

Project Title: Client-Server Chat Application

Description

The Simple Chat Application is a multi-user chat program that allows users to communicate in real-time over a network. This application is designed to facilitate group discussions among a limited number of participants, making it ideal for small team collaborations or casual conversations.

Features

User Authentication:

Users must enter a valid user-name and corresponding password to access the chat.

The application supports six predefined users with usernames and respective passwords:

Users: Abadanaf, Abdi, Amanuel, Ayana, Bana, Abubaker

Passwords: 01, 02, 03, 04, 05, 06

Real-Time Messaging:

Once authenticated, users can send messages to all participants in the chat. Messages are broadcasted to all connected clients, creating a real-time chat experience.

User Limit:

The chat room is limited to a maximum of three concurrent users. If the limit is reached, subsequent users will receive a notification that the chat room is full.

Graceful Exit:

Users can exit the chat by typing "bye," which notifies other participants of their departure.

Technology Stack

Java:

The application is built using Java, leveraging its networking capabilities to create a client-server architecture.

Sockets: The program utilizes Java Sockets for network communication between the client and server.

Architecture

The application consists of two main components:

1. Server Component:

- Listens for incoming client connections.

- Handles user authentication and manages the chat sessions.

- Broadcasts messages to all connected clients.

2. Client Component:

- Connects to the server and handles user inputs.

- Displays messages received from the server and allows users to send their messages.

Usage

1. Running the Server:

- Compile and run the **ServerSideChat.java** file.

- The server will start listening for client connections.

2. Running the Client:

- Compile and run the **ClientSideChat.java** file.

- Each client will be prompted to enter their username and password.

- Upon successful authentication, users can start messaging.

Example Interaction

1. User enters their username:

Enter username: **Abadanaf**

2. User is prompted for their password:

Enter password for Abadanaf:

3. Upon successful login, the user receives a welcome message:

**Welcome to the chat, Abadanaf! You have successfully connected to the server.
You can start the discussion...**

4. Users can send messages:

Abadanaf: Hello everyone!

Conclusion

The Simple Chat Application demonstrates fundamental principles of network programming in Java, including client-server communication, user authentication, and real-time messaging. This project serves as a foundational example for understanding and building upon networked applications.

Collaborators	ID
• ABADANAF CHALA	1848/14
• ABDIBIYA ABADURA.....	1875/14
• AMANUEL ABATE	2122/14
• AYANA FILE	2226/14
• BANA DAWIT	2245/14
. Abubaker Mohammed.....	2045/14