A logo of a university

Description automatically generated

|  |  |
| --- | --- |
| **Subject** | **Object Oriented Programming** |
| **Assignment** | **Messaging App(use of sockets)** |
| **Submitted To** | **Sir Shahid Bhatti** |

**Submitted By:**

**Zaina Hassan SP24-BSE-126**

**Abeera Rehman SP24-BSE-005**

**Assignment 2**

**1. Application Overview**

* A client-server messaging system that supports:
  + **Sending, receiving, and managing messages** in real time.
  + **Interaction between multiple clients** and the server through TCP sockets.
* Built in Java, utilizing ServerSocket and Socket for network communication.

**2. Key Components**

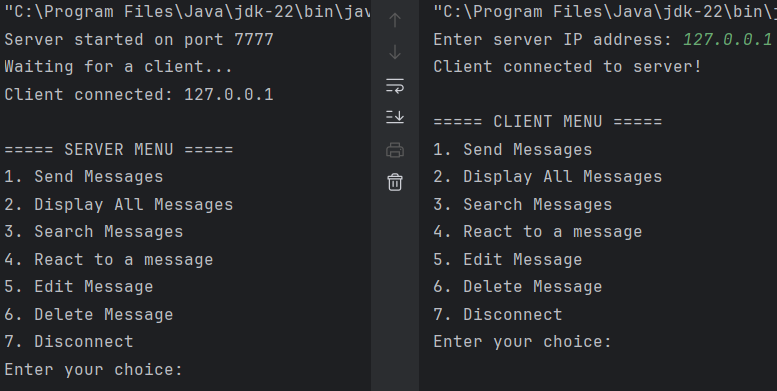
* **Server Class**: Manages incoming client connections and processes messages.
* **Client Class**: Sends messages to and receives responses from the server.
* **Message Class**: Represents individual messages with unique IDs, sender info, content, and timestamps.

**3. Server Side Implementation**

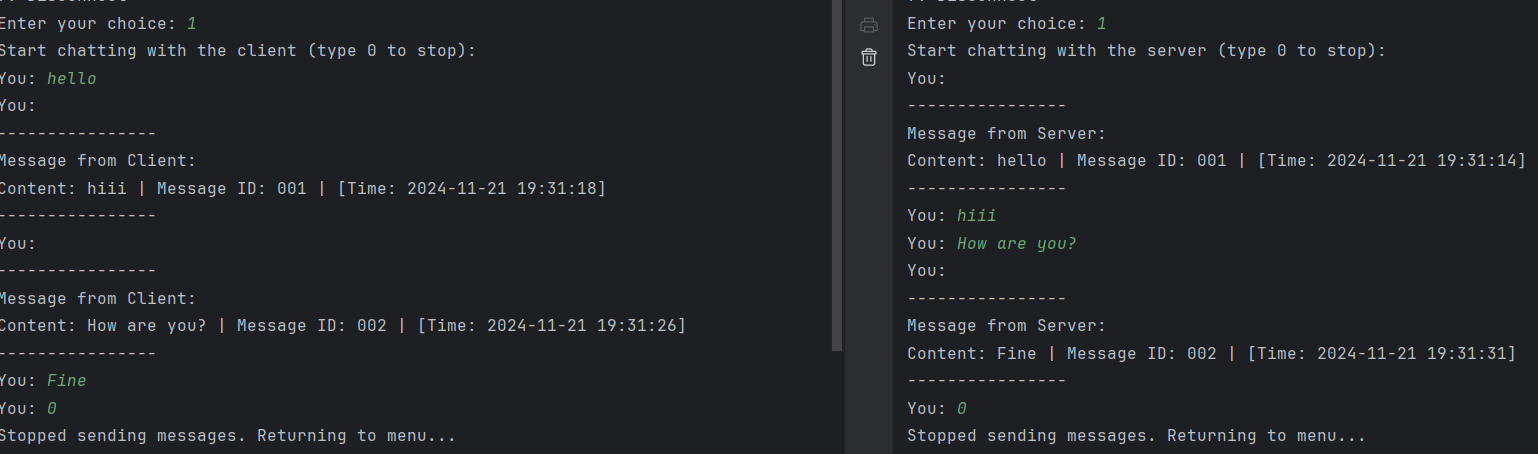
* **ServerSocket**: Listens on port 7777 for incoming client connections.
* **Message Management**:
  + **Add**: Adds a new message to the server.
  + **Edit/Delete**: Allows editing or deleting messages by ID.
  + **Display/Search**: Searches or displays messages based on specific criteria.
* **Multithreading**: Each client is handled in a new thread, allowing multiple clients to connect simultaneously.
* Listens to client messages, processes operations like deletion and editing, and responds accordingly.
* Uses arrays to keep track of messages sent and received.

**4. Client Side Implementation**

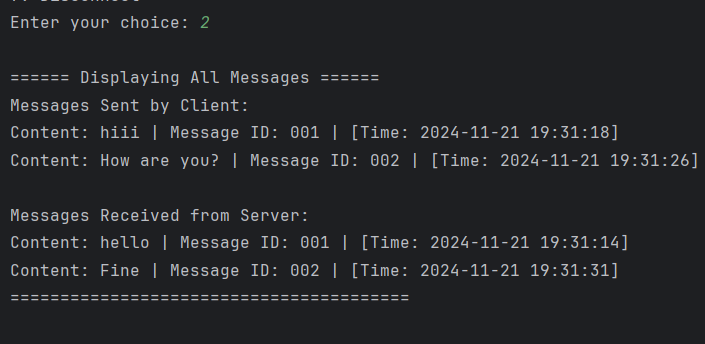
* **Socket Connection**: Connects to the server IP on port 7777.
* **User Interaction**:
  + **Options Menu**: Provides options to send, display, search, edit, or delete messages.
  + **Input Handling**: Prompts the user for IDs and content when editing or searching messages.
* **Listener Thread**: Receives messages from the server continuously in the background.
* Supports sending messages (sendMessagesContinuously), displaying them, searching, editing, and deleting messages.
* Contains arrays to store client and server messages locally.

**Initial Menu:**

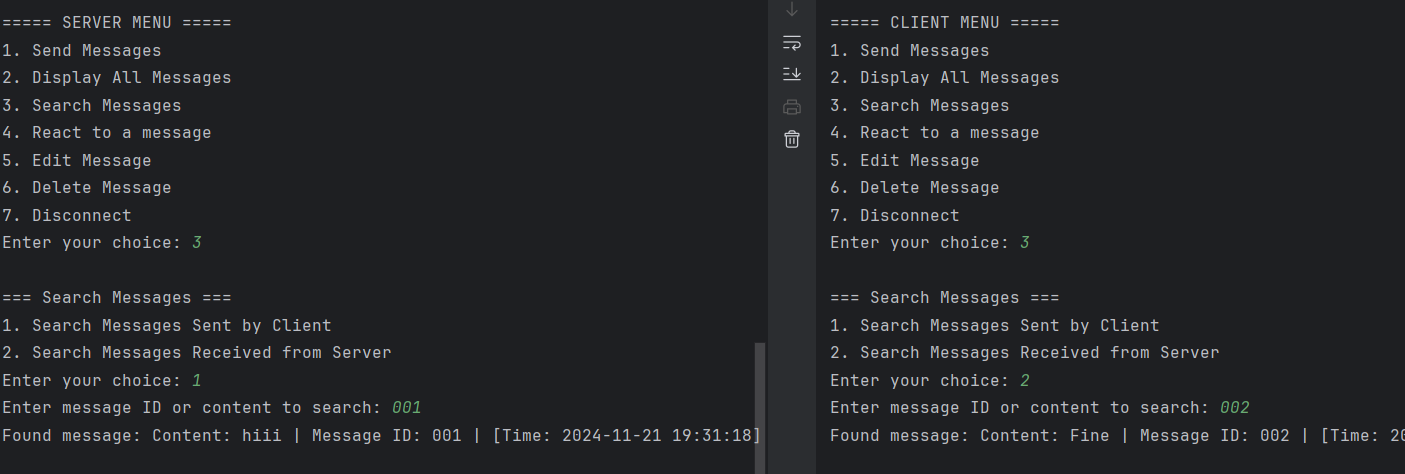
**When 1 pressed (To Send a Message):**

**Server Client**

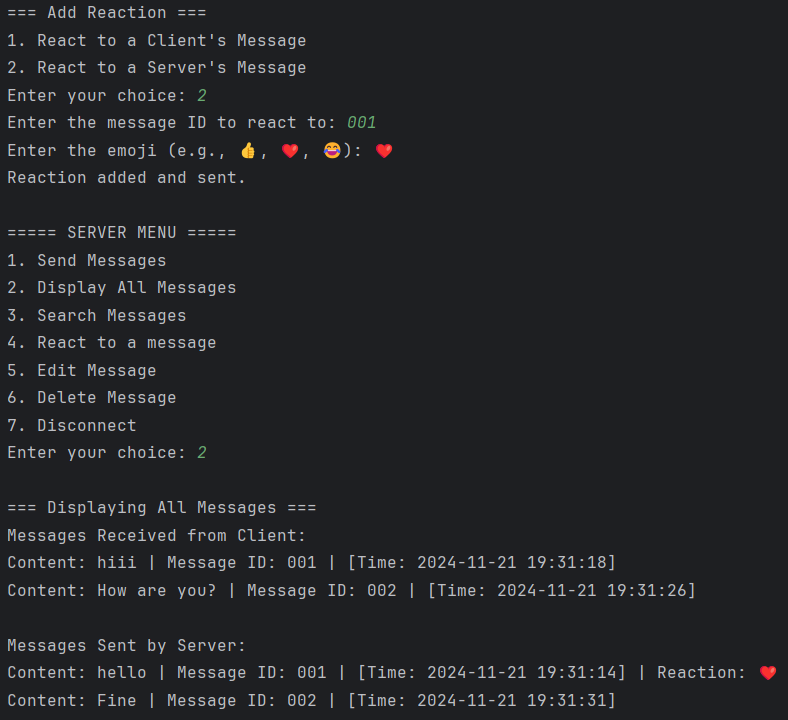
**When 2 Pressed (To Display All Messages):**



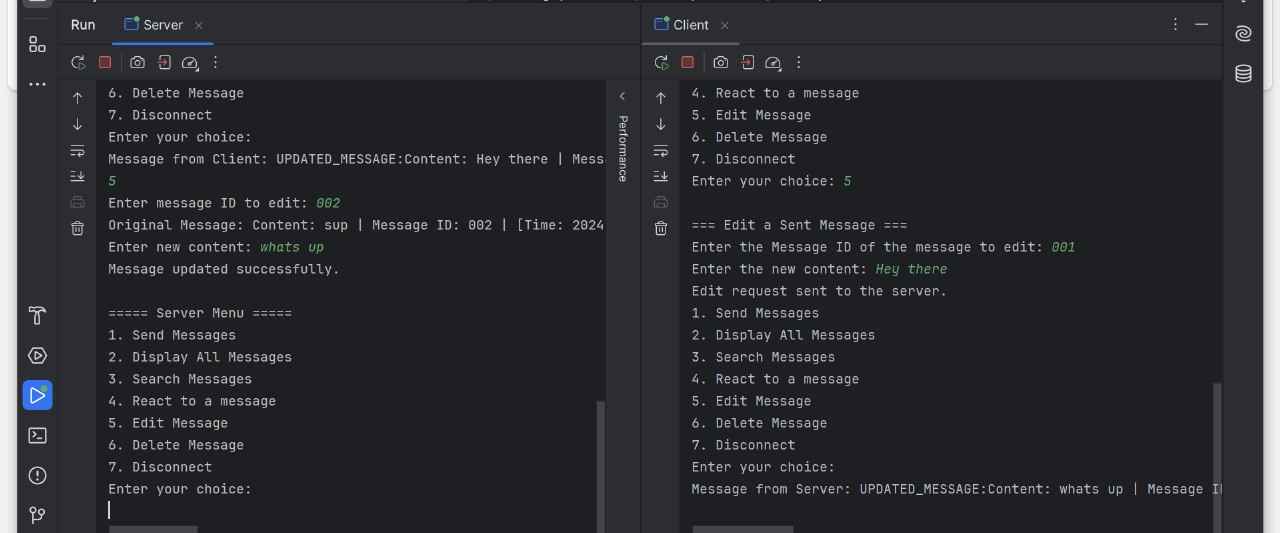
**When 3 pressed (To Search a message from either console):**

**Server Client**

**When 4 Pressed (To React to a Message):**

****

**When 5 Pressed (To Edit a message)**



**When 6 Pressed (To delete a message)**

A screenshot of a computer

Description automatically generated

**UML Diagram**

**A screenshot of a computer

Description automatically generated**