Vol 8. No. 3 Issue 2 – October, 2015
African Journal of Computing & ICT

www.ajocict.net

Internet Chat Application: A Solution to Reduce Cost of Procuring and Maintaining a PABX Phone in an Enterprise

J. Odiagbe & O.I. Oyemade
Research Student, M.Sc. Information Technology
National Open University of Nigeria
Sokoto Study Center – Nigeria
justusodiagbe@gmail.com

B. A. Buhari
Department of Mathematics, Computer Science Unit
Usmanu Danfodiyo University
Sokoto – Nigeria
buhari.bello@udusok.edu.ng

ABSTRACT

The Internet as an application development platform emerged rapidly from obscurity to the dominant position it now holds in enterprise and inter-enterprise computing. This research designs and implements internet Chat application as a solution to reduce cost of procuring and maintaining a PABX phone in an enterprise. Two-tier client server architecture is employed in the design of the internet chat application. And it is implemented using Visual basic programming under Microsoft Visual Studio 2013 development environment.

Keywords – Chat Application, Modeling, Visual Basic, PABX

African Journal of Computing & ICT Reference Format:
J. Odiagbe, B.A. Buhari & O.I. Oyemade (2015): Internet Chat Application: A Solution to Reduce Cost of Procuring and Maintaining a PABX Phone in an Enterprise. Afri J Comp & ICTs Vol 8, No.3 Issue 2 Pp 183-192

1. INTRODUCTION

The Internet as an application development platform emerged rapidly from obscurity to the dominant position it now holds in enterprise and inter-enterprise computing. Over the brief life of the Internet, Web applications have grown in prominence and capability. Each successive wave of client and Web server technology has upped the ante on the previous generation, increasing capability, integration and responsiveness. Text and video conferencing have gained popularity as they allow instantaneous human friendly communication. Save and edit contacts, storage of conversations and other user information are some common features in these applications.

Many of these chat applications are based on server-client architecture. That is, a centralized server is used to maintain all the information necessary to authenticate the user and relay data or connection information between users. Most of the chat applications existing today require user created profiles containing personal information before being able to chat. All this information is stored on a server. This method of connecting users leads to the server being a store-house of personal information.

Such a system has many potential disadvantages. It creates a performance bottle-neck as vast amounts of data must be processed by server. Connectivity issues between the clients and the server can interrupt connections between users that could otherwise be avoided. Likewise, in a corporate environment, hospital or schools' libraries, where noise of any form can cause a lot of disadvantages ranging from distraction which reduces productivity, distorted recuperation process, distractions from understanding concepts explained in textbooks etc., the widely used intercom telephone line is undesirable, and coupled with the fact that the cost of procuring this communication device and its maintenance is relatively high, the need for a cheaper and less intrusive means of communication is imminent and expedient.

Thus, we need a chat application that overcomes these drawbacks; one which can work without the complications of having a centralized server." [1]. This research designs and implements and internet Chat application as a Solution to Reduce Cost of Procuring and Maintaining a PABX Phone in an Enterprise. Two-tier client server architecture is employed in the design of the internet chat application. And it is implemented using Visual basic programming under Microsoft Visual Studio 2013 development environment.

2. RELATED WORKS

The need for communication in an enterprise is of utmost importance as the different departments of the enterprise need to share and transfer information between themselves.
A Private Branch eXchange, or PBX, is a circuit-switching system which provides service to one user organization. Usually located at the users' sites, PBXs have traditionally provided basic voices witching services.

The PBX usually routes incoming calls to attendant positions, from which they may be extended to station users; allows for station-to-station calling without the use of the telephone network; and allows station users to access the telephone network for outgoing calls.

A lot of researches has been conducted in internet chatting. Peris et al. perform study of interpersonal relationships in cyberspace using the chat channel as an interaction medium [2]. Results suggest that relationships developed online are healthy and a complement to face-to-face relationships. In addition, Dewes et al. performs an analysis of Internet chat systems [3]. They show how to separate chat traffic from other Internet traffic and present the results of an extensive validation of their methodology. Further, a Study of Internet Instant Messaging and

Chat Protocols has been conducted [4]. This analysis helps bridge this gap by providing an overview of the available features, functions, system architectures, and protocol specifications of the three most popular network IM protocols: AOL Instant Messenger, Yahoo! Messenger, and Microsoft Messenger.

3. METHODOLOGY

Client/Server architecture is used in this research instead of Sever/Client architecture. Client/Server computing involves two or more computers sharing tasks related to a complete application. Ideally, each computer is performing tasks appropriate to its design and stated function. This implies that computing resources and data storage resources are located where they will do the most good in fulfilling the computing task at hand.

Client/Server describes a program architecture and development process and is not tied to any particular operating system, database engine, programming language or networking environment. The main advantages of client/server applications are task specificity and independence.

The proposed system's architecture is a two-tier client-server architecture. A two-tier client/server application architecture is implemented when a client talks directly to a server, with no intervening server. It is typically used in small environments of less than 50 users. Generally two-tier architecture separates the user interface and the business logic onto one computer (Tier1) and the database server is onto another computer (Tier2). This can be shown in figure 1.

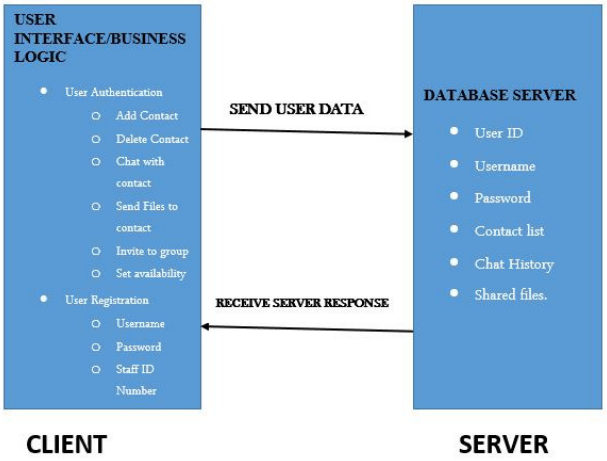


Figure 1: Two-tier client-server architecture for the proposed internet Chat application

The proposed design is implemented using Visual basic using Microsoft Visual Studio 2013 development environment.

4. DESIGN OF PROPOSED INTERNET CHAT SYSTEM

The proposed system would be able to do the following on each individual user's terminal:

i. Set status: With this, other users can check the availability of the user. The status could be available, out of office, busy, in a meeting etc. and the user can set personal status message which would be seen by every other user on the contact list.
ii. Accept user: Users can accept request to connect from anyone within their workgroup.
iii. Delete User: This can be used when the contact is no more available on the network and/or the organization.
iv. Add user: This button would be used to add a new user to the contact list.
v. Attach: This can be used to send files like pictures, Chats, reports or records sheet.
vi. User Authentication/Registration.

4.1 Design Architecture

Representation of the various parts of the application that makes up the design architecture can be easily done using Unified Modeling Language (UML) modeling. Use case diagram and activity diagram are going to be use to model the design specifications of the proposed internet Chat system. The UML is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system [5].

4.1.1 Use Case Diagram

A use case [6] is a sequence of transactions performed by a system that yields an outwardly visible, measurable result of value for a particular actor. A use case typically represents a major piece of system functionality that is complete from beginning to end [7].

Figure 2 is a User Interface Representation of the user. It shows the interactions with the user interface. On the user interface, each user can login with already registered credentials or can register if a new user. Once the authentication credentials have been verified to be correct, the user can have access to any of the following options like adding a new contact, deleting old contacts, selecting a contact from the list to chat with. In addition, one-to-one communication with users can be shown in figure 3.

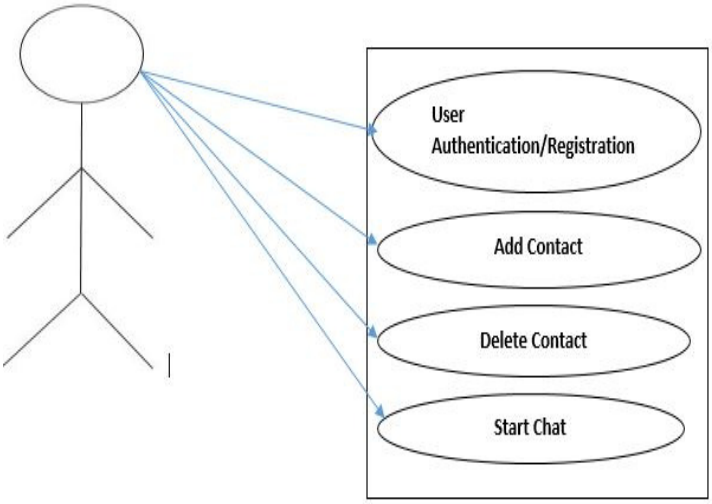


Figure 2: A Use-Case Diagram showing the interaction of the user with the application.

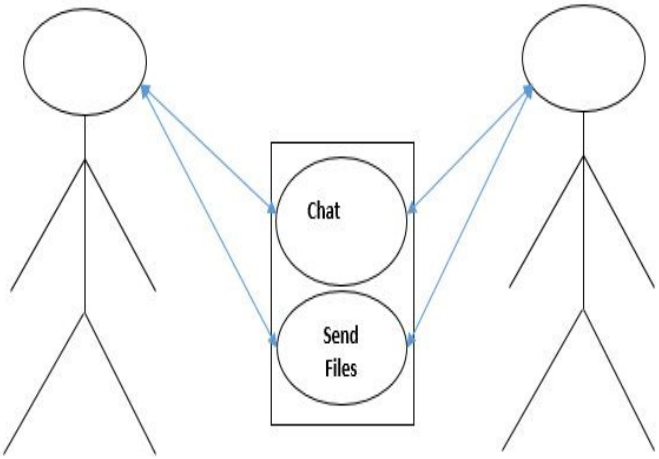


Figure 3: A Use-Case Diagram showing one-to one communication between two users through the chat application.

4.1.2 Activity Diagram

Activity diagrams are mainly used as a flow Chat consisting of activities performed by the system. But activity diagram are not exactly a flow Chat as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc. Activity diagram showing what happens at each stage of the process is shown in figure 4. For the user authentication, the proposed system gives limited number of tries for a wrong login combination. If after 3 trials, the combination still isn't correct, the system locks out. Some recovery questions inserted when registering would be asked and if the user still can't access the account due to wrong input, the system administrator would have to be called upon to validate the user with his priority password. Last login attempt would be displayed to the user, so, in case of any account breach, the user can raise an alarm over the issue in order to vet the system for compromise of user's data.

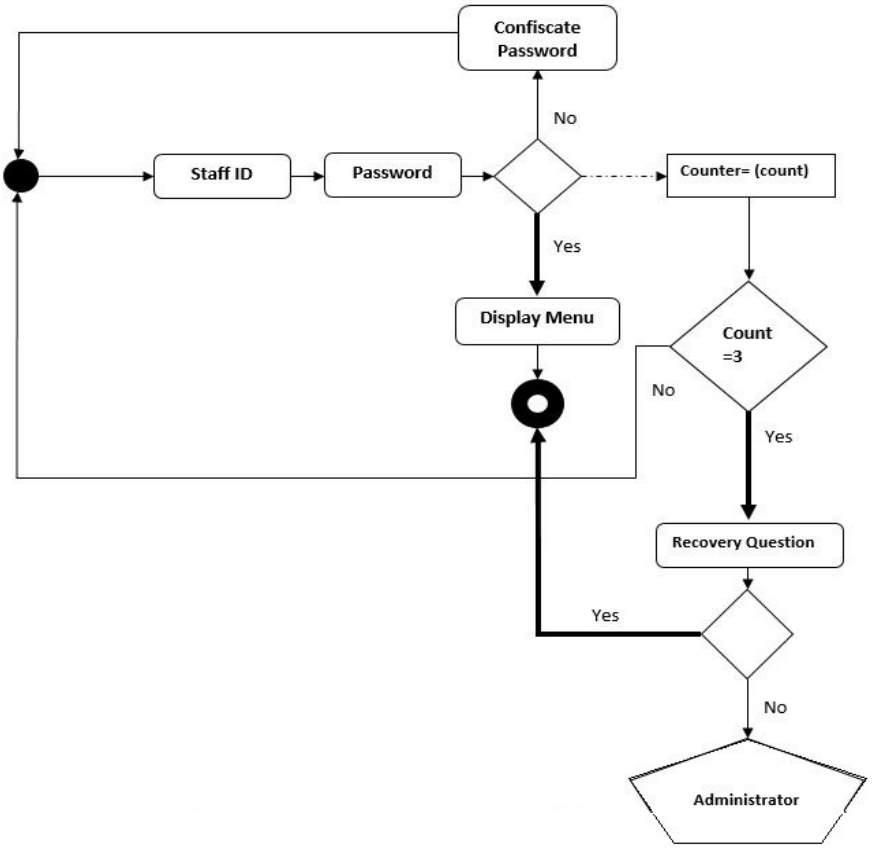


Fig. 4: Activity diagram explaining the proposed system.

5. IMPLEMENTATION OF THE PROPOSED INTERNET CHAT SYSTEM

During the implementation, the actual system is built. Building a successful information system requires performing some steps like Hardware requirement, Software Requirement, Implementation Procedure, Algorithms and Input and Output Snapshot.

5.1 Hardware Requirement

i. At least 2.2GHz Processor
ii. 1GB of RAM
iii. 20GB Hard drive
iv. Local Area Network
v. CD/DVD ROM

5.2 Software Requirements

i. Windows Operating System
ii. Microsoft Visual Studio 2013

5.3 Implementation Procedure

There are two important classes implemented in this program that made it possible for the two parties to communicate. These classes are:

i. **TCPListener:** This is the channel through which both parties communicate. It specifies the IP address of the server along with the Port number. The port number is the exact location where communication would be taking place. This is a number between 0 and 65535 and it is specified by the server user.
ii. **TCPClient:** This is like the TCPListener, but used by the client user. This takes the destination IP address and Port number from the client of the server to complete the half-duplex channel of communication.

Some system's libraries were also employed (imported) in the program, without which basic operations of the chat application like text, network address with port number and other basic simple Input/Output operations will not be possible.

i. **System.Net:** Used to initiate the network properties of the machine. This allows us to specify the IP address.
ii. **System.Net.Sockets:** This allows us to specify the port address on which the server would be listening to for data from the client.

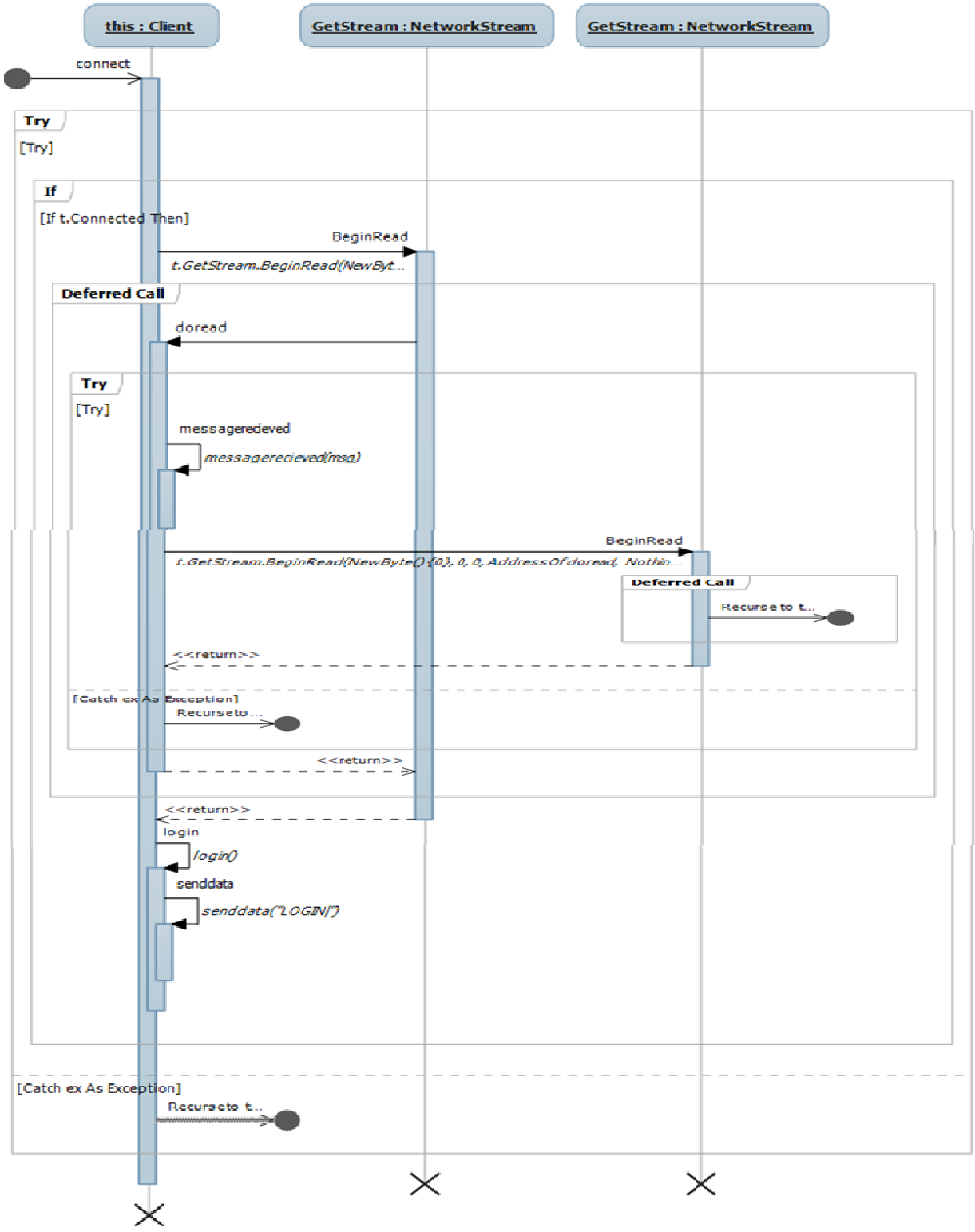


Figure 5: A sequence diagram showing the activities performed at each stage of the application

i. **System.Text:** This initializes the text properties of the system, since our chat application is a text based application.

ii. **System.IO:** This initializes the basic input and output operations. This allows us to give inputs as text to be sent to the other party on the network and also expect outputs as text in reply the message sent.

The activities performed at each stage of the application, starting from when the client connects till when data is being sent by the client can be shown in figure 5. Also activities performed by the server since when connection has been received till when data is being sent by the server can be shown in figure 6. Lastly, a system generated sequence diagram showing how the module interfaces with both the client and the server can be shown in figure 7.

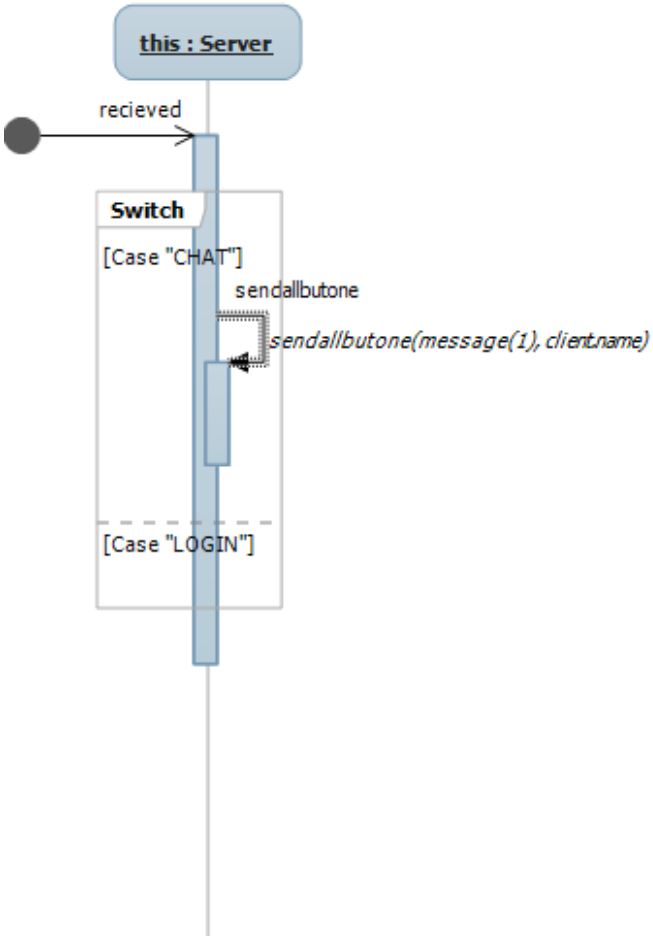


Figure 6: A sequence diagram showing the activities performed by the server

When the parameters for the two independent communicating parties have been set up, there has to be a module between the two parties that does the work of connect, get data and data forwarding. This module is called the ClientConnection. For the purpose of this project, a random name of generated for each of the clients connected to the server, with their IP addresses appended to their names. This is done by getting some ASCII characters that falls between 65 and 89. Also, when a client's chat window is closed, it is disconnected from the server and the name is no longer seen on the listbox.

The listbox holds the array of clients connected to the server. Once the client's window is closed, an Event (disconnected) is raised and the client is delisted from the listbox. Another event used in this module is the GotMessage event which takes the message from the connectedclient. To ensure that the message is well read, the message is read line by line into a buffer called ReadData. The read data will be read by the streamreader, which is a function of the System.IO library of the system and the message is transferred to the streamwriter of the same library for onward dispatch of the message to the appropriate address.

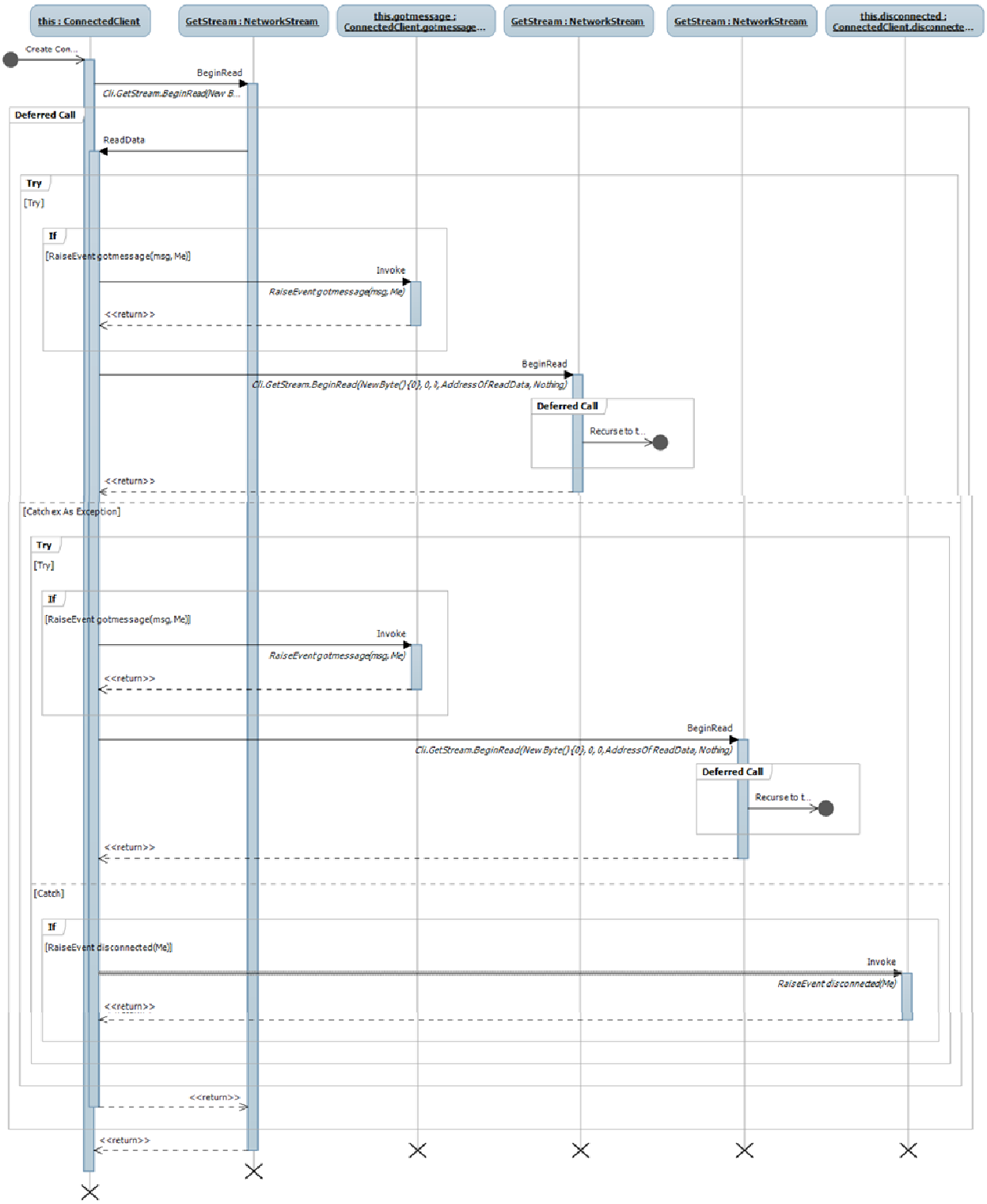


Figure 7: A sequence diagram showing how the module interfaces with both the client and the server

Sample snapshot of the system can be shown in figure 8 (a) – (c). Figure 8(a) is a Server Window showing the input area for the port specification, and message input. Also showing the listbox for the clients connected to the server, and also chat history. Figure 8(b) is the client window showing the specified IP address and the Port location specified for communication. Figure 8(c) is the client connecting to the specified port by the server and after connecting, it is showing on the server as a connected client in the listbox with the IP address of the client displayed along with the randomly generated name of the client.

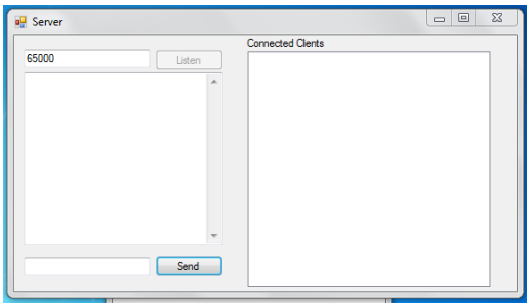


Figure 8 (a): Server Window

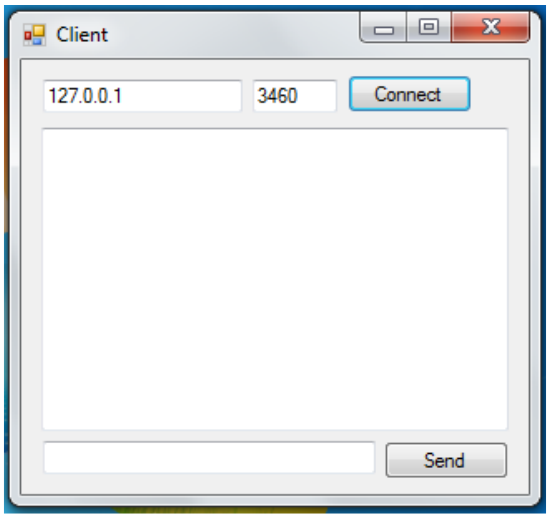


Figure 8 (b): Client Window

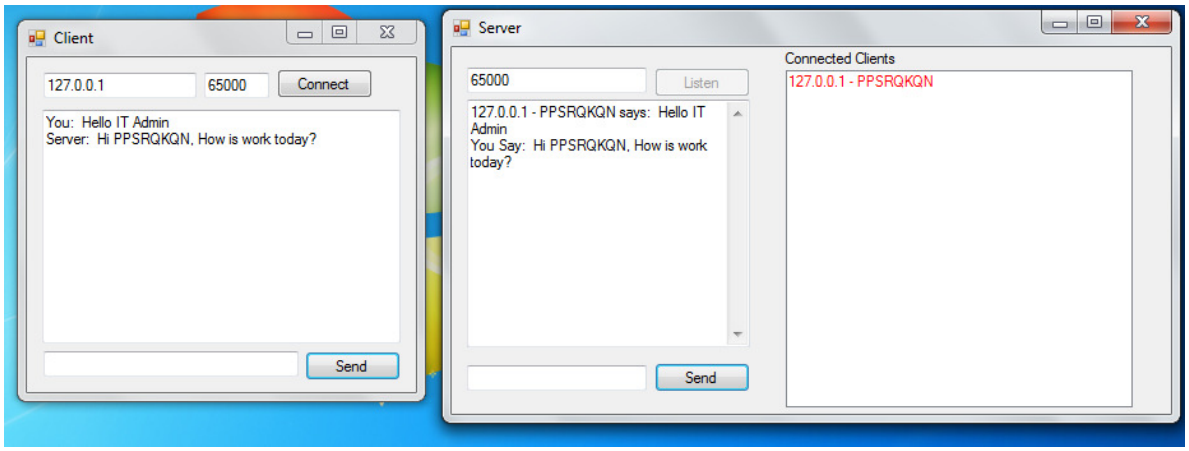


Figure 8 (c): A client connecting to the specified port by the server and after connecting

6. CONCLUSION AND RECOMMENDATION

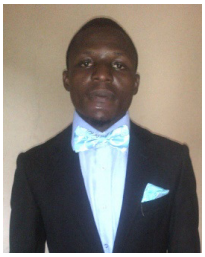
The development and implementation of this chat application proposed an alternative way in how we run our corporate environment is much better when compared with the old way of communicating through Intercom lines, which is quite intrusive, costly to acquire and maintain. Due promises proposed by this chat application, we therefore recommend that banks, hospitals, schools' administrative body, governmental organizations and non-governmental organizations, who are looking for means of cutting cost and budgets, can look in the line of this proposed system, build on the existing knowledge it portrays and adapt it to their mode of operations in their respective organizations .

It can also be observed that staffs' productivities are increased since they can keep working while attending to a chat message, with little or no distraction.

REFERENCES

[1] Dalwadi, P. (2011). Video chat on LAN
[2] Peris, R., Gimeno, M. A., Pinazo, D., Ortet, G., Carrero, V., Sanchiz, M., & Ibanez, I. (2002). Online chat rooms: Virtual spaces of interaction for socially oriented people. *CyberPsychology & Behavior*, *5*(1), 43-51.
[3] Dewes, C., Wichmann, A., & Feldmann, A. (2003, October). An analysis of Internet chat systems. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement* (pp. 51-64). ACM.
[4] Jennings III, R. B., Nahum, E. M., Olshefski, D. P., Saha, D., Shae, Z. Y., & Waters, C. (2006). A study of internet instant messaging and chat protocols. *Network, IEEE*, *20*(4), 16-21.
[5] Addison-Wesley,*Unified Modeling Language User Guide, The* (2 ed.). 2005. p. 496. ISBN 0321267974.
[6] Jacobson, I., M. Christerson, et al. (1992). Object-Oriented Software Engineering: A UseCase Driven Approach. Wokingham, England, Addison-Wesley.
[7] Bruegge, B. and A. H. Dutoit (2000). Object-Oriented Software Engineering:Conquering Complex and Changing Systems. Upper Saddle River, NJ, PrenticeHa

Authors' Profile



ODIAGBE, Justus Oluwaseun, was born on the 7th of April, 1990, in Ijebu ode Local Government of Ogun state. He hails from Edo state of Nigeria. He obtained his Primary school leaving certificate at St. Anthony's Nursery and Primary school, Ijebu-Ode, between 1993-2001. He proceeded to Sacred Heart Catholic College, also in Ijebu-Ode, between 2001 and 2007, thereby obtaining his Secondary school leaving certificate. He went ahead to obtain a Bachelor's degree in Computer Science and Information Technology from Bowen University, Iwo, Osun state from 2008-2012. He has just completed his Masters' degree in Information Technology at National Open University of Nigeria and he is currently rounding off his Masters' in Educational Leadership and Administration, at University of Nicosia, the course which being taken in an online environment. He is currently an ATM Engineer (NCR and Hyosung brands) at Inlaks Computers Ltd, a position he has held since 2013.

Bello Alhaji Buhari was born on 20th October 1974 in Sokoto North Local Government of Sokoto State. He obtained his Primary certificate at Model primary school Wurno road Sokoto from 1981 – 1987. He proceeded to G.S.S.S Yelwa Yauri where he obtained his junior leaving certificate from 1988 – 1990. He also obtained his senior secondary school certificate at Nagarta College Sokoto from 1991 – 1993. He then obtained a B.Sc. Degree in Computer Science at Usmanu Danfodiyo University Sokoto from 1996 – 2000. He further obtain an M.Sc. Degree in Computer Science at Ahmadu Bello University Zaria from 2006 – 2009. He is now undergoing Ph.D in Computer Science Research at Ahmadu Bello University Zaria. He started his career as a lecturer at Sokoto Polytechnic from sept 2003 to Dec 2003. He is presently lecturing at Usmanu Danfodiyo University Sokoto from Jan 2004 to date.

OYEMADE, Olufemi Isaiah, was born on the 18th of March, 1985, in Ife East Local Government of Osun state. He hails from Osun State of Nigeria. He obtained his Primary school leaving certificate at All Saint Nursery and Primary school, Ile-Ife, between 1991-1996. He proceeded to Urban Day Grammar School Ile-Ife, between 1996 and 2002, thereby obtaining his Secondary school leaving certificate. He went ahead to obtain a Bachelor's degree in Computer Science and Information Technology from Usman Danfodiyo University, Sokoto, Sokoto state from 2008-2011. He has just completed his Masters' degree in Information Technology at National Open University of Nigeria. He is currently Regional Passive Planned Manager (IHS Towers) at BISWAL LIMITED, a position he has held since 2012.