**ISCG7436 – Enterprise Java Programming**

**Socket Programming Lab Session**

Use the example client / server chat project provided and setup code for the following changes to the baseline functionality. A key requirement with client-server based programs is to make sure that all of the messages that are sent and received are completely synchronized. If any part of the protocol becomes unsynchronized then the whole application will fail as the client / server will be sending data / receiving data that it cannot handle.

1. Setup code to allow a user to input a user handle. Send the user handle to the server using a new socket message defined using a new Server constants parameter.
2. Setup a new JList component to allow the chat client to display in list format the names / handles of the other users logged into the Server. Setup an appropriate client / server message to retrieve the list of users logged into the server.
3. Allow a user to select a client from the list of logged in users, and allow the system to send that user a private message.

**Challenge question 1**

Re-develop system completely using Java NIO Socket Channels. Note: This is not something we have looked at in class and is therefore a research question.

**Challenge question 2**

Re-develop system so it does not use threads and instead uses Java NIO’s non blocking event driven mode. Note: This is not something we have looked at in class and is therefore a research question.

**Resources**

* <http://docs.oracle.com/javase/6/docs/api/java/net/Socket.html>
* <http://docs.oracle.com/javase/tutorial/networking/sockets/index.html>
* <http://onjava.com/pub/a/onjava/2002/09/04/nio.html>
* <http://docs.oracle.com/javase/6/docs/api/java/io/DataOutputStream.html>
* <https://blogs.oracle.com/slc/entry/javanio_vs_javaio>
* <http://rox-xmlrpc.sourceforge.net/niotut/index.html>
* <http://www.mailinator.com/tymaPaulMultithreaded.pdf>