#### Assignment 1

|  |  |
| --- | --- |
| **Due date:** |  |
| **Value:** | 25% of total assessment |

#### Expected Learning Outcomes Assessed

#### On successful completion of this unit, students should be able to:

LO1: discuss the networking and operating systems requirements for enterprise architecture systems

LO2: describe and compare the models of enterprise architectures

LO3: describe, compare and use the core technologies that underpin enterprise systems

LO4: analyze a business process problem and design enterprise architecture systems for distributed computing applications

LO5: describe the likely trends in enterprise architecture systems

LO6: use technical and non-technical oral and written communication strategies in presenting design features of enterprise architecture systems

#### Rational

This subject offers students an understanding of enterprise architecture and its domains in connectivity, portability and interoperability.

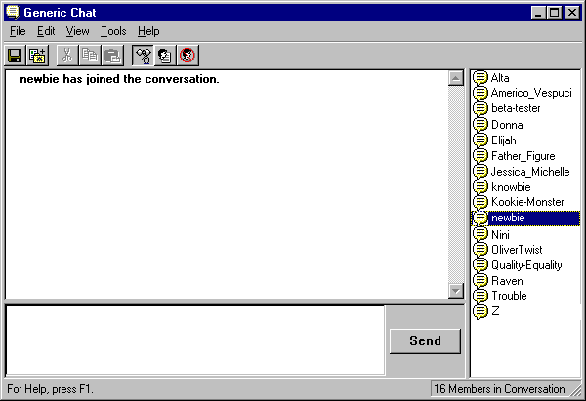
**Instructions**

**What is a chat room?**

There are different kinds of chat applications available, but for this assignment's purposes, a chat room is a program (or collection of programs) that does the following:

* Identified by a URL.
* People can "subscribe to" or "enter" a chat room.
* You can send messages, and they instantly appear on the screens of everyone else in the room.

(Think ICQ, Instant messenger, Zwrite, ytalk, etc.) Here's a picture:

  
*You type into the entry line at the bottom, hit Enter, and everyone else in the chat room gets your message. The server sends a message whenever anyone logs on.*

**Message protocol**

For this assignment, chat messages will have the following format:

|  |  |
| --- | --- |
| **field name** | **content** |
| name | name of person sending message |
| text | text of message |

The idea behind this standard is that everyone's chat room should end up being compatible. That is, every client should work with every server.

The protocol is stateless, and clients are not expected to be able to show messages that were sent before they connected to the server. The server will discard messages once they are sent, and there is no way to get them back.

The client and server don't exchange any information other than these chat messages. That is, the client doesn't need to send any requests to the server, and the server will send no acknowledgment. The client needs to watch for this in case the server goes down, and the server needs to watch for this because clients are allowed to disconnect at any time.

**The monitor client**

Write a monitor client. It should contact a chat server and display all messages that come from it. It shouldn't send any messages, just monitor the ones that get sent. This program is useful for debugging and should be easy to write.

**The server**

Write a chat room server that uses TCP/IP streams. When it receives a message from a client, it should send a copy of it to all its clients (including the one it came from). Clients should be able to connect to and disconnect from the server at will. When a client connects, the server should broadcast a message from "server" stating that someone has logged in.

**The GUI client**

Write a fully functional client with a graphical user interface something like the screen shot above. There should be at minimum a text field that shows all the messages that have been broadcast so far (including who sent them and the text) and a text field where you can type a message and send it. You should be able to specify a name that will be sent with your messages, either with a command line option, a system property, a configuration file, or with a dialog box that pops up. You should display this name somewhere on the screen.

**Other Requirements**

The host and port where the clients look for the server should not be hard coded. It's okay to put in a default value, but it should be possible to change the server host and port either with a command line option, a system property, a dialog box, a configuration file, something like that. Be sure to document how this works in your README file.

The clients and server must be able to run on any machine and any port. Server should be able to connect to other server. A cross servers broadcast and cross server chatting should be allowed.

**Tasks To Do**

Submit a softcopy and hardcopy of your documentation (1000 – 2000 words or 5 – 7 pages) consists of the following tasks.

**Part 1 (Research)**

**Task 1**

As needed for information gathering, survey and review on network and operating system requirements.

**Task 2**

Produce a list of requirements and justify why it is needed for the chat room system.

**Task 3**

Conduct a research on current available similar solutions. Discuss on the strengths and weaknesses of each and review on the technologies used. The main objective is to gather information and ideas to propose a better solution.

**Part 2 (Design)**

**Task 1**

Present your overall project plan, timescale and milestones.

**Task 2**

Design an overall system diagram showing the connections and components which able to fulfill the requirements.

**Task 3**

Develop a prototype of the chat room solution. It should consist of a set of application interfaces or screens that cover the basic design of the system including screens for all propose functionalities.

These screens need not have any real functionality, but suitable form elements should be selected to indicate how the interaction with the user might work. At this stage the concern should only be with the design. The screens are not expected to link together to form a complete application, because there need be no functionality.

Provide a brief statement stating the role of each screen (e.g. illustrates a possible design for a page where the user can delete messages. It does that using a simple list of message titles, with a check box next to each.).

# Marking Criteria

|  |  |  |
| --- | --- | --- |
| Criteria | Mark | Grade |
| **Part 1** |  |  |
| Task 1 – review on network and operating system requirements | 10 |  |
| Task 2 – requirements list | 10 |  |
| Task 3 – Review on similar solutions (minimum 3)   * Comparing * Describe current trends * Include critical reviews and conclusions | 10  5  5 |  |
| **Part 2** |  |  |
| Task 1 - Project plan, timescale and milestones. | 10 |  |
| Task 2 - Overall system diagram | 10 |  |
| Task 3 - Prototype / Interfaces   * User-friendly * With clear & detail description. | 25 |  |
| Reference | 5 |  |
| Presentation | 10 |  |
| Total | 100 |  |