

ChatManager Project Report

Silent0Wings

August 18, 2024

Contents

1	Introduction	1
2	Project Overview	1
2.1	Core Components	2
3	Functionality	2
3.1	Chat Server	2
3.2	Chat Client	2
4	Code Snippets	2
4.1	Server Class	2
4.2	Client Class	3
5	Conclusion	3
6	References	3

1 Introduction

The **ChatManager** project is a Java-based application designed to manage real-time communication between multiple clients through a server. This report provides an overview of the project's components, functionality, and design rationale.

2 Project Overview

The primary goal of the **ChatManager** is to facilitate communication between clients in a network. This involves setting up a server to handle multiple client connections and enabling message exchanges between them.

2.1 Core Components

- **Chat Server:** Manages incoming client connections and facilitates message routing between clients.
- **Chat Client:** Allows users to connect to the server, send, and receive messages.
- **Message Handling:** Ensures that messages are delivered to the intended recipients efficiently.

3 Functionality

This section discusses the specific functionalities offered by the **ChatManager** project.

3.1 Chat Server

The server class is responsible for accepting client connections, managing client threads, and routing messages between connected clients.

3.2 Chat Client

The client class connects to the server and enables users to send and receive messages. It also handles the display of incoming messages.

4 Code Snippets

Below are some critical code snippets from the project to illustrate how the functionality is implemented.

4.1 Server Class

```
public class Server {
    private ServerSocket serverSocket;

    public void start(int port) {
        serverSocket = new ServerSocket(port);
        while (true) {
            new ClientHandler(serverSocket.accept()).start();
        }
    }

    public void stop() {
        serverSocket.close();
    }
}
```

4.2 Client Class

```
public class Client {
    private Socket clientSocket;
    private PrintWriter out;
    private BufferedReader in;

    public void startConnection(String ip, int port) {
        clientSocket = new Socket(ip, port);
        out = new PrintWriter(clientSocket.getOutputStream(), true);
        in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
    }

    public void sendMessage(String msg) {
        out.println(msg);
    }

    public String receiveMessage() {
        return in.readLine();
    }

    public void stopConnection() {
        in.close();
        out.close();
        clientSocket.close();
    }
}
```

5 Conclusion

The **ChatManager** project is designed to provide a robust framework for managing real-time chat applications. By implementing core components like the chat server and client, the project ensures efficient and reliable communication between users. This report has outlined the key features and provided insight into the underlying code, demonstrating how the project achieves its goals.

6 References

Include any references to external libraries, resources, or documentation used in the project.