Chat Room

ȘTREANGĂ NADINA

UPT AC CTI ENG, 2ND YEAR, GR.2.2

Introduction

- Chat Room is a client-server chat application that allows users to connect to a chat room and send text messages to each other.
- It is an application that mirrors the chat applications around the 2000's.

Design and Implementation

- The client-server side was implemented in the following manner:
- - Client side: C# with Windows Forms for GUI
- - Server side: C# Console Application

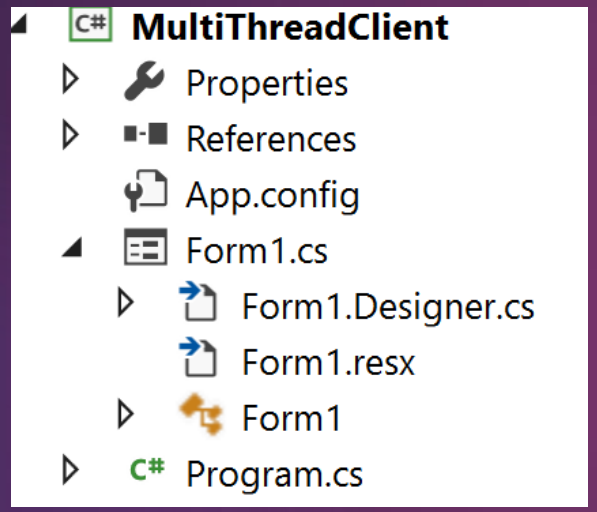
- The **client** side has 5 UI components:

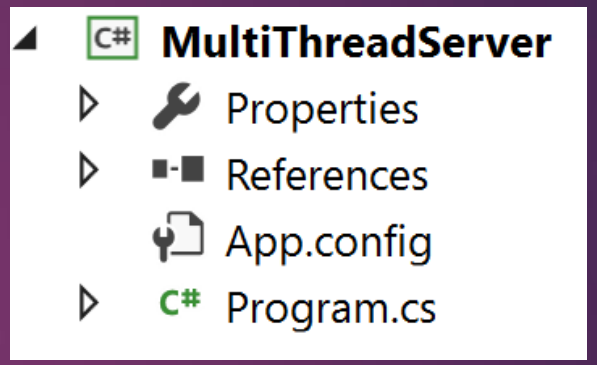
- Dynamic label showing if the client is connected to the server
- Chat box in which messages are displayed
- Message box where the user inputs his/her messages
- The buttons where you can send and receive the data

- The **server** side:

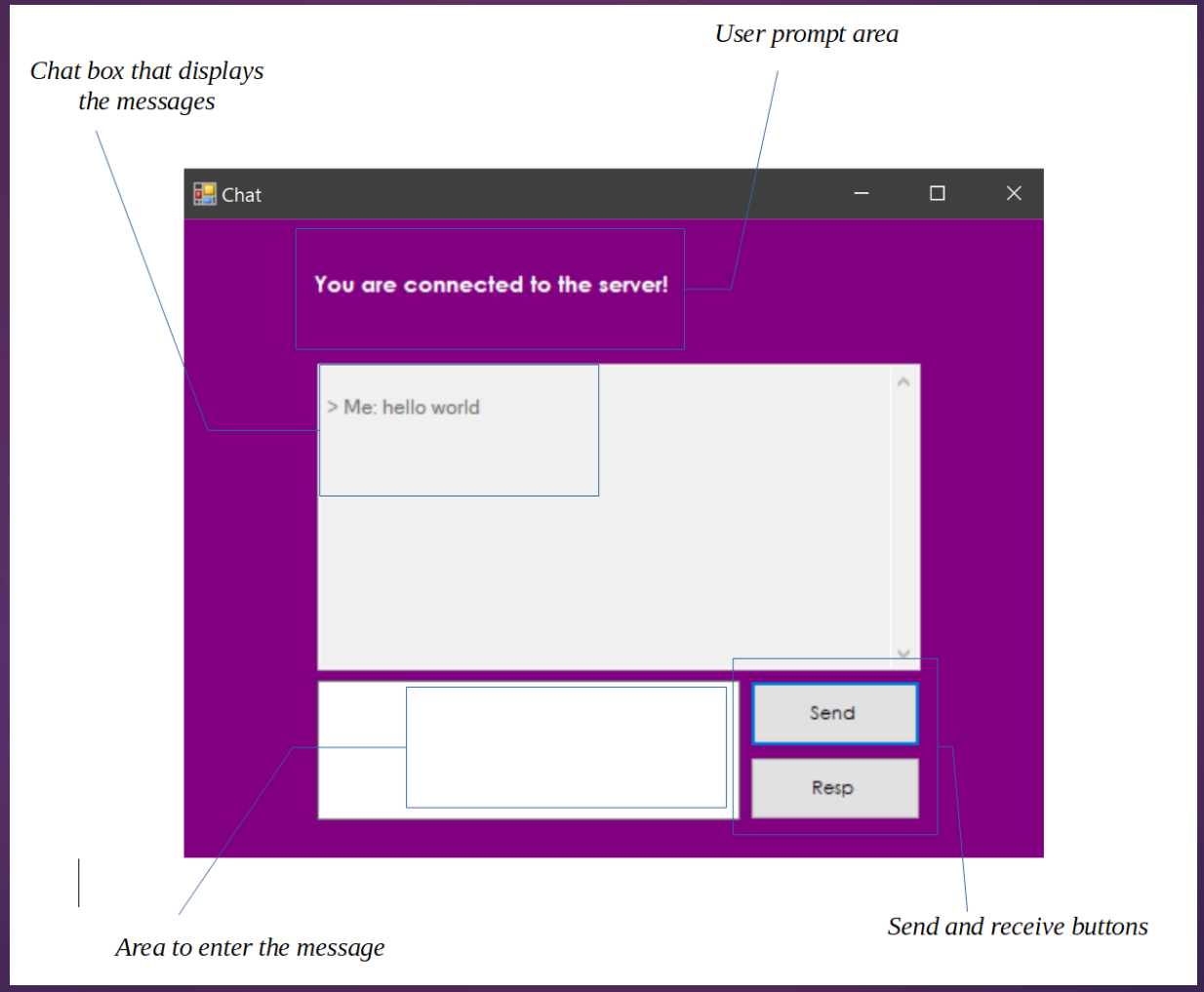
- prints in the terminal all established connections and all messages sent to the server
- establishes a connection with the client, creating a separate thread on which communication happens
- each thread opens a stream of communication with the client and waits to receive messages

The project structure:





Usage

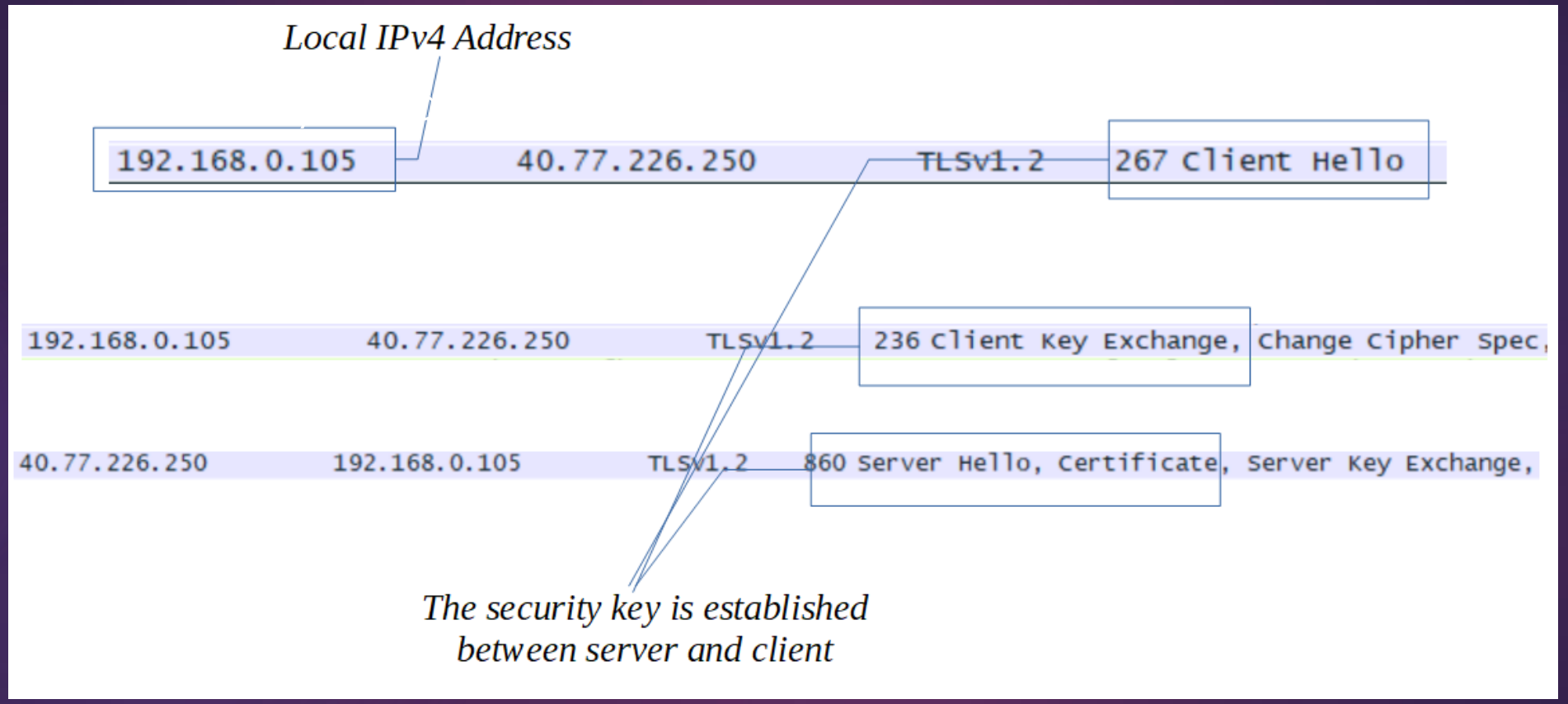

User prompt area
Chat box that displays the messages
Chat
You are connected to the server!
> Me: hello world
Send
Resp
Area to enter the message
Send and receive buttons

Server side console application logs

State of the art

- The Chat Room had two sources of inspiration: Yahoo Messenger.
- It offers only the basic text messaging, while others also offer file upload, emojis etc.
- The application does not allow P2P communication, compared to the other messaging apps.
- Compared to other apps, the Chat Room is much easier to use.

Analysis


Local IPv4 Address
192.168.0.105 40.77.226.250 TLSv1.2 267 Client Hello
192.168.0.105 40.77.226.250 TLSv1.2 236 Client Key Exchange, Change Cipher Spec,
40.77.226.250 192.168.0.105 TLSv1.2 860 Server Hello, Certificate, Server Key Exchange,
The security key is established between server and client

Packets exchanged while the connection the is active

192.168.0.105	40.77.226.250	TLSv1.2	331	Application Data
40.77.226.250	192.168.0.105	TCP	56	443 → 58145 [ACK] Seq=3754 Ack=4933 Win=131840 Len=0
192.168.0.105	40.77.226.250	TCP	1474	58145 → 443 [ACK] Seq=4933 Ack=3754 Win=66560 Len=1420
192.168.0.105	40.77.226.250	TLSv1.2	75	Application Data
40.77.226.250	192.168.0.105	TLSv1.2	155	Application Data
192.168.0.105	40.77.226.250	TCP	1474	58143 → 443 [ACK] Seq=673 Ack=3855 Win=66304 Len=1420
40.77.226.250	192.168.0.105	TCP	56	443 → 58145 [ACK] Seq=3754 Ack=6374 Win=131840 Len=0

- The TCP and TLS protocols are used in order to establish the connection between the client and the server.
- Both server and client, agree through a “handshake” on the security key, as dictated by the TLS protocol
- Through TCP, the data and the TCP header are sent and received to and by the client from and to the server