TCP CLIENT SERVER EXCHANGE OF MESSAGES AND FILES

TEAM MEMBERS

Shreyaa Sridhar (Student ID #16251190)

Naga Venkata Satya Pranoop Mutha (Student ID #16245066)

Vinay Maturi (Student ID #16242466)

INTRODUCTION

A computer running a program that makes a request for services is called **Client**. A computer running a program that offers requested services from one or more clients is called **Server.**

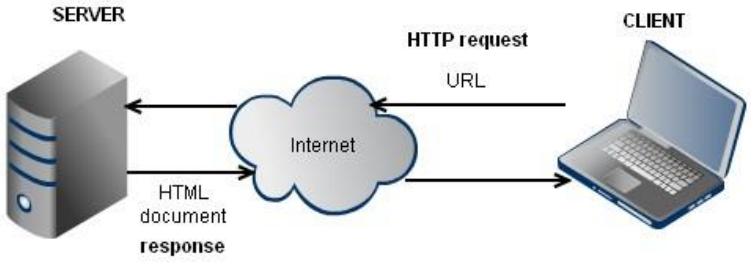


Figure: Client-server Block diagram

The media between these two machines may be wired or wireless network. Generally, programs running on client machines make requests to a program (often called as server program) running on a server machine. They involve networking services provided by the transport layer, which is part of the Internet software stack, often called Transport Control Protocol/Internet Protocol) stack. A socket is an endpoint of a two-way communication link between two programs running on the network. Sockets provide an interface for programming networks at the transport layer. Socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent. Network communication using Sockets is called Socket Programming. Here we have created a client server model which exchanges of messages or files transfer.

MOTIVATION

The client-server model is one of the most used communication paradigms in networked systems. To make a communication in a network we need at least two network devices. One network device for managing the connections known as Server and remaining devices connects to the server known as clients. All these devices use their ports and IP addresses to make the connections and utilizes the transport layer protocols for connection establishment(TCP/UDP). While sending/receiving the data uses other layer protocols for data security, data integrity, error checking, retransmission etc. Sockets

are bidirectional, meaning that either side of the connection is capable of both sending and receiving data. This model can be implemented on a single pair of machine or on multiple pairs machines. Present almost all the corporate companies are using this client and server model architecture for developing the successful applications.

TOOLS USED

Technology : .Net Framework

Development Environment : Visual Studio 2017

Operating System Used : Windows OS

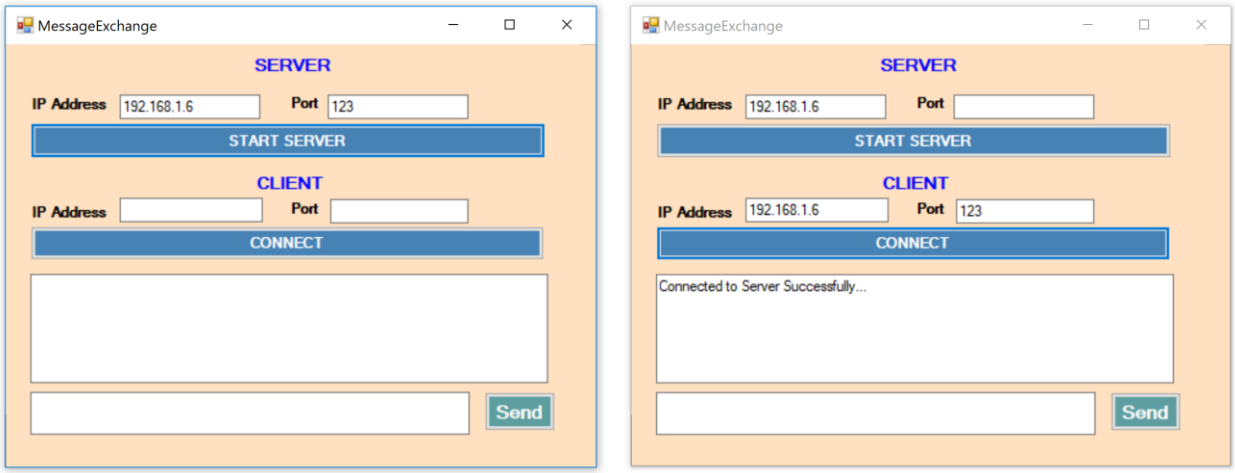
Languages Used : C# .net

Compiler : csharp compiler (csc)

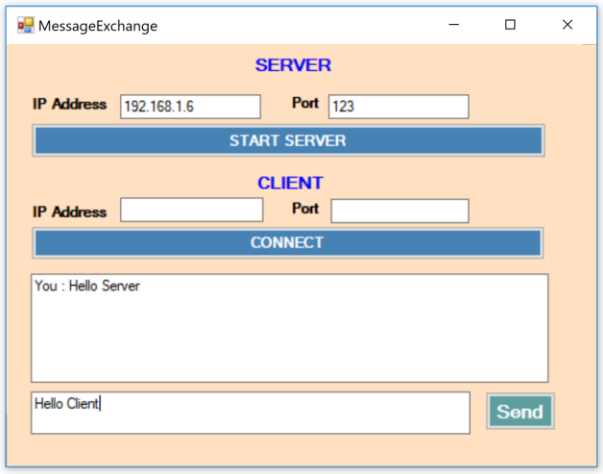
PROCEDURE

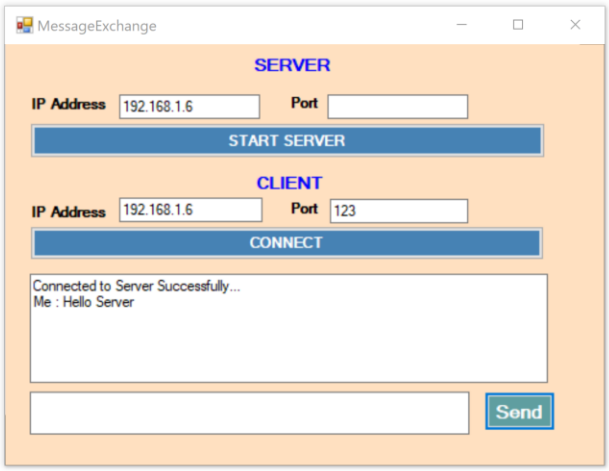
Task 1: Client Server Communication or Exchange of messages

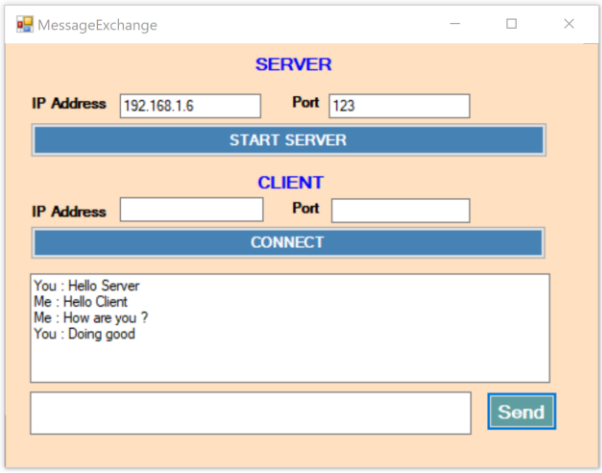
First we open application twice, for one as server (left image) and other as client (right image). Click on **START SERVER** in Server window and then **CONNECT** on the Client window to connect the server and client successfully.

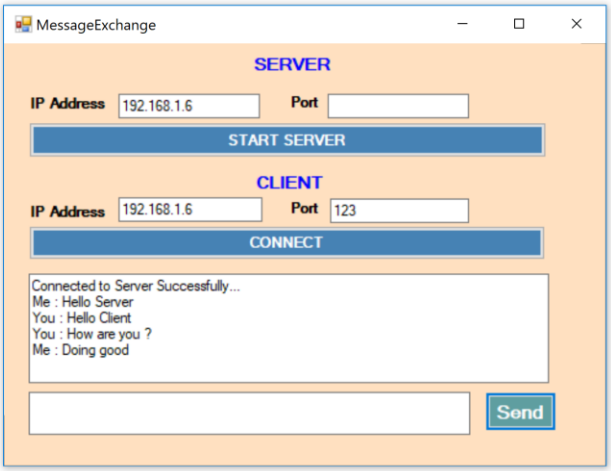


Now server and client can exchange messages between them. Here **'Me'** is the person sending the message and **'You'** is the message received from other person. Clicking on **SEND ,** will send your message to the other person.





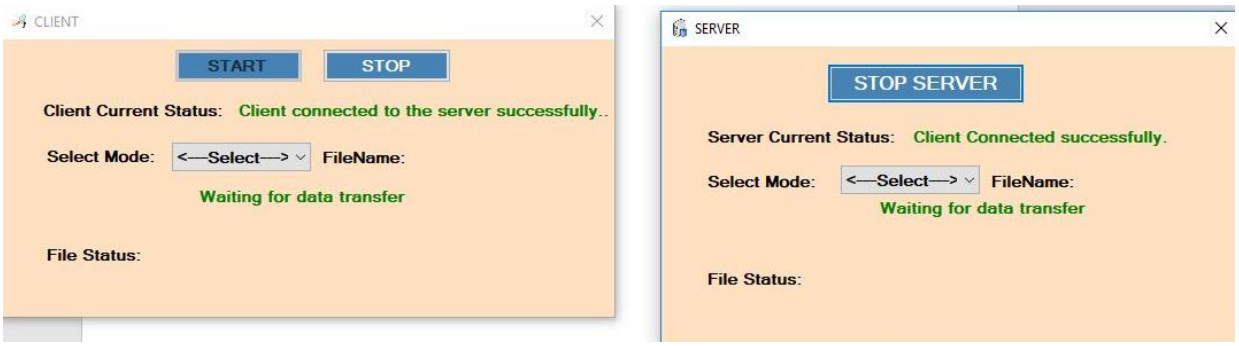




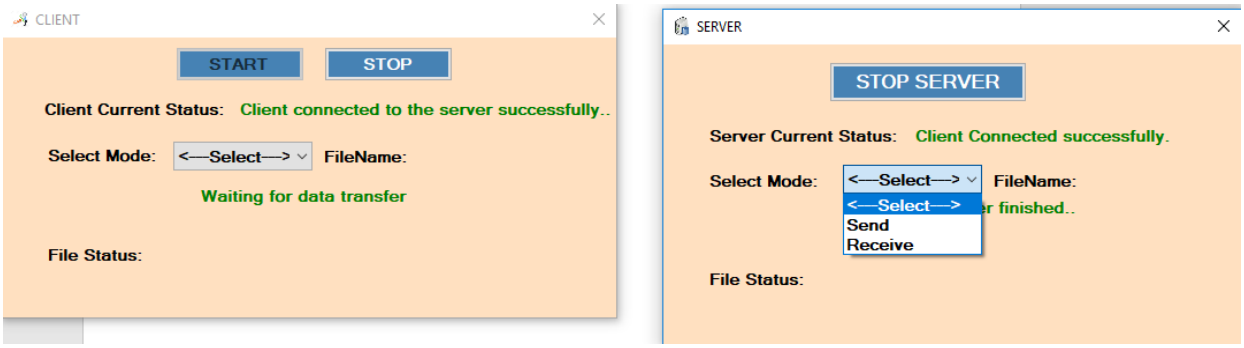
Task 2: File transfer between Client and Server

First, we initialize Server and Client applications.

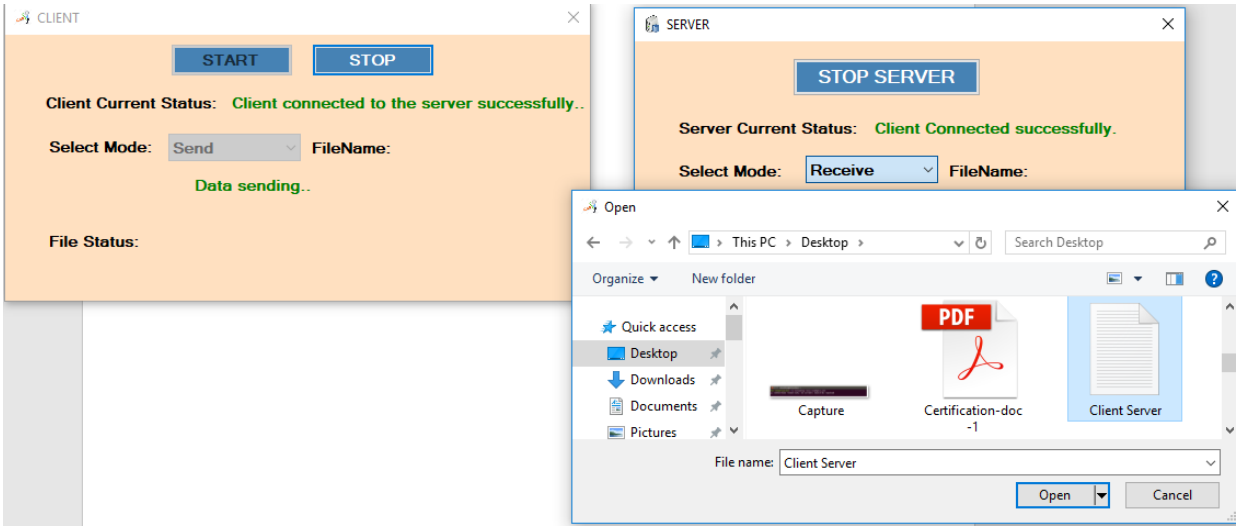
Click on **START SERVER** to start the server and then Click on **START** in Client window to start the client and connect to the server.

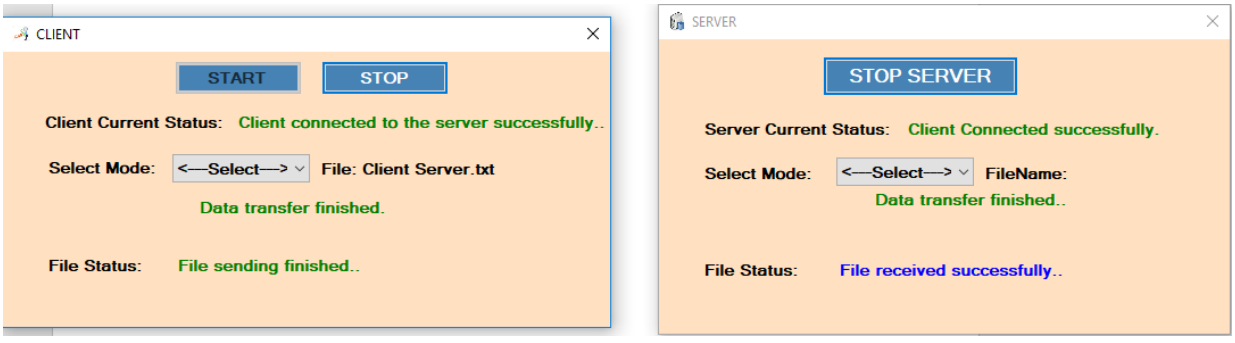


Here you can select mode **SEND** or **RECEIVE** for both Client and Server.



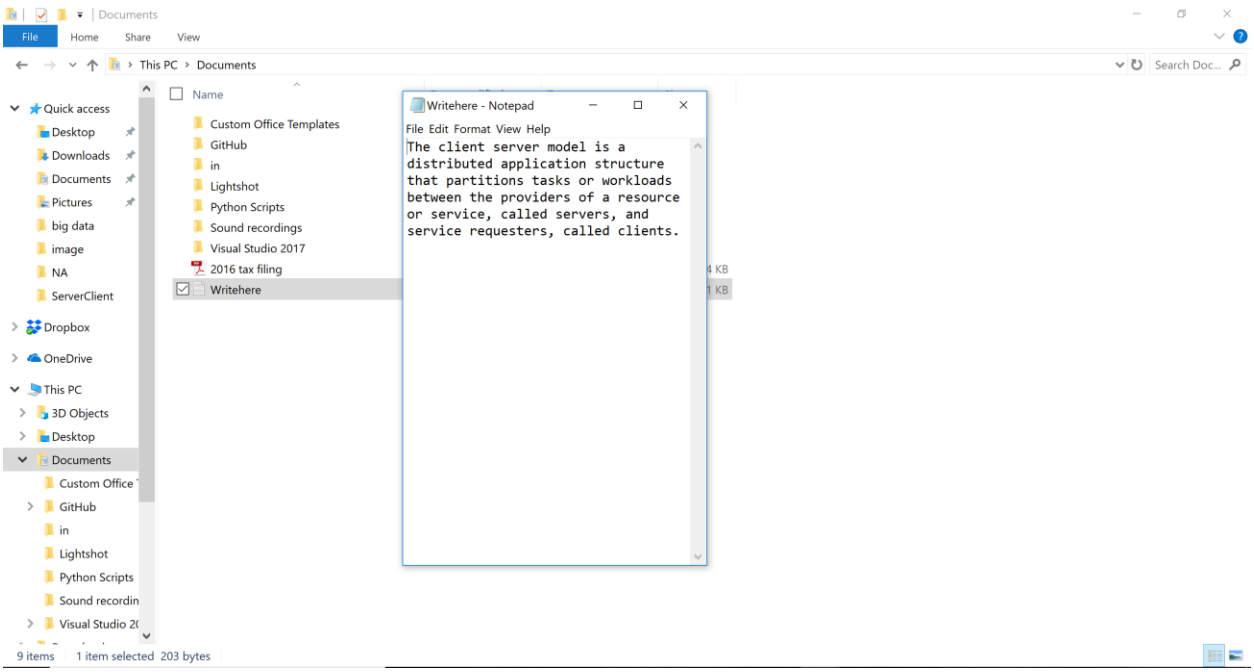
When you mention **SEND** mode , it allows you to browse for the file to be transferred.





File was successfully transferred to the location mentioned.

```
(@"C:\Users\Shreya\Documents\Writehere.txt");
```



TEAM CONTRIBUTION

Shreyaa Sridhar

Client Server Communication or Exchange of messages

Documentation

Naga Venkata Satya Pranoop Mutha

File transfer between Client and Server

Documentation

Vinay Maturi

File transfer between Client and Server

Documentation

CONCLUSION

This project mainly focused on client–server model to exchange messages and send/receive files, which accomplished successfully.