TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A PROJECT REPORT
ON
"CHAT SYSTEM OVER LOCAL NETWORK"

SUBMITTED BY:-

1. **RAHUL SHAH (075BCT063)**
2. **ROSHAN SUBEDI (075BCT068)**
3. **RUPAK RAJ PANTHA (075BCT070)**

Submission Date: March 4, 2020

SUBMITTED TO

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

Acknowledgement:

In the accomplishment of this project successfully, many people have best owned upon their blessings and the heart pledged support. I am utilizing this time to thank all the people who have been concerned with this project. I would like to appreciate the effort of our respected professor Mr. Daya Sagar Baral who encouraged us to proceed with this idea and helped us in completing this project.

Eventually, I would like to thank all my classmates who has helped us.

ABSTRACT

Chat refers to the process of communicating, interacting and exchanging messages over the Internet. It involves two or more individuals that communicate through a chat-enabled service or software. Chat may be delivered through text via the Internet.

A chat application has basic two components, server and client. A server is a computer program or a device that acts as a medium for client to connect with each other. Clients are the people trying to chat with each other through the means of a program.

The chat service we are going to make will be more like a chat room, rather than a peer to peer chat. So this means that multiple users can connect to the chat server and send their messages. Every message is broadcasted to every connected client in the server system.

Table of Content

1.INRODUCTION

Chat system is very common communication tools that have been used in human in this modern cutting edge technology world. This chat system has become one of the important intermediate tools for everyone to share knowledge and materials via network. The technology has been available for years but the acceptance is quiet recent. The chat system is a distributed programming which consist two distributed components, chat server and chat client. Chat client supports for all communication that displays received chat messages.

Several network systems are built to communicate with one another and are made available through service-oriented architectures. In this project, we use the client server architecture to develop a secured Client-Server chat application. A chat application is created based on Transmission Control Protocol (TCP) where TCP is connection oriented protocol.

Our chat server will be developed with optimized technique to make fast and of less size. Even though it is a small application, it is eligible to meet client requirements. Our chat program allows for two-way conversations between users in real time. Typically, the users will connect to a chat server using a chat client and meet in a chat room. Once the users are in the same chat room, they can converse with one another by typing messages into a window where all of the other users in the chat room can see the message. The user can also see all of the messages entered by the other users. Conversations are then carried on by reading the messages entered by the other users in the chat room and responding to them.

2.OBJECTIVE

The objectives of the chat system includes are as follows:

- To develop an instant messaging solution to enable users to communicate with each other seamlessly
- To develop a user friendly application so that any person can use the program without any complications

3.APPLICATION

This project is applicable in all the fields where we need to communicate which means it has wide ranges of opportunities in real world application.

- This will be helpful for office staffs, students and all those who need to communicate over a network. Companies would like to have a communication software wherein they can communicate instantly within their organization.
- Schools and colleges would like to publish notice instantly to students. This can also be useful for the students to discuss between themselves.

4.LITERATURE REVIEW

We have gone through many projects on chat systems which are readily available in the internet. In their project we have found that their program mainly consists of two programs: one for the server and another for client. Moreover, they run the program in the same computer i.e. they used the same computer for both server and client. Moreover there were no graphics used in their project which we found a bit less user friendly and hard to use.

Our project will also be similar to theirs but in our project, we will try to improve the things that we felt were lacking in that project. First of all, absence of graphical user interface made us uncomfortable to use which we will be including with the help of wxWidgets. There are different features in the wxWidgets header file such as creating windows, inserting pictures and audios, creating buttons for specific tasks, etc. which we will be implementing in our project.

Also, in their project, there weren't provision for separate profile of each user since they used single computer for both server and client. In our case, there will be username for each username and when the user runs the program, they get connected to the server and there displays the window asking for them to log in. We will simply click log in option and type username to enter chat screen. In the chat screen the user will be able to see the members online and be able to send message in the group chat.

5.EXISTING SYSTEM

Chat application is one of the most used programs in today's world. Messenger, WhatsApp, Viber, etc. are some of the major chat app used. We have gone through some projects on chat systems which are available in the internet. We are trying to make a basic version of the major chat applications where a person can chat with a group of person in the same LAN.

6.METHODOLOGY

This project is basically an example of chat system (i.e. sending and receiving messages over a certain network). It is made up off two network: the client which runs on the user PC and the server application which runs on any PC on the network. To get started chatting, client must get connected to the server where they can communicate with each other. Our project contain server code and client code for connection and (if possible) database may be used for storing messages being sent or received and for future reference. At the same time, there can be more than one client which can be handled by the TCP connection. In order to handle with networking concepts, we will be using numbers of C++ built in features to have optimized codes like threading, networking console, Input and Output console and more. Among all, it may be the application which can be run, using the concept of WLAN (commonly WIFI) and all clients need to be connected to same server or use their dedicated IP address to receive and send messenger through broadcasting central server.

Server will act as a broadcasting center which will handle the sending and receiving of messages. It's the part which have to deal with TCP/IP and port number. Its predefined logic handles all the messages being received and sent.

On the other hand, using the client panel, user can login using their username. And another panel where message exchange process takes place.

7. IMPLEMENTATION:

The working of the network chat system is demonstrated in the flow chart below:

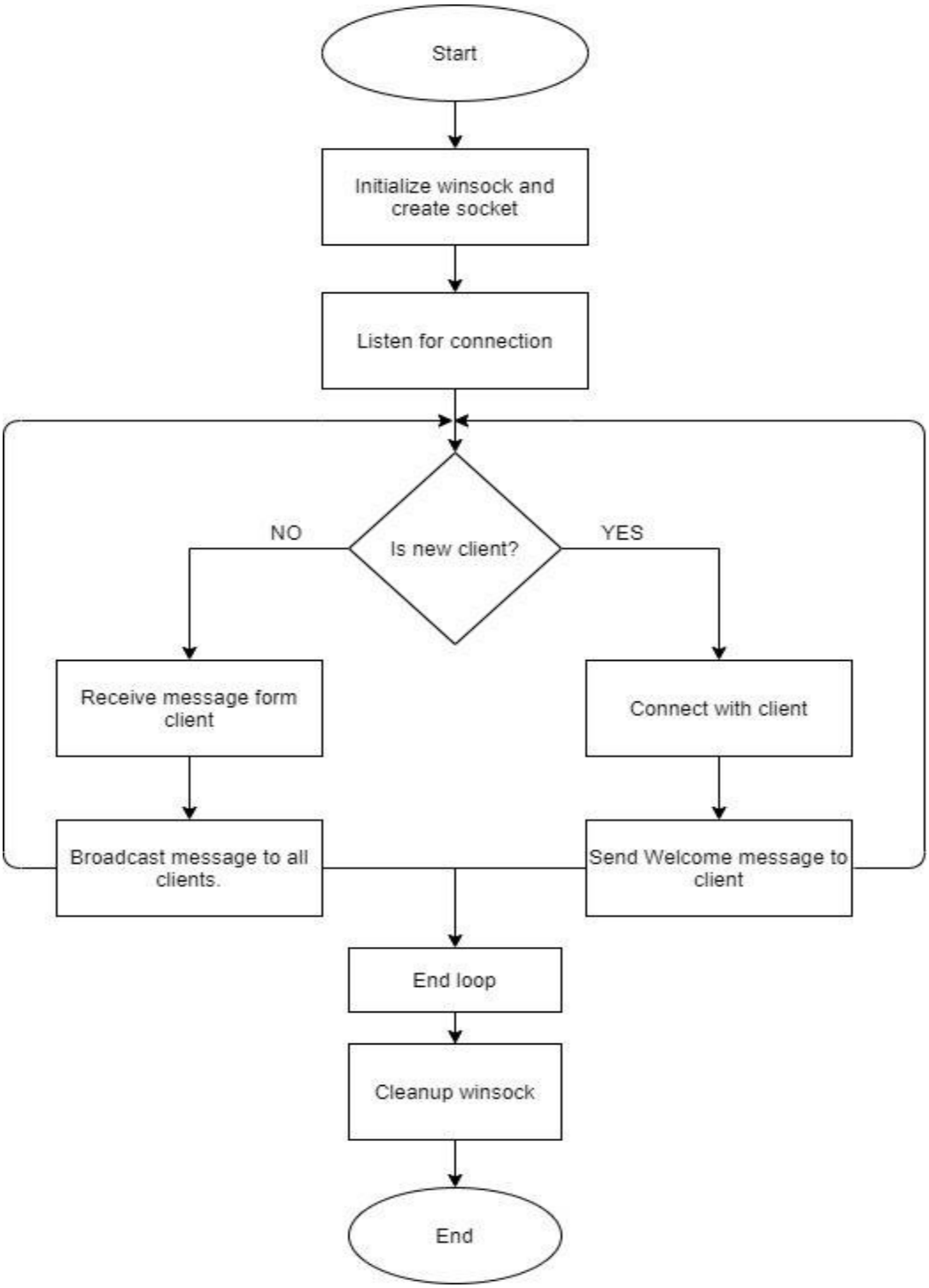

Fig: Server Flow Diagram

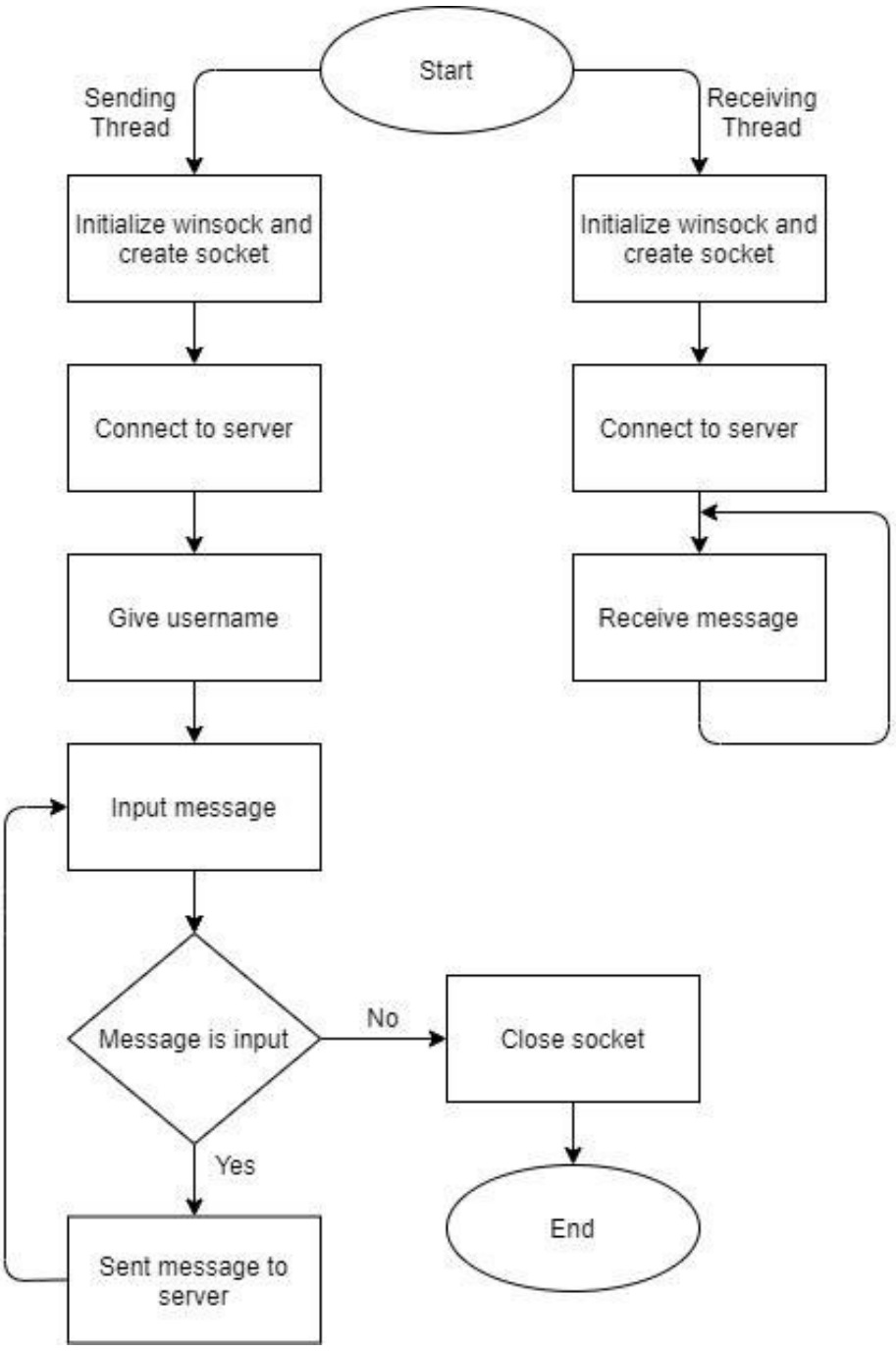
Start
Sending Thread
Receiving Thread
Initialize winsock and create socket
Initialize winsock and create socket
Connect to server
Connect to server
Give username
Receive message
Input message
Message is input
No
Close socket
Yes
End
Sent message to server

Fig:Client Flow Diagram

8. RESULTS

The chat system we were trying to make was a group chatting service with a Graphical User Interface from wxwidgets. But, after all the research, coding, testing and error checking we weren't able to combine the backend code for the chat with the wxWidgets framework. So, we are submitting a console application that allows a number of users to chat with each other. Also we are submitting the framework created.

9. PROBLEM FACED

While doing this project we ran into many problem. We were able to solve some of the problems but not all of them. We were doing a program using the wxWidgets for the GUI and network programming using Winsock. These were fairly new topics for us but we're able to make a good GUI and a client server network over a LAN and working perfectly individually. The main problem was faced while integrating the "Winsock" into the

wxWidgets framework we created. Here we tried many different ways to integrate but we weren't able to do it. Then we tried using the built in network in wxWidgets i.e. wxSocket. Here we face the problem of not being able to send messages between many clients over a same LAN. We're also not able to completely understand some terms in the language so we moved back to using "Winsock".

10.CONCLUSION AND RECOMMENDATION

In a nut shell, following one month of research, coding, testing, debugging and hardship after we failed to achieve out expected goal in building our purposed Chat System but regardless of our failure we gained lots of knowledge about GUI framework using “wxWidgets” and network programming using “Winsock”. This project taught us how to create a basic chat application by creating a local network and using TCP.

References

- www.google.com
- www. wxwidgets.org
- docs.microsoft.com ‘s Winsock documentation