

# Assignment 2a (*Networking*)

Oliver Makins MKNOLI003, Masixole Ntshinga NTSMAS016, Daniel Schwartz SCHDAN037

April 6, 2017

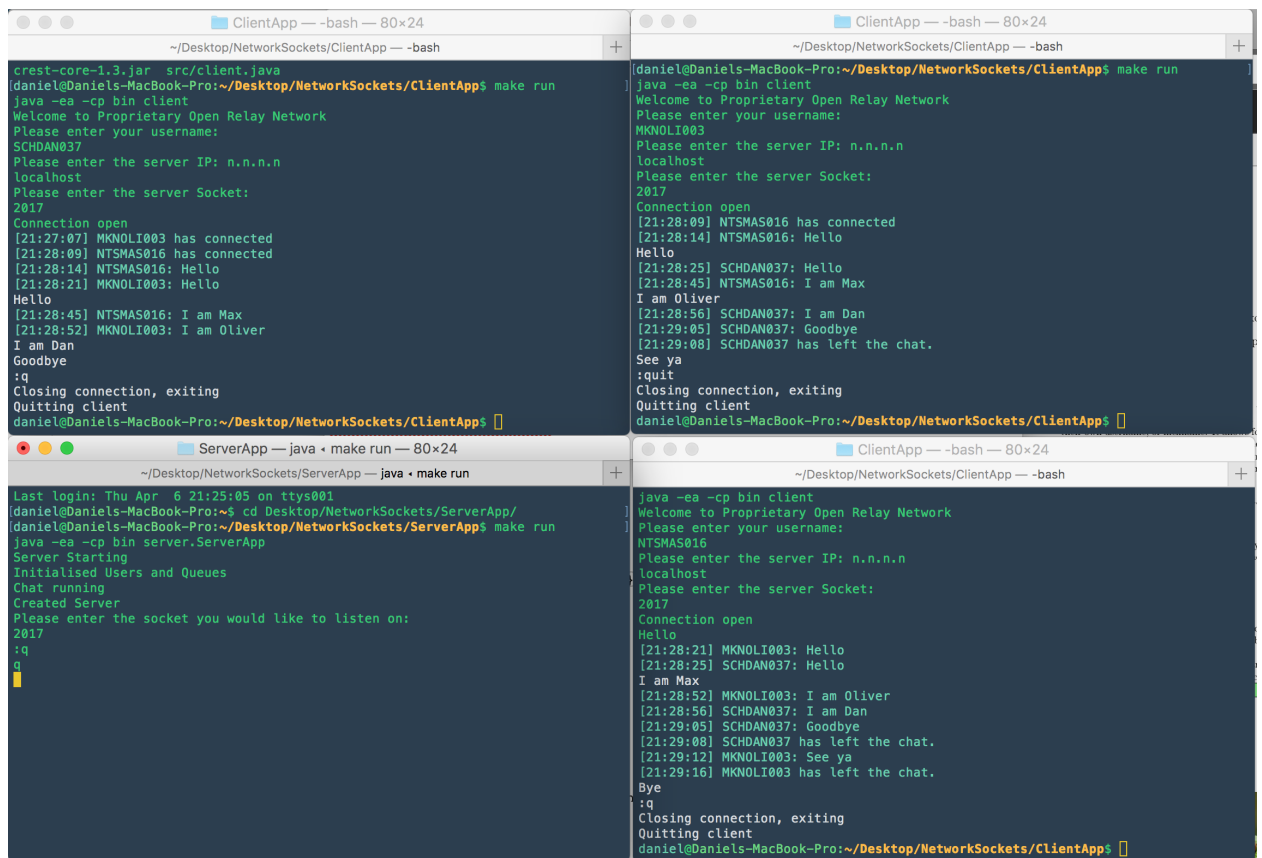
## 1 Summary

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



The figure displays four terminal windows arranged in a 2x2 grid, illustrating the operation of a multi-user chat application. The top-left window shows the 'ClientApp' running on a Mac, where a user enters their username 'SCHDAN037', server IP 'localhost', and socket '2017'. The top-right window shows another 'ClientApp' where a user enters 'MKNOLI003', 'localhost', and '2017'. The bottom-left window shows the 'ServerApp' running on a Mac, which logs connections from 'MKNOLI003' and 'NTSMAS016', and displays messages sent by 'SCHDAN037'. The bottom-right window shows a third 'ClientApp' where a user enters 'NTSMAS016', 'localhost', and '2017'. The server window shows the connection from 'NTSMAS016' and the messages received from all three clients.

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

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

initialized and a port chosen to open up to listen for clients to connect on. Then as the ClientApps are initialized 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.

## **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.