CEN322 – Network Programming

Project 02: Multi-Client Chat Application

Student Information

Name: İbrahim Emre YILDIZ
Student ID: 2020555069
Course: CEN322 – Network Programming
Date: 31.10.2025

---

1. Introduction

This project aims to develop a simple **multi-client chat system** using Python socket programming.
The server communicates with multiple clients through TCP connections.
Each client can send and receive messages in real time, use private messaging, and the system prevents spam using a rate-limiting feature.

---

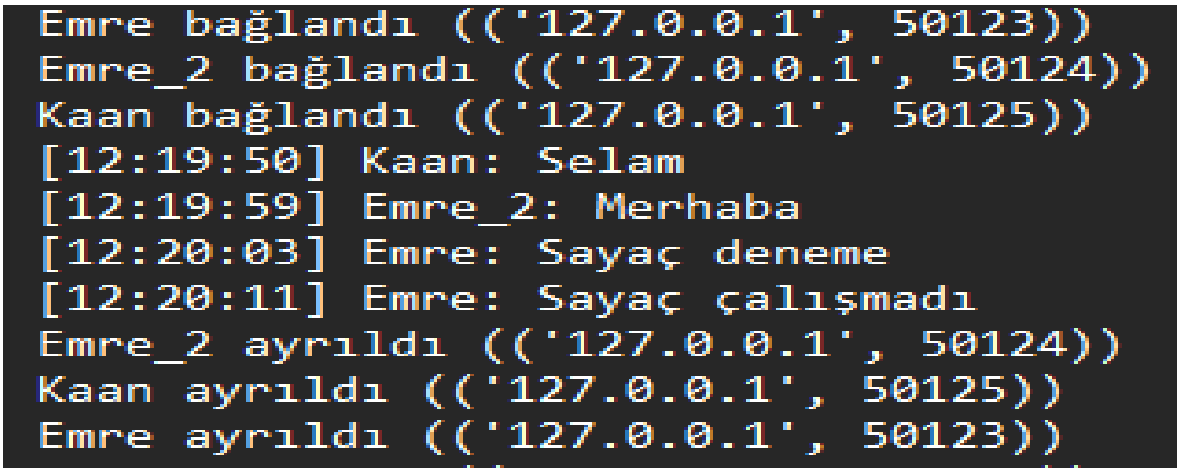
2. System Architecture

The system is based on a **client-server model** using TCP sockets.
Each client runs in a separate thread on the server side, allowing simultaneous communication.

---

3. Modules

Module Name	Description
chat_server.py	Manages all client connections, broadcasts messages, applies rate limit, logs events, and displays statistics.
chat_client.py	Sends and receives messages, supports /msg command for private messaging.
server_log.txt	Contains connection, disconnection, and message logs automatically recorded by the server.



4. Implemented Features

Feature	Description
Multi-Client Support	The server accepts multiple clients at the same time using threading.
Nickname System	Each client chooses a nickname. Duplicates get _2, _3, etc.
Private Messaging (/msg)	Users can send private messages using /msg <name> <message>.
Rate Limiting	Prevents spam by limiting 8 messages per 5 seconds. Violations lead to 10 seconds mute.
Logging	All activities are written into server_log.txt.
Active User List	Shows connected users to everyone when someone joins or leaves.
Performance Monitoring	Shows statistics like message count and total connections on the server terminal.
Error Handling	Handles disconnections and socket errors safely.

```
[STATS] Aktif kullanıcı: 1 | Toplam bağlantı: 1
Emre_2 (('127.0.0.1', 58757)) bağlandı.
[STATS] Aktif kullanıcı: 2 | Toplam bağlantı: 2
Emre_3 (('127.0.0.1', 58760)) bağlandı.
[STATS] Aktif kullanıcı: 3 | Toplam bağlantı: 3
[STATS] Toplam işlenen mesaj: 1
[13:35:36] Emre: Selam
[STATS] Toplam işlenen mesaj: 2
[13:35:41] Emre_2: Merhaba
[STATS] Toplam işlenen mesaj: 3
[13:35:44] Emre_3: Hi!
```

5. Code Overview

Main functions used in the project:

- **handle_client()** – Handles each connected client in a separate thread.
- **broadcast()** – Sends messages to all connected clients.
- **get_socket_by_nick()** – Finds a client socket using the nickname.
- **rate limit logic** – Tracks message timestamps and applies mute if too many messages are sent.
- **log_message()** – Saves logs to server_log.txt.

```
1
[13:39:16] Emre: 1
1
[13:39:17] Emre: 1
1
[UYARI] Çok hızlı mesaj yolluyorsunuz. 10 sn susturuldunuz.


11
[UYARI] Spam nedeniyle susturuldunuz, 9 sn sonra tekrar deneyin.
```

```
[13:39:16] Emre: 1
[13:39:16] Emre: 1
[13:39:17] Emre: 1
[RATE-LIMIT] Emre susturuldu (10s)
```

6. Execution Steps

a) Starting the Server

python chat_server.py

Expected Output:

[SERVER] Sunucu çalışıyor... 5000 portunda

Starting the Client

Open multiple terminals and run:

python chat_client.py

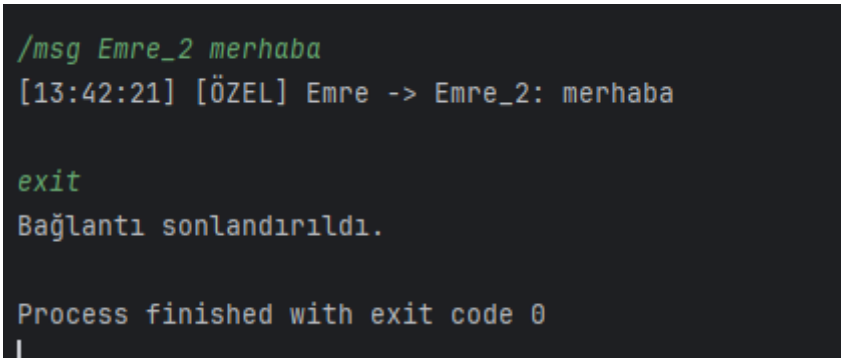
Example Output:

Sunucuya bağlanıldı. 'exit' yazarak çıkabilirsiniz.

Kullanıcı adınızı yazın: Emre

Commands

- Normal message:
  Hello everyone!
- Private message:
  /msg Ali Hi!
- Exit:
  exit



7. What I Learned

Through this project, I learned:

- How to use sockets for TCP communication in Python.
- How to handle multiple clients using threading.
- How to manage real-time message broadcasting and logging.
- How to apply rate limiting and performance monitoring.

It also improved my understanding of concurrency, socket APIs, and network stability.

8. References

- Python Official Documentation – Socket and Threading libraries
- CEN322 Lecture Notes
- Stack Overflow – Examples of Python chat servers
- GitHub Repository- https://github.com/IbrahimEmreYildiz/diagnostic_tool (our first project)