Server overvejelser og læring:

På min server oprettede jeg en ClientHandler klasse som implementerede Runnable interface. Server klassen afventer klienter opretter forbindelse og tilføjer dem til min liste. Jeg startede ud med at have en liste der indeholdt Threads, men på den måde kunne jeg ikke tilgå objekterne af ClientHandler fra andre steder. Så derfor lavede jeg en List<ClientHandler> i stedet og så startede thread.

ClientHandler klassen bliver ret stor så jeg overvejede at lave en overklasse der hed ”Protokol” som ClientHandler kunne nedarve fra. Så langt kom jeg dog aldrig.

På min server har jeg forsøgt og lave en ClientList klasse som følger pattern Singleton som er synchronized så kun en thread kan tilgå denne klasse af gangen. Efter jeg havde lavet min ClientList klasse stoppede min thread cleanActiveClientsList med at virke. Jeg tænker at der er en thread som bliver ved med at holde på klassen og derved kan ovenstående thread ikke tilgå denne.

Jeg fik aldrig lavet min log, men havde planlagt og bruge Logger klassen.

Client overvejelser og læring:

I mit program er det klienten der verificere brugerinputtet på JOIN og DATA, men der burde også være en eller anden verificering på serveren. Jeg har verificeret disse med regex på klienten.

Oprettet separat Thread til heartbeat. Hvis server ikke modtager heartbeat lukkes socket og brugeren fjernes fra listen.(virker dog ikke helt).

Når brugeren skriver JOIN message udledes brugernavn af denne og det sendes til serveren. Jeg synes ikke det var relevant og sende serverip:port med da det kun skal bruges på klientsiden til forbindelsen.