

Wi-Fi Communication platform

Final Report

Version 1.0

140323F

Abstract

This is about sharing files and communicating without data charges using Wi-Fi in a user-friendly environment. So, the users can communicate with this like a messaging application with file sharing functionality in a local area network using Wi-Fi. Main task is to develop windows application for personal computers. It is hope to developed for android and Linux in future. In this system user's messages or data are the inputs and output is to send it to the destination person or the group of people.

Objective of this project is to provide a simple and easy interface to the users to share the data and communicate via Wi-Fi. This system will provide a solution to fast file sharing and free chatting in a Local Area Network

Table of Contents

Abstract	1
1. Introduction.....	3
1.1. Background of the application domain/ problem	3
1.2. Motivation for the selected system development	3
1.3. Importance and main purpose of the system	3
1.4. Overview/ summary of the system and used approach and outcome	3
2. Literature Review	4
3. System Models.....	5
3.1. System requirements	5
3.1.1. Functional and non-functional requirements	5
3.1.2. Use case view.....	5
3.2. System Design	6
3.3. Architectural design	6
3.3.1. Class diagram.....	6
3.3.2. Main activity	7
3.3.3. Database Design.....	7
4. System implementation	8
4.1. Implementation Procedure	8
4.1.1. Tools and technologies	8
4.1.2. Methodology	8
4.2. Materials	9
4.3. The Algorithm	9
4.4. Main Interfaces	10
5. System Testing and Analysis.....	11
5.1. Testing approach	11
5.2. Unit Testing, Results and analysis of testing	11
5.3. Aspects related to performance, security, failures	12
6. Conclusion and Future work.....	12
6.1. Conclusion	12
6.2. Observations, new questions, and future work	12
6.2.1. Windows application	12
6.2.2. Android version	13
7. References.....	13

1. Introduction

1.1. Background of the application domain/ problem

Domain of this application is data sharing and communicating. Here it focuses on LAN. People use Bluetooth functions for wireless data transferring in PCs which is slow compared with Wi-Fi.

1.2. Motivation for the selected system development

Nowadays people are using devices like laptops and mobile phones that has Wi-Fi functionality. But in computers it is still been used Bluetooth stock application if needed to share some files wirelessly. It is slow and not a much user-friendly process. If this is done by Wi-Fi it will be faster than Bluetooth. Another problem in such applications that are already exist ask for another connection for that. But in this system, will be developed to use sharing functionality without having another network connection by using Wi-Fi hotspot of a using device.

1.3. Importance and main purpose of the system

It is important to have a system for sharing data through LAN because it is free of charge and speed of the transfer is also high. In a local group of workplace communication purpose also can be achieve with this system.

1.4. Overview/ summary of the system and used approach and outcome

It acts like a normal messenger application except it works in LAN without data charges and it limited into the LAN.

Anyone in the same network can use this system without any registration.

To achieve the target, it provides a built-in hotspot creator for a situation that Wi-Fi network is not around. So that one of the pc can act as a router and start a network. Then it can communicate with others who have connected to it.

Proper data transfer and communication is the outcome of this system.

2. Literature Review

This system has developed using socket programming. It creates socket between every user it connected to and communicate via that socket and a waiting socket is always run and looking for new connection in the server device.

This system has some system that do the job in various manner with different similarities and differences.

Other systems that looks similar are Feem [1], Skype [2], Whatsapp [3], Viber [4], Facebook messenger [5]. Except Feem all the other applications use internet for communication. Feem is a LAN communication platform that is very similar to this.

Considering Skype, Whatsapp, Viber and Facebook messenger those applications provides the same functionalities this application provides and those are much user friendly systems. As those systems started and transformed from a considerable time they have more improved functionalities. But there are some limitations they keep. Those applications cannot be used as data transferring applications when the data file is bigger than the size those applications limited.

“flyBird” doesn’t keep data size limitations for the data it transfers and it can handle large sizes of data files without error or application freezing.

But those applications provide data transferring around the world using internet. In the considering expectations of this application it needed to use it in a LAN. So, this application is not capable of communicating using internet.

When considering LAN this “flyBird” (this application) is fast and free of charge. It works without using any ISP. So, it can use Wi-Fi connection without using internet by using only the LAN.

When considering Feem it is a reliable application for offline communication purposes and yet it is a costly application that come up with a free version with limited functionalities. In “flyBird” it has no any limitation for the communication and it is free of charge.

Another main aspect that “flyBird” provides is its low utilization of resources. Application is small in size compared to Feem and its RAM usage also lower (under 50mb). So comparing resource utilization flyBird useful for run on any PC without hesitation.

Comparison of similar systems shows considering problem is solve better by “flyBird” in the given environment (LAN).

3. System Models

3.1. System requirements

3.1.1. Functional and non-functional requirements

A user should be able to add the data (name, profile picture) to his/her application profile. This is not compulsory. User should be able to use the application either without making the profile. A user should be able to edit his/her profile details.

User should be able to create and stop Wi-Fi hotspot to make a network if they are not already in a network or they need to create a private network themselves.

User should be able to see the connected devices in the network as contacts names and start a chat with connected users to the network. This is done by selecting the required receiver from contact list.

User should be able to create a group chat with the connected users in the network.

Then the main function of the application users can send text messages and file messages to connected devices as a normal chat application software. User should be able to check old messages from a connected user.

3.1.2. Use case view

Use case diagram of the system is showing here.

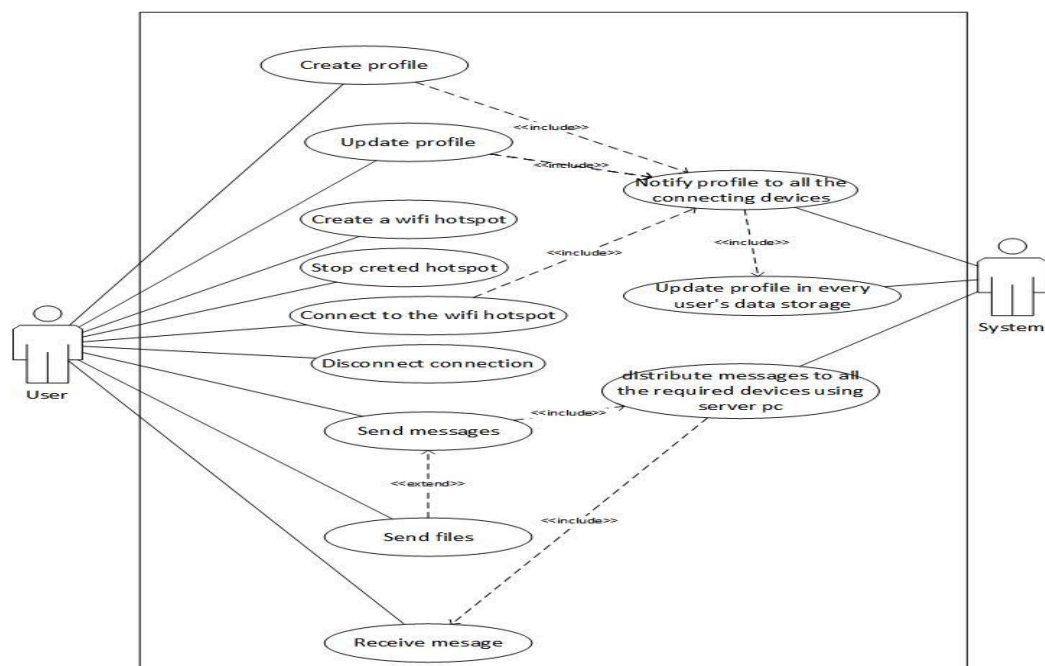


Figure 1-Use case diagram

3.2. System Design

3.3. Architectural design

Architecture of the system is described using class diagram, package diagram and a deployment diagram.

3.3.1. Class diagram

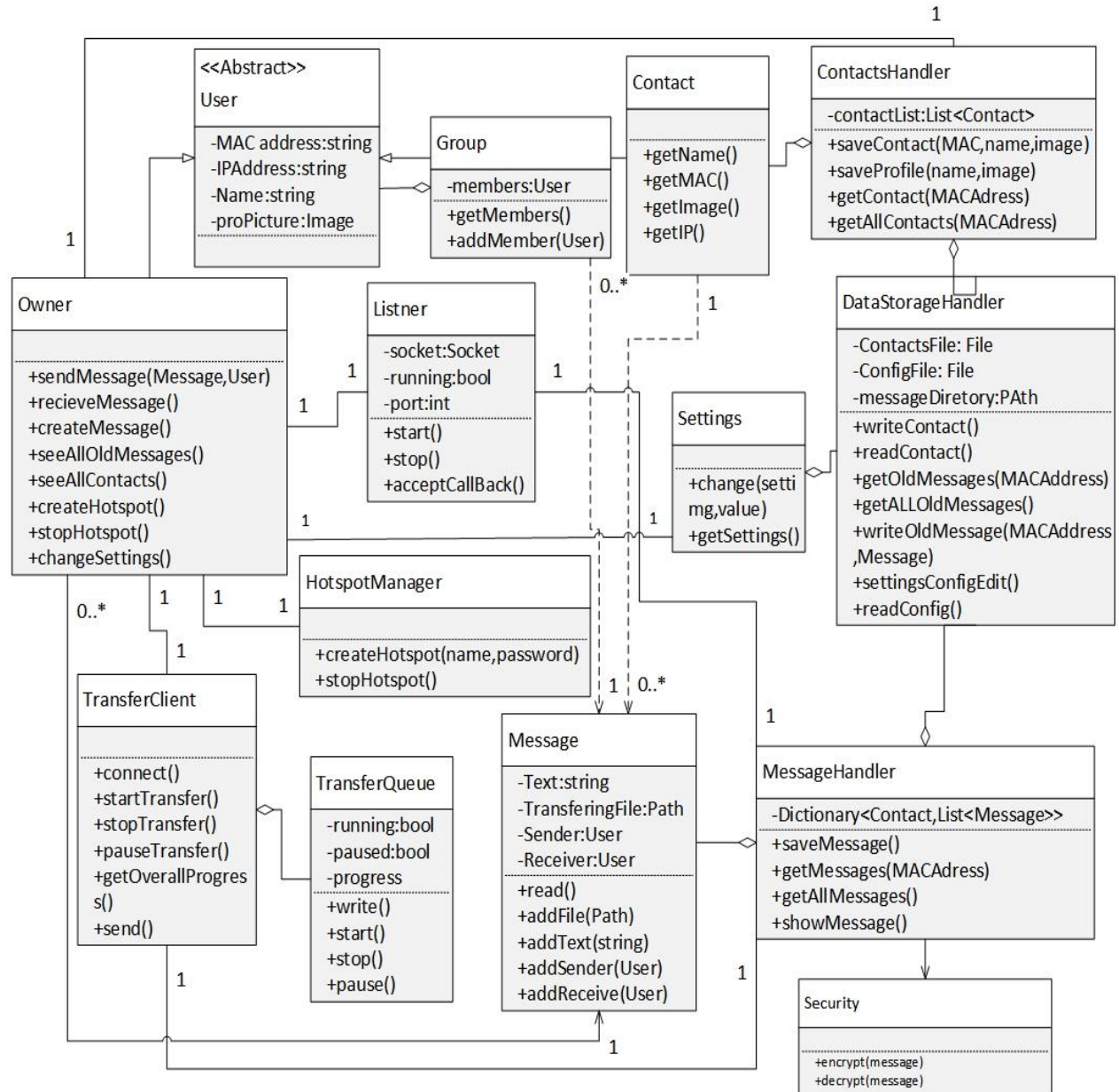


Figure 2-Class diagram

3.3.2. Main activity

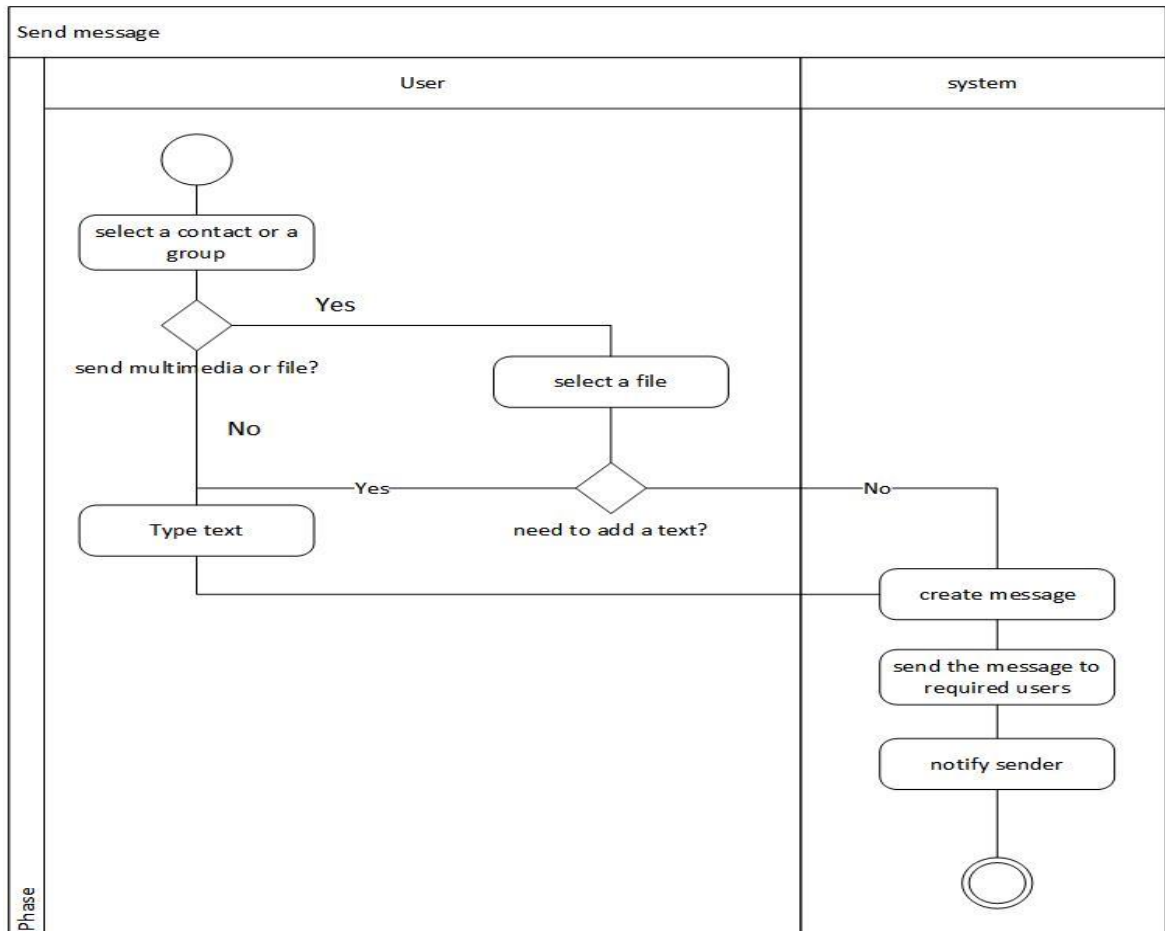


Figure 3- Send message activity diagram

Figure-3 shows the activity diagram of message sending. It after selecting needed contact system send the message to the IP address of the selected contact using the port that application use. After sending the message the sender gets a delivered notification in the message bubble in the chat display.

3.3.3. Database Design

This application doesn't use a database.

4. System implementation

4.1. Implementation Procedure

4.1.1. Tools and technologies

This system has developed using:

1. C# [6]
2. Visual studio 2017[7]
3. Metro UI framework [8]
4. GitHub [9].

Using C# language, it can be easily implemented the network handling part of this system and visual studio has used as the IDE.

Metro UI framework has used to create a user friendly smooth User interface. GitHub is used for version control.

4.1.2. Methodology

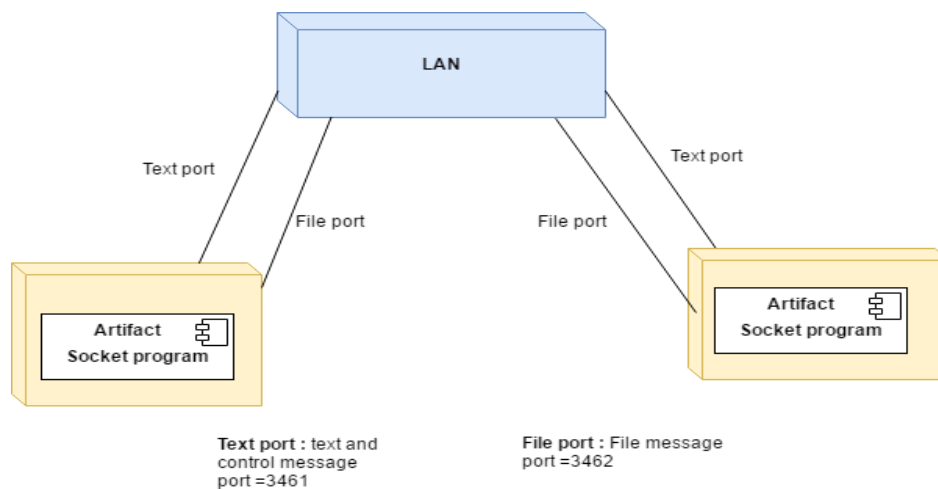


Figure 4-Deployment diagram

This application uses two ports to communicate via LAN. Those are for,

1. Text and control message sending and receiving.
2. File sending and receiving

It is easy and reliable to use a different port for the file transfer.

4.2. Materials

No materials used in the system.

4.3. The Algorithm

One of the main method in the application is starting a listening thread for every socket it created as a connecting device.

```
public void OnNewSocketAdded(object source, SocketAddedEventArgs e)
{
    string id = e.id;
    Socket socket = e.socket;

    confirmConnectionToConnectedDevice(socket);

    new Thread(() =>
    {
        //listening
        {
            while (true)
            {
                if (socket != null)
                {
                    receiveTextMessage(id, socket);
                }
            }
        }
    }).Start();
    Console.WriteLine("new listening thread started for id=" + id);
}
```

Here it become a running thread for listening. For every connection threads created for listen on that socket. That's how the application handles the multiple users in the system.

recieveTextMessage(id, socket) function is the listening function that catch the received text messages. File receiving function also looks similar to this method.

It should have different threads for that purpose because this application runs without a middle server like internet applications. So, the application cannot check a middle server for messages. It keeps track of every connection it has on the moment and runs different thread for every connection listening for new messages.

4.4. Main Interfaces

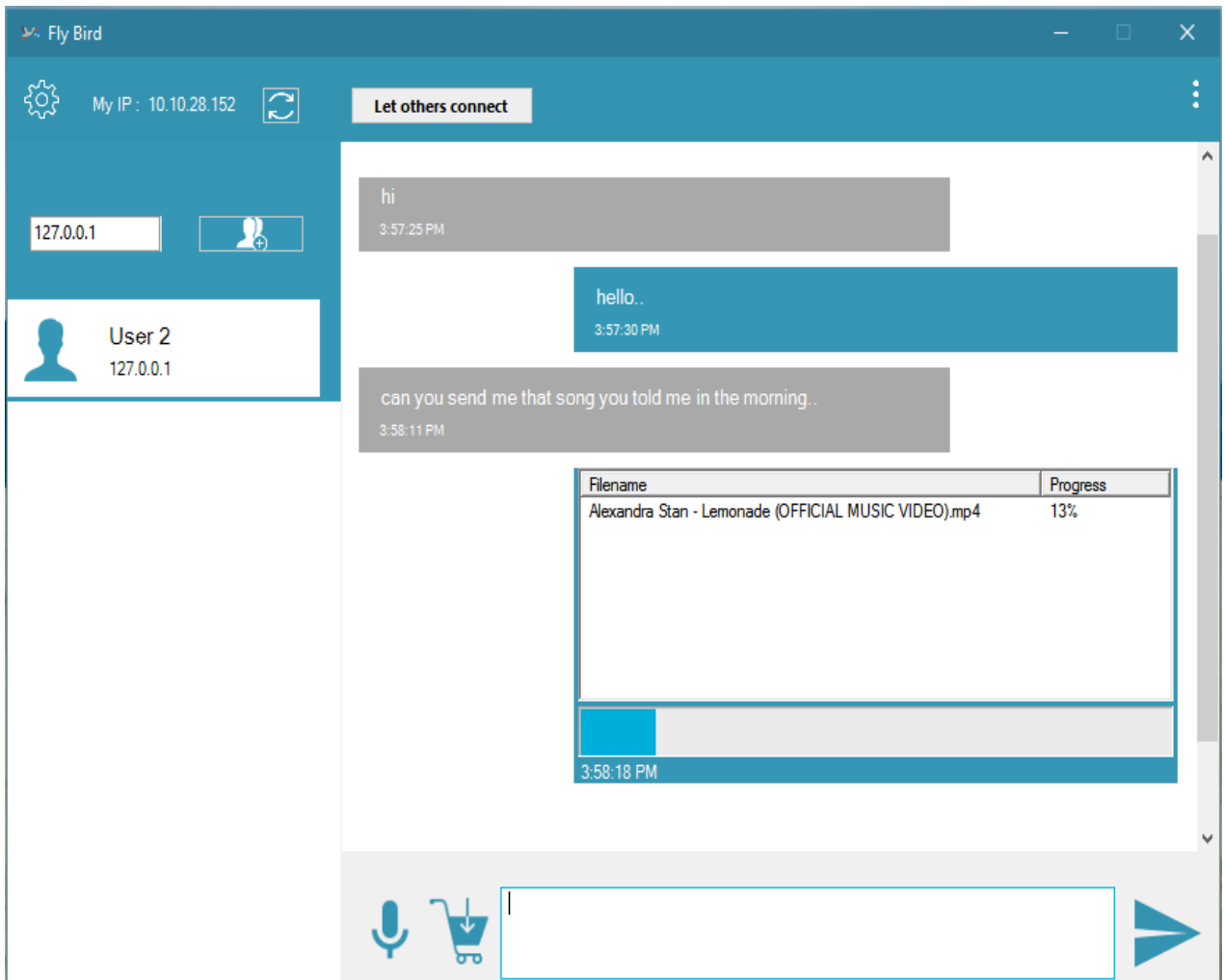


Figure 5-Chat display UI

Figure-5 shows the main User interface that users spend most of their time on this application. It is the chat display that shows sending and receiving messages of a conversation. By add to chart icon users can add files to send or they can type the text in the textbox to send as text message.

Messages are shown with the time they sent or received. File messages shows the current progress of the transferring the data.

5. System Testing and Analysis

5.1. Testing approach

This system has been testing from the beginning. It is done with every function adding. After completion of the system it is expected to test performance testing and failover testing.

Testing is done according to these topics,

- Data and Database integrity types
- Function Testing
- User Interface Testing
- Performance Profiling
- Load Testing
- Security and Access Control Testing
- Failover and Recovery Testing
- Configuration Testing

Testing tools given by Visual Studio has been using for the system testing needs.

- MSTest
- Visual studio
- Performance profiler
- Diagnostic tool

5.2. Unit Testing, Results and analysis of testing

This system has been testing from the beginning. It is done with every function adding.

Various system functions have been tested using Unit testing. MSTest was used as the testing framework. As a testing framework MSTest was really useful for this project because it saves time and effectively do the job inside Visual Studio.

Mainly encrypting decrypting functions were tested using MSTest and verified the correctness of the functions.

And another main unit test was control message testing. As this application shares several control messages between connected users it verified the correctness of the functions using unit test and after that whole system was tested for those functions.

5.3. Aspects related to performance, security, failures

This system has utilized into its expected level. Tests shows that the system runs below 50mb of RAM usage and below 25% of CPU usage.

As encryption functions work properly the system is a protected in the network to share information without security threats.

6. Conclusion and Future work

6.1. Conclusion

“flyBird” is a data sharing and communicating application using LAN. Main target is wireless LAN. It tries to minimize the complexity of the network based scenarios and give a very user friendly interface to the users to transfer the data by using their Wi-Fi devices.

In PC's there isn't a built-in application for this purpose like Bluetooth. The problem in Bluetooth is it is very slow compared to Wi-Fi and the application doesn't provide a communicating interface except only file transferring. “flyBird” tries to be the solution for these problems. After it expand for mobile platforms it can do the job more widely.

6.2. Observations, new questions, and future work

future work of the application has planned by the topics shown below.

6.2.1. Windows application

In the application that already developed has some areas to be improved. So the next version of the application will come up with improvements of the following areas.

1. UI.

It is expected to develop the application UI using WPF UI method in the next build and hope to improve the UI look and feel to more user-friendly manner.

2. Architecture

Some architectural issues are there in the system. So, it is expected to develop the architecture of the application for more reliable and less coupling manner.

3. More user-friendly design

Creating the server also can automate so that next version will be more similar to modern social chat applications that hides all the network scenarios.

6.2.2. Android version

This system is expected to develop for android mobiles also in the next step. Experiences gained from the first version of the application for Windows will be useful for the designing the application for mobile platforms. It should communicate between mobiles-mobiles and mobile-PC.

Later it is expected to develop this for Linux also.

7. References

- [1] FeePerfect “Faster Transfers” Internet: <http://tryfeem.com/> [Feb. 08, 2017]
- [2] Skype and/or Microsoft. “Celebrate the season together, wherever you are” Internet: <https://www.skype.com/en/> 2017 [Feb. 08, 2017]
- [3] WhatsApp Inc. “Simple. Secure. Reliable messaging.” Internet: <https://www.whatsapp.com/> 2017 [Feb. 08, 2017]
- [4] Viber Media S.à r.l. “Viber Connect. Freely.” Internet: <http://www.viber.com/en/> 2017 [Feb. 08, 2017]
- [5] Facebook. “Messenger” Internet: <https://www.messenger.com/> 2016 [Feb. 08, 2017]
- [6] Microsoft. “C#” <https://msdn.microsoft.com/en-us/library/kx37x362.aspx> 2017 [Feb. 08, 2017]
- [7] Microsoft. “Any Developer, Any App, Any Platform” Internet: <https://www.visualstudio.com/> 2017 [Feb. 08, 2017]
- [8] Dennis Magno. “Metro Modern UI - Metro Framework 1.4.0” Internet: <https://www.nuget.org/packages/MetroModernUI/> July. 19,2016[Feb. 08, 2017]
- [9] GitHub, Inc. “Github” Internet: <https://github.com/> 2017 [Feb. 17, 2017]