KTN1, TTM4100

Audun Vigdissønn Nytrø (759535, audunvn@stud.ntnu.no)

Spring 2017

Contents

List of Figures

1 System design concept

1.1 Description

My plan for the assignment is essentially to write a Python-based server and client, where the client implements both a bare-bones chat client plus a web server for providing a browser-based UI to the user. I initially considered writing the entire client in JavaScript using WebSockets, but this would unfortunately make the client incompatible with the specified standards as WebSockets requires a few special (and non-standard) responses and headers to initialize the connection, so having a simple Python chat client store the client state and handle messages seemed like the easiest solution.

The chat client (and web server) thus continously runs in the background and handles client state to preserve it between browser refreshes and restarts, while the web client continously polls the chat client for new responses or data. A polling method must unfortunately be used as JavaScript has no provision for receiving responses without first doing a request using AJAX or similar methods outside the aforementioned unusable WebSockets API.

The chat server itself will be a simple and pretty straightforward implementation directly based upon the framework code that was handed out with the assignment.

A possible class diagram for the proposed system may be found in figure 1.1, while a sequence diagram showing an user logging in, sending a message and logging out may be found in figure 1.2.

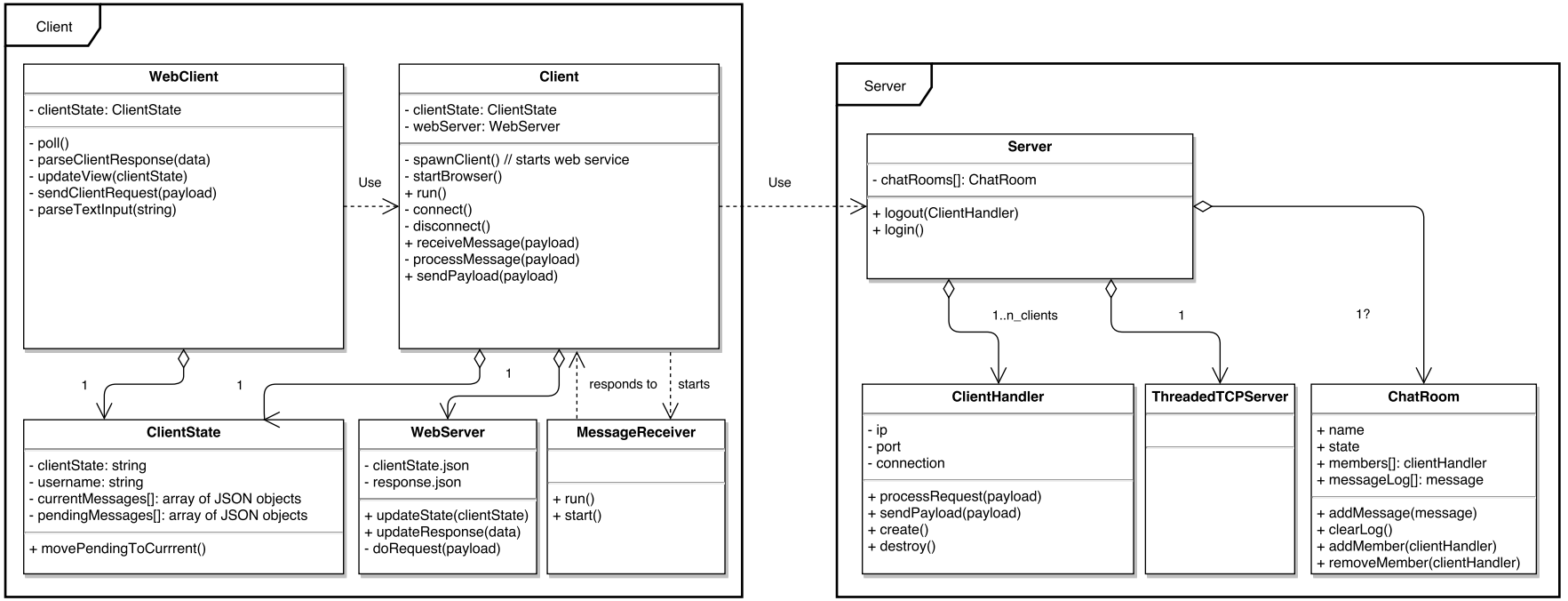

Figure 1.1: Class diagram for the chat system

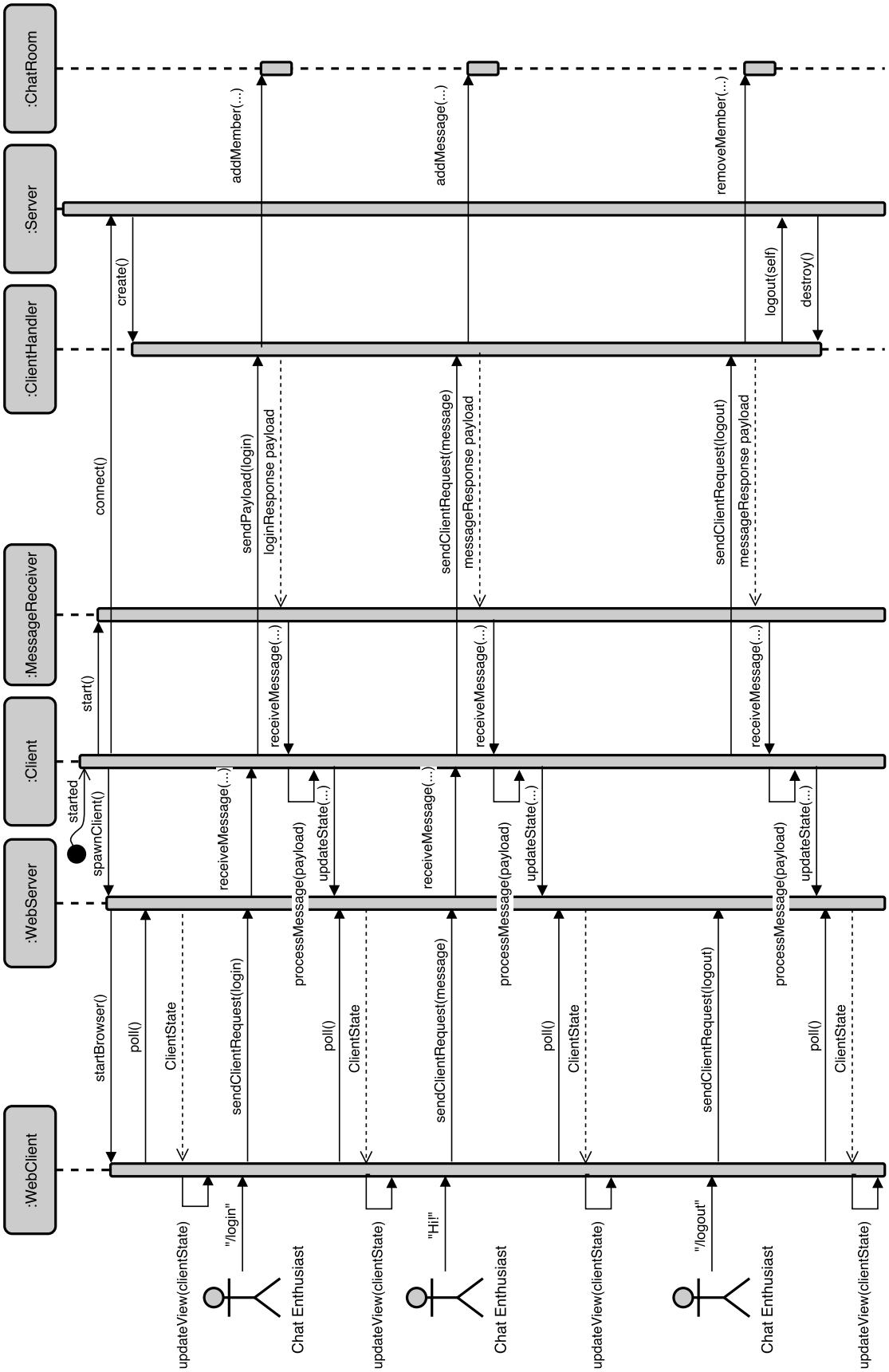

Figure 1.2: Sequence diagram for the chat system