Course: B.Sc. in Commercial Computing
Module: Client-Server Distributed Programming
Assignment: 1 – Messaging System based on the Socket API
Credit: 20%

Objective(s):

- Design, implement and test a client-server messaging application using Java sockets.
- Document and present the solution in a screen cast.

Requirements

You have been asked to design and develop a ***messaging application*** that will allow employees in an organization to send and receive messages to one another.

The messaging application will need to cater for the following requirements:

- Compose and send a message to one or more users asynchronously (i.e. the delivery of a message is independent of the sending, so that the sender's process will not be blocked until the message is received by the receiver).
- Receive and read any message(s) that have been sent to them since the last time that the user logged into the system.

For this assignment you will need to consider:

- Interface Design
- Protocol Design
- Application Design and Implementation
- Testing
- Screen cast

Interface Design

You must design and develop a DOS command-line interface.

(Marks – 10%)

Protocol Design

A protocol is required to specify the rules that must be observed by the client and the server during a session of a particular service (in this case a mail service). Remember that the protocol is independent of implementation. The data exchanged should be in text. You will need to provide an overview of the protocol, a description of the identification scheme, one or more sequence diagrams illustrating the interaction between the participating client and server and a description of the format of each type of message.

(Marks – 10%)

Application Design and Implementation

The example programs from the slides covering Sockets and the Client-Server Model can be used to assist you with the design and development of the application. You should pay particular attention to the 3-tier design and implementation of the echo service in the Client-Server model. You will need to consider:

- The functionality of the server:
  - How is the server going to accept messages sent from the user to one or more user(s)?
  - How are you going to store the messages? For simplicity, it is sufficient for the server to maintain in transient memory messages that have not yet been delivered, i.e. you do not need to write these messages to a file/database.
  - Multiple clients will need to be able to interact with the server.
  - Mutual exclusion will need to be considered if a common data object is being accessed concurrently by multiple clients (i.e. co-ordinate access to a shared resource).
- The functionality of the client:
  - How is a client going to send and receive messages?
- What messages will you need to exchange between the client and the server? What is the protocol?
- Appropriate error handling.

(Marks – 60%)

Testing

Once the application has been designed and implemented it needs to be tested appropriately to determine if the stated requirements have been meet. You will need to consider ***functional testing***. For functional testing, does the application send and receive messages appropriately? Test cases should be determined and documented appropriately. Screenshots of the application should be provided. The client and server components should be tested locally on the same machine first and then deployed on different machines.

(Marks – 10%)

Documentation

This is a group project (pairs) but each individual is required to produce and submit their own screen cast detailing what they contributed to the project. The screen cast should include the following:

Section	Description
• *Problem Definition*	• Outline the requirements (provided in this document). If you decide to provide additional functionality ensure you document it here.
• *Methodology*	• How did you approach the assignment? What steps did you follow?
• *Design*	• Interface design • Protocol design (messages exchanged between client and server). • Application design – UML diagrams, etc.
• *Implementation*	• Source file summary. • Demonstrate your program running • Discussion of the technical details with appropriate code segments (that is, how did you realise the requirements).

• *Testing*	• Functional Testing: o Sample screen shots to demonstrate that the tests were conducted
• *References*	• Any resources that you used to assist you with the assignment (notes, textbooks, web references, etc.).

Other sections can be added to the screen cast at your discretion. Your documentation should be clear, concise, complete and correct.

(Marks – 10%)

Suggested Approach

Decide how you intend to tackle the assignment (i.e. methodology). As a suggestion:

- Design and implement the interface(s). What do you need to consider?
- Design and develop the application. This should be done ***incrementally***. Begin by developing and testing an application with minimal functionality. Then gradually introduce extra functionality with each successive round of implementation and testing. Remember the information found in the first set of notes on the Client-Server Model.

Submission Requirements

1) The source code must be submitted in source code (*.java). Note that you should not provide projects from either Netbeans or Eclipse, only the .java files are required.
2) Assignment screen cast should be submitted in a file called ScreenCastURL.txt.
3) All source files and documentation must be submitted in a single (zip format) file.
4) Assignments should be submitted through Moodle.

The deadline for this assignment is:

November 16th 2012 at 11:55pm