

Mandatory Assignment 1 Dat18c Tech 3 og SWC 3

# Mandatory Assignment 1 - Chat system

(SWC3 + TECH2)

## Dat18C hand in date:

This is one of the compulsory assignments you need to hand in, and pass, to qualify for 3rd semester exam.

**Hand in date: 27? September 2019, latest at 8:25, on Fronter.**

**Presentation: 27? September 2019.**

**Hand in info: You must submit a link to your git repo.**

## Overview:

You should code a Chat system (client and server), with a chat client that can connect to a chat server.

You must use threads in client and/or in server. The client should at the start ask the user his/her chat-name and then send a join message to the server.

The server should accept clients to join the chat system, using a protocol specified below. When a client joins, the server should maintain and update a list of all active clients. The server will need to save for each client the user name, IP address and Port number.

If a new user tries to join with the same name as an already active user, then an error message should be sent back to client. Client can try again with a new name.

An active client can send user text message to the server that will just send a copy to all active clients in the client list.

The Client must send a "heartbeat alive" message once every minute to the Server. The server should (maybe with a specialized thread) check the active list, and delete clients that stop sending heartbeat messages. Maybe the active list should include last heartbeat time.

The Client must send a Quit message when it is closing.

Use a thread pool and limit the number of clients to e.g. 5.

Enable the system to log transaktions <<timestamp>> + <<request>> and <<timestamp>> + <<response>>

Requests not following the protocol should give an error response back to the client and of course log the event.

Mandatory Assignment 1 Dat18c Tech 3 og SWC 3

## Protocol between Chat server and client:

List of allowed messages (and their meaning):

**JOIN** <<user\_name>>, <<server\_ip>>:<<server\_port>>

From client to server.

The user name is given by the user. Username is max 12 chars long, only letters, digits, '-' and '\_' allowed.

**J\_OK**

From server to client.

Client is accepted.

**J\_ER** <<err\_code>>: <<err\_msg>>

From server to client.

Client not accepted. Duplicate username, unknown command, bad command or any other errors.

**DATA** <<user\_name>>: <<free text...>>

From client to server.

From server to all clients.

First part of message indicates from which user it is, the colon(:) indicates where the user message begins. Max 250 user characters.

**IMAV**

From client to server.

Client sends this heartbeat alive every 1 minute.

**QUIT**

From client to server.

Client is closing down and leaving the group.

**LIST** <<name1 name2 name3 ...>>

From server to client.

A list of all active user names is sent to all clients, each time the list at the server changes.

### Note:

This notation <<info>> indicates a placeholder, and they need to be replaced with appropriate content.

E.g.:

**JOIN** <<user\_name>>, <<server\_ip>>:<<server\_port>>

Might look like this:

**JOIN** alice\_92, 172.168.168.12:4578