

**An-Najah Nation University**

**Faculty of Engineering and**

**Information Technology**

# جامعة النجاح الوطنیة

**كلیة الھندسة وتكنولوجیا المعلومات**

**Computer Engineering Department**

**Networks1 (10636454)**

**Programming HW project**

**Chatting Between Peers**

Instructor name: Student#1 Name: Ibrahim Mashaqi

Dr. Eng. Saed TARAPIAH Student#2 Name: Nassar Barahmeh

Eng. Ola Mardawi

Academic Year: **2024-2025** Registration Number#1: 12218206

Semester: **Spring** Registration Number#2: 12218619

Credit Hours: **3**

Date: **5/April/2025**

### **. Introduction**

This project aims to build a peer-to-peer (P2P) chat application using UDP socket programming in Java, with an interactive graphical user interface (GUI). The application enables users to exchange text messages over a network while fulfilling additional requirements such as message formatting, archiving, and log management.

### **. System Design**

#### **Functional Requirements**

* Send and receive messages via UDP with manual input of IP addresses and ports.
* Display sent and received messages with timestamps and distinct colors (yellow for sent, orange for received).
* Conversation management buttons: Delete All, Delete Selected, Archive.
* Store deleted messages in an archive and automatically remove them after 2 minutes.
* Create a log file to record all activities.

#### **Architecture Overview**

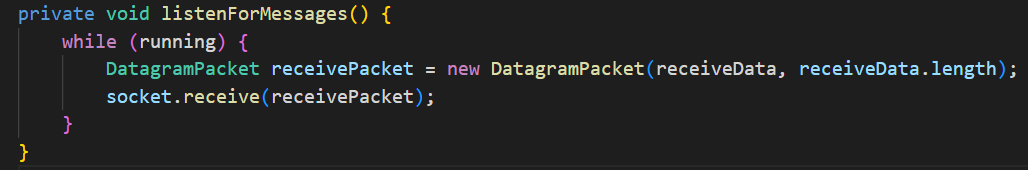
* **Graphical User Interface (GUI):**
  + Input fields for local and remote IP addresses and ports.
  + Message display list with visual formatting.
  + Control buttons for sending, deleting, and archiving messages.
* **Backend Logic:**
  + Use DatagramSocket to establish UDP connections.
  + Separate threads for listening to incoming messages and sending outgoing messages.
  + Archive and deleted message management.

### 

### **0. Implementation**

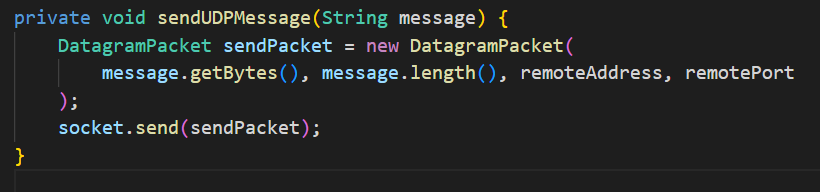
#### **Key Code Components**

1. **Socket Initialization:**



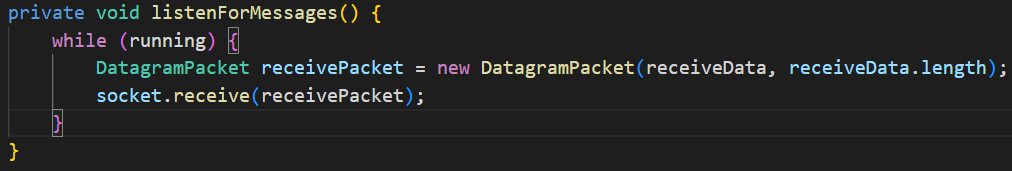
* + Creates a UDP connection using the specified local port and IP address.

1. **Sending Messages:**



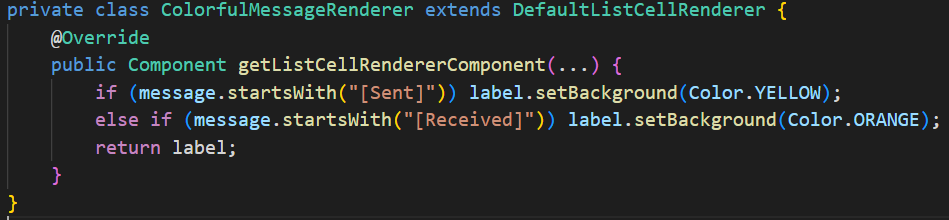
* + Sends messages as UDP packets to the specified remote address.

1. **Receiving Messages:**



* + A dedicated thread for continuous listening to incoming messages.

1. **Message Formatting:**



* + Colors messages based on type (adjusted to meet requirements).

1. **Archive System:**
   * Deleted messages are moved to archivedMessages.
   * Archived messages are displayed in a separate window via showArchiveDialog().

### **. Testing**

* **Connection Testing:**
  + Messages were exchanged between peers on the same machine (localhost) and different machines.
  + Verified message delivery with correct timestamps and colors.
* **Archive Testing:**
  + Deleted specific messages and confirmed their transfer to the archive.
  + Reopened the archive to view messages (restore functionality not yet implemented).

### **. Challenges and Solutions**

* **Thread Synchronization:**
  + Used SwingUtilities.invokeLater() to update the GUI from sub-threads.
* **Socket Management:**
  + Closed sockets upon window closure to prevent resource leaks.
* **Color Mismatch:**
  + Adjusted colors in ColorfulMessageRenderer to yellow and orange.

### **. Improvements**

* Add a log file to record events such as message sending, deletion, and archiving.
* Implement automatic removal of archived messages after 2 minutes using TimerTask.
* Enhance error handling for invalid IP/port inputs.

### **Conclusion**

A peer-to-peer chat application using UDP sockets was successfully implemented with an interactive GUI that meets core requirements. Additional features (e.g., log files) require further development.

**Appendix:**

* GUI Screenshot:

