

Enhancing Chat Application: Multi-Client Support

Hammoudi Sarra

1 Introduction

In the initial version of our chat application, the server could handle only one client at a time. To support multiple clients, we introduced multithreading in the server. Now, the server can accept multiple client connections and broadcast messages to all connected clients.

This document highlights the changes in class design, how they interact, and explains the multi-client functionality.

2 Modifications to the System

To enable multi-client communication, we made the following modifications:

1. The Server now maintains a list of connected clients. 2. The ClientHandler class is introduced to handle each client in a separate thread. 3. The broadcast method allows sending messages to all clients. 4. Clients receive messages in real-time without blocking input.

3 Class Modifications

3.1 Changes in the Server Class

Previous Version: *The server could handle only one client.*

Updated Version: *The server now:*

- *Uses a `Set<ClientHandler>` to store connected clients.*
- *Creates a new thread for each client connection.*
- *Calls the `broadcast()` method to send messages to all clients.*

3.2 Introduction of the ClientHandler Class

Purpose: *Handles individual client connections in separate threads.*

Responsibilities:

- Reads messages from the client.
- Calls `broadcast()` to share messages with all clients.
- Closes the connection when a client disconnects.

3.3 Client Class Modifications

Previous Version: The client communicated directly with the server.

Updated Version: The client now:

- Can receive messages while typing (using a separate thread).
- Continuously reads server messages and displays them.

4 How Multi-Client Support Works

1. The server starts and waits for incoming client connections. 2. When a client connects, a new `ClientHandler` thread is created. 3. Each client sends messages to the server. 4. The server broadcasts the message to all connected clients. 5. If a client disconnects, it is removed from the active clients list.

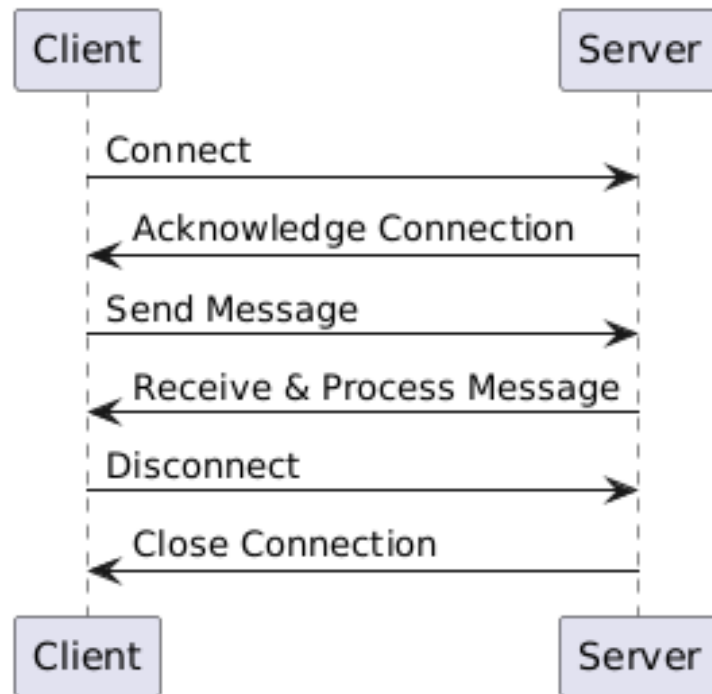


Figure 1: Multi-Client Chat System Flow

5 Advantages of Multi-Client Support

This new implementation allows:

- **Real-time communication** between multiple users.
- **Multithreading** to manage concurrent client connections.
- **Scalability**, as more clients can join dynamically.