Flow-2 CA - Chat-Server/Client

For this CA you must write a simple Chat-server and a corresponding (non-GUI) client.

It is expected that other companies will start to use this chat system and write their own clients (and servers) in languages/platforms other than Java, so ALL implementations must obey the protocol given below.

Commands from clients to the server		
Command	**Content**	**Example(s)**
CONNECT	Name of the client (use a hardcoded map to hold users)	CONNECT#LarsM
	CONNECT can only be followed by an ONLINE command from the server If a user is not found, the server will send CLOSE#2 and close the connection.	
SEND	Receiver followed by the message. (receiver must be a name received via an ONLINE command)	SEND#Peter#Hello Peter SEND#Peter,Hans#Hello Peter and Hans SEND#*#Hello everybody
	After a SEND command, the server can send an ONLINE, MESSAGE or a CLOSE command	
CLOSE	Nothing	CLOSE#
	After having sent a CLOSE command the client should discard all messages received until the server responds with a corresponding CLOSE command on which the client should close the connection.	

Commands from Server to client(s)		
Command	**Content**	**Example**
ONLINE	Name of all clients, currently online. • The server should send this message to **all** clients, each time a client has connected or disconnected) • The list should include the name of the specific client that receives this message	ONLINE#LarsM,Peter,Hans
MESSAGE	Sender followed by the message	MESSAGE#Peter#Hello Hans
CLOSE	**0** for a normal close **1** illegal input was received **2** User not found	**CLOSE#0** or **CLOSE#1** or **CLOSE#2**
	Having sent a CLOSE command the server should close the connection and release all resources attached to that client.	

Every message is sent as a string, including both the command and content.

Requirements

- The protocol must follow the rules given above and be 100% (unit) tested.
- (The server must include a log-file to hold important run-time information)
- The server and the client must be written in Java. Your server must be designed to handle many clients simultaneous
- The system must include a simple non-GUI client

Getting Started with this exercise

We recommend that you clone this very simple maven project

https://github.com/cph-dat-sem2/chat-server-startcode

and use it as a start. It doesn't provide much Java (almost nothing) but it will make the final deployment to your Droplet a "piece of cake2" :-)

See the link in the README-file for information related to how you deploy to a droplet.

Daily tasks for this exercise

Daily log for Monday to Thursday (don't forget this part if you want your studypoints)

You must create a log-document, readable by us in some way (Google doc, git, or whatever) and send the link to your teacher (Daniel, Lars or Thorbjørn) no later than Friday 05-03 (in week 2).
This document must CLEARLY state your GROUP NAME and the NAMES and STUDENT-ID's of each member.

By the end of each day (Monday - Thursday) you must update this document with a small description including:

- How far you have come
- Problems you have had
- Who did what
- Do you need help

For the last day it should include the final status of your project, and a description of how you have tested your system, link to github, link to your deployed server etc.

This documentation can be in Danish or English.

Demonstration and what to hand in for the Chat System

We will add a few more details here regarding the group presentation Friday 12-03.

- The server must be demonstrated using your own client + a client from at **least of other team**
- Your client must be demonstrated up against your own server, + a server from **at least one other team**.
- The code must be made available via GIThub.
- Your Repo must include a short description including (either in your README-file, or in a document linked to from the readme-file):
  - A short design description of the chosen design.
  - A section stating who did what, backed up by your git-logs
  - A Description of the results of your acceptance tests with other groups
- The project must deployed and accessible on a droplet

Note: This Course Assignment will form the background for one or more examination questions. See the list of learning goals/exam-questions for period 2

Study-points for this Period and the CA

You can get a maximum of 20 studypoints for this project