TCP-Chat

This programming assignment gives you the opportunity to implement a simple chat application using the TCP-protocol. It counts 5% towards the final grade.

Instructions: You are allowed to work alone or in teams of two students.
You are asked to develop an Internet chat system, based on a client/server architecture, using the Java language, and exploiting the TCP protocol:

- Your system should allow one to connect *multiple* remote clients to a single central server.
- When a user enters a text message on his/her client, the message is delivered through the server and displayed by any other client that is currently connected to the server, including the original sending client.
- Users can join and leave the chat at any time, provided the server is up.

Features

For simplicity, the client can run in a single text-mode window. The following implementation features are required:

- When the user is typing, incoming messages must be buffered in order to avoid any overlap. They will be displayed later, once the user has entered his/her message.
- The overall system must be robust with respect to client crashes and/or blocked connections, i.e., any problem with a single client must not affect the other users.

Tasks

You are asked to:

1. Describe your algorithm design, showing what interactions occur between the clients and the server.

2. Develop a TCP-based Java implementation of both client and server.

3. Test your implementation and report on possible bugs and/or unexpected behaviors you should find.

4. Shortly discuss a possible alternative implementation exploiting the UDP protocol, making a critical comparison between the two.

Deliverables

Submit a single zip file ChatSystem.zip which contains

1. A 2–3 pages technical report in PDF format discussing items 1, 3 and 4 from above

2. Two classes Client.java and Server.java which implement the client and the server functionalities in their main method.

Submission: Friday, 15th of April, 10am via Moodle.