

LAN BASED CHAT SYSTEM

Created by –

Shreayaas S Iyer – 2018506115

Shriram G – 2018506116

V Hariharakrishna – 2018506035

About our project –

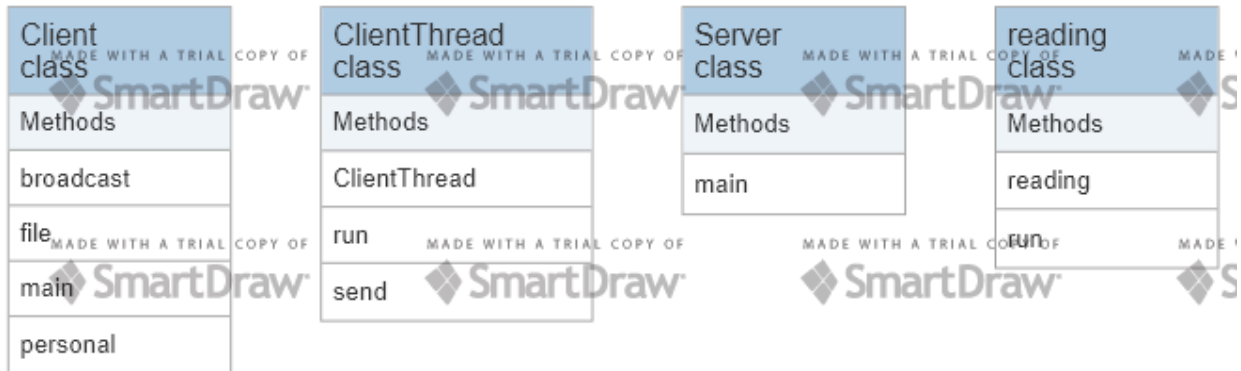
As the name suggests, we implemented a chat room using *socket programming* in Java. The chat system can do the following –

1. Allow users to chat in a public chat room.
2. Allow users to send private messages to other users.
3. Allow users to send files to other users.

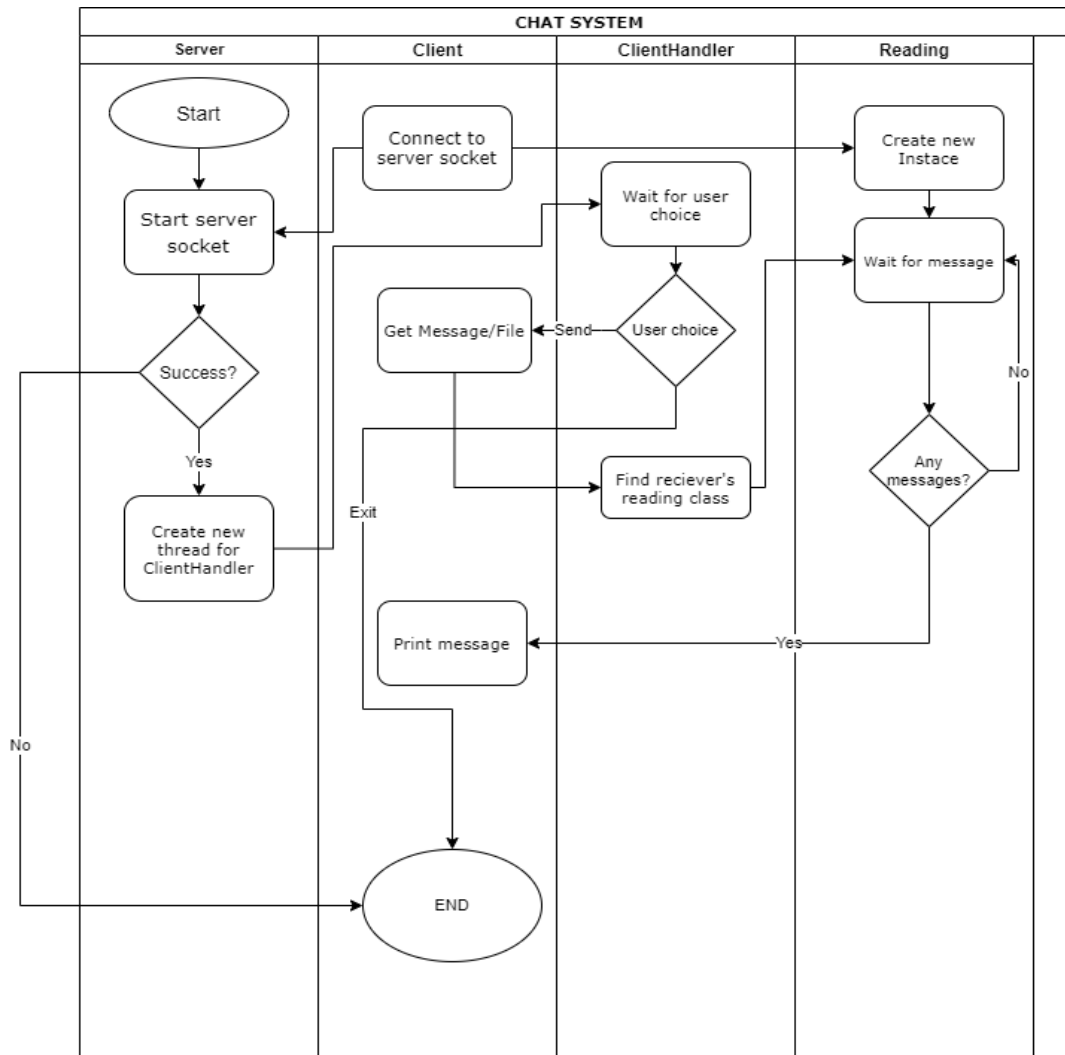
The concepts we used to create this project are –

1. Multithreading – Every time a user logs on to the server, a new thread is created running a new **ClientHandler** instance.
2. Socket Programming – The clients are connected to the Server program(which runs all the time) using the port – 1500.
3. File Handling – The users can send files to each other provided they mention the paths of the files.
4. Exception Handling – Since this project pretty much deals with Input and Output streams along with files, exceptions had to be handled at every point.

CLASS DIAGRAM –



Design Swimlane diagram –



MODULES –

The application contains two java files – Client and Server. The Server program must run before any client tries to connect.

Every time a Client program is run on a terminal, a new instance of “ClientHandler” is created as a thread. It has three modules –

1. **personal** – Used to create a private chat between two users. If a user wants to send a private message to another user, he must enter his userid. The Server side checks whether that user is present in the list of active users, and then sends the message to that user.
2. **broadcast** – Used to send message to all active users in that chat room. If a user wants to send a broadcast, he types the message and the server iterates through the list of active users and sends the message to all of them.
3. **file** – The user can send files to each other on different terminals of the same device – multiple users over local host. Every user must specify a path at the beginning where the received file must be saved. The file at that path must exist – it may or may not be empty. When a user wants to send a file, he must specify the path of the file to be sent – again it must exist but must not be empty; along with the userid of the receiver. The sent file will get saved in the path of the file specified by the receiver.
4. **send** – Every ClientHandler has a send module which is used as an interface for receiving messages from other users. For eg – `client1.send("//message of client 2//");` sends the message of client2 to client1.
5. **reading** – Every ClientHandler will run an instance of the Reading class, which runs infinitely until the user exits. Its used to read incoming messages sent by other users using the “send()” function.

COST ESTIMATION –

The application consists of 2 files – Server and Client.

We have estimated the cost of our project using the LOC(Lines of Code) technique.

Assuming an employee drawing a salary of Rs.10,000 p/m can code up-to 600 lines of core code in a month, the cost of every line of that code is Rs.17

Our project consists of 300 lines of essential code. So, the total cost of our project is $300 \times 17 = \text{Rs.5100.}$

OUTPUT TEST CASES –

1. Basic test case – two users using broadcast.

Server-

```
F:\Java Project_1\src>java Server
Enter name of chat room :
WT
Chat room WT created...
Waiting for clients to connect...

Shreayas connected to chatroom...
Shriram connected to chatroom...
Shriram is on group chat
[group -Shriram] : Hello
Shreayas is on group chat
[group -Shreayas] : Hi
[group -Shriram] : How are you?
[group -Shreayas] : Im fine.
```

Client 1 –

```
F:\Java Project_1\src>java Client
Enter the username:
Shreayas
Enter the path where you want to download files :
F:\fileRecieved.txt
Joined to chat room : WT
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
[Shriram] : Hello
1
Type exit for go to menu...
Hi
[Shriram] : How are you?
Im fine.
```

Client 2 –

```
F:\Java Project_1\src>java Client
Enter the username:
Shriram
Enter the path where you want to download files :
F:\file1.txt
Joined to chat room : WT
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
1
Type exit for go to menu...
Hello
[Shreayas] : Hi
How are you?
[Shreayas] : Im fine.
```

2. Two users on private chat

Server –

```
Shreayas is on private chat with :Shriram
Shreayas is on private chat with :Shriram
Shreayas is on private chat with :Shriram
Shriram is on private chat with :Shreayas
Shriram is on private chat with :Shreayas
Shriram is on private chat with :Shreayas
Shriram is on private chat with :Shreayas
Shreayas is on private chat with :Shriram
Shriram is on private chat with :Shreayas
```

Client 1 –

```
exit
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
2
Enter username :
Shriram
Type exit for go to menu...
Hello. How is the project?
[ Private- Shriram] : It is going well.
[ Private- Shriram] : We almost finished it. Taking screenshots of output.
Okay. So this is part of documentation?
[ Private- Shriram] : Yes.
```

Client 2 –

```
[ Private- Shreayas] : Hello. How is the project?
exit
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
2
Enter username :
Shreayas
Type exit for go to menu...
It is going well.
We almost finished it. Taking screenshots of output.
[ Private- Shreayas] : Okay. So this is part of documentation?
Yes.
```

3. File Sharing –

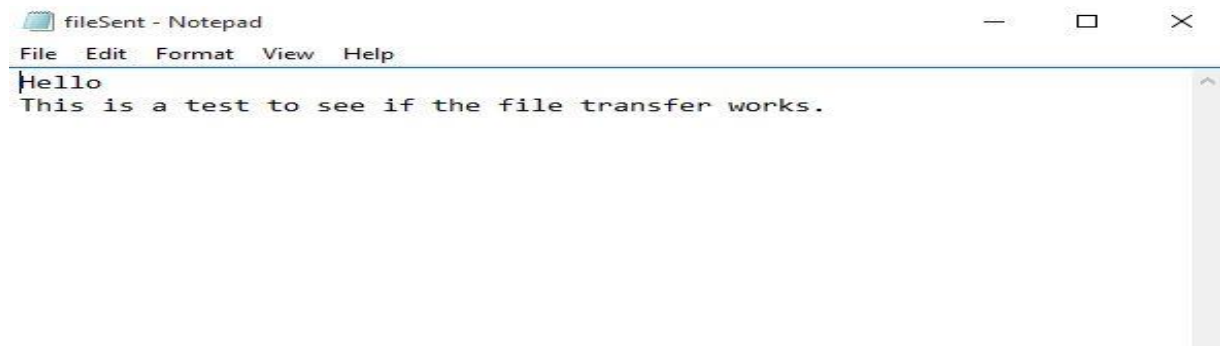
Sender –

```
exit
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
3
Enter the userid of the person to send file :
Shreayas
Enter the path of the file to send :
F:\fileSent.txt
File sent
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
```

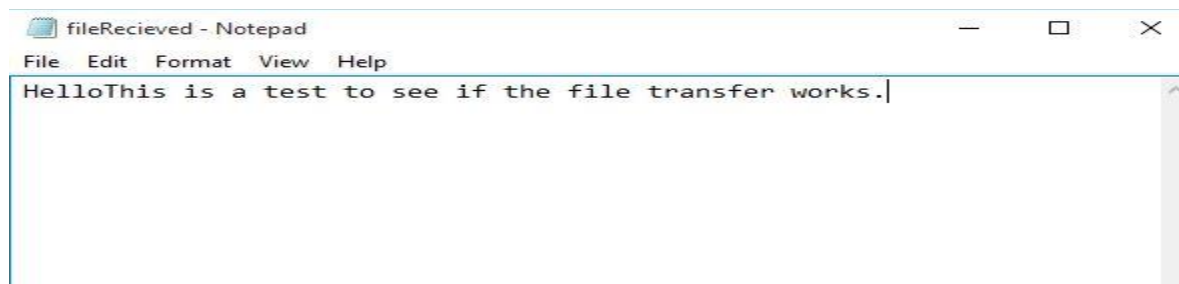
Receiver –

```
exit
Press 1 to type broadcast message...
Press 2 to type personal message...
Press 3 to send files...
Press 0 to exit...
File recieved from - Shriram. Check the file whose path you mentioned at the start
```

File sent –



File received –



HOW TO RUN THIS APPLICATION –

THIS APPLICATION CONTAINS TWO FILES – SERVER AND CLIENT. BOTH OF THEM MUST BE RUN ON THE **COMMAND LINE TERMINAL**. THE SERVER PROGRAM MUST BE STARTED FIRST AND A NAME FOR THE CHAT ROOM MUST BE GIVEN.

THEN MULTIPLE USERS CAN ACCESS MULTIPLE TERMINALS ON THE SAME LOCALHOST OR LAN AND START THE CLIENT PROGRAM. THEY MUST ENTER THEIR USERIDS AND THE LOCATION TO DOWNLOAD ANY FILES RECEIVED.

THEN THE USER CAN EITHER CHOOSE TO BROADCAST HIS MESSAGE OR SEND PRIVATE MESSAGES. **REMEMEBER TO SPECIFY WHICH MODE OF MESSAGING YOU WANT TO USE BEFORE SENDING THE MESSAGE.** A USER CAN RECEIVE A PRIVATE MESSAGE FROM ANOTHER USER WHILE BEING ON GROUP CHAT, BUT IN ORDER TO RESPOND PRIVATELY, THE USER MUST EXIT FROM THE GROUP BROADCAST FIRST, SELECT THE PRIVATE MODE, ENTER THE USERID OF THE OTHER USER AND THEN SEND THE MESSAGE.

FOR FILE TRANSFERS, MODE 3 CAN BE USED. **YOU MUST ENTER THE CORRECT ABSOLUTE FILE PATH OF THE FILE TO SEND.** THE RECEIVER WILL THEN RECEIVE A MESSAGE ABOUT THE FILE TRANSFER AND CAN CHECK THE PATH WHICH HE SPECIFIED IN THE BEGINNING.

YOU CAN CHOOSE 0 TO EXIT ANYTIME.

CONCLUSION –

THIS PROJECT WAS IMPLEMENTED USING THE CONCEPTS OF OBJECT-ORIENTED PROGRAMMING USING JAVA ALONG WITH OTHER BASIC NETWORKING IDEAS.

THE WORKING OF THE PROGRAM IS VERIFIED AND IT PASSES ALL THE BASIC TEST CASES WHICH WE THOUGHT OF.