# Table of Contents


| | Name | Page |
|---|---|---|
| * | Abstract | 2 |
| * | Acknowledgements | 2 |
| 1 | Introduction | 3 |
| 2 | Project Overview (2.1-2.6) | 3-8 |
| 3 | Technology | 9 |
| 4 | Technical Issues and Difficulty (4.1-4.6) | 9-12 |
| 5 | Future Work | 12 |
| 6 | Conclusion | 13 |
| 7 | Reference | 13-14 |


# Table of Figure


| | Name of Figures | Page |
|---|---|---|
| **1** | User Window and chat window | 4 |
| **2** | Status/public Message Update | 4 |
| **3** | Public Message/Status and News Feed | 6 |
| **4** | Buzz | 7 |
| **5** | Leave and Client is notified when Server breakdown. | 8 |
| **6** | About Our Application Details | 8 |
| **7** | Custom User Control | 11 |

## Abstract

This project is to build an instant messaging system that has similar functions to popular instant messaging applications. It is a Local area Network (LAN) chat application with TCP/IP socket programming technology in C#. In our Networking System, there must be at least one server to create a network. One can join with an existing server or create one them self and then join with that. We focused on the user experiences when he/she is in a chat session – in particular, the functionality to send and receive textual messages.



## Acknowledgements

Thanks to our Software Development Sessional Course Instructor Fahim Hasan Khan for helping us to make this software.

# 1. Introduction:

Our Chat Messenger is developed for chatting in a Local Area Network (LAN). To communicate with other computers we have used TCP (Transmission Control Protocol) socket, The main advantage of our messenger is User flexibility.

# 2. Project Overview:

This Project has the following features and task,

## 2.1 Chat with Friends:

In our software, single chat is possible. The name in Buddy List, are online, someone can chat with him/her by clicking on the name only . When User click on the buddy's name, A chat window will open. In the chat window there is An area for writing something. There is two option for sending message. One by clicking on the "send" button or by pressing "Enter". In which date and time, user and his/her friend sent/receive message are also given with the chat message.

When a message is arrived, it makes a sound. User can change on/off it by going Options->Message->On/Off.

Again it is very annoying that someone working in a window and suddenly a message Come, appear a chat window and take the focus. To avoid this sort of problem we show the number of message arrive from a buddy with his/her name in the list. When a message come the color of buddy list is changed, so user knows about message very quickly. Again if User opens the chat window and remains it open, then the number of message or buddy list color will not change as the buddy's window is open and User is seeing it, threr is no need to show those.

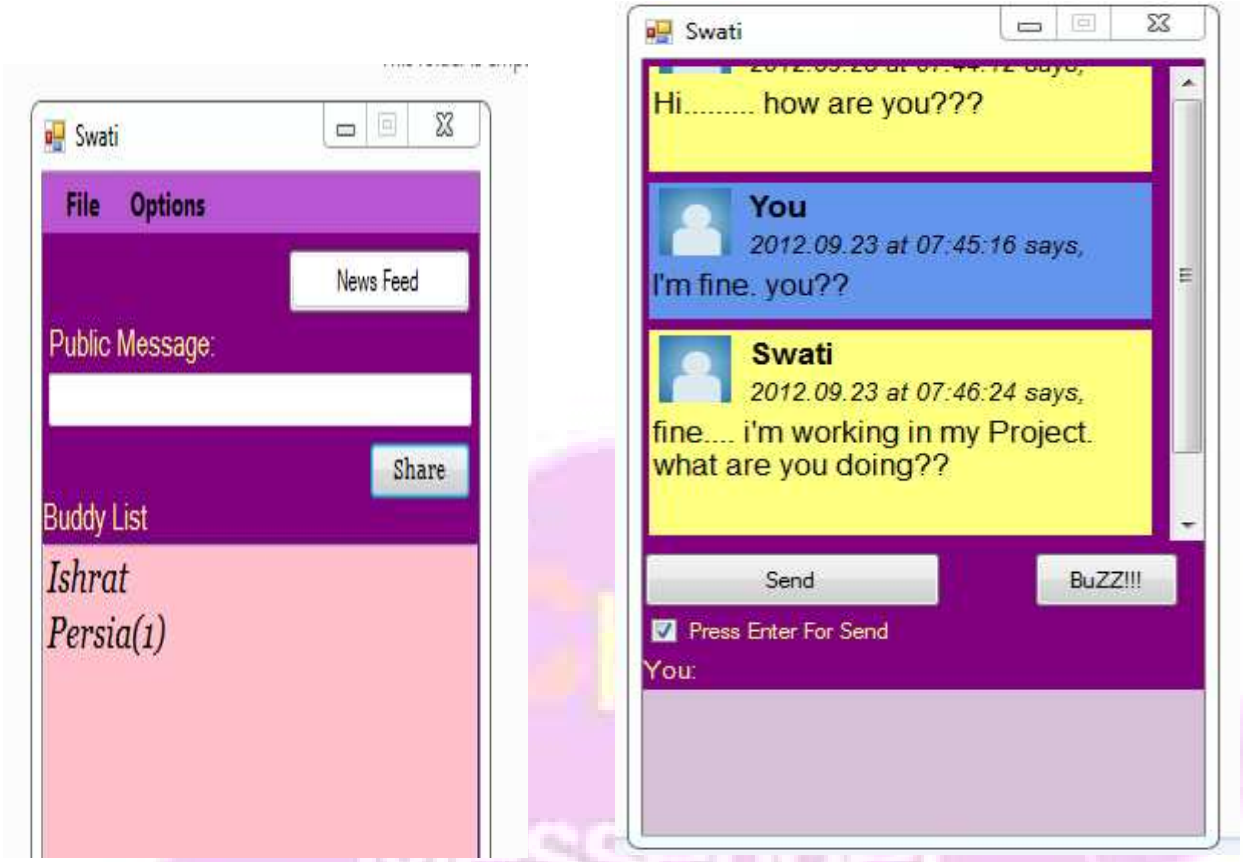


**Fig 1:- User Window and Chat Window.**

## 2.2 Public Message/Status:

There is also status/public messaging option. Public Message/status means a message which is send all the Buddy in the list.(Fig-2)


**Fig 2:- Status/Public Message Update.**

## 2.3 News Feed:

In the News Feed, User gets the news like buddy's joining or leaving News, public message. In the User Window News feed is a Button. When User click it, the user Window expand and show the news.(Fig-2)

Now, how will User know there is arrive a new news about one buddy? In our software we have make it simple, When a new news arrive the button color will be changed (White to yellow) and the number of new news will be shown in like this, "News Feed(1)" .

The time and date is also shown with the news.

To show the news in the expanded area we have added there a Richtextbox, As it supports partial formatting. The name of buddy, time and date, and the news are in different color. So it makes more easier to see.

There is also a Hide button to go back to its previous and compact size.

Again if the expanded area is opened and then new news is come, News Feed button color or number of message is not shown. As it is not needed anymore.

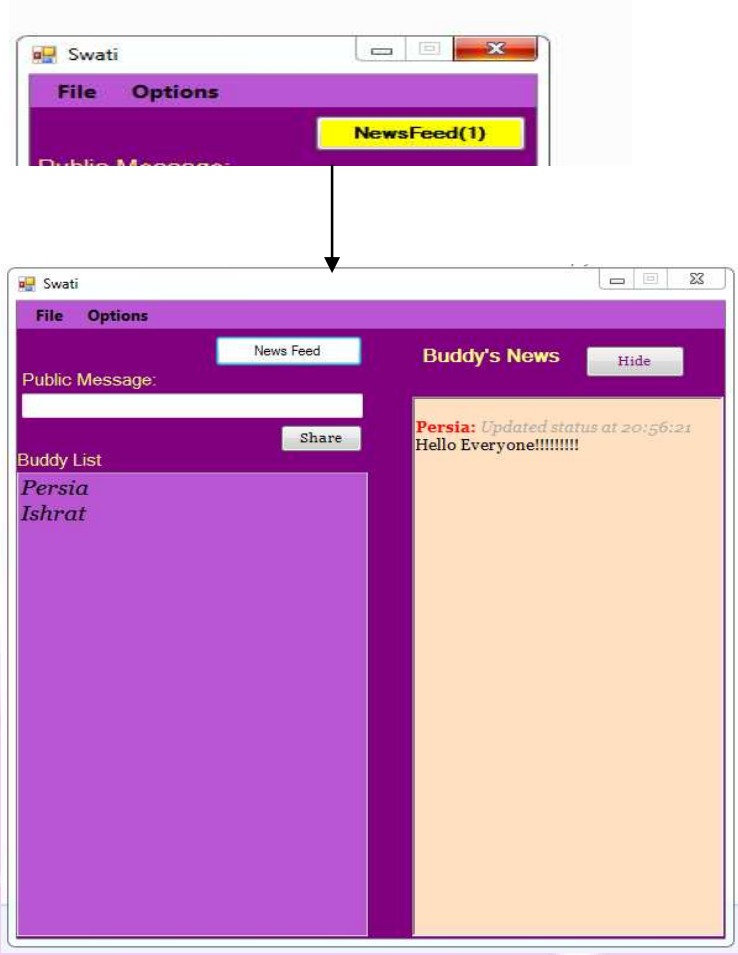


**Fig 3:- Public Message/status and News Feed.**

## 2.4 Buzz:

In the chat window there is a Buzz Option. When user gives buzz to his/her friend, On the other end, the buddy's chat window will suddenly appear, then the chat window vibrate for some moment with a great sound. It also draw the focus in the writing area for writing something.


**Fig 4:- Buzz.**

## 2.5 Leave chat & Unexpected disconnection:

User can leave chat by going File->Leave in the menu or simply click in "cross" option in the User Window . When user leaves, there a message will automatically send to server, so server remove that clien and notify all the existing clients in the list that this client has leaved. Again if the User who start server is leaved, the network will breakdown, the clients connected with server will be disconnected and get the Message "SERVER IS NOT FOUND".

If the Power Supply or net is Disconnected suddenly, the server will know this and remove that Client. We have done this by sending a alive message to server from client in 3 seconds. Server checks in 6 seconds which client does not send alive message, then server remove that client from list and notify others. Here we can see server is giving 2

chances (as it checks in 6 seconds) to the clients.

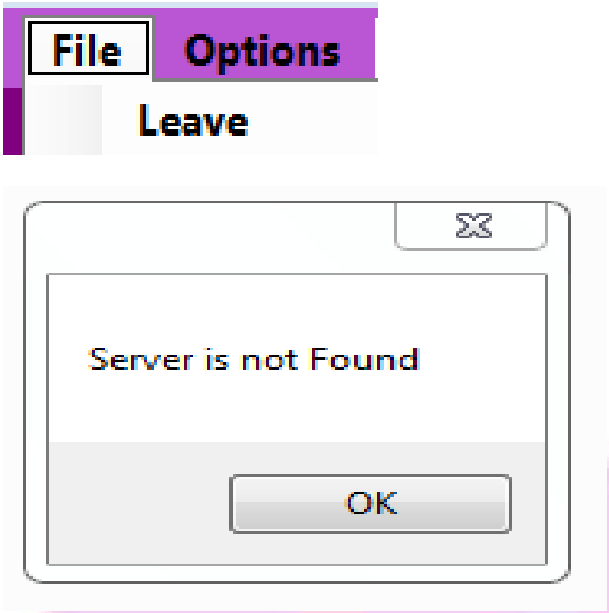


**Fig 5:- Leave and Client is notified when Server breakdown.**

2.6 Help Option: (This is newly added). In help->Manual, this report's pdf is added and Help->About, our project name, version and description is given.


**Fig 6:- About our Application Details.**

# 3 Technology:

- IDE Microsoft Visual Studio 2010
- Language C#
- .NET Framework 4
- Windows Form Application

# 4 Technical Issues and Difficulties:

## 4.1 Socket Programming:

The Socket Programming is a technique through which we establish communication between client and server.

TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet.

- TCP (Transmission Control Protocol): TCP is responsible for verifying the correct delivery of data from client to server. TCP will make sure that data will reached to server correctly. It also detects the error that arrives between data transmission.
- Port: A port is an application-specific or process-specific software construct serving as a communications endpoint in a computer's host operating system. A port is associated with an IP address of the host, as well as the type of protocol used for communication. The protocols that primarily use the ports are the Transport Layer protocols, such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) of the Internet Protocol Suite. A port is identified for each

address and protocol by a 16-bit number, commonly known as the port number.

## 4.2 Delegates:

A delegate is a type that references a method. Once a delegate is assigned a method, it behaves exactly like that method. The delegate method can be used like any other method, with parameters and a return value.

## 4.3 Multithreading Programming:

Thread is simply a “Execution Unit”. A Multithread Program contains two or more execution unit that can run concurrently.

- ### Cross Thread Communication:

  There is many UI thread and a socket thread in our application. As we know threads cannot communicate normally (if we do it, there will be cross thread Exception) .

  There is three different ways to perform this cross thread communication:

  1. Backgroundworker Class
  2. SynchronizationContext Class
  3. Invoke/BeginInvoke methods available from the UI controls.

  We have used in our application Invoke/BeginInvoke methods.

- ### Race Condition :

  Some Race Conditions have arise during runtime when two or more thread access the common resource at a time and change it.

  Because the thread scheduling algorithm can swap between threads at any point, we cannot know the order at which the threads will attempt to access the shared data. Therefore, the result of the change in data is dependent on the thread scheduling algorithm, i.e. both threads are 'racing' to access/change the data. The reason for having more race conditions among threads than processes is that threads can share

their memory while a process cannot.

To avoid this type problem, We use "Monitor". Before accessing in common resources ,a thread lock that object and then change/access it.

## 4.4 Custom User Control:

User Control is the most Power full features of windows Forms. In other words User controls are a way of making a custom, reusable component. A user control can contain other controls but must be hosted by a form. They enable to Encapsulate User Interface Design into lies reusable packages that can be plugged into project after project.

In our software, we have use it in the chat window to show the messages. It is a box with Name ,time and date and message. We reuse it to show buddy (Yellow) and user (blue) Message box.

To contain of this User Control, we have used Flow layout, where controls are added top to bottom.

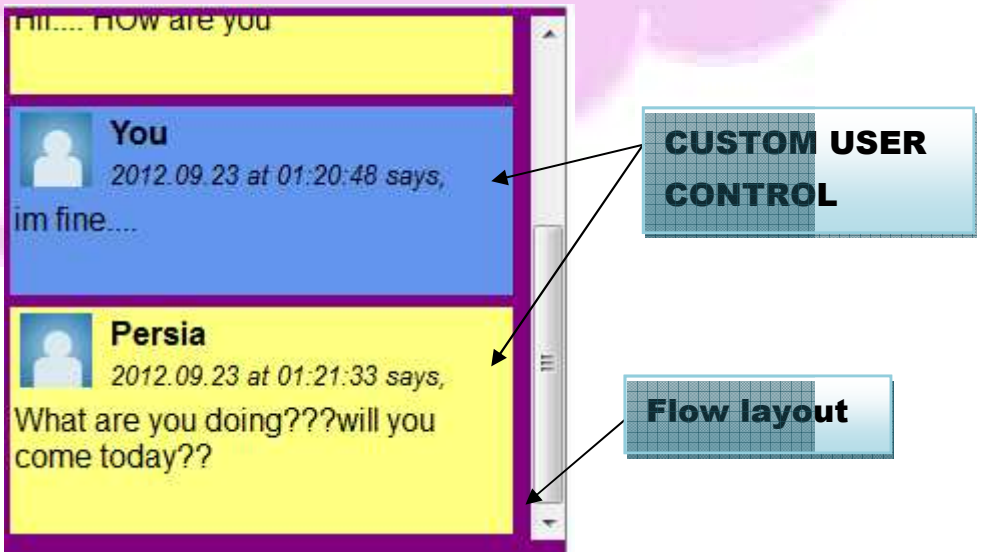


**Fig 7:- User Control.**

## 4.5 Object Serialization:

We define our Message Protocol in a class and send its object to others. For sending object, we need to serialize it and on the other end de-serialize it. We use XML to convert object to string. Then again string to byte before sending (As for sending something we need to make everything).

Serialization can be defined as the process of storing the state of an object instance to a storage medium. During this process, the public and private fields of the object and the name of the class, including the assembly containing the class, is converted to a stream of bytes, which is then written to a data stream. When the object is subsequently de-serialized, an exact clone of the original object is created.

## 4.6 Custom Event Design:

Events are mechanisms that allow application-specific code to execute when an action occurs. Events happen either before the associated action occurs (pre-events) or after the action occurs (post-events). For example, when a user clicks a button in a window, a post-event is raised allowing application-specific methods to execute.

Every Control in C# is full of events like MouseButtonClick, KeyDown, KeyPress etc. But We have used some events that are not already built in. We made it when a new message, status, client List or other update have come.

# 5 Future Work:

We left here scope for developing this software. The code is simple and easy to understand.

In our Wish list, there are,

- Broadcasting: Our Software is server IP dependant, that means, if a person wants to join network

he/she have to know the server IP. We want to show the currently existing server's IP. But TCP does not support it. We have to use UDP.

- Add Picture.
- File Transferring.
- Voice and Video Call.

# 6 Conclusion:

We have learnt a lot of thing by working in this project. It clears our concept about Networking(socket programming) ,some basic features of object oriented language like multithreading programming ,encapsulation of code . We have also learnt how a software should be designed. As we worked in a group we learnt how a group work is done which increase our skill as we share our ideas with each other.

# 7 References:

These are the References, where we got help,


- C# Tutorials-MSDN
  http://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx
- Multithreading Tutorial- Code Project.
  http://www.codeproject.com/Articles/14746/Multithreading-Tutorial\
- Delegates (C# Programming Guide)-MSDN
  http://msdn.microsoft.com/en-us/library/ms173171(v=vs.80).aspx
- Managed Threading- MSDN
  http://msdn.microsoft.com/en-us/library/3e8s7xdd.aspx
- How To: Manage cross-thread WinForms controls access-Code Project.
  http://www.codeproject.com/Articles/405947/How-To-Manage-cross-thread-WinForms-controls-acces
- Object Serialization in .Net Framework-MSDN
  http://msdn.microsoft.com/en-us/library/ms973893.aspx

- An Introduction to Socket Programming in .NET using C#- Code Project
  http://www.codeproject.com/Articles/10649/An-Introduction-to-Socket-Programming-in-NET-using

- Learn C# Socket Programming- Developer Fusion
  http://www.developerfusion.com/article/3918/socket-programming-in-c-part-1/

- TCP/IP Chat Application Using C# - Code Project
  http://www.codeproject.com/Articles/12893/TCP-IP-Chat-Application-Using-C

- Event Design-MSDN
  http://msdn.microsoft.com/en-us/library/vstudio/ms229011(v=vs.100).aspx