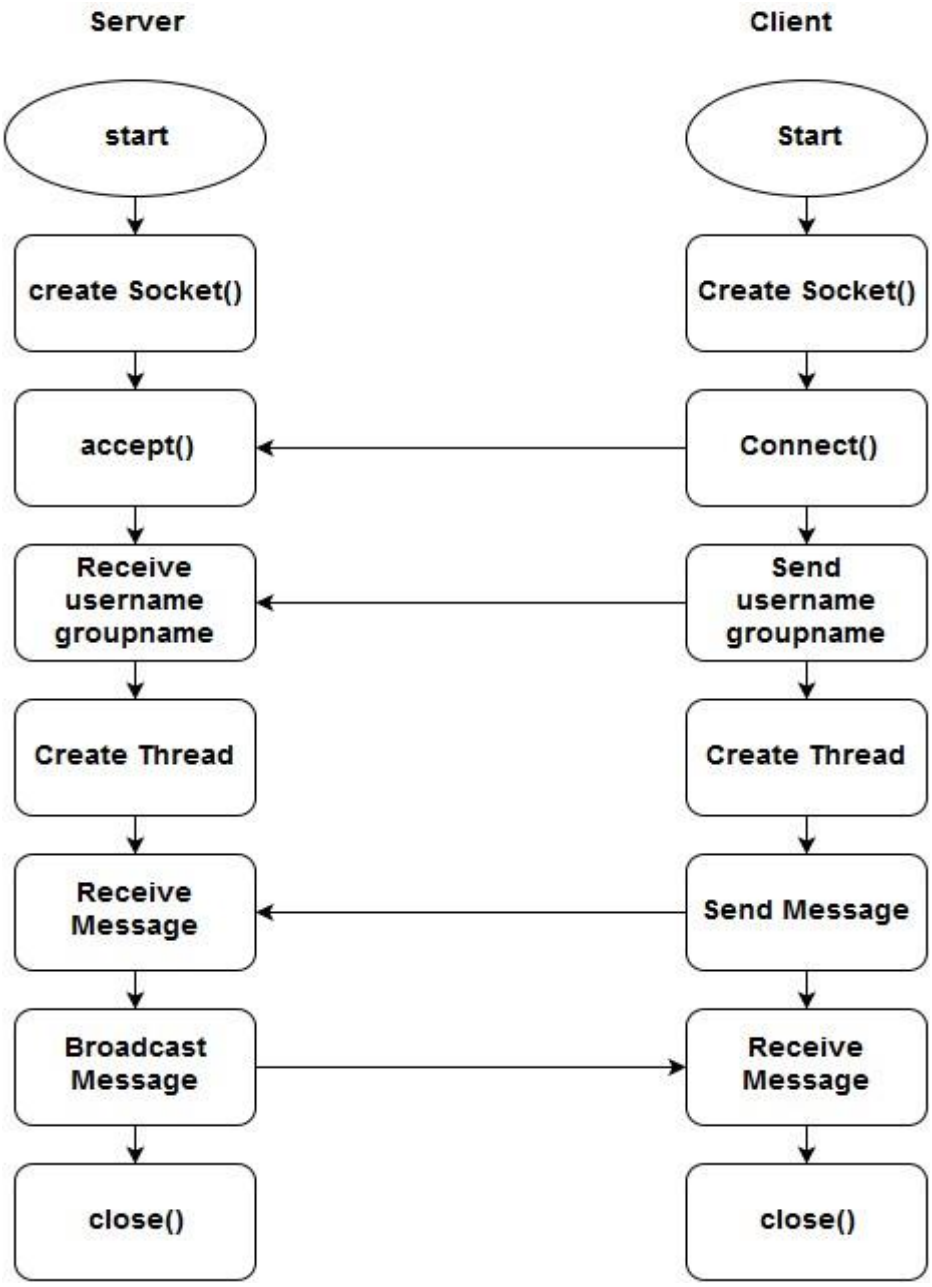
Introduction and Problem Statement:

This is a Centralized chat application project. Here multiple chat rooms are available. It consists of implementing two components: a chat client and a chat server. In this implementation, the server maintains information about the clients that have registered with a specific chat group and pass the messages sent by the clients. Here a client can join a group at a time that means joining in multiple groups are not available. He can leave the group and can join another group.

Flow chart:



Socket()

A socket is one endpoint of a two-way communication link between two programs running on a network.

Accept()

The accept() is used to retrieve a connect request and convert that into a request.

- Carries out the passive open
- Blocking operation
- Does not return until a remote participant has established a connection
- When it does, it returns a new socket that corresponds to the new established connection and the address argument contains the remote participant's address

Connect()

The connect function is used by a TCP client to establish a connection with a TCP server.

- Does not return until TCP has successfully established a connection at which application is free to begin sending data
- Address contains remote machine's address

Close()

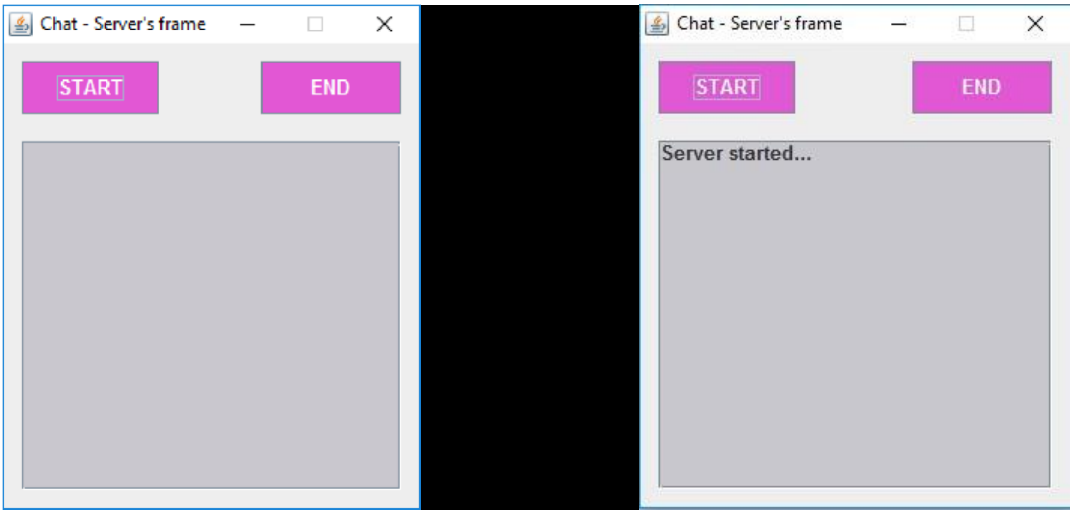
This method is used to destroy the socket.

Implementation:

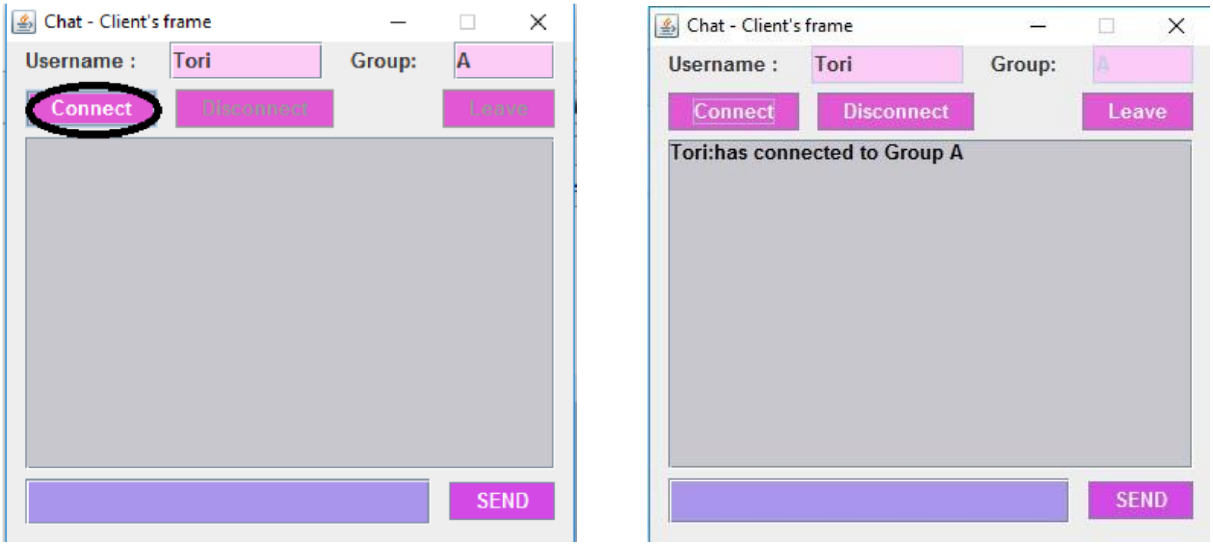
We use java swing to create the interface. There are two classes. Client and Server. We use arraylist to store usernames of client. We have a 2D string name data. Where we store username, message, type (connect/disconnect/chat), group, leave (in a group or left a group). If the type is chat, server send the message only to the same group members. If the type is connect, it will add the user, and if the type is disconnect, it will remove the user from the arraylist.

Screenshots:

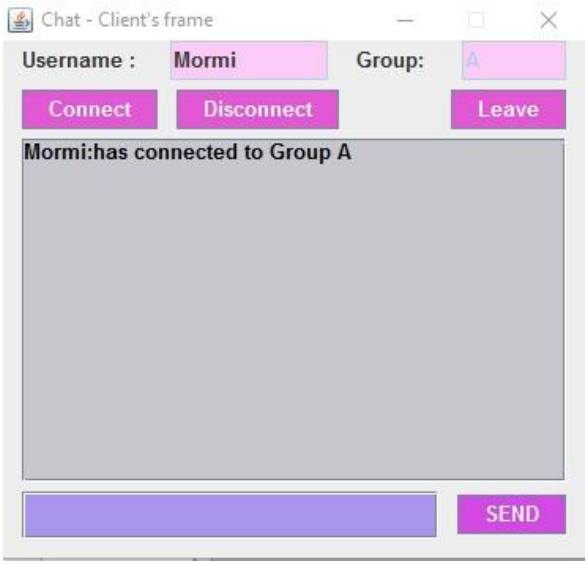
Server is started

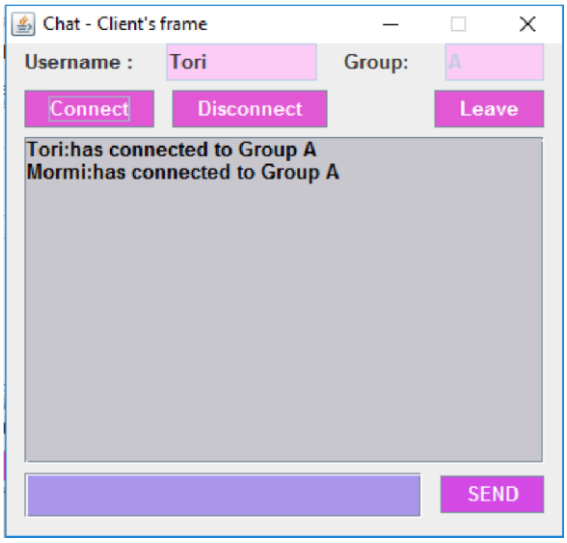


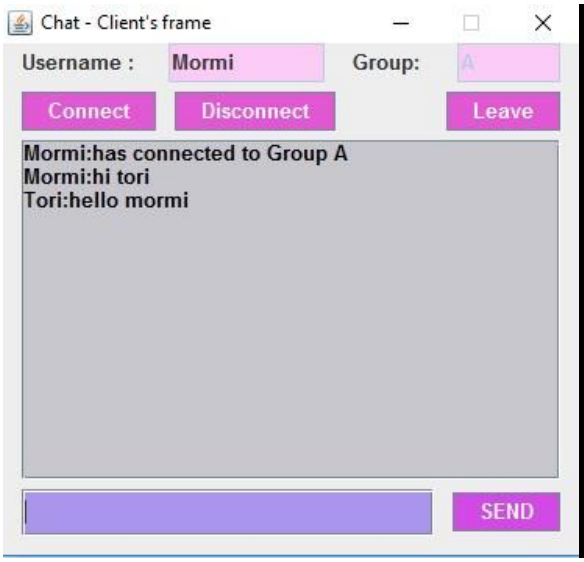
Initially Disconnect and Leave button are not visible. When a client is connected then the buttons are visible. If username field is empty, you are asked to enter username first. Without group name you cannot connect. After giving username and groupname you are able to connect on that group.

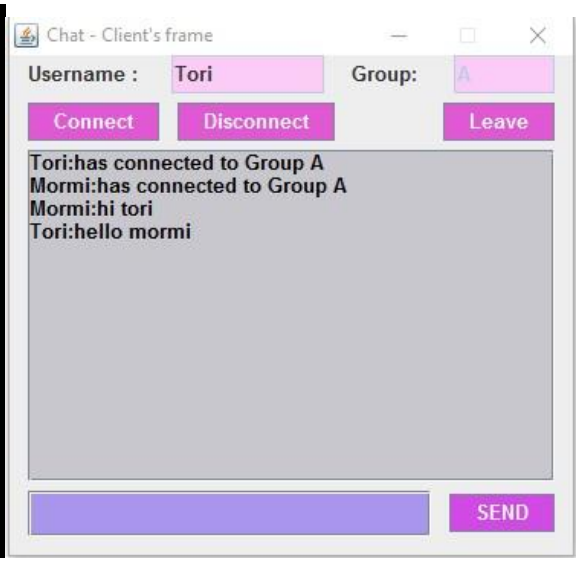


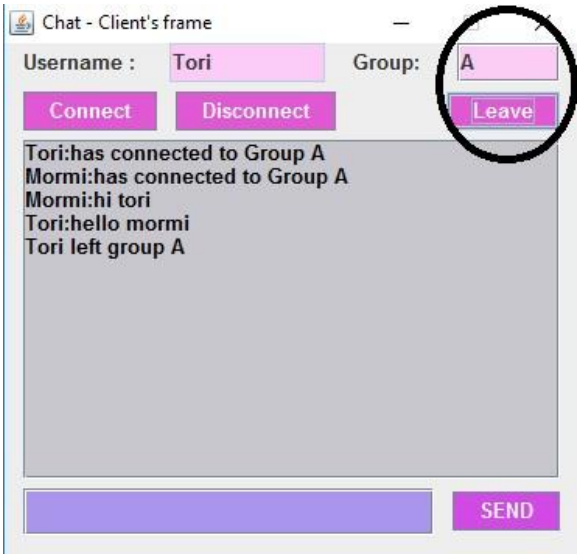
When another member is connect to your group you are able to see that. Only group members are able to see the messages send by the group members. Other group members are not able to see messages.

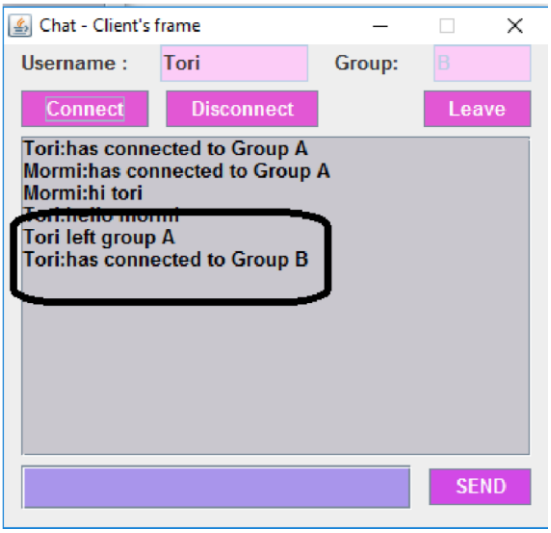

Chat - Client's frame
Username :
Mormi
Group:
A
Connect
Disconnect
Leave
Mormi:has connected to Group A
SEND


Chat - Client's frame
Username :
Tori
Group:
A
Connect
Disconnect
Leave
Tori:has connected to Group A
Mormi:has connected to Group A
SEND

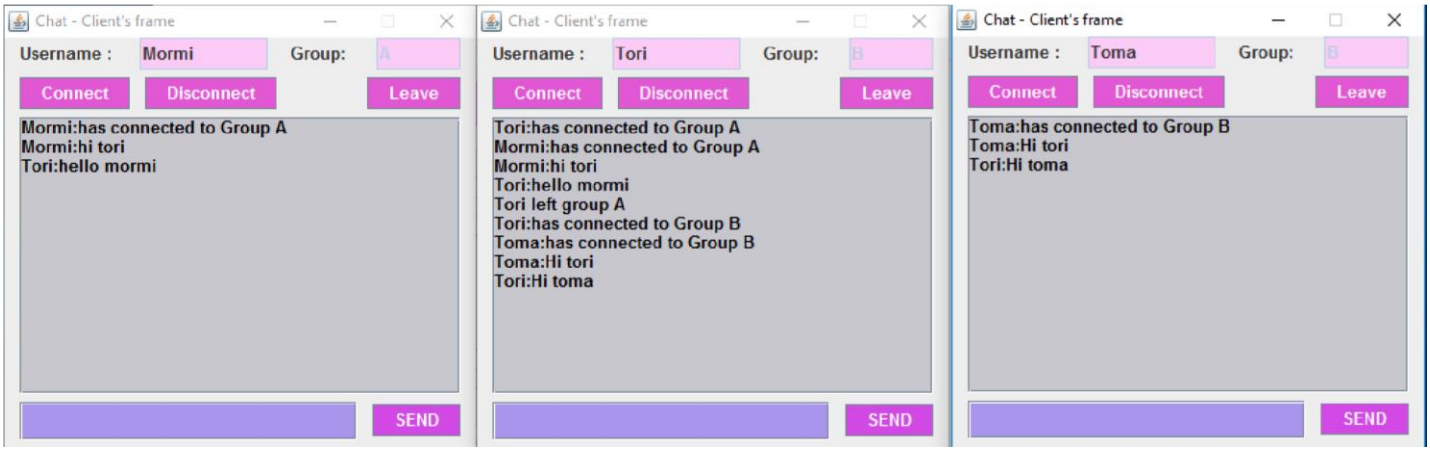

Chat - Client's frame
Username :
Mormi
Group:
A
Connect
Disconnect
Leave
Mormi:has connected to Group A
Mormi:hi tori
Tori:hello mormi
SEND


Chat - Client's frame
Username :
Tori
Group:
A
Connect
Disconnect
Leave
Tori:has connected to Group A
Mormi:has connected to Group A
Mormi:hi tori
Tori:hello mormi
SEND


Chat - Client's frame
Username :
Tori
Group:
A
Connect
Disconnect
Leave
Tori:has connected to Group A
Mormi:has connected to Group A
Mormi:hi tori
Tori:hello mormi
Tori left group A
SEND


Chat - Client's frame
Username :
Tori
Group:
B
Connect
Disconnect
Leave
Tori:has connected to Group A
Mormi:has connected to Group A
Mormi:hi tori
Tori left group A
Tori:has connected to Group B
SEND

You can leave a group and can join another group.



Conclusion and future improvements:

We have learned Java GUI using Swing Framework and could able to chat between multiple clients in chat rooms through a server. In Future we can take our work further and send files from one client to another client. It has been a great learning experience.