Operating System – Multi Threaded Client Server project

**Project Outline**

As outlined in the project specifications I have created a Multi-Threaded application that is designed with the capability of communicating between Client and Server by using a Socket connection. I have chosen Java as my programming language for this project and Eclipse as the IDE. This project has been prepared as part of my Operating Systems Module for 3rd year in GMIT for lecturer Martin Hynes.

**GitHub**

As part of my own individual initiative and in keeping with current Version control trends, I have been using GitHub to regularly commit my work. Here is the link:

<https://github.com/willhogan11/Multi-threaded_TCP_Server_Application>

**Design Choices**

As mentioned I have opted for Java to complete the project. With regard to setup it’s important to note that I tested my server and client locally using loopback address localhost. It’s important to note that the port number on the EchoServer will need to be changed as it’s been statically initialised to 9999.

The Application is broken up into two main packages, one which contains the Client Side code and the other which contains the server side code.

The Client side has no access to Authentication information or any other sensitive information as upon Successful Authentication, the user is Automatically working from their path, issued from the server side, for example if the user logs in as a valid user present on the server, then they will be logged into this example path C:\Users\william\Desktop\ProjectUsers\will

The Server side contains all the sensitive information. I have decided for convenience to have a separate folder that contains the Credentials.txt, which for security reasons is located at the root and all user folders which contains their files etc are after that. There are two other classes located in the Server package;

**FileClass** – Has two methods, one that is used to list the directory Path and output the contents of the users folder and one that creates a New Directory(folder) inside the users folder.

**Authentication** – I use this class for loading in the username and password authentication details straight from File into a HashMap, I have chosen this methods, as I feel it is particularly useful dealing with Key Value Pairs. Uncomment the DEBUG section to display to screen the contains of the populated HashMap.

Regarding configuration, the conection between the Server and the Client was made using a Socket connection, which connects both usign agreed port numbers and IP Addresses.

**Contents of Application**

**EchoServer**

* Echoserver.java
* FileClass.java
* AuthenticationParser.java
* Test.java (to be sent to the client)

**Client**

* Client.java
* **ProjectUsers**
  + credentials.txt
    - will
      * Files etc
    - John
      * Fields etc
    - Mark
      * Fields etc
    - Mike
      * Fields etc
* FileToUploadToServer.txt

**Instructions for use**

To make things easier during the development stages, I have opted for testing Client / Server code using localhost, this can be done by opening each project in Eclipse seperately or by using command line. The other alternative is to use the actual server remotely, ie Windows Azure, and use the server code there.

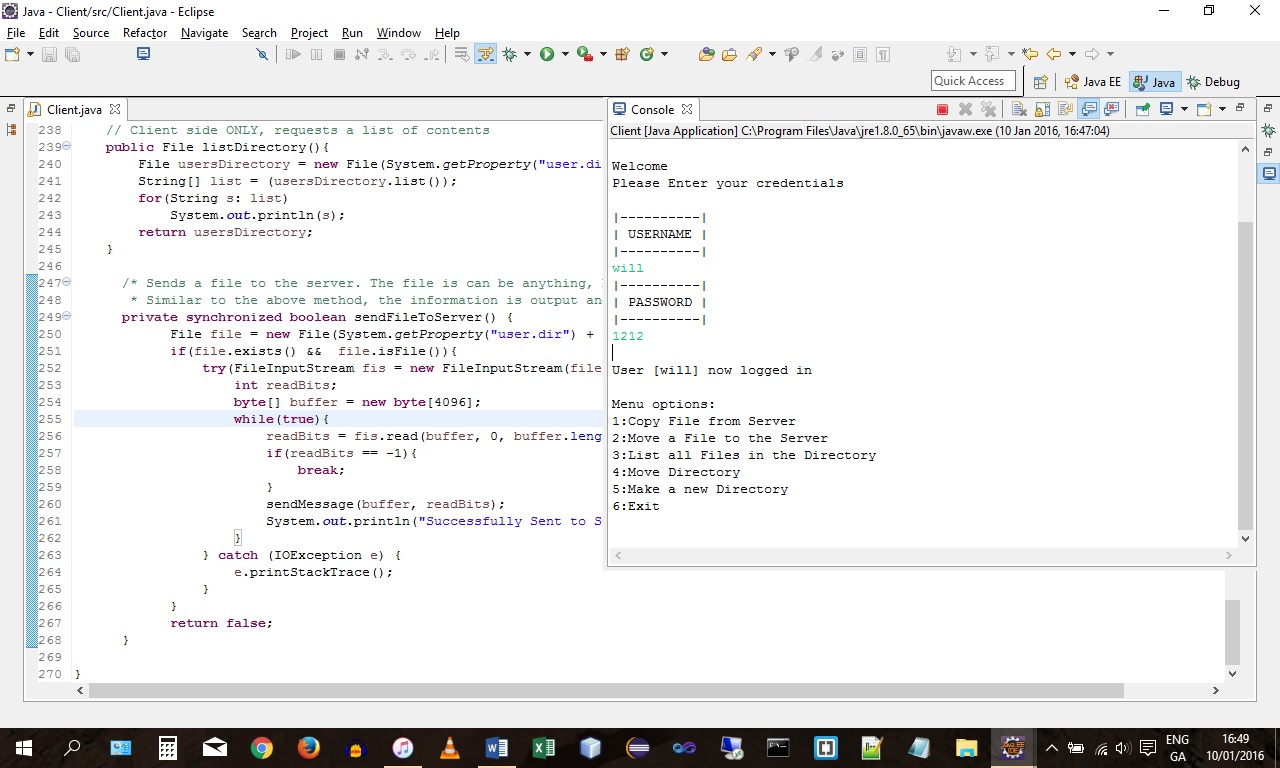
Steps:

* Launch / Run Server side code
* Launch / Run Client side code
* User prompted for the IP Address
* User prompted for Credentials, username and password located Server Side
* If successful, user logged in and has access to their folder only
* Menu shown to the use
  + 1: Copy a file from the server to the Client
    - Copies a textfile from a static location on the server, to the Clients folder
  + 2: Move a File to the Server
    - Uploads a static file from the client side to the users folder, server side
  + 3: Lists all files in the users directory plus displays the users absolute path
    - List all Files in the Directory
  + 4: <Not Implement>
  + 5: Make a new Directory
    - Inside the specified users directory
  + 6: Exits the Application

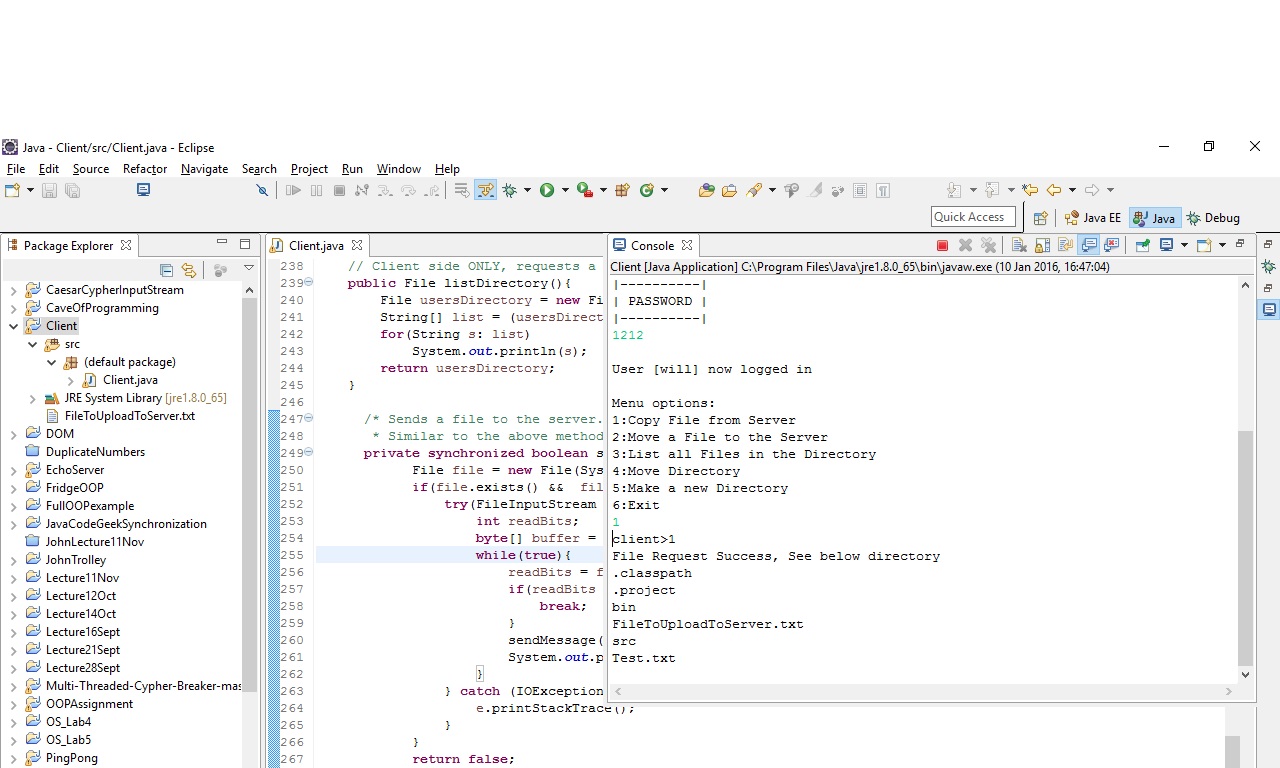
***Note:*** Unfortunately as I was having issues with my code on the Client Side (do while loop), every time a selection is made and the code has been executed, the application will have to be re-run. [ctrl +F11] for quick Re-Run shortcut!

**Screenshots**

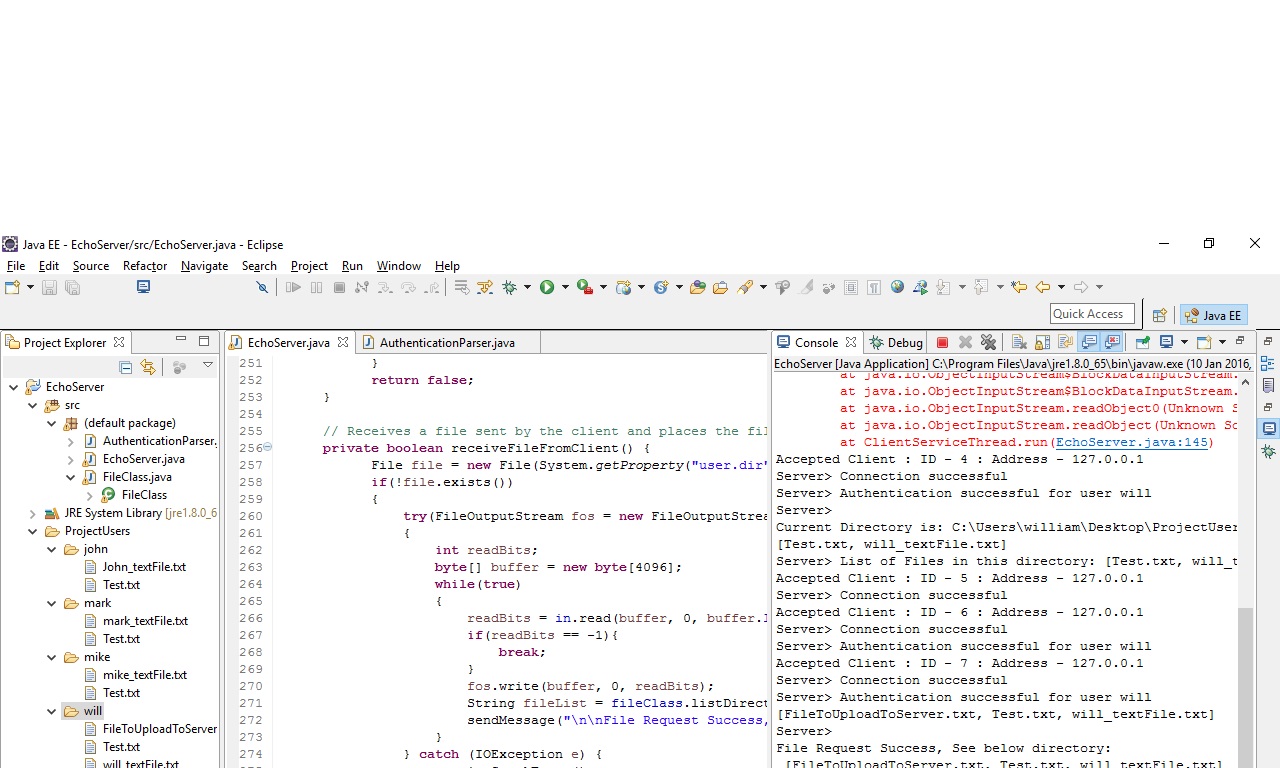
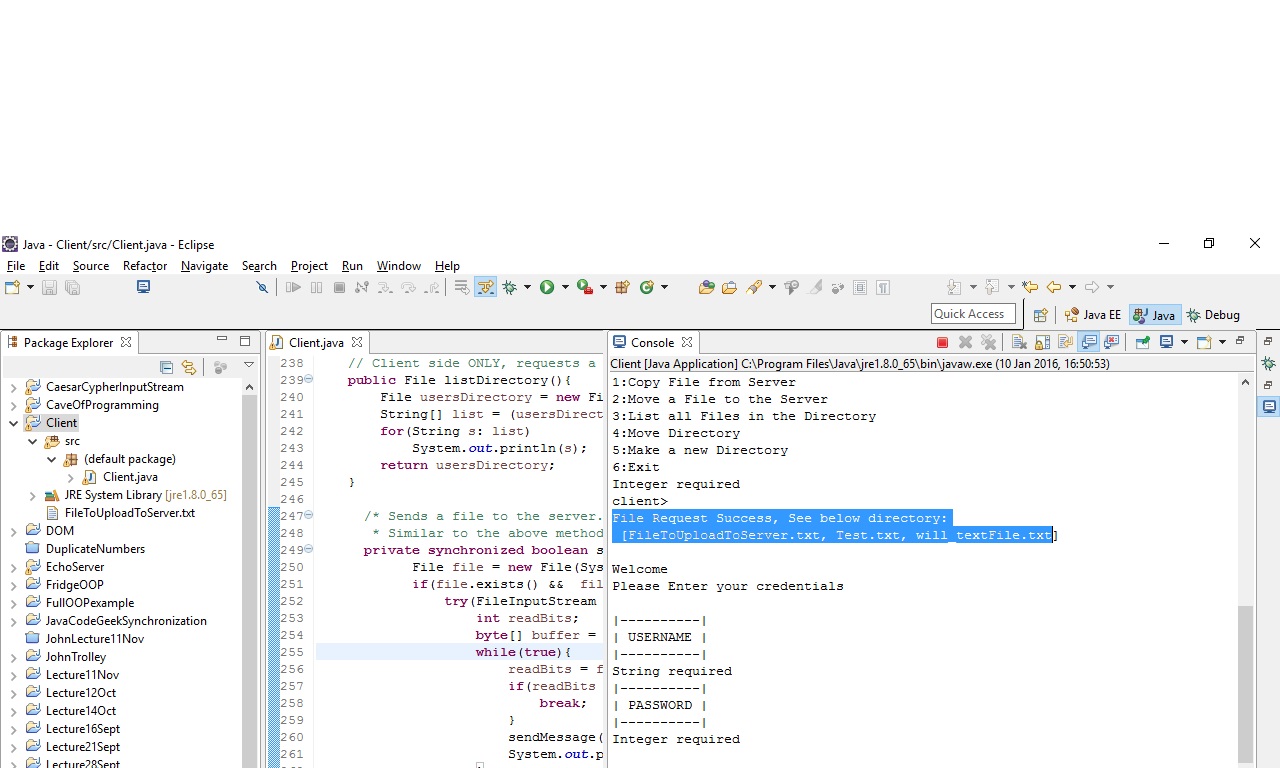
**Authentication:**

****

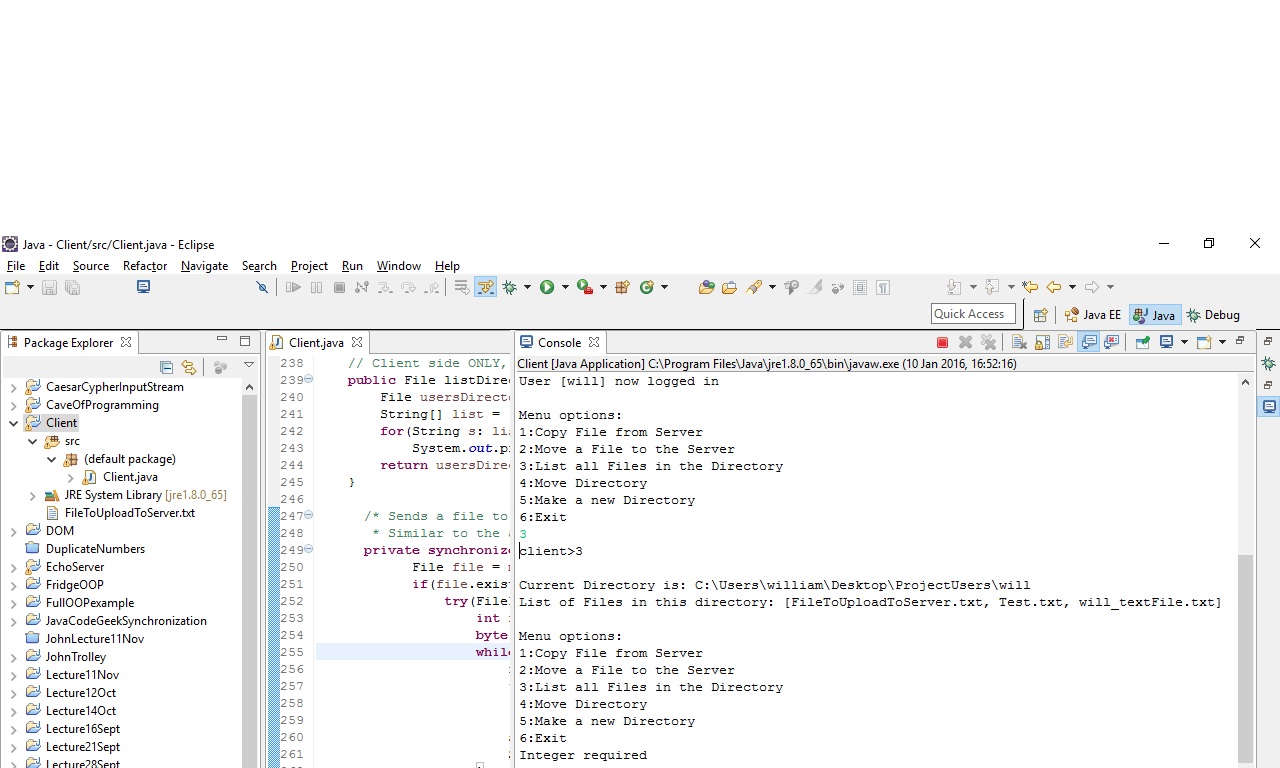
**Copy File:**

****

**Send File**

****

**List Directory**

****

**Make Directory**

