GIT LINK

Git repository

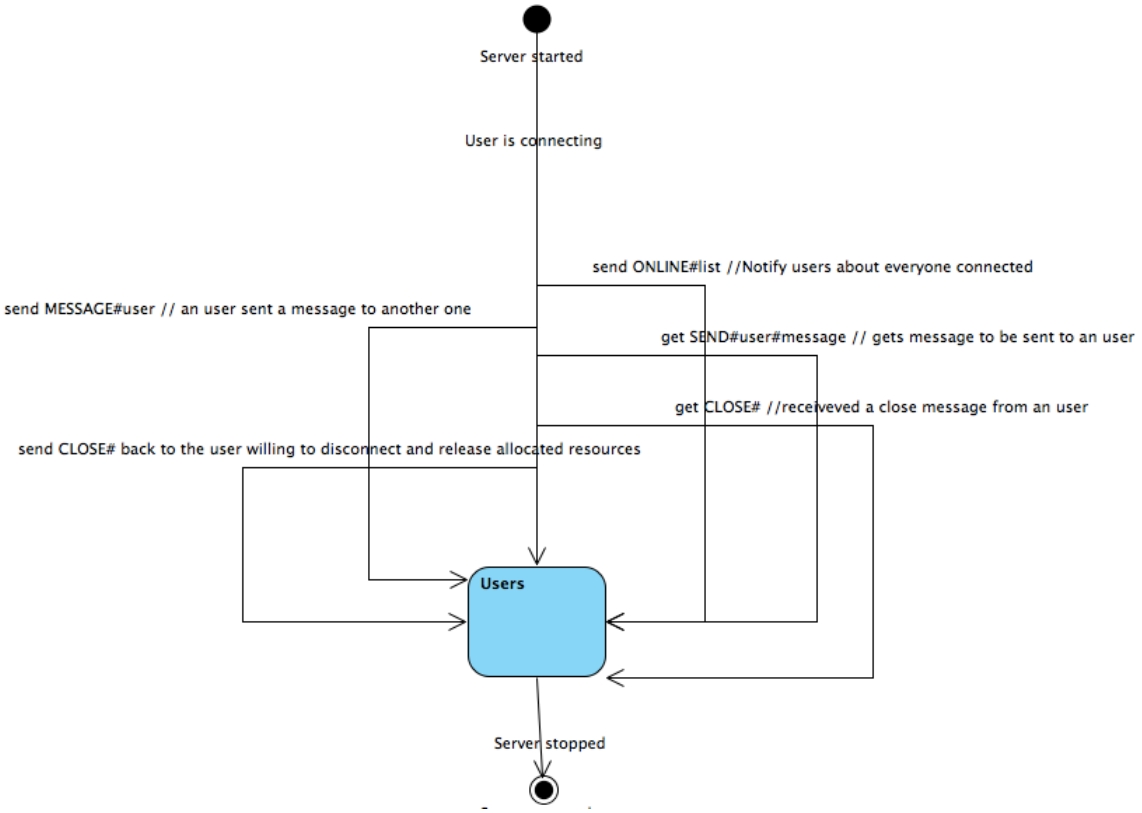
Description of design

For our design we have chosen to userJavascript using Node.js on the server and Node-Webkit for the client, which basically has an embedded Node.js server in it and uses the Webkit Engine to create desktop applications for all the platforms.

State Behaviour

Our state behaviour is the following. The server starts having an open predefined port through which the users can connect to. The server is storing the incoming sockets as objects used to identify the users by the raw socket object instead of other primitive methods like username/ip/port/ip&port/etc. When a user is connected to the server, the server encapsulates the incoming socket in a socket storage variable. From that point on, the client can send the "messages" which are commands and the server will interpret them.

State diagram for the server



Flow Graph diagram

```
|Time     | 192.168.1.110                        |
|         |
|0.000000000|         SYN       |Seq = 0
|         |(51210)  ------------------>(3333)
|0.000065000|         SYN, ACK  |Seq = 0 Ack = 1
|         |(3333)   ------------------>(51210)
|0.000080000|         ACK       |Seq = 1 Ack = 1
|         |(51210)  ------------------>(3333)
|0.000095000|         ACK       |Seq = 1 Ack = 1
|         |(3333)   ------------------>(51210)
|0.498450000|         PSH, ACK - Len: 15Seq = 1 Ack = 1
|         |(51210)  ------------------>(3333)
|0.498482000|         ACK       |Seq = 1 Ack = 16
|         |(3333)   ------------------>(51210)
|0.501262000|         PSH, ACK - Len: 14Seq = 1 Ack = 16
|         |(3333)   ------------------>(51210)
|0.501293000|         ACK       |Seq = 16 Ack = 15
|         |(51210)  ------------------>(3333)
|8.339147000|         PSH, ACK - Len: 7Seq = 16 Ack = 15
|         |(51210)  ------------------>(3333)
|8.339180000|         ACK       |Seq = 15 Ack = 23
|         |(3333)   ------------------>(51210)
|8.339875000|         PSH, ACK - Len: 6Seq = 15 Ack = 23
|         |(3333)   ------------------>(51210)
|8.339907000|         ACK       |Seq = 23 Ack = 21
|         |(51210)  ------------------>(3333)
|8.340268000|         FIN, ACK  |Seq = 21 Ack = 23
|         |(3333)   ------------------>(51210)
|8.340292000|         ACK       |Seq = 23 Ack = 22
|         |(51210)  ------------------>(3333)
|8.340301000|         ACK       |Seq = 22 Ack = 23
|         |(3333)   ------------------>(51210)
|8.347476000|         FIN, ACK  |Seq = 23 Ack = 22
|         |(51210)  ------------------>(3333)
|8.347542000|         ACK       |Seq = 22 Ack = 24
|         |(3333)   ------------------>(51210)
```

The Client sends a TCP SYNchronize packet to the server The server receives the client's SYN The server sends a SYNchronize-ACKnowledgement The client receives the server's SYN-ACK The server sends ACKnowledge The Server receives ACK. TCP socket connection is ESTABLISHED.

The server sends a FIN-ACK package to end the connection between them. The client acknoledges the ending of the connection.