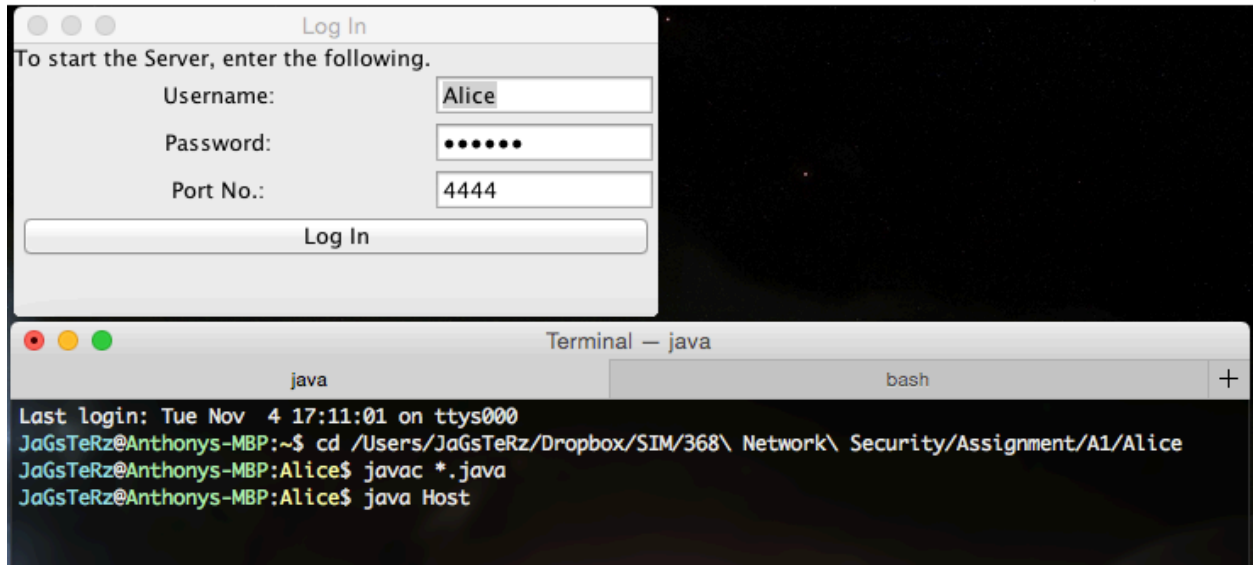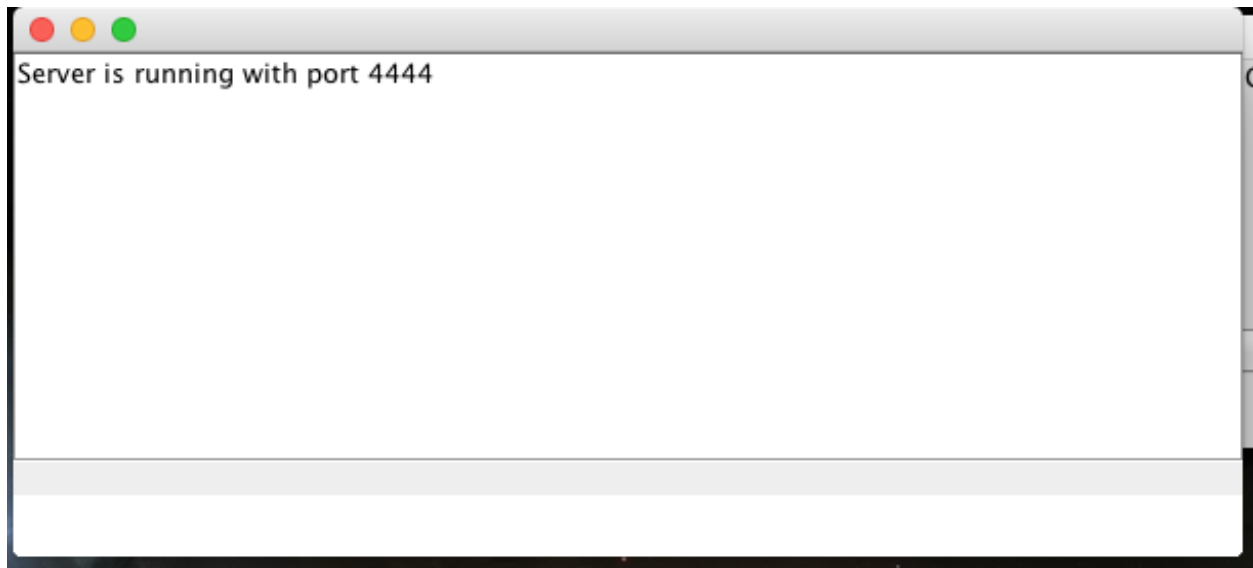# CSCI368 – A1

by Garcia Anthony John Abril (4321819)
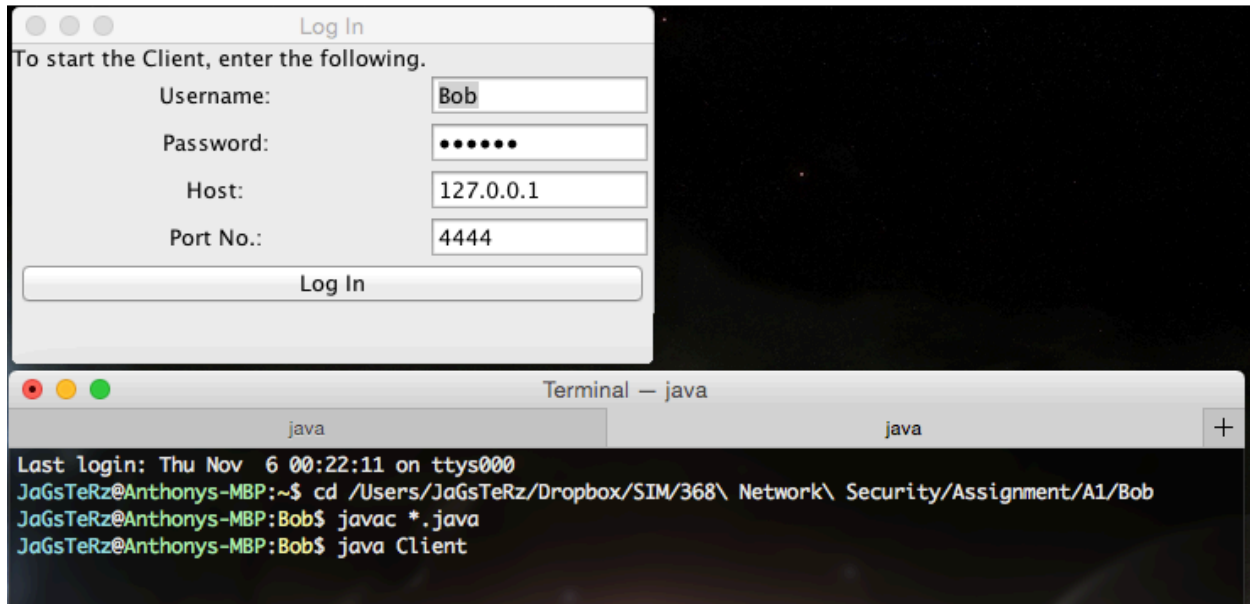
First, we compile the host files using "javac *.java" and we run the host using "java Host"
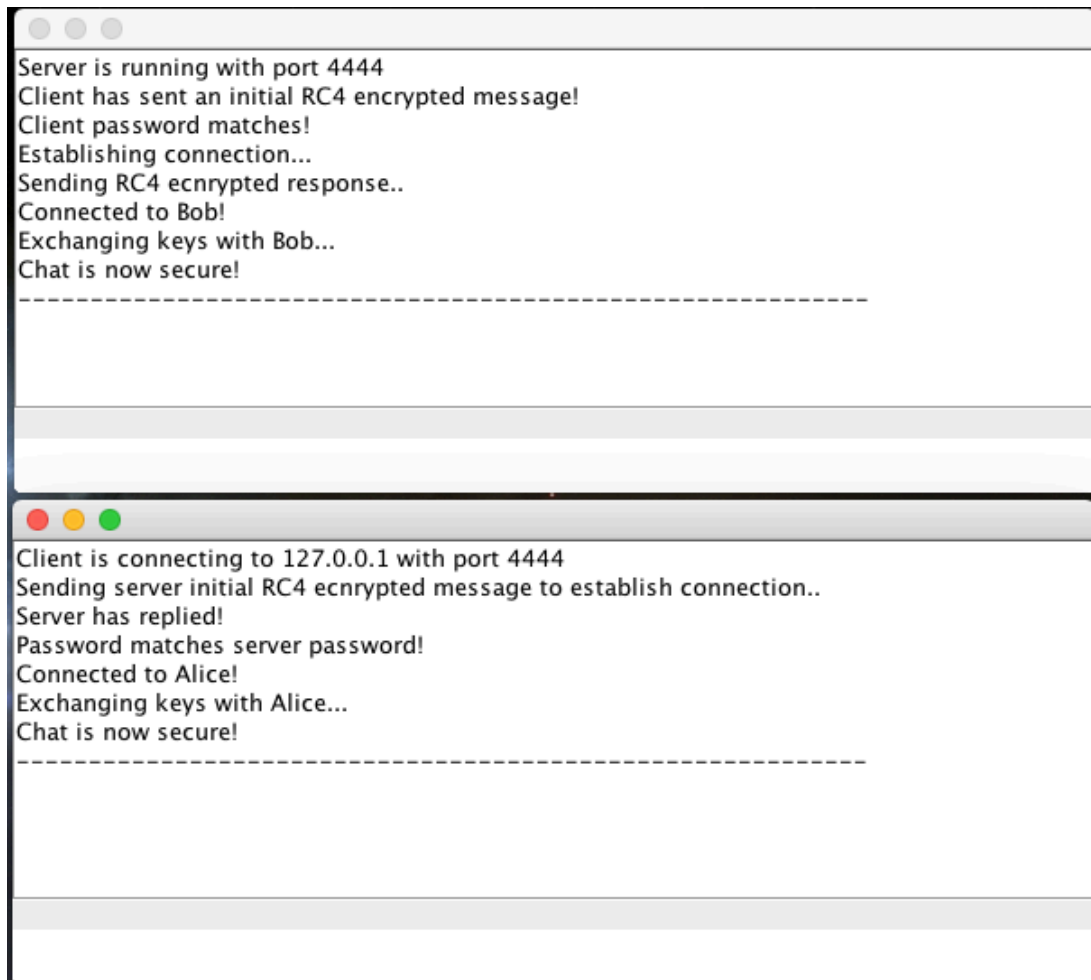

Here we have the host running after clicking the "Log In" button and listening to port 4444. The default password is "123456".
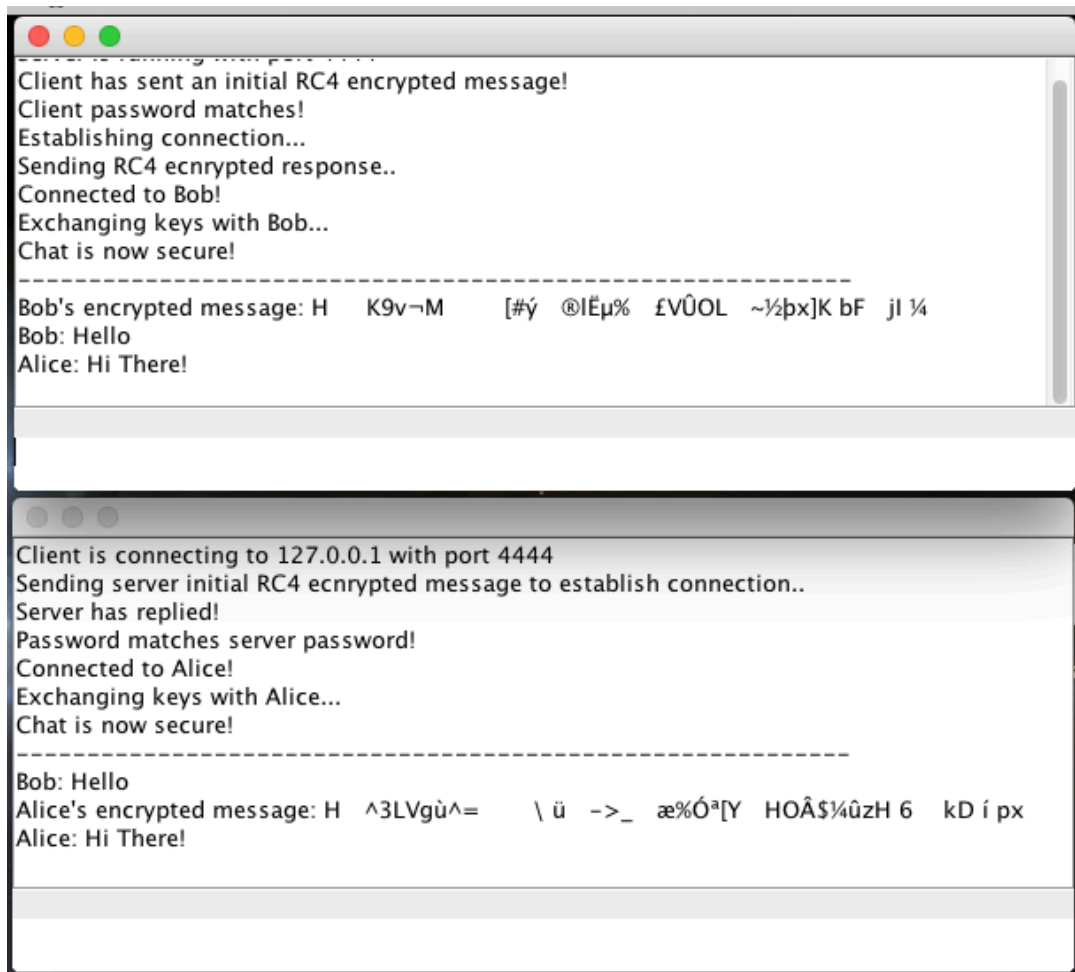
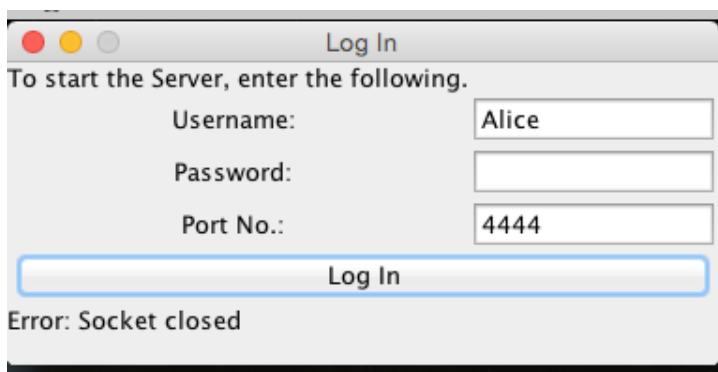Next, we compile the client program using "javac *.java" and run it using "java Client"

Here, we have logged in using the client with the password "123456" and both the server and client have send their initial messages.

Next, we try to send some messages between the host and client. We can also see the encrypted messages.

Client has sent an initial RC4 encrypted message!
Client password matches!
Establishing connection...
Sending RC4 ecnrypted response..
Connected to Bob!
Exchanging keys with Bob...
Chat is now secure!

Bob's encrypted message: H   K9v¬M   [#ý ®IËµ% £VÛOL ~½þx]K bF jI ¼
Bob: Hello
Alice: Hi There!

Client is connecting to 127.0.0.1 with port 4444
Sending server initial RC4 ecnrypted message to establish connection..
Server has replied!
Password matches server password!
Connected to Alice!
Exchanging keys with Alice...
Chat is now secure!

Bob: Hello
Alice's encrypted message: H   ^3LVgù^=   \ ü ->_ æ%Óª[Y HOÂ$¼ûzH 6   kD í px
Alice: Hi There!

When either the client or host closes, the program is pushed back to the login screen with the error of socket closed.

Log In
To start the Server, enter the following.
Username: Alice
Password:
Port No.:   4444
Log In
Error: Socket closed

Here we have the client and the host not having the same password. This check is done during the initial messages and kicks both the client and host back to the login screen to enable them to type in their passwords again.
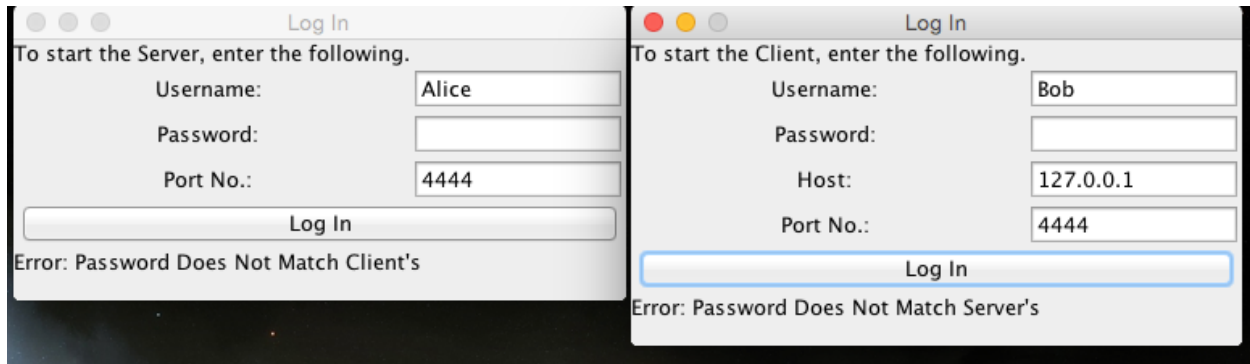
The program is also using threading, which allows for multiple messages to be sent concurrently.
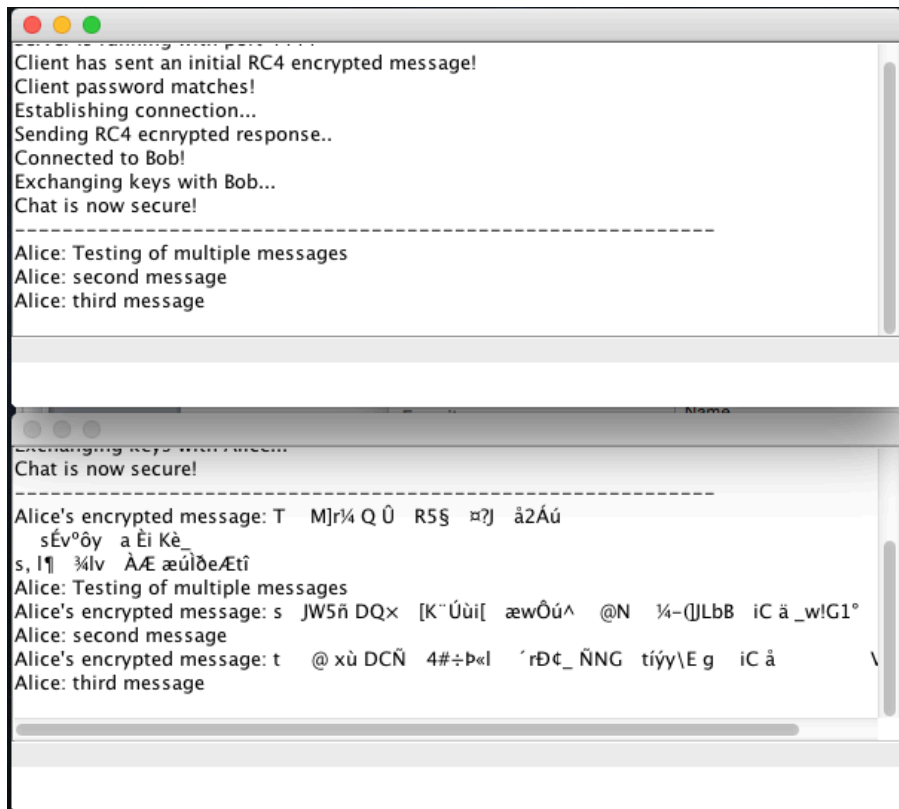
References:

RC4 codes were obtained from:

http://www.theaccents.org/ijacr/papers/current_june_2014/45.pdf

http://stackoverflow.com/questions/20001772/implementation-of-rc4-encryption-and-url-encoding-in-objective-c