

Java Chat Application – Project Report

Introduction

The Java Chat Application is a real-time peer-to-peer messaging system designed to simulate basic functionalities of modern chat platforms. It supports both group and private messaging, user nicknames, and connection logs. Built using Java's networking and multithreading capabilities, the project demonstrates core concepts of socket programming and concurrent processing.

Abstract

This project showcases the implementation of a multi-client chat system using Java. The server listens for incoming connections and delegates each client to a separate thread for parallel communication. A simple JavaFX-based GUI enhances usability, while basic encryption ensures message privacy. The application is scalable and serves as a strong foundation for more advanced messaging platforms.

Tools Used

- Core Programming Language: Java (JDK 17)
- Networking: Socket Programming (ServerSocket and Socket for client-server communication)
- Concurrency: Java Threads
- GUI Framework: JavaFX
- Encryption: Base64 Encoding (For basic encryption of messages)
- IDE: IntelliJ IDEA

Steps Involved in Building the Project

Server Setup:

- Implemented a ServerSocket to listen for incoming client connections.
- Created a ClientHandler class to manage each client in a separate thread.
- Maintained a ConcurrentHashMap of connected clients for efficient message broadcasting.

ClientHandler Implementation:

- Developed a ClientHandler class that runs on a separate thread per client.
- Managed incoming and outgoing messages, nickname assignment, and disconnection handling.

GUI Design:

- Built a JavaFX interface with a text area to display messages, input field for typing messages, and a send button.
- Used dialogs to input server host, port, and nickname.

- Handled message sending and receiving asynchronously in a separate thread to keep the UI responsive.

Messaging Logic:

- Supported group messages: broadcasted to all connected clients.
- Supported private messages: sent to a specific user using `/w <nickname> <message>`.
- Encoded messages using Base64 to ensure safe transmission of special characters.

Connection Management:

- Verified unique nicknames before allowing users to join.
- Broadcasted information messages when a user joined or left the chat.
- Maintained server-side logs of all messages and user activity.

Encryption Layer:

- Implemented Base64 encoding for all messages as a basic encryption mechanism.
- Ensured client-server communication is consistent and error-free.

Conclusion

The Java Chat Application successfully demonstrates real-time communication using sockets and threads. It provides a foundation for building more advanced chat systems, including features like online user lists, file sharing, and secure message encryption. The project reinforces understanding of client-server architecture and JavaFX-based interface design.

It integrates GUI design, encryption, and user management into a cohesive system. This project is a valuable addition to any developer's portfolio, especially for roles involving networking, concurrency, or desktop application development.

Author: Aashi Vishwakarma

Date: 08-09-2025

Email: aashivishwakarma1121@gmail.com