Chat Room Design

Yasser Afifi
ID: 14303154

1- Introduction

This design is based on the client-server model. The server forms a central point for clients to connect to, performing the required message delivery/multiplexing and other functions.

2- Components

The following section defines the basic components of the design.

2.1 Servers

The server is the backbone of this design as it is the only component of the design which is able to link all the other components together. It provides a point to which clients may connect to talk to each other. Also, a point where other servers can connect to access a service provided by this server. Depending on the number of users, many servers can be combined to be able to support a large number of requests.

2.2 Clients

A client is anything connecting to a server that’s not another server itself. In the context of this design, a client is an application with a text based interface that is used to communicate interactively with the server.

2.3 OAuth

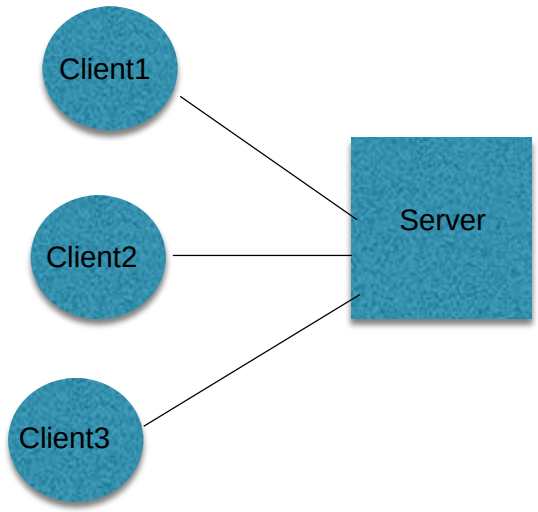
OAuth is a protocol that allows third-party to get a limited access to users’ resources without sharing their passwords. It is a popular way to authenticate users without the overhead

of creating new accounts. In the context of this design, Google's OAuth will be used to authenticate users.

3- Architecture

The architecture of this design is defined by a group of servers connected to each other. A single server forms the simplest possible case of this design. However, as the number of users and the number of services grow, more servers will be required.
It is important to note that this design provides no mean for two clients to directly communicate. All communication between clients is relayed by the server(s).

Picture of the simplest version possible of the design



4- Server Design

This section is devoted to describing the design of the server and how different classes of messages between clients and the server are delivered. There are 2 types of communication that are supported by this design; One to One communication and one to many communication.

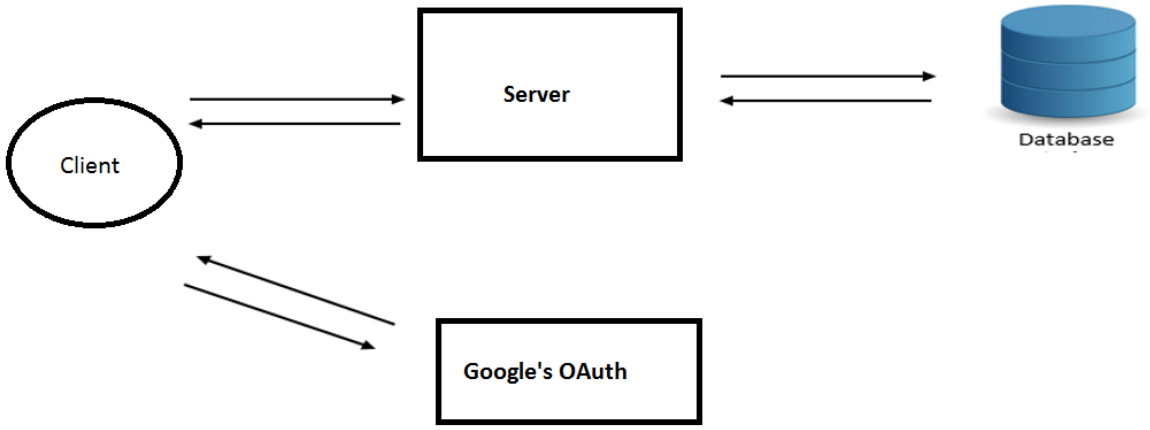
4.1- One to One Communication

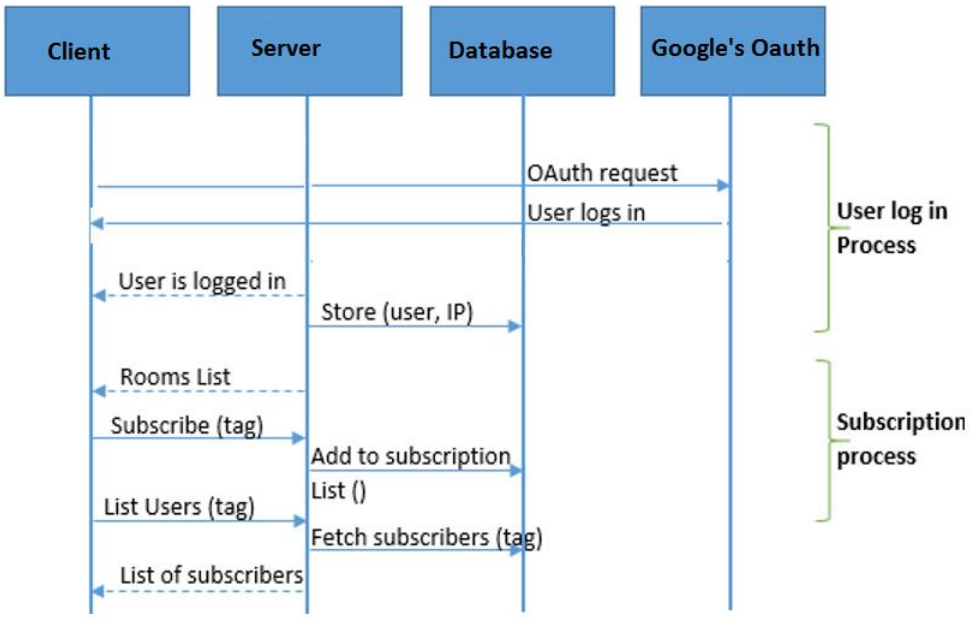
One to one communication between clients (also known as private messaging). This kind of communication is started by clients, and the server must be able to send the message in ONLY one direction. For example if Client1 from the above picture wants to send a message to client2, it sends the message to the server, the server sees the message and send it directly to Client2.

4.2- One to Many Communication

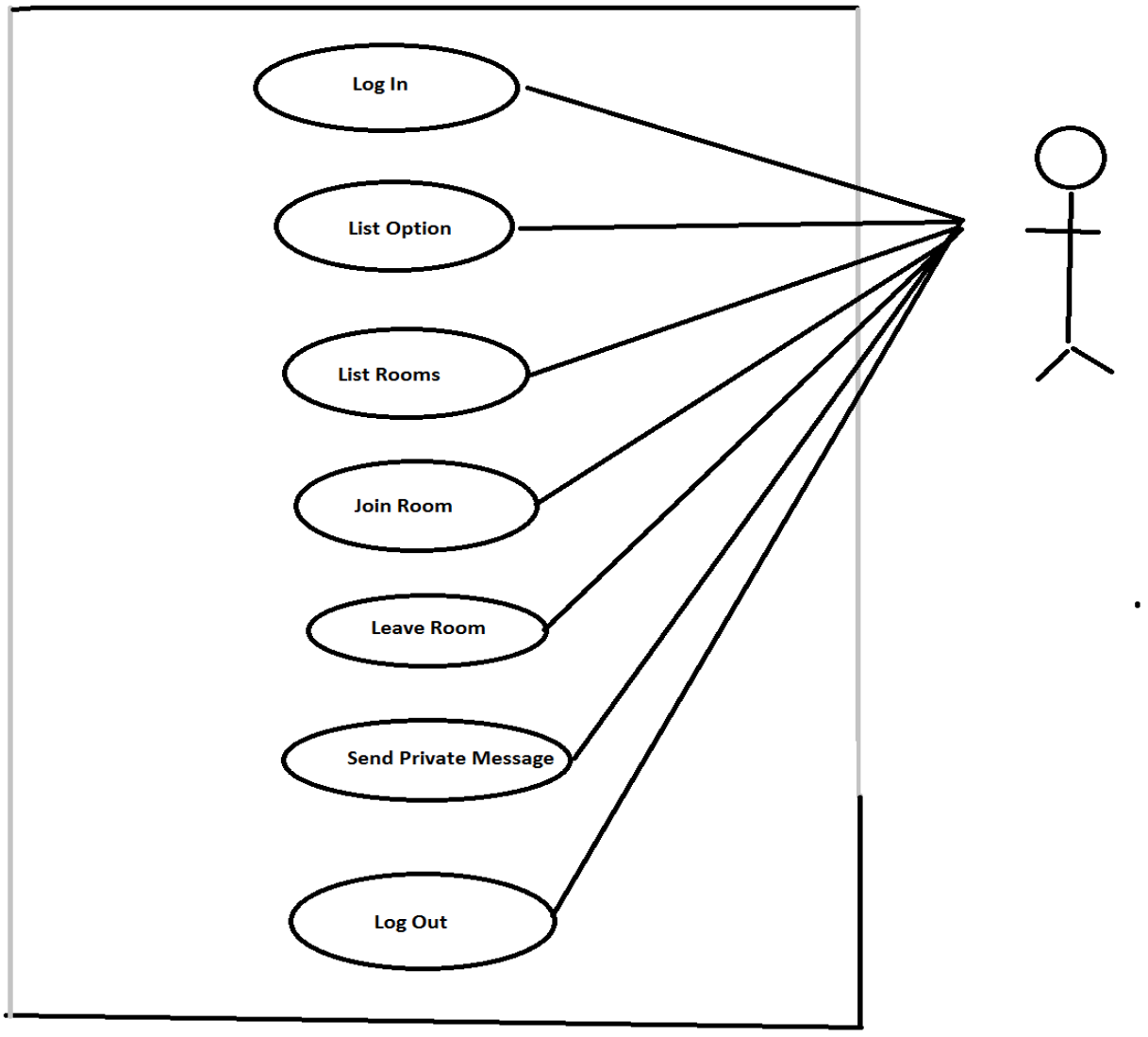
One to many communication (group messaging) is also supported in this design by the notion of tags. Tags can be thought of as a repository inside the server that clients can join (subscribe to). Any message directed to a certain tag is forwarded by the server to all the clients inside that tag. Thus the notion of group messaging would be possible.

5- More Sophisticated Architecture




Client
Server
Database
Google's Oauth
OAuth request
User logs in
User log in Process
User is logged in
Store (user, IP)
Rooms List
Subscribe (tag)
Add to subscription List ()
List Users (tag)
Fetch subscribers (tag)
List of subscribers
Subscription process

At the beginning, the user has to log in using the google's Oauth, then the user will be interfaced with a list of options such as chat room list, private chat, or looking at chat history. A user can also join or leave rooms.



Evaluation

This design is good and scalable, the server doesn't need to store information about users and who is subscribed to what rooms. This is why it is a good and scalable design that is able to handle a lot of requests