

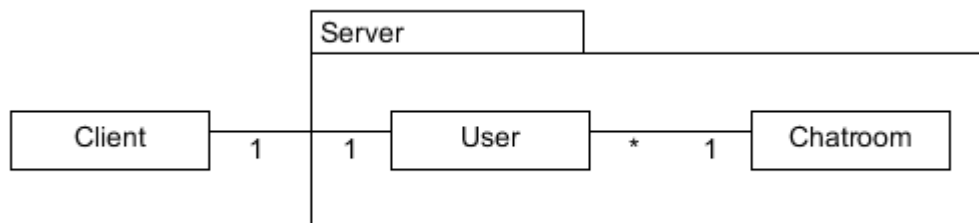
Chat Server

Task:

The task is to set up a chat server offering the following:

- login, providing a username
- chronological display of messages, containing name and timestamp
- login and logout notifications
- a \list command, listing the names of all active users
- private messages
- sending multi-media files

Component Diagram



Each client creates a socket on the serverside providing access and identification.

Each user is connected to the chatroom, that is shared by all users.

Mockup

The mockup is drawn on graph paper and consists of two main sections connected by a downward-pointing arrow.

Top Section (Login Form):

- A rectangular box contains the text "Name:".
- Below "Name:" is a rectangular input field containing the text "Username".
- Below the input field is a rectangular button labeled "Enter".

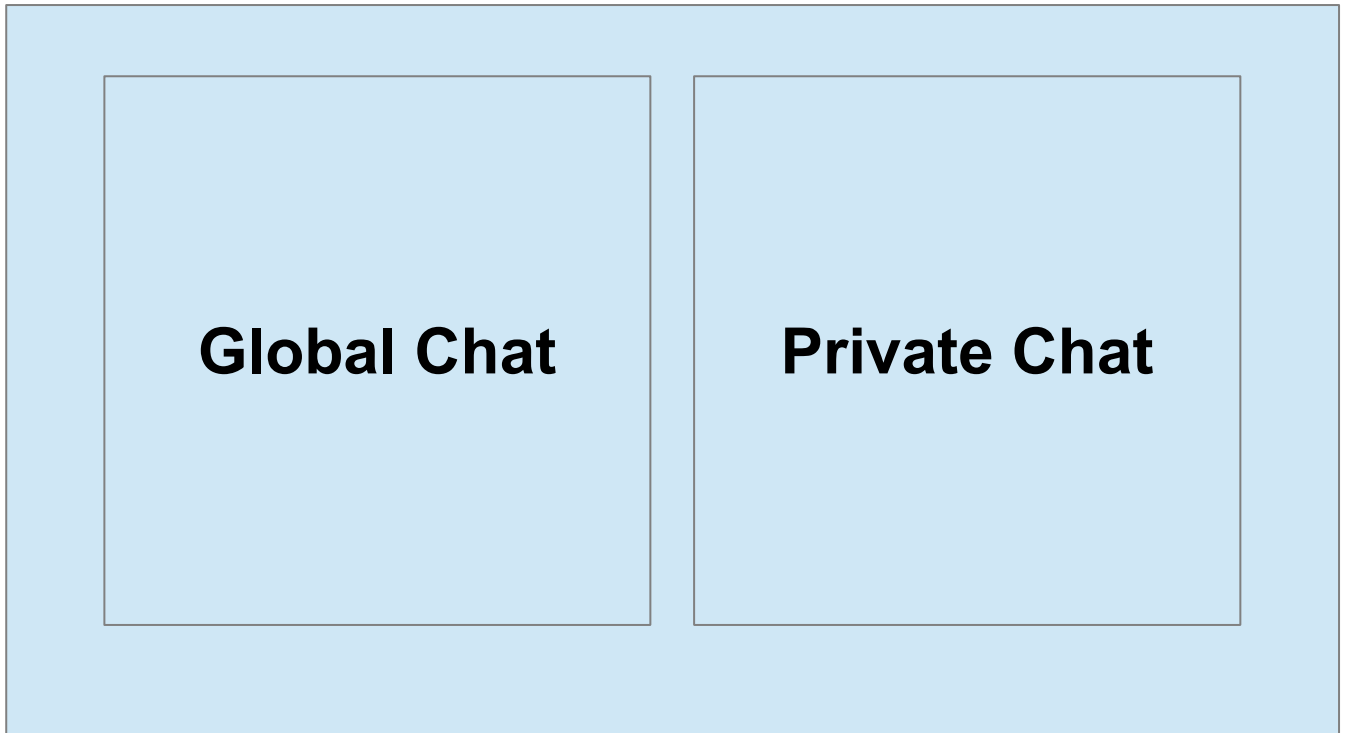
Bottom Section (Chat Interface):

- A larger rectangular box represents the chat window.
- At the top left of the chat window is the text "Page Name".
- At the top right of the chat window is a rectangular input field containing the text "Username".
- Inside the chat window, there is a list of three messages, each preceded by the text "Time Name:":
 - Time Name: message 1
 - Time Name: message 2
 - Time Name: message 3
- At the bottom of the chat window is a rectangular input field containing the text "message".
- To the right of the input field is a rectangular button labeled "send".

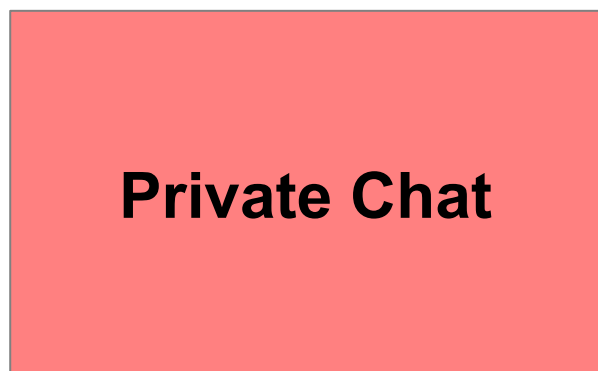
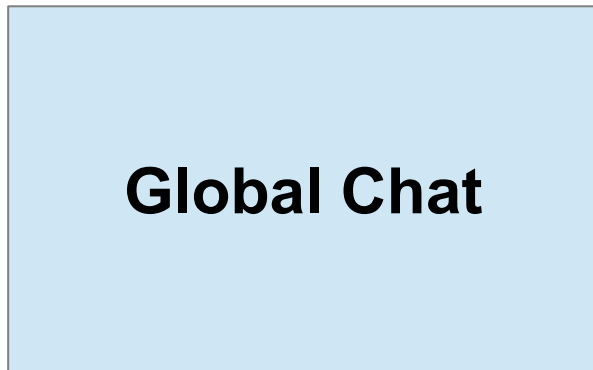
Simple and based on SocketIO example

Mockup - Scrapped Ideas

Multiple Chats Side-by-Side:

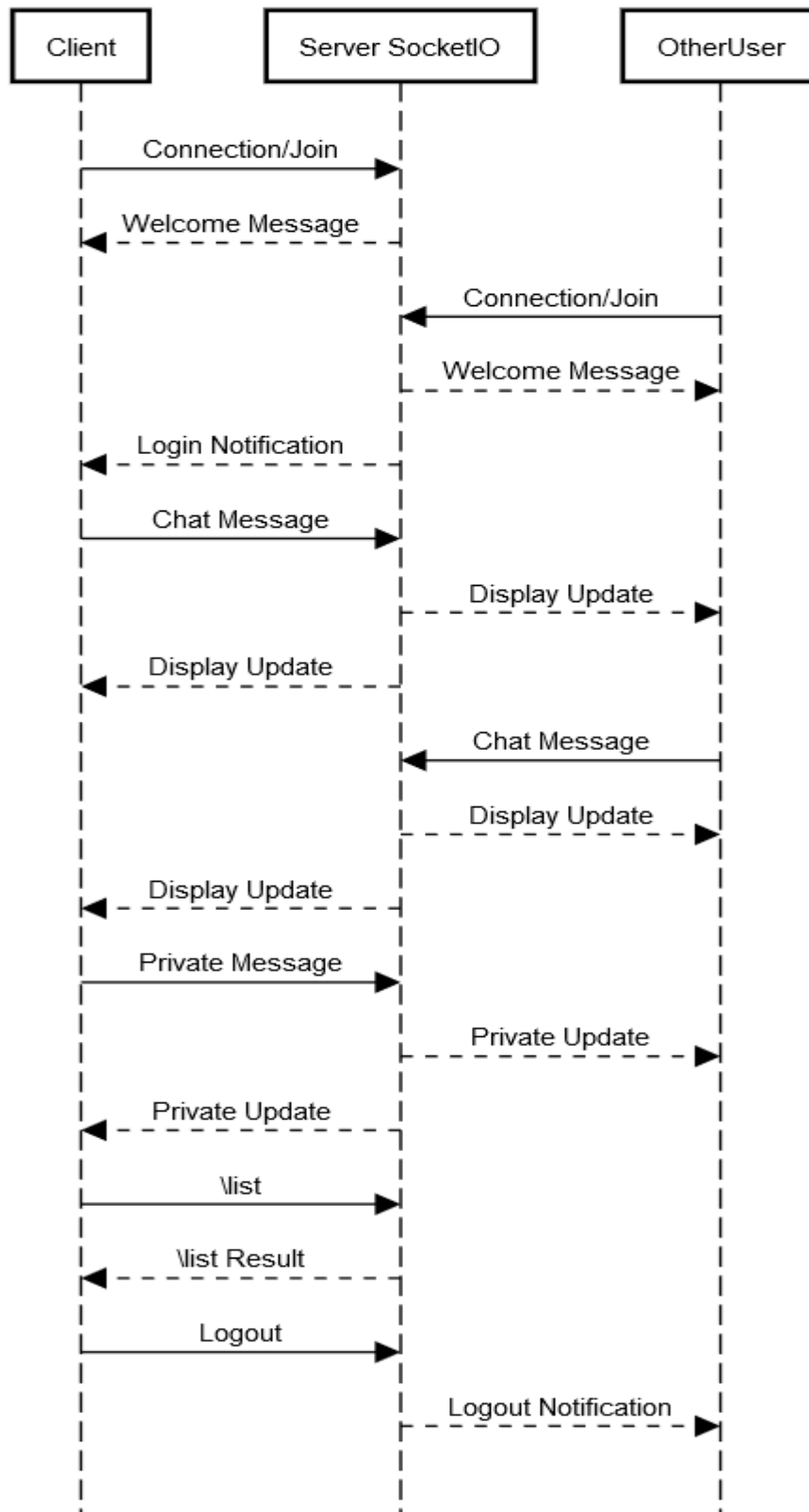


Different colors for different chats:



Sequenzdiagram

Chat Server SocketIO



Lesson-Learnt

- usefull and easy to use modules
- a lot of function already given

Todo for next exercise:

- media messages
- server-side username validation