

# KTN1

Fredrik V. Aagaard & Håvard Storvold

March 2017

## Introduction

In this assignment we were asked to make a chat service where a clients can communicate with a server over TCP connection. All communication will be done using the JSON format.

## Client

The client is a 'stupid' client, and most of the logic is located in the server. The client connects to the server, gets user input, parses payload(eg. user input) to the server, receives a response from the server, and prints this info to the screen.

We want to be able to receive and send messages concurrently, but with one thread this would not be possible. To solve this we run `MessageReceiver` in its own thread.

## Server

The server is responsible for most of the logic. The server accepts one (or multiple) client connection(s) and keeps track of them in the list `users`. It also keeps track of the chat history in `history` so that new users can see the chathistory when they connect. When the server receives a payload from the client it handles it with the `handle()` function and sends a response back to the client.

We want the server to be able to handle multiple clients at once, so we run one thread for each client that is connected to the server.

## Note

The server's logic is not shown in the sequence diagram, only the connections and interfaces. All types of requests follow the sequence diagram.