Assignment 2a (*Networking*)

Oliver Makins MKNOLI003, Masixole Ntshinga NTSMAS016, Daniel Schwartz SCHDAN037

April 7, 2017

1 Summary

The web chat application we designed for this assignment implements features that most chat applications use today. It is a simple chat room that supports multiple users, which each have their own username, or nickname. It allows for users to transmit messages over the chat application through the chat server. It records the times and who sent each message.

The chat application runs off of a server which each user client connects to. As a user connects it notifies the other users that that person has connected. And the same for disconnection of users.

2 Features

2.1 Multi-user messaging

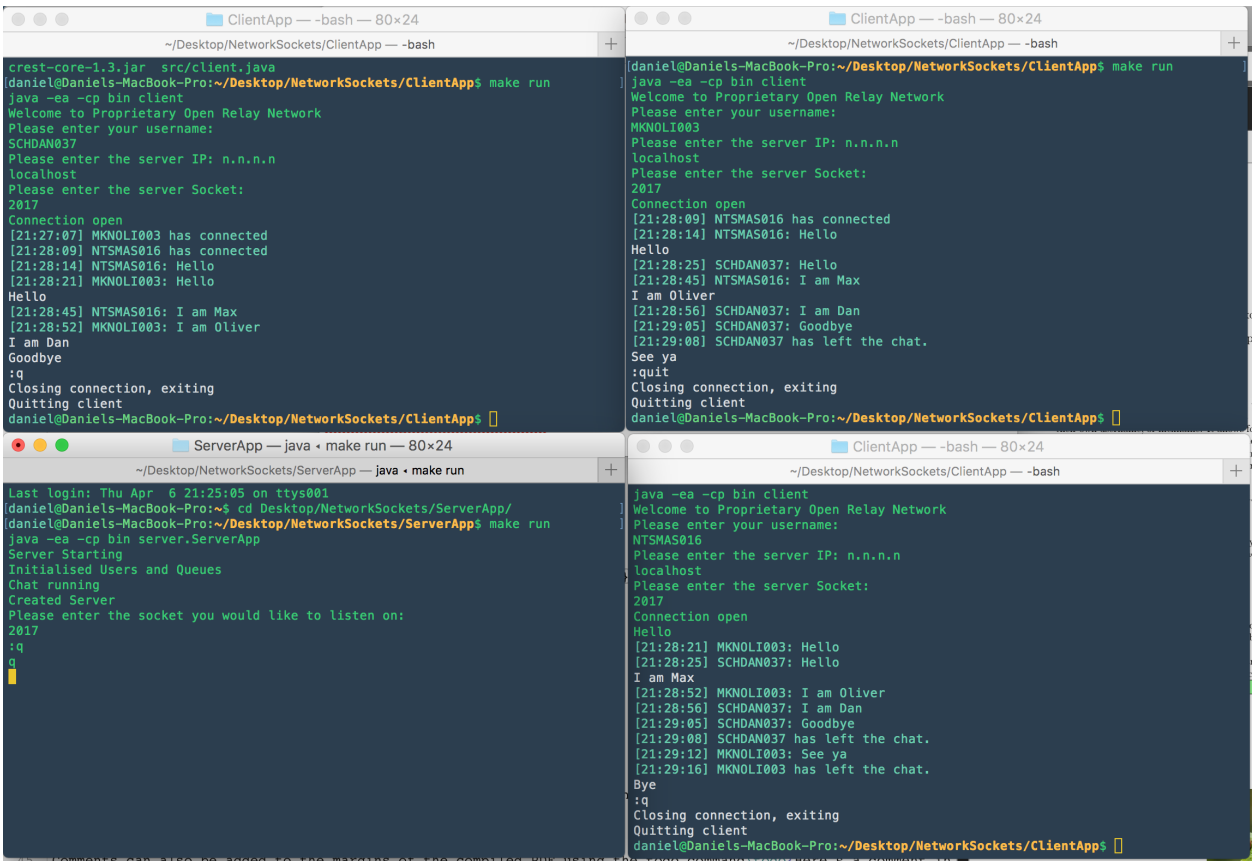

Figure 1: 4 terminal windows showing the server and 3 clients working

As one can see in Figure 1, the server can connect multiple users at the same time and one can see when, and who is sending each message as they come through. To connect: first the ServerApp must be initialized –

```
make run
```

– and a port chosen to open up to listen for clients to connect on. As the ClientApps are initialized – again using the makefile – the user chooses a username, the server IP (n.n.n.n format or localhost) and the port number that the server is listening on. Then as seen in Figure 1, The users can message each other and have all the messages sent to all clients. To quit a user needs only type either

```
:q
or
:quit
```

The server and all clients will be notified that the user is disconnecting and the client will close those ports and data streams and quit the process.

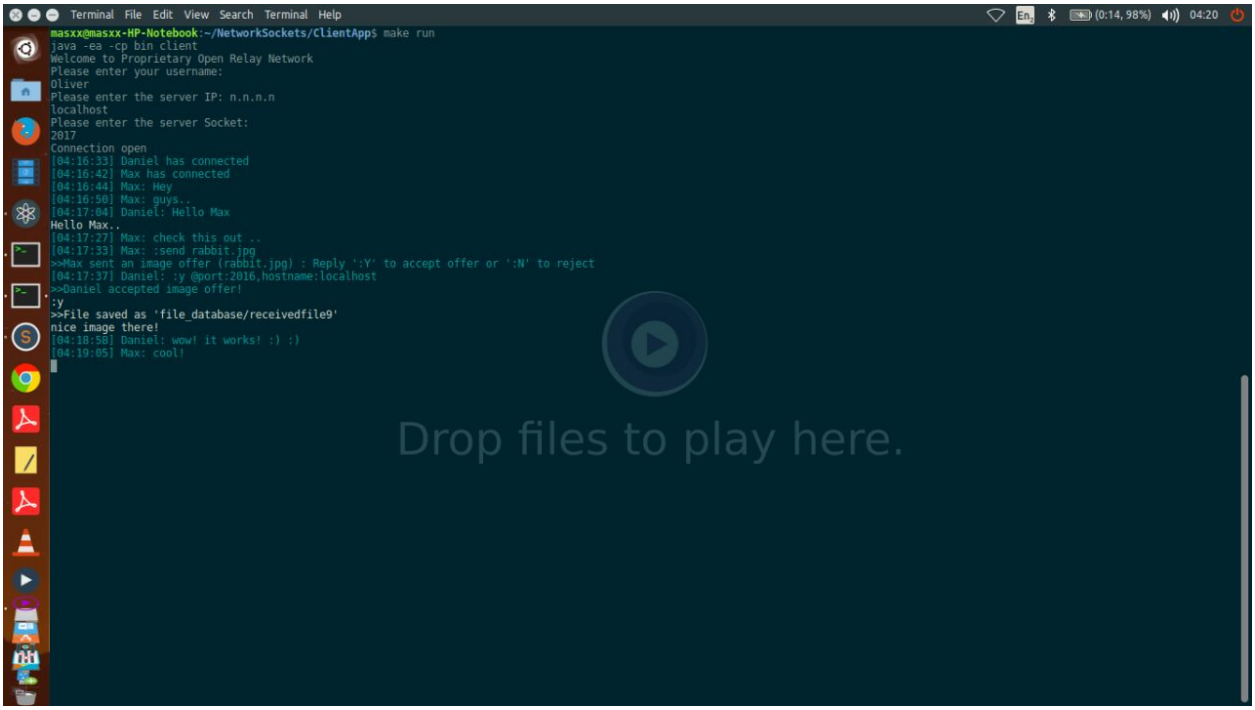

Figure 2: Terminal showing client user named Oliver

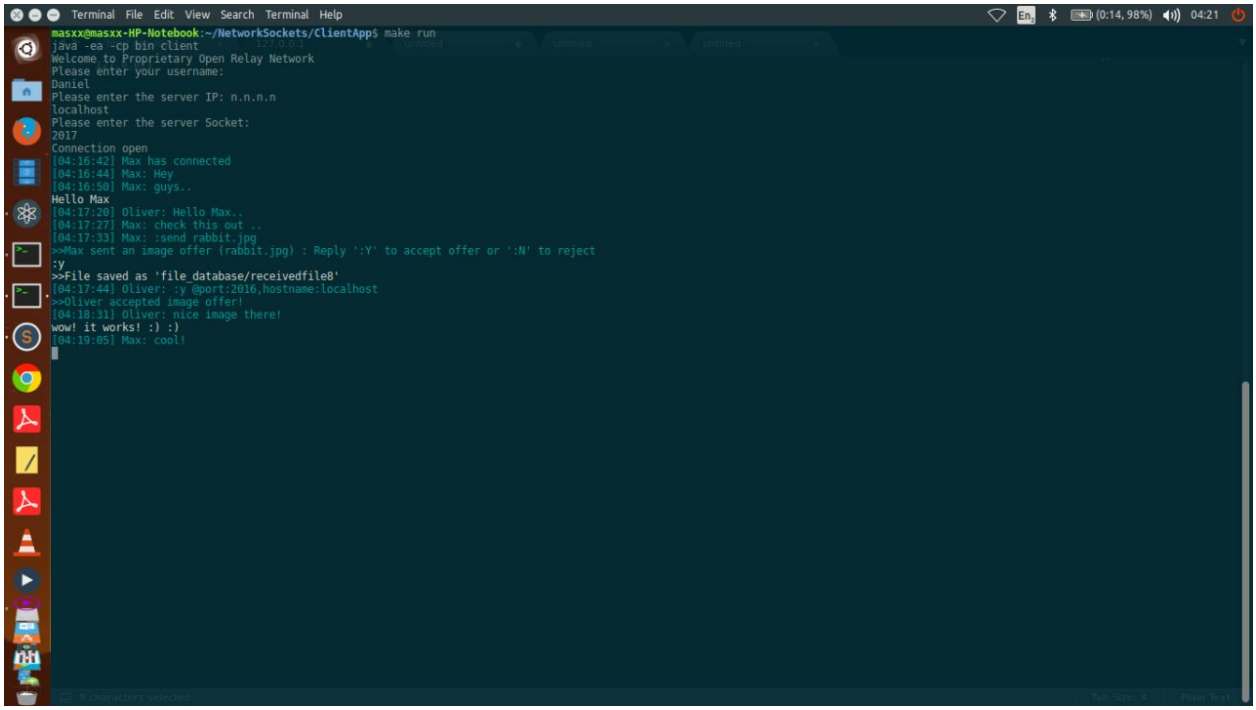

Figure 3: Terminal showing client user named Daniel

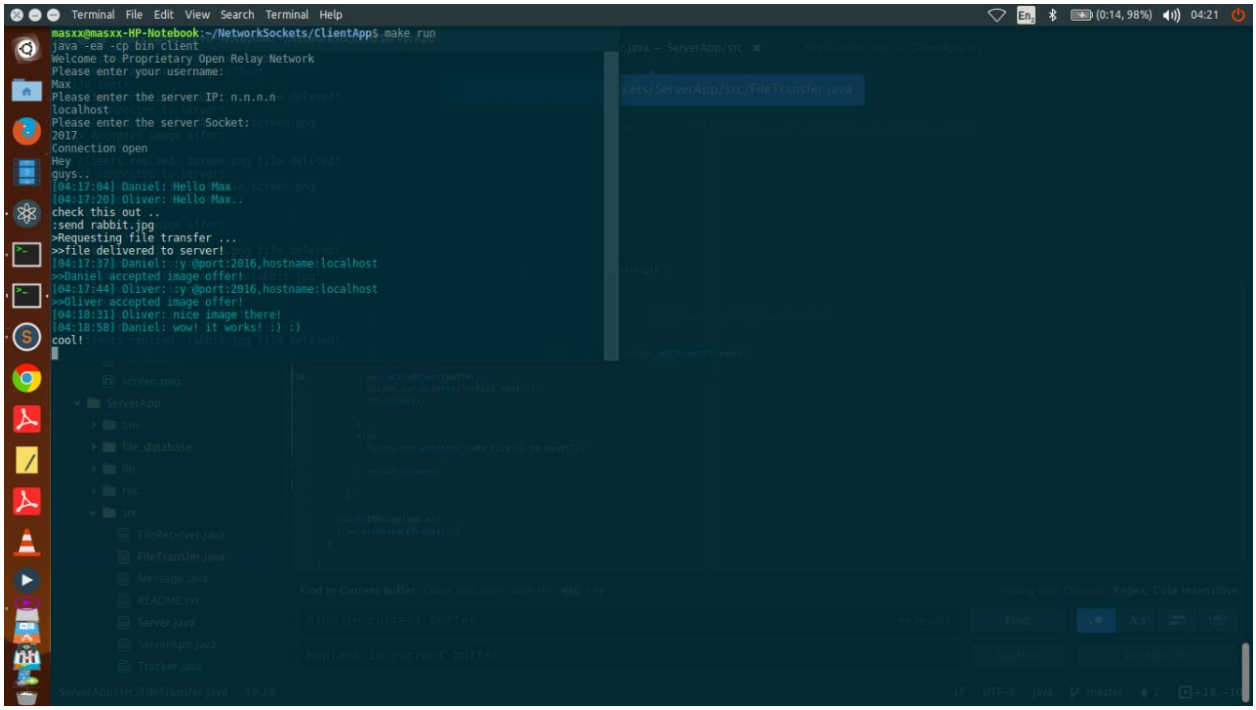

Figure 4: Terminal showing client user named Max

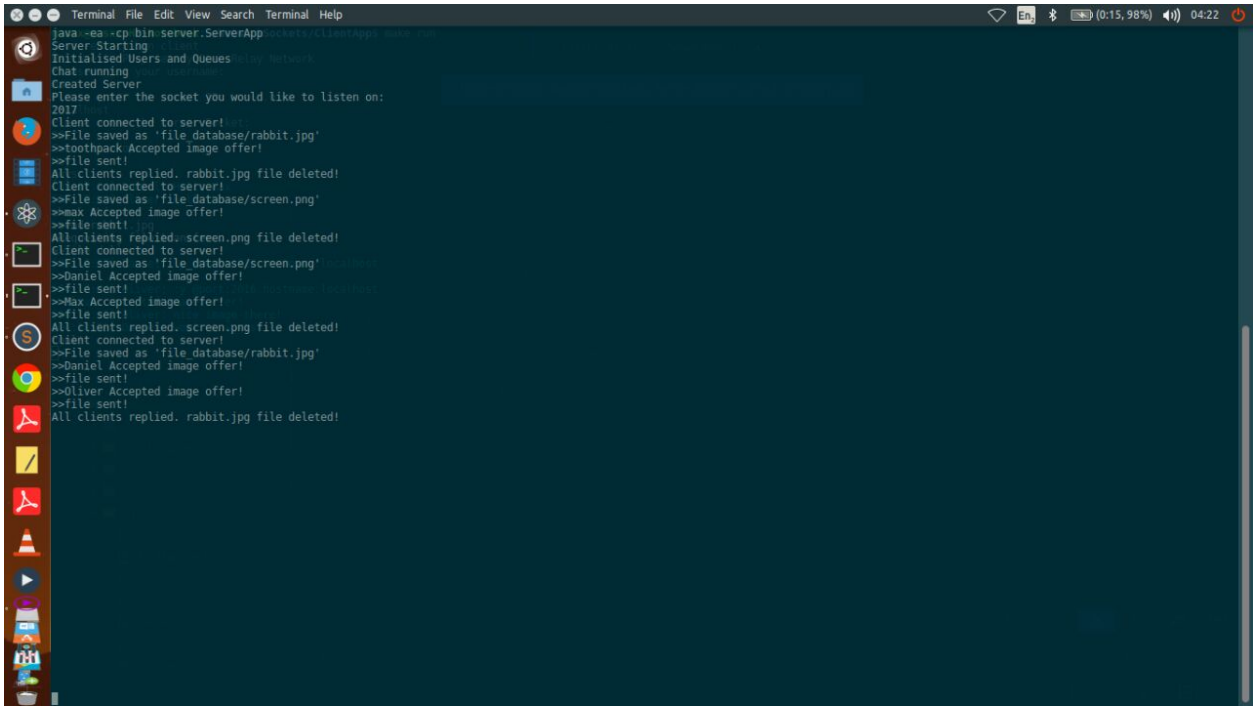

Figure 5: Terminal showing the Server running

2.2 Image transferring

The applications also accommodated for image transferring between users. The user could write a command,

```
:send <filename>
```

to send a file to the server. That image file will then be sent to the server over the network completely. Then the server will notify all the users that there is an image available to be downloaded. The users may accept or decline the image using the command ':Y' or ':N', respectively.

Once all the users either accept or decline the image, except the client that sent the image, the image is deleted off the server. In other words, for N clients, once the number of answered clients is N-1 it deletes the image off the server. The Server keeps a log of the image transfer stages. This approach was chosen to keep the server lightweight and maximise user privacy by mitigating the risks associated with retaining data.