

Chatroom Project

Contents

Operating system, code language and version:	1
Theoretical background:	1
<i>The transport layer and the application layer (layers 4-5)</i>	1
<i>Client-server architecture</i>	1
<i>Socket Programming in python</i>	2
Setting up a listening socket	2
Running the programs	3
<i>Example: Running the socket example from tergul</i>	3
Rerunning the program:.....	5
Explanation about the code files, functions, and classes:	6
<i>Server</i>	6
<i>Client</i>	7
Explanation about the socket handshake:	7
Running examples:	10

Operating system, code language and version:

Operating system: Windows 11

Code language and version: Python, PyCharm community Edition 2021.2.2

Theoretical background:

The transport layer and the application layer (layers 4-5)

The transport layer and the application layer are two of the five layers in the OSI model, which is a framework used to understand and describe the different functions and protocols involved in the process of transmitting data between two devices over a network.

The transport layer is the fourth layer in the OSI model, and it is responsible for establishing end-to-end communication between devices and ensuring that data is delivered reliably and in order from the sender to the receiver. It does this by providing error-checking mechanisms, flow control, and segmentation/reassembly of data into smaller units for transmission. The transport layer also provides multiplexing capabilities, which allow multiple applications on a device to share the same network connection and use the same port numbers.

There are two main transport layer protocols: TCP and UDP.

TCP (Transmission Control Protocol) is a connection-oriented protocol that provides reliable, ordered delivery of data between devices. It does this by establishing a connection between the sender and

the receiver, sending data in small chunks called segments, and using a variety of error-checking mechanisms to ensure that the data is delivered correctly. TCP is a popular transport layer protocol because it provides reliable delivery of data, but it can be slower than other transport layer protocols due to the overhead required for error-checking and flow control.

UDP (User Datagram Protocol) is a connectionless protocol that provides a way for devices to send data to each other without establishing a connection. It is a simple protocol that allows devices to send small packets of data called datagrams to each other without the overhead of error-checking or flow control. UDP is a good choice for applications that require fast transmission of data and do not need the reliability provided by TCP.

The application layer is the highest layer in the OSI model, and it is responsible for providing the interface between the network and the applications that run on the devices connected to the network. It defines the protocols and services that are used by applications to communicate over the network and access network resources. The application layer also provides the means for applications to communicate with each other and with the user.

There are many different application layer protocols, including HTTP (Hypertext Transfer Protocol), which is used for web browsing; FTP (File Transfer Protocol), which is used for transferring files; and SMTP (Simple Mail Transfer Protocol), which is used for sending and receiving email.

Overall, the transport layer and the application layer play important roles in the transmission of data over a network, with the transport layer handling the delivery of data between devices and the application layer providing the interface between the network and the applications running on the devices.

Client-server architecture

The client-server architecture has two entities: client who consume services, and server who provides services. In our case:

- The client is a user whose interested at group chatting.
- The server is the computer whose responsible of managing the group chats. Its services include: access to group chats, print new messages, etc.

Socket Programming in python

Setting up a listening socket

A listening socket listens for connections from clients.

.bind() is used to associate the socket with a specific network interface and port number.

For example:

```
HOST = '127.0.0.1' # Standard loopback IP address (localhost)
PORT = 5000 # Port to listen on (non-privileged ports are > 1023)
FORMAT = 'utf-8' # Define the encoding format of messages from client-
server
ADDR = (HOST, PORT) # Creating a tuple of IP+PORT
```

```
# Function that starts the server
```

```
def start_server():
    server_socket.bind(ADDR) # binding socket with specified IP+PORT tuple
```

.listen() enables a server to accept connections. It makes the server a “listening” socket.

For example:

```
# Function that starts the server
def start_server():
    server_socket.bind(ADDR) # binding socket with specified IP+PORT tuple

    print(f"[LISTENING] server is listening on {HOST}")
    server_socket.listen() # Server is open for connections
```

When a client connects, the server calls .accept() to accept, or complete, the connection.

For example:

```
# Function that starts the server
def start_server():
    server_socket.bind(ADDR) # binding socket with specified IP+PORT tuple

    print(f"[LISTENING] server is listening on {HOST}")
    server_socket.listen() # Server is open for connections

    while True: # the server should always listen
        print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}\n") # printing the amount of threads working

        connection, address = server_socket.accept() # Waiting for client to connect to server (blocking call)
```

Running the programs

First we run the server. Then we run the clients.

A possible way to run the clients in python is to create a different python file for each client we want to run. So if we want N clients to connect to the servers, we will create N python client files.

Example: Running the socket example from tergul

Note that in the tergul example, the code for each client is different. In the assignment, the code for each client is exactly the same, we only need to duplicate the file in order to create more clients.

First we will run the server, then Client1 and then Client 2. Afterwards we will start to work with Client1 and then Client2. A possible output is:


```

File Edit View Navigate Code Refactor Run Tools VCS Window Help SocketTergulExample - C:\Users\RonIV\Downloads\Server.py
C: \Users\RonIV\Downloads\Server.py
Project: SocketTergulExample C:\Users\RonIV\PycharmProjects\48
main.py Client1.py Server.py Client2.py
view history run start_server() while True : if threading.activeCount() == 1
PRINT("Received name from client: %s + name + %s")
start_server() while True : if threading.activeCount() == 1
Run: Server Client1 Client2
C:\Users\RonIV\PycharmProjects\pythonProject14\SocketTergulExample\venv\Scripts\python.exe C:/Users/RonIV/Downloads/Client1.py
[CLIENT] Started running
Please enter # of messages you would like to send to server: 2
2
count: 0, total: 2
Please enter message for server: cat
[SENT DATA] cat
[RECEIVED DATA] cat

count: 1, total: 2
Please enter message for server: dog
[SENT DATA] dog
[RECEIVED DATA] dog

[CLOSING CONNECTION] client closed socket!
Goodbye client:

Process finished with exit code 0

```



```

File Edit View Navigate Code Refactor Run Tools VCS Window Help SocketTergulExample - C:\Users\RonIV\Downloads\Server.py
C: \Users\RonIV\Downloads\Server.py
Project: SocketTergulExample C:\Users\RonIV\PycharmProjects\48
main.py Client1.py Server.py Client2.py
view history run start_server() while True : if threading.activeCount() == 1
PRINT("Received name from client: %s + name + %s")
start_server() while True : if threading.activeCount() == 1
Run: Server Client1 Client2
C:\Users\RonIV\PycharmProjects\pythonProject14\SocketTergulExample\venv\Scripts\python.exe C:/Users/RonIV/Downloads/Client2.py
[CLIENT] Started running
[RECEIVED DATA] Welcome! This is your server:
Please enter your name:
Jason
[SENT DATA] Jason
[RECEIVED DATA] Hello Jason!
What is your age?
10
[SENT DATA] 10
[RECEIVED DATA] What is your profession?
doctor
[SENT DATA] doctor
[RECEIVED DATA] Nice to meet you:
Goodbye for now...

[CLOSING CONNECTION] client closed socket!
Goodbye client:

Process finished with exit code 0

```

Rerunning the program:

Note that, the example from class terminates client1 and Client2 but doesn't kill the server process. So even after we close PyCharm, the server still listens on port 5000!

That way, if we try to open again the code and rerun the server, we will get an error:

```
OSErr: [WinError 10048] Only one usage of each socket address
(protocol/network address/port) is normally permitted
```

To fix this error, and rerun the server, we need to first kill the old process. To do this, we can go to windows command prompt, and type:

```
netstat -ano|findstr 5000
```

The output would be something like:

Protocol	Local Address	Remote Address	Status	PID
TCP	127.0.0.1:5000	0.0.0.0:0	LISTENING	19556
TCP	127.0.0.1:5000	127.0.0.1:50727	ESTABLISHED	19556
TCP	127.0.0.1:50727	127.0.0.1:5000	ESTABLISHED	17100
UDP	0.0.0.0:5000	*:*		4192

To kill the process, we can type:

```
taskkill /F /PID 19556
```

the output would be:

```
SUCCESS: The process with PID 19556 has been terminated.
```

And then we would be able to rerun the process.

Screenshot from prompt:

```
C:\Users\Roniv>netstat -ano|findstr 5000
C:\Users\Roniv>
TCP    127.0.0.1:5000          0.0.0.0:0              LISTENING      19556
TCP    127.0.0.1:5000          127.0.0.1:50727      ESTABLISHED   19556
TCP    127.0.0.1:50727        127.0.0.1:5000      ESTABLISHED   17100
UDP   0.0.0.0:5000           *:*                  4192

C:\Users\Roniv>taskkill 19556
'tskill' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Roniv>taskkill /F /PID 5000
'tskill' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Roniv>taskkill /F /PID 19556
SUCCESS: The process with PID 19556 has been terminated.

C:\Users\Roniv>
```

Explanation about the code files, functions, and classes:

An explanation about any file you created – including any classes and functions.

We have created two files: ServerFinal.py, which has the code for the Server, and ClientFinal.py, which has the code for the client.

Server

The server code has three functions:

```
1) def broadcast(message, c)
```

This function is called whenever we want to transfer a message to all the clients in a certain group chat. This function gets the message and client that sends the message as parameters.

```
2) def handle_client(client):
```

This function handles clients' connections. It gets the client as a parameter. It listens if there are messages to be sent and if there are it calls the function broadcast.

```
3) def receive():
```

This function is responsible for getting establishing the connection with the clients, receiving the required information for each option (password, alias, group ID). It is also responsible for making sure that the information that was received is valid.

For more information about the implementation of the functions see the code files.

Client

The client code has two functions:

```
1) def client_receive():
```

This function is used to handle messages that are sent from the server to the client. If the messages that are sent are one of the server keywords, specific actions are taken, otherwise, the message is printed.

```
2) def client_send():
```

This function is used to handle messages that are sent from the client to the server.

Explanation about the socket handshake:

5. An explanation about the socket handshake?

which protocol you used?

what are the commands?

are they blocking commands?

The Handshake

When the client runs the following command:

```
client.connect(('127.0.0.2', 59000))
```

.connect() establishes a connection to the server and initiates the three-way handshake. The handshake step's purpose is to ensure that each side of the connection is reachable in the network. If the handshake was successful, it means that the client can reach the server and vice-versa.

The three-way handshake works like so:

The Three Steps of a Three-Way Handshake:

Step 1: A connection between server and client is established

First, a connection between server and client is established, so the target server must have open ports that can accept and initiate new connections. The client node sends a SYN (Synchronize Sequence Number) data packet over an IP network to a server on the same or an external network.

This SYN packet is a random sequence number that the client wants to use for the communication (for example, X). The objective of this packet is to ask/infer if the server is open for new connections.

Step 2: The server receives the SYN packet from the client node

When the server receives the SYN packet from the client node, it responds and returns a confirmation receipt – the ACK (Acknowledgement Sequence Number) packet or SYN/ACK packet. This packet includes two sequence numbers.

The first one is ACK one, which is set by the server to one more than the sequence number it received from the client (e.g. X+1).

The second one is the SYN sent by the server, which is another random sequence number (for example, Y).

This sequence indicates that the server correctly acknowledged the client's packet, and that is sending its own to be acknowledged as well.

Step 3: Client node receives the SYN/ACK from the server and responds with an ACK packet

The client node receives the SYN/ACK from the server and responds with an ACK packet. Once again, each side must acknowledge the sequence number received by incrementing it by one.

So now it's the turn of the client to acknowledge the server's packet by adding one to the sequence number (in this case, Y+1), and resend it to the server.

Upon completion of this process, the connection is created and the host and server can communicate.

All these steps are necessary to verify the serial numbers originated by both sides, guaranteeing the stability of the connection.

Since both hosts must acknowledge the connection parameters of the other side, a missing or out-of-order segment can be quickly detected before the actual data transfer process is initiated.

[Based on the article on <https://www.techopedia.com/definition/10339/three-way-handshake>]

We have created a socket object using the following command:

```
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Specifying the socket type as socket.SOCK_STREAM cause it to use the Transmission Control Protocol (TCP) by default.

The Transmission Control Protocol (TCP)

TCP covers parts of Layer 4, the transport layer, and parts of Layer 5, the application layer.

TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules that define the internet.

TCP performs the following actions:

- determines how to break application data into packets that networks can deliver;
- sends packets to, and accepts packets from, the network layer;
- manages flow control;
- handles retransmission of dropped or garbled packets, as it's meant to provide error-free data transmission; and
- acknowledges all packets that arrive.

Each packet in the transmission has the same source and destination IP address. However, packets may be sent along multiple routes. The packets are numbered. This is how the TCP works:

1. First, the TCP program layer in the client computer waits until all packets have arrived.
2. Then it acknowledges those it receives. It checks if there are missing packets based on the packets numbers and asks for the retransmission of the packets it did not receive.
3. Finally, the TCP layer assembles the packets into a file and delivers the file to the receiving application.

The commands used on the server/client connection:

Server side:

- `socket.bind()` - This method binds the socket to a specific address and port.
- `socket.listen()` - This method listens for incoming connections from clients.
- `socket.accept()` - This method accepts a connection request from a client and returns a new socket object that can be used to communicate with the client.
- `socket.recv()` - This method receives data from the socket.
- `socket.send()` - This method sends data over the socket.
- `socket.close()` - This method closes the socket.

Client side:

- `socket.connect()` - This method establishes a connection to a server.
- `socket.recv()` - This method receives data from the socket.
- `socket.send()` - This method sends data over the socket.
- `socket.close()` - This method closes the socket.

Blocking vs Non-blocking sockets

By default, TCP sockets are placed in a blocking mode. This means that the control is not returned to the program until some specific operation is complete.

When we run the line:

```
client.connect(('127.0.0.2', 59000))
```

the connection blocks the program until the operation is complete.

(Sometimes, we don't want to keep our program waiting forever, or we want to have a stop functionality that can cancel an active connection process in the middle of its operation. This can be achieved by placing the socket in the non-blocking mode).

Running examples:

Now I am going to show examples for each case. I will create overall 9 Clients, who all have the same code:

File Name	User Name	Case
Client33	Noya	
Client34	Eliana	
Client35	Sara	
Client36	Shlomo	
Client37	Ariel	(Right id, unrecognized password)
Client38	Rinat	(Unrecognized ID, right password)
Client39	Roni	(Unrecognized ID, unrecognized password)
Client40	Yuval	(Recognized ID, recognized but unmatching password)
Client41	- (chooses option 3)	-

We start by running the server:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** Computer Networks and Internet 2
- File:** ServerFinal.py
- Code Editor:** The code is a Python script for a server. It defines a class `Client` with methods `__init__`, `send`, and `recv`. It also defines a function `main` that handles multiple clients. The code uses `socket` and `select` modules.

```
    aliases.append(alias)
    clients.append(client)
else:
    alias = alias + ('+' + str(resultAlias) + '-')
    aliases.append(alias)
    clients.append(client)

password = 'initial'
client.send('password?'.encode('utf-8'))
password = client.recv(1024).decode('utf-8')

receive() > while True
```

- Run:** The run configuration is set to "ServerFinal" and the command is "C:\Users\RoniV\AppData\Local\Programs\Python\Python39\python.exe". The output shows "Server is listening".
- Status Bar:** Shows the date and time as 30/12/2022 11:15, and the Python version as Python 3.9.

Now we run a client. The client file we run is named Client33:

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and ServerFinal.py - Client33.py. The Project tool window on the left shows a folder named 'Computer Networks and Internet 2' containing files like Client33.py, Client35.py, Client36.py, Client37.py, Client40.py, Client41.py, Client42.py, and Client39.py. The main code editor window displays Python code for a client application. The terminal window at the bottom shows the command: C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n roniv/Computer Networks and Internet 2/PythonDraft30.12/Client33.py". The terminal output shows a menu: Choose an option: 1) Connect to a group chat 2) Create a new group chat 3) Disconnect. The status bar at the bottom right indicates the date as 30/12/2022, time as 11:15, and battery level as 19%.

Client33 choose option 2, to create a new group chat. He is asked to enter a name:

The screenshot shows the PyCharm IDE interface. The terminal window at the bottom has the command: C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n roniv/Computer Networks and Internet 2/PythonDraft30.12/Client33.py". The terminal output shows the same menu as before: Choose an option: 1) Connect to a group chat 2) Create a new group chat 3) Disconnect. However, the user has entered '2' and the terminal now prompts: ENTER YOUR NAME:. The status bar at the bottom right indicates the date as 30/12/2022, time as 11:15, and battery level as 19%.

Then a password:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client33.py
Computer Networks and Internet 2 PythonDraft30.12 Client33.py
Project Computer Networks and Internet 2 D:\Users\Shaiel\Bar-Ilan\Computer Networks and Internet 2\PythonDraft30.12\Client33.py
External Libraries Scratches and Consoles
client.send(message.encode('utf-8'))
else:
    break

receive_thread = threading.Thread(target=client_receive)
receive_thread.start()

send_thread = threading.Thread(target=client_send)
send_thread.start()

if option == "2":
    print("Choose an option:")
    print("1) Connect to a group chat")
    print("2) Create a new group chat")
    print("3) Disconnect")
    print("4) Exit")

    name = input("ENTER YOUR NAME: ")
    password = input("CHOOSE A PASSWORD FOR THE GROUP: ")

    print("Noya has connected to the chat room")
    print("YOU ARE NOW CONNECTED TO THE CHAT")
```

After typing the password, the following appears:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client33.py
Computer Networks and Internet 2 PythonDraft30.12 Client33.py
Project Computer Networks and Internet 2 D:\Users\Shaiel\Bar-Ilan\Computer Networks and Internet 2\PythonDraft30.12\Client33.py
External Libraries Scratches and Consoles
client.send(message.encode('utf-8'))
else:
    break

receive_thread = threading.Thread(target=client_receive)
receive_thread.start()

send_thread = threading.Thread(target=client_send)
send_thread.start()

if option == "2":
    print("Choose an option:")
    print("1) Connect to a group chat")
    print("2) Create a new group chat")
    print("3) Disconnect")
    print("4) Exit")

    name = input("ENTER YOUR NAME: ")
    password = input("CHOOSE A PASSWORD FOR THE GROUP: 123")

    print("THE NEWLY CREATED CHAT ID IS 2")
    print("Noya has connected to the chat room")
    print("YOU ARE NOW CONNECTED TO THE CHAT")
```

Now anyone who wishes to enter the chat can do so with the newly created group ID, 2 and chosen password, 123.

Now we will connect another client to this chat.

We run Client34, and chose option 1. After choosing option 1, Client34 is asked to write his name:

The screenshot shows the PyCharm IDE interface. The code editor displays Python code for a client application. The terminal window below shows the application's interaction with the user:

```
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n roniv/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
ENTER YOUR NAME: |
```

After writing his name, Client44 is asked to enter the chat id of the group he wishes to join:

The screenshot shows the PyCharm IDE interface. The code editor displays Python code for a client application. The terminal window below shows the application's interaction with the user:

```
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n roniv/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
ENTER YOUR NAME: Eliahu
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: |
```

Now Client 44 is asked to write the password:

The screenshot shows the PyCharm IDE interface. The code editor displays Client34.py, which contains Python code for a chat client. The terminal window below shows the execution of the script and the user's interaction:

```
C:\Users\Roni\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n rono/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4) 
ENTER YOUR NAME: Eliana
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 2
ENTER THE PASSWORD: |
```

After typing the password we get:

The screenshot shows the PyCharm IDE interface. The code editor displays Client34.py, which now includes the password input logic. The terminal window shows the completed connection process:

```
C:\Users\Roni\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n rono/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4) 
ENTER YOUR NAME: Eliana
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 2
ENTER THE PASSWORD: 123
Eliana has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
|
```

Now Client34 (Eliana) and Client33 (Noya) can speak:

Client34 (Eliana) types “Hello” and press enter:

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal - Client34.py
Computer Networks and Internet 2 / PythonDraft30.12 / Client34.py
Project Run ServerFinal Client33 Client34
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_renno/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
|
ENTER YOUR NAME: Eliana
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 127
ENTER THE PASSWORD: 127
Eliana has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Hello
Eliana: Hello
|
```

Client33 (Noya) also sees the message:

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal - Client33.py
Computer Networks and Internet 2 / PythonDraft30.12 / Client33.py
Project Run ServerFinal Client33 Client34
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_renno/Computer Networks and Internet 2/PythonDraft30.12/Client33.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
|
ENTER YOUR NAME: Noya
CHOOSE A PASSWORD FOR THE GROUP: 127
THE NEWLY CREATED CHAT ID IS 2
Noya has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Eliana has connected to the chat room
Eliana: Hello
|
```

Client33 (Noya) types "Hi":

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client34.py
Computer Networks and Internet 2 PythonDraft30.12 Client34.py
Project Run ServerFinal Client33 Client34
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n ronono/Computer Networks and Internet 2/PythonDraft30.12/Client33.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
|
ENTER YOUR NAME: Eliana
CHOOSE A PASSWORD FOR THE GROUP: 123
THE NEWLY CREATED CHAT ID IS 2
Noya has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Eliana has connected to the chat room
Eliana: Hello
Noya: Hi
|
```

Client 34 (Eliana) receives the message:

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client34.py
Computer Networks and Internet 2 PythonDraft30.12 Client34.py
Project Run ServerFinal Client33 Client34
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n ronono/Computer Networks and Internet 2/PythonDraft30.12/Client34.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
|
ENTER YOUR NAME: Eliana
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: |
ENTER THE PASSWORD: 123
Eliana has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Hello
Eliana: Hello
Noya: Hi
|
```

Now we will connect another client, Client35, named Sara, who choose to create a new group chat with password 456. She is automatically given the group id 3:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client35.py
Computer Networks and Internet 2 PythonDraft30.12 Client35.py
Project PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ClientFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
Run: ServerFinal Client33 Client34 Client35
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client35.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Sara
CHOOSE A PASSWORD FOR THE GROUP: 436
THE NEWLY CREATED CHAT ID IS 3
Sara has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
```

Now a new Client, Client36, whose name is Shlomo, connects to group chat 3:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client36.py
Computer Networks and Internet 2 PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ClientFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
Project PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ClientFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
Run: ServerFinal Client33 Client34 Client35 Client36
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client36.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Shlomo
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 3
ENTER THE PASSWORD: 436
Shlomo has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
```

Sara is notified that Shlomo had joined the chat:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client36.py
Computer Networks and Internet 2 PythonDraft30.12 Client36.py
Project PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
  receive_thread = threading.Thread(target=client_receive)
  receive_thread.start()
  send_thread = threading.Thread(target=client_send)
  send_thread.start()
  if option == "2"
Run: ServerFinal Client33 Client34 Client35 Client36
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client35.py"
Choose an option:
  1) Connect to a group chat
  2) Create a new group chat
  3) Disconnect
  4)
ENTER YOUR NAME: Sara
CHOOSE A PASSWORD FOR THE GROUP: 123
THE NEWLY CREATED CHAT ID IS 3
Sara has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Shlomo has connected to the chat room
19°C 10:17 AM 13:1 CRLF UTF-8 4 spaces Python 3.9 Event Log
Run TODO Problems Terminal Python Packages Python Console 11:33 30/12/2022 20
```

Now they can speak: Sara type “How are you?”

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client36.py
Computer Networks and Internet 2 PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
  receive_thread = threading.Thread(target=client_receive)
  receive_thread.start()
  send_thread = threading.Thread(target=client_send)
  send_thread.start()
  if option == "2"
Run: ServerFinal Client33 Client34 Client35 Client36
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client35.py"
Choose an option:
  1) Connect to a group chat
  2) Create a new group chat
  3) Disconnect
  4)
ENTER YOUR NAME: Sara
CHOOSE A PASSWORD FOR THE GROUP: 123
THE NEWLY CREATED CHAT ID IS 3
Sara has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Shlomo has connected to the chat room
How are you?
Sara: How are you?
19°C 10:17 AM 15:1 CRLF UTF-8 4 spaces Python 3.9 Event Log
Run TODO Problems Terminal Python Packages Python Console 11:33 30/12/2022 20
```

Shlomo sees the message:

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client36.py
Computer Networks and Internet 2 PythonDraft30.12 Client36.py
Project PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
96 receive_thread = threading.Thread(target=client_receive)
97 receive_thread.start()
98
99 send_thread = threading.Thread(target=client_send)
100 send_thread.start()
if option == "2":
    Sara: How are you?
|
```

Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Shlomo
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 1
ENTER THE PASSWORD: 456
Shlomo has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Sara: How are you?

Shlomo respond: "I am fine, thank you"

```
File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client36.py
Computer Networks and Internet 2 PythonDraft30.12 Client38.py Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
96 receive_thread = threading.Thread(target=client_receive)
97 receive_thread.start()
98
99 send_thread = threading.Thread(target=client_send)
100 send_thread.start()
if option == "2":
    Sara: How are you?
    I am fine, thank you
    Shlomo: I am fine, thank you
```

Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Shlomo
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 1
ENTER THE PASSWORD: 456
Shlomo has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Sara: How are you?
I am fine, thank you
Shlomo: I am fine, thank you

Sara gets the messege:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** Computer Networks and Internet 2 > PythonDraft30.12 > Client36.py
- Code Editor:** Client36.py (Line 96-100) shows the creation of two threads: `receive_thread` and `send_thread`, both using the `target` parameter to point to the `client_receive` and `client_send` functions respectively.
- Run Tab:** Shows runs for ServerFinal, Client33, Client34, Client35, and Client36.
- Terminal:** Displays the following output:

```
C:\Users\Ronit\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n_ronit/Computer Networks and Internet 2/PythonDraft30.12/Client35.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
/
ENTER YOUR NAME: Sara
CHOOSE A PASSWORD FOR THE GROUP: 454
THE NEWLY CREATED CHAT ID IS 3
Sara has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Shlomo has connected to the chat room
How are you?
Sara: How are you?
Shlomo: I am fine, thank you
```
- Bottom Bar:** Includes Run, TODO, Problems, Terminal, Python Packages, Python Console, Event Log, and system status icons.

Notice that, Client33(Noya)and Client34(Eliana) don't get the messeges that Shlomo and Sara are sending in the chat:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** PythonDraft30.12
- Files:** Client38.py, Client39.py, Client40.py, Client41.py, Client42.py, ServerFinal.py, Client33.py, Client34.py, Client35.py, Client36.py.
- Code Snippet (Client36.py):**

```
receive_thread = threading.Thread(target=client_receive)
receive_thread.start()

send_thread = threading.Thread(target=client_send)
send_thread.start()

if option == "2":
```

- Run:** ServerFinal, Client33, Client34, Client35, Client36.
- Terminal Output:**

```
C:\Users\Ronit\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n ronono/Computer Networks and Internet 2/PythonDraft30.12/Client33.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
2
ENTER YOUR NAME: Noya
CHOOSE A PASSWORD FOR THE GROUP: 123
THE NEWLY CREATED CHAT ID IS 2
Noya has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Eliana has connected to the chat room
Eliana: Hello
Hi
Noya: Hi
```

- Bottom Status Bar:** 19°C, 7:22, CRLF, UTF-8, 4 spaces, Python 3.9, Event Log, 11:34, ENG, 30/12/2022.

Likewise, if Client33(Noya) suddenly typed a message, like "Aloha", only Client34(Eliana) would get the message:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** Computer Networks and Internet 2 > PythonDraft30.12 > Client36.py
- Code Editor:** The code for Client36.py is displayed, showing the implementation of a client using threads for receiving and sending messages.
- Run Tab:** The run configuration is set to "Client36" with the command "C:\Users\Ronit\AppData\Local\Programs\Python\Python39\python.exe" "D:/Users/Shaiel/Bar-Ilan/n_ronit/Computer Networks and Internet 2/PythonDraft30.12/Client34.py".
- Output Window:** The output shows the client connecting to a group chat, sending a message, and receiving a response from another user.
- Bottom Bar:** Includes icons for Run, TODO, Problems, Terminal, Python Packages, and Python Console, along with system status icons like battery level (19%), temperature (19°C), and network connection.

Now Client37, Ariel, tries to connect to chat 2:

However, he entered the wrong password, 122 instead of 123:

The screenshot shows a PyCharm IDE interface with the following details:

- Project:** PythonDraft30.12
- Files:** Client37.py (active), Client33.py, Client34.py, Client35.py, Client36.py, Client37.py, Client39.py, Client40.py, Client41.py, Client42.py, ServerFinal.py.
- Terminal Output:**

```
C:\Users\Ronit\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Bar-Ilan/n ronomo/Computer Networks and Internet 2/PythonDraft30.12/Client37.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Alon
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 22
ENTER THE PASSWORD: 22
Sorry, we are unable to connect you to the group chat
Disconnecting you from the server . .
Disconnection completed
```
- Bottom Status Bar:** Shows the date (30/12/2022), time (13:13), and file encoding (UTF-8).

He is disconnected from the chat. If he tries to send more messages, he will get:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client37.py
Computer Networks and Internet 2 PythonDraft30.12 / Client37.py
Project PythonDraft30.12 Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
  96     receive_thread = threading.Thread(target=client_receive)
  97     receive_thread.start()
  98
  99     send_thread = threading.Thread(target=client_send)
100     send_thread.start()
101
102     if option == "2":
103
104     ENTER YOUR NAME: arg1
105     ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 123
106     ENTER THE PASSWORD: 123
107     Sorry, we are unable to connect you to the group chat
108     Disconnecting you from the server . .
109     Disconnection completed
110     can leave here we?
111     Disconnection completed, you can not send messages!
112
113 Process finished with exit code 0
```

Client33(Noya) and Client34(Eliana) can still continue to chat regularly:

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client37.py
Computer Networks and Internet 2 PythonDraft30.12 Client37.py
Project PythonDraft30.12 Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
96     receive_thread = threading.Thread(target=client_receive)
97     receive_thread.start()
98
99     send_thread = threading.Thread(target=client_send)
100    send_thread.start()
if option == "2":

```

Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)

ENTER YOUR NAME: Eliana
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 123
ENTER THE PASSWORD: 123
Eliana has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Hello
Eliana: Hello
Noya: Hi
Arieh
Eliana: Aloha

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ServerFinal.py - Client37.py
Computer Networks and Internet 2 PythonDraft30.12 Client37.py
Project PythonDraft30.12 Client39.py Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py
96     receive_thread = threading.Thread(target=client_receive)
97     receive_thread.start()
98
99     send_thread = threading.Thread(target=client_send)
100    send_thread.start()
if option == "2":

```

Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)

ENTER YOUR NAME: Noya
CHOOSE A PASSWORD FOR THE GROUP: 123
THE NEWLY CREATED CHAT ID IS 2
Noya has connected to the chat room
YOU ARE NOW CONNECTED TO THE CHAT
Eliana has connected to the chat room
Eliana: Hello
Hi
Noya: Hi
Eliana: Aloha

Client37(Ariel) doesn't get the messeges:

```
C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shalel/Baer-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client37.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
1
ENTER YOUR NAME: argel
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 123
ENTER THE PASSWORD: 123
Sorry, we are unable to connect you to the group chat
Disconnecting you from the server . .
Disconnection completed
can leave here we?
Disconnection completed, you can not send messages!

Process finished with exit code 0
```

On the server side, we can see the error is “No such password”:

```
3
3
b'Shalomo: I am fine, thank you'
connection is established with ('127.0.0.1', 50967)
2
122
result

0
No such password
Server is listening
tempo-temp
<socket.socket fd=372, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.2', 59000), raddr=('127.0.0.1', 50958)>
<class 'str'>
2
2
b'Eliana: Aloha'
tempo-temp
<socket.socket fd=360, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.2', 59000), raddr=('127.0.0.1', 50959)>
<class 'str'>
2
2
b'Eliana: Aloha'
```

Client38(Rinat) now wants to join chat 2. However, she accedlty entered the wrong group id:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client38.py
Computer Networks and Internet 2 PythonDraft30.12 Client38.py
Project Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py Client38.py Client39.py Client40.py
Run: ServerFinal Client33 Client34 Client35 Client36 Client38
C:\Users\Roni\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Ban-Ilan/n_ronon/Computer Networks and Internet 2/PythonDraft30.12/Client38.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
4)
ENTER YOUR NAME: Client
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 123
ENTER THE PASSWORD: 123
Sorry, we are unable to connect you to the group chat
Disconnecting you from the server . .
Disconnection completed
```

She also gets disconnected. On the server side, we can see the error was “No such id”:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client38.py
Computer Networks and Internet 2 PythonDraft30.12 Client38.py
Project Client40.py Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py Client38.py Client39.py Client40.py
Run: ServerFinal Client33 Client34 Client35 Client36 Client38
connection is established with ('127.0.0.1', 58971)
22
123
result
2
index5

5
passoriginal

123
current password

123
matching password
flag
22
result2 is:

0
No such id
Server is listening
```

Client39(Roni) now tries to join chat 2. He accidentally gets both the ID and the password wrong, and gets disconnected from the group chat:

```

File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client39.py
Computer Networks and Internet 2 PythonDraft30.12 Client39.py
Project Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py Client38.py Client39.py Client40.py
100
receive_thread = threading.Thread(target=client_receive)
receive_thread.start()
send_thread = threading.Thread(target=client_send)
send_thread.start()
if option == "2"
    print("Sorry, we are unable to connect you to the group chat")
    Disconnecting you from the server . .
    Disconnection completed

```

ENTER YOUR NAME: *Yuval*
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: *22*
ENTER THE PASSWORD: *123*
Sorry, we are unable to connect you to the group chat
Disconnecting you from the server . . .
Disconnection completed

the error from the server side “No such password”:

```

File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client39.py
Computer Networks and Internet 2 PythonDraft30.12 Client39.py
Project Client41.py Client42.py ServerFinal.py Client33.py Client34.py Client35.py Client36.py Client37.py Client38.py Client39.py Client40.py
100
receive_thread = threading.Thread(target=client_receive)
receive_thread.start()
send_thread = threading.Thread(target=client_send)
send_thread.start()
if option == "2"
    print("5")
    print("passoriginal")
    print("123")
    print("current password")
    print("123")
    print("matching password")
    flag
    22
    result2 is:
    0
    No such id
    Server is listening
    connection is established with ('127.0.0.1', 51046)
    22
    222
    result
    0
    No such password
    Server is listening

```

The already connected clients can continue to chat regularly.

Client38 and Client39 don't get their messages.

If clients38 and 39 would try to send more messages, they would get the same output as client37.

Client40(Yuval) wishes to connect to the chat with ID 3. However, he accidentally confused the password for the chat ID with 2 with the chat with ID 3. The ID and the password exist in the system, but don't match, so he gets disconnected from the server:

```

File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client40.py
Computer Networks and Internet 2 PythonDraft30.12 Client40.py
Project Client42.py Client40.py
  Client35.py
  Client36.py
  Client37.py
  Client38.py
  Client39.py
  Client40.py
  receive_thread = threading.Thread(target=client_receive)
  receive_thread.start()
  send_thread = threading.Thread(target=client_send)
  send_thread.start()
if option == "2"
Choose an option:
  1) Connect to a group chat
  2) Create a new group chat
  3) Disconnect
  4)
ENTER YOUR NAME: ronel
ENTER THE ID OF THE GROUP YOU WANT TO JOIN: 4
ENTER THE PASSWORD: 456
Sorry, we are unable to connect you to the group chat
Disconnecting you from the server ...
Disconnection completed

```

The server prints the error om his side:

```

File Edit View Navigate Code Befactor Run Tools VCS Window Help ServerFinal.py - Client40.py
Computer Networks and Internet 2 PythonDraft30.12 Client40.py
Project Client42.py Client40.py
  Client35.py
  Client36.py
  Client37.py
  Client38.py
  Client39.py
  Client40.py
  receive_thread = threading.Thread(target=client_receive)
  receive_thread.start()
  send_thread = threading.Thread(target=client_send)
  send_thread.start()
if option == "2"
222
result
8
No such password
Server is listening
connection is established with ('127.0.0.1', 51050)
2
456
result
2
index5
1
passoriginal
123
current password
456
unmatching password
Server is listening

```

The Clients who previously connected can continue to chat regularly.

Client41 chooses option 3 and get automatically disconnected:

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help ServerFinal.py - Client41.py
Computer Networks and Internet 2 PythonDraft30.12 Client41.py
Project Run: C:\Users\Roniv\AppData\Local\Programs\Python\Python39\python.exe "D:/Users/Shaiel/Ban-Ilan/n_roniv/Computer Networks and Internet 2/PythonDraft30.12/Client41.py"
Choose an option:
1) Connect to a group chat
2) Create a new group chat
3) Disconnect
Process finished with exit code 0
```

The Clients can continue to chat regularly.

Like the previously disconnected clients, Client40 and Client 41 won't get any messages.