Documentation

General Explanation of the System and Code Structure

The system we built is a chat application based on TCP sockets, enabling communication between a server and multiple clients. The system supports public messages (broadcast to all connected clients) as well as private messages (directed to a specific client).

Code Structure

Server Features

1. **Multithreading Support**
    - The server can handle multiple clients simultaneously using separate threads.
2. **Broadcast Messages**
    - The server broadcasts messages to all connected clients except the sender.
3. **Private Messaging**

Supports private messages using the format:

/pm <username> <message>

    -
    - Allows clients to send messages intended only for a specific user.
4. **Connection Management**
    - The server maintains a list of all connected clients and their usernames.
    - In case of unexpected client disconnection, the server removes them from the list and notifies the other clients.
5. **Graceful Shutdown**
    - The server can broadcast a message to all clients when it shuts down (an optional feature).

Client Features

1. **Connecting to the Server**
    - The client prompts the user to enter a username and connects to the server.
    - The username is sent to the server for identification.
2. **Sending Public Messages**
    - Users can send public messages that are received by all connected clients.
3. **Sending Private Messages**

Users can send private messages to a specific client using the format:

/pm <username> <message>

    -
4. **Receiving Real-Time Messages**
    - The client listens for messages from the server and displays them to the user.
5. **Disconnecting from the Server**

The client allows the user to disconnect from the server by typing:

quit

    -
6. **User Notifications**
    - The client displays clear instructions to the user, such as how to send a private message.

Installation and Execution Instructions

Prerequisites

- Make sure Python (version 3 or above) is installed.
- Download the files `server.py` and `client.py` into a local folder on your computer.

Running the Server

Open a terminal and run:

```
python server.py
```

-
- The server will display a message indicating it is listening for incoming connections.

Running the Client

Open a new terminal and run:

```
python client.py
```

-

The client will prompt you to enter a username. For example:

```
Enter your username:
```

-

Input and Output Examples

5 Clients Connecting

- **Sam** sends a group message:

On the server terminal:

[Broadcast message from Sam...]

Sending a Private Message (Eliya → Sam)

/pm Sam Hello!

Disconnecting

- **Sam** types `quit` and disconnects from the server.

On the server terminal:

[Sam has disconnected]