

## Multi-threaded client/server chat system

### Beskrivelse af funktionalitet

#### Server

Serveren lytter på en ServerSocket i main tråden hvor der accepteres forbindelser fra clients. Hver client der forbinder får tildelt en tråd på serveren, som er tildelt af en threadpool, til at behandle data der kommer ind. Til at håndtere data på tværs af tråde, bruger serveren to singleton objekter; en der holder styr på alle chat beskeder fra clients der skal sendes til alle andre, og en der holder styr på hvilke clients der er forbundet, og er aktive.

For at behandle dette, er der udover main tråden og threadpool'en, en 'watcher' tråd, som har til ansvar at broadcaste chat beskeder, og holder styr på hvilke clients der stadig er aktive. Dem der ikke længere er aktive fjernes automatisk af 'watcher' tråden. 'Watcher' tråden har også til ansvar, at informere om hvilke clients der er forbundet, til alle clients (LIST delen).

Serveren accepterer forbindelser, og inkommende data, via TCP. Men, til at broadcaste chat beskeder og listen af aktive clients, benyttes der UDP med datagram og datasocket. Af denne grund gemmes hvert clients ip adresse og port nummer, og username.

#### Client

En client har en main tråd hvor der skal indtastes en JOIN meddelse der indeholder username, server ip adresse, og server port nummer som man ønsker at forbinde til. Hvis forbindelsen bliver godkendt af serveren, oprettes to daemon tråde.

Den ene tråd sender automatisk en 'IMAV' besked hvert minut for at holde forbindelsen i live. Den anden tråd opretter en datagram socket til at lytte på. Denne tråd tjekker om der er data fra serveren at behandle. Main tråden tager herfra imod bruger input, eksempel at sende en chat besked.

Da et username har en begrænsning på kun at måtte indeholde almindelige bogstaver, tal, '-' og '\_', tjekkes et ønsket username med en regular expression (regex), om hvorvidt der er ikke godkendte karakterer som indgår.