

# TTM4100

## Chat-prosjekt

### Gruppe 82

Rubin Gerritsen

Simen Haugo

Erik Liland

Kristian Roaldsnes

Øyvind Wilson

### Innhold

<b>1</b>	<b>Beskrivelse</b>	<b>1</b>
1.1	Generelt . . . . .	1
1.2	Klient . . . . .	1
1.3	Server . . . . .	1
<b>2</b>	<b>Klassediagram</b>	<b>2</b>
<b>3</b>	<b>Sekvensdiagram</b>	<b>3</b>

# 1 Beskrivelse

## 1.1 Generelt

Vi har valgt å implementere dette prosjektet i Go, da dette er et språk som har en god idiomatisk støtte for flere-tråd applikasjoner. Kommunikasjonen mellom trådene er basert på kanaler, som gjør deling av data mellom flere tråder enkelt og sikkert. Figur (1) viser hovedoppdelingen av prosjektet. Figur (2) og (3) viser hvordan meldingsflyten typisk vil se ut på klient- og server-siden.

## 1.2 Klient

Klienten kommer til å bestå av tre tråder:

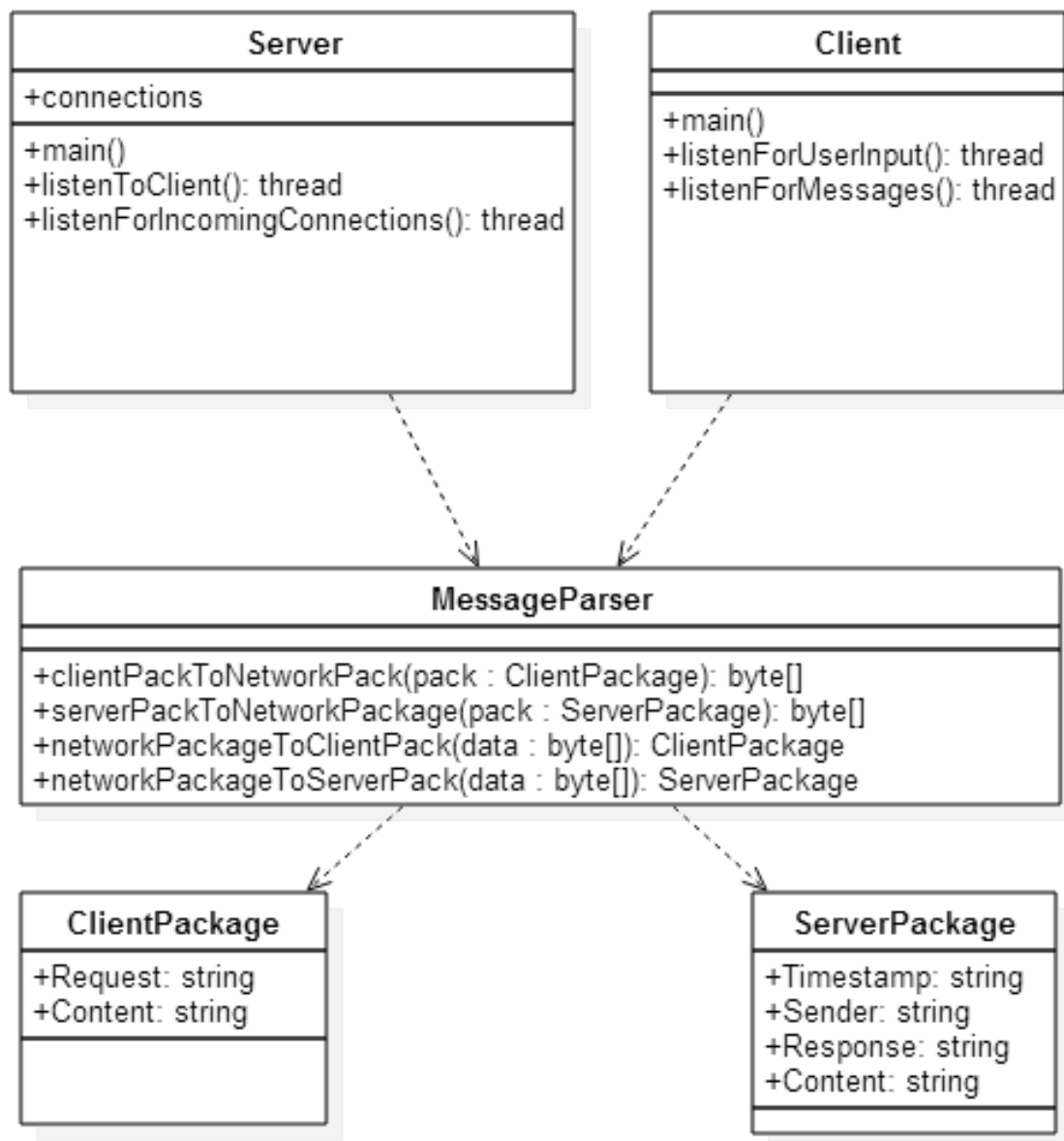
1. (main) Kjører initialisering av kommunikasjonskanaler mellom trådene, oppretter TCP-forbindelse med serveren og starter de to andre trådene. Tar imot nettverkspakker fra serveren og håndterer disse, tar imot input fra brukeren (gjennom en kanal) og sender dette videre til serveren.
2. (nettverkslytter) Blir startet med en opprettet TCP-socket og en kommunikasjonskanal som argumenter. Kjører loop på lesing av socketen, og returnerer ferdig dekodete pakker til "main" gjennom en dedikert kanal.
3. (bruker-input-lytter) Blir startet med en kanal som argument, og bruker denne kanalen til å returnere strenger som brukeren gir gjennom tastaturet.

## 1.3 Server

Serveren kommer til å bestå av 2 tråder + 1 tråd per bruker:

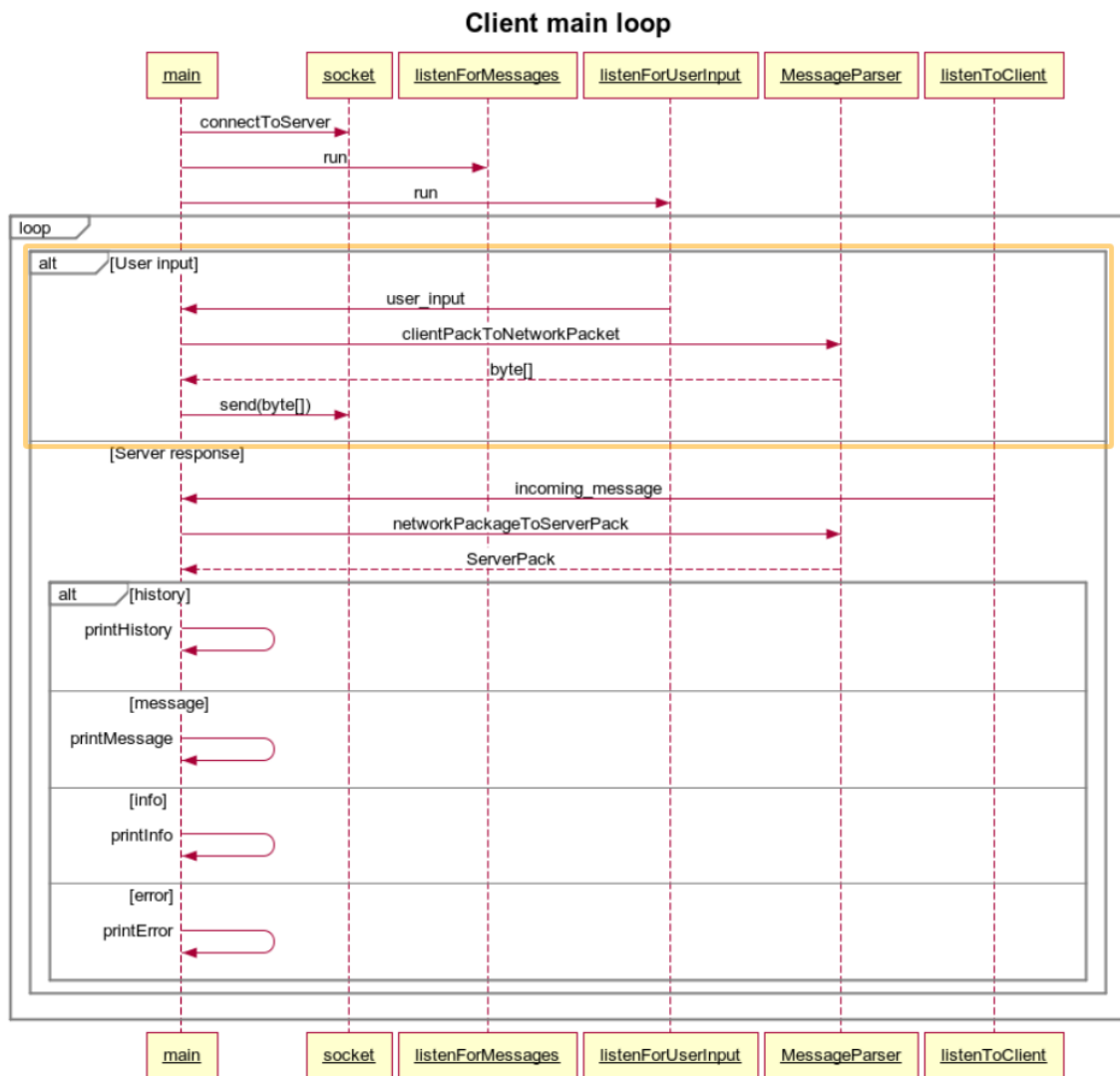
1. (main) Kjører initialisering av kommunikasjonskanaler og starter en ny tråd som tar seg av TCP socket og innkommende forbindelser. Etter initialiseringen tar main seg av all server-logikken:
  - (a) For hver nye TCP-forbindelse som tråd2 får inn, starter main en ny tråd som tar seg av meldinger fra denne brukeren.
  - (b) For hver melding som kommer fra en bruker, returneres respons etter spesifikasjonen.
2. (TCP-lytter) Oppretter en TCP-socket som tråden lytter på. Kjører loop: Når en ny klient ønsker å koble seg på, aksepteres denne tilkoblingen og den nye socketen blir sendt til main for videre håndtering.
3. (Bruker-tråd) Opprettes en tråd per bruker som leser socketen til denne brukeren, og returnerer en dekodet pakke til main når den får inn meldinger.

## 2 Klassediagram

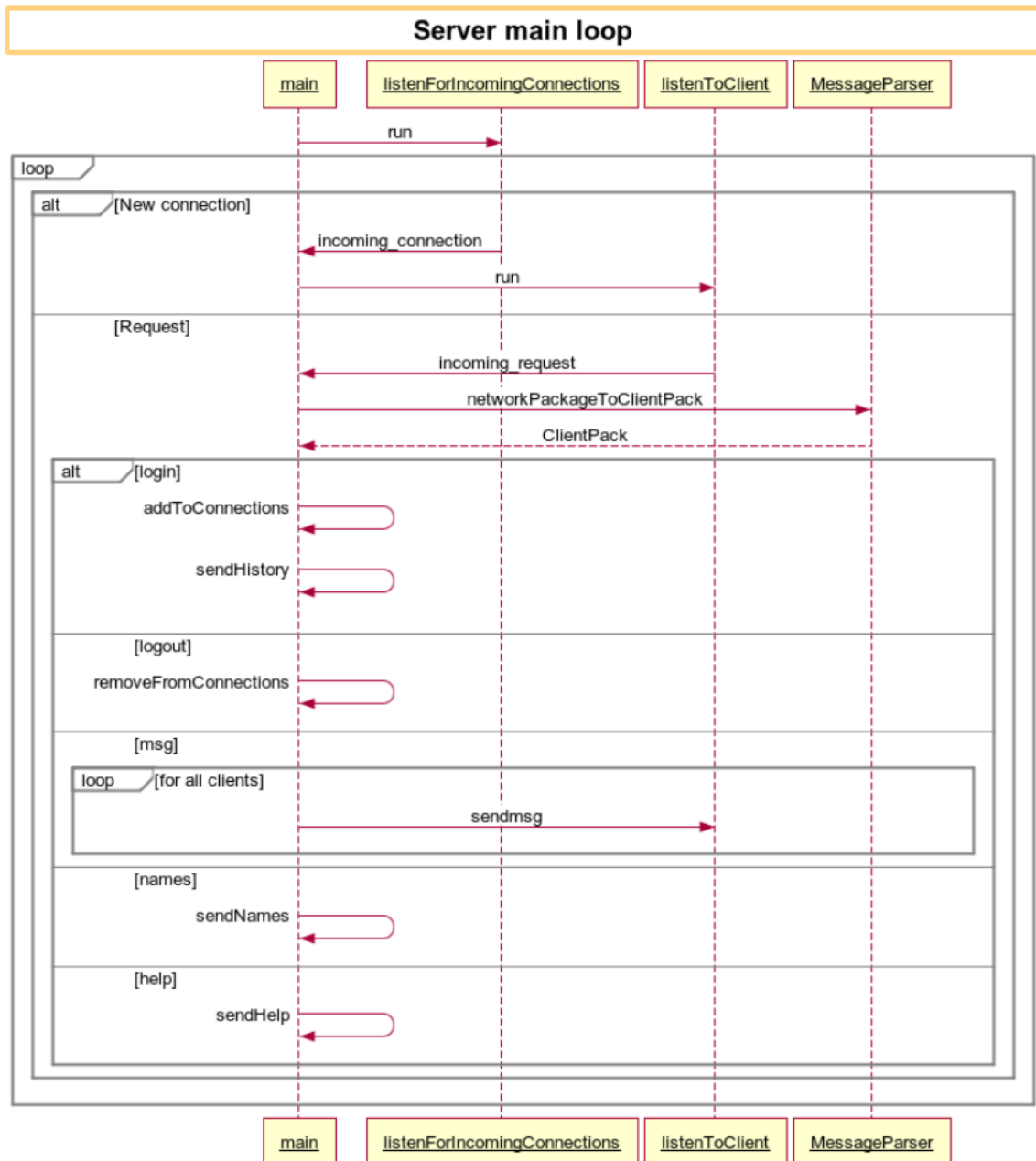


Figur 1: Hovedoppdeling av prosjekt

### 3 Sekvensdiagram



Figur 2: Sekvensdiagram for klient



Figur 3: Sekvensdiagram for server