

**DEPARTMENT OF COMPUTER SCIENCE AND IT**

Społeczna Akademia Nauk

Diploma’s Thesis on Computer Science and IT

Prajwol Devkota

</>



**DEPARTMENT OF COMPUTER SCIENCE AND IT**

Społeczna Akademia Nauk

Diploma’s Thesis on Computer Science and IT

Author: Prajwol Devkota

Supervisor: Dr. Piotr Bilski

Submission Date: 10. Oct 2019

I hereby confirmed that this diploma’s thesis is my research and hard work.  
I have mentioned all the sources and materials that were used during the preparation of this documentation.

Warsaw, 10. Oct 2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

This thesis becomes a truth with the kind effort and help of many people. I am deeply grateful to all of them who helped me.

I am indebted to **Społeczna Akademia Nauk and Faculty of Computer Science and IT** for providing all the help and consultation to conduct this thesis. I have no words that can adequately carry out sincere gratitude to my supervisor and project coordinator for their valuable suggestions and encouragement for completing this thesis.

I would also like to thanks and appreciation my colleagues and people who helped me with ideas and supports.

**ABSTRACT**

MessengerLAN is an android application based on Wi-Fi Direct (also called P2P or peer to peer) architecture that uses the Wi-Fi Direct connection to allow users to share a message, audio message, video message, drawing and document with nearby devices, at a range where Bluetooth is not capable of making a connection. This application allows users to create a common group for the purpose of communication and sharing files. Within that particular created group, message, audio message, video message, drawing, and documents have been made common to all the users. This project ensures a private communication between a small group of people in a limited area and has focused communication between users without the need for internet connection. This application can be used as like chatting or file transferring application between two people or group file sharing.

**CONTENTS**

1. Introduction
   1. Background
   2. Problem Statement
   3. Objectives
   4. Scope of the application
   5. Applications
2. Literature Review
3. Algorithm
   1. Java language
   2. Android Studio
   3. Wi-Fi Direct
      1. IEEE 802.11
      2. Security
4. Implementation
   1. Software Development Process Model
   2. Algorithm
   3. Block Diagram
   4. Flow Chart
   5. Use Case Diagram
   6. Class Diagram
   7. Sequence Diagram
5. Future Work
   1. Conclusion
   2. Future Enhancements
   3. Gantt Chart
6. Reference
7. APPENDIX

**LIST OF FIGURES**

**FIGURES**

Figures 4.1 Prototype Process Model

Figure 4.3 Block Diagram of The System

Figure 4.4 Flow Chart

Figure 4.5 Use Case Diagram

Figure 4.6 Class Diagram

Figure 4.7 Sequence Diagram

Figure 5.3 Gantt Chart

**LIST OF ABBREVIATION**

ADT – Android Development Tools

IDE – Integrated Development Environment

CLI – Command-Line Interfaces

GUI – Graphical User Interfaces

LAN – Local Area Network

MP2P – Mobile Peer to Peer

P2P – Peer to Peer

SSID – Service Set Identifier

WPA – Wireless Protected Access

WBA – Wireless Broadband Alliance

IEEE – Institute of Electrical and Electronics Engineers

PSK – Pre-Shared Key

PLANCOM – Public Local Area Network Communication

JRE – Java Runtime Environment

JDK – Java Development Kit

API – Application Program Interface

OS – Operating System

GMS – Google Mobile Services

**CHAPTER 1: INTRODUCTION**

**1.1 Background**

Currently, we can observe that mobile users are increasing rapidly in both the commercial sector and the business sector. Increasing the use of mobile in the current mobile market is demanding a vast majority of applications. A mobile application helps the user to come up with a solution for tasks. In the current mobile market, we can find an application that can make out tasks easier and convenient. There are various types of apps like games, business apps, educational apps, lifestyle apps, entertainment apps, utility apps, travel apps, health and fitness apps, food and drinks apps, etc. For example, we have a calculator app that is natively available on every mobile device. We can make a calculation of some numbers within the touch of a few numbers and symbols on the mobile screen, we do not need to search for the physical calculator device. Nowadays, we all use social network applications to communicate with our friends and family from far away. Before the development of a social network application, it was difficult to communicate. We need to wait for a person whom we want to talk to and even out network provider charge sum of money for call duration. But nowadays if a person we are trying to contact is busy, then we can simply drop text message. We can have a voice or also video call from far away, this is because of development in a social network application. That is how mobile application makes our life easier.

In the current situation, we have two dominant platforms, those are Android and IOS. Application development is possible in both platforms with similar functions and features. Many common days today application and games can be found in both Android and IOS but on a closer look, Android application development has a majority.

App development is one of the most leading markets in recent times. Each day many apps are being released with their unique function. Some apps may be used by a lot of users; others might be not. One of the most important aspects of computer engineering is programming, which in turn is the prime necessity for app development.

Recently there has been a lot of app development regarding messenger types of applications like Facebook, Viber, WeChat, WhatsApp and many more.

A Wi-Fi Direct is a simple connection network that can be set up by one user for allowing other users to be connected within the same network. It can also be regarded as one form of a private network but with the limited number of people that can be connected to it as per the capability of the device set up the Wi-Fi Direct group.

When a user is connected to an intended network, a real-time message can be sent that can be seen by only a single person in the virtual room. A mobile device will transmit packets containing the message written to the server, which then collects, organizes and reassembles the data, down to the very font, text size, and color used in some instances. The message is then echoed by the server to the other using only one at a time in the chatroom. However, files are shared commonly to every user in the same network with the app being opened.

**1.2 Problem Statement**

In rural areas where there are many people with the devices that can use wireless technology but no internet access to make use of the technology, this application enables people in such areas to create a local network and be able to communicate

with the people that are connected in the Wi-Fi Direct group. Despite having a connection in small areas, it creates a huge difference in day to day operation and function of general people. Not only in a rural area, but it can also be hugely beneficial in private communication in small areas like offices, libraries, schools and even in a business organization. Thus, this application tackles the basic need of internet and privacy by using Wi-Fi Direct and can create a huge impact on the day to day functioning of general people.

**1.3 Objectives**

To provide private communication between two users and allow file sharing between all users in the same network using the Wi-Fi Direct.

**1.4 Scope of the application**

* The application user can be the general population, office premises or College Library.
* This Application can be also be used in place, where there is no internet connection available.
* This application can be used for students in the rural area where there are devices that use Wi-Fi Direct but there is no internet connection.

**1.5 Applications**

* This app will allow many users to connect on the same Wi-Fi Direct group.
* Users can send a message to any of the users connected to the group.
* Users can also share files with others.

**CHAPTER 2: LITERATURE REVIEW**

Wi-Fi Direct initially called Wi-Fi P2P (Peer to Peer), is a Wi-Fi standard enabling devices to easily connect with each other without requiring a wireless access point. Wi-Fi Direct allows two devices to establish a direct Wi-Fi connection without requiring a wireless router. Hence, Wi-Fi Direct is single radio hop communication, not multihop wireless communication, unlike wireless ad hoc networks and mobile ad hoc networks. Wi-Fi ad hoc mode, however, supports multi-hop radio communications, with intermediate Wi-Fi nodes as packet relays. Wi-Fi becomes a way of communicating wirelessly, much like Bluetooth. It is useful for everything from internet browsing to file transfer and to communicate with one or more devices simultaneously at typical Wi-Fi speeds. One advantage of Wi-Fi Direct is the ability to connect devices even if they are from different manufacturers. Only one of the Wi-Fi devices needs to be compliant with Wi-Fi Direct to establish a peer-to-peer connection that transfers data directly between them with a greatly reduced setup. Wi-Fi Direct negotiates the link with a Wi-Fi Protected Setup system that assigns each device a limited wireless access point. The "pairing" of Wi-Fi Direct devices can be set up to require the proximity of near field communication, a Bluetooth signal, or a button press on one or all the devices [1].

Wi-Fi Direct is built upon the same Wi-Fi technology used by most modern consumer electronic devices to communicate with wireless routers. It allows two devices to communicate with each other, provided that at least one of them is compliant with the standard to establish a peer-to-peer connection. Before Wi-Fi Direct, it was possible to establish a similar connection with ad-hoc networking, a decentralized type of wireless network that doesn’t rely on a pre-existing infrastructure, but WiFi Direct makes decentralized wireless networking accessible even to people who are not technically inclined [2].

Wi-Fi Direct device connections can happen anywhere, anytime - even when there is no access to a Wi-Fi network. Wi-Fi Direct devices emit a signal to other devices in the area, letting them know a connection can be made. Users can view available devices and request a connection, or may receive an invitation to connect to another device. When two or more Wi-Fi Direct-certified devices connect directly, they form a Wi-Fi Direct Group using Wi-Fi Protected Setup™ and the latest Wi-Fi security [3].

**CHAPTER 3: ALGORITHM**

**Tools used**

**3.1 Java language**

Java is one of many high-level programming languages that use abstraction (reducing and factoring details) to create software for computers and many other electronic devices. Compared to low-level programming languages, high-level programming languages use a mixture of natural language (any language used by humans) and typical computer terms to create an easier to understand and simpler form of programming. Like human languages, programming languages have different levels of writing difficulty; some are easy to write, while others require extensive computer knowledge to understand. In programming, high-level languages are easier to understand, learn and write because of the use of natural language. But, as the use of natural language decreases and more words and computer statements are added, the difficulty level of language increases.

**3.2 Android Studio**

Android studio is the official IDE for Google's Android operating system, built on Jet

Brains' IntelliJ IDEA software and designed specifically for Android development. It is

available for download on Windows, Mac OS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as a primary IDE for native Android application development [4].

**Features:**

* Android-specific refactoring and quick fixes.
* Lint tools to catch performance, usability, version compatibility and other problems.
* ProGuard integration and app-signing capabilities.
* Template-based wizards to create common Android designs and components.
* A rich layout editor that allows users to drag-and-drop UI components, an option to preview layouts on multiple screen configurations.

**3.3 Wi-Fi Direct**

Wi-Fi Direct (also known as peer-to-peer or P2P) allows your application to quickly find and interact with nearby devices, at a range beyond the capabilities of Bluetooth.

The Wi-Fi peer-to-peer (P2P) APIs allow applications to connect to nearby devices without needing to connect to a network or hotspot. If your app is designed to be a part of a secure, near-range network, Wi-Fi Direct is a more suitable option than traditional Wi-Fi ad-hoc networking for the following reasons:

* Wi-Fi Direct supports WPA2 encryption. (Some ad-hoc networks support only WEP encryption.)
* Devices can broadcast the services that they provide, which helps other devices discover suitable peers more easily.
* When determining which device should be the group owner for the network, Wi-Fi Direct examines each device's power management, UI, and service capabilities and uses this information to choose the device that can handle server responsibilities most effectively.
* Android doesn't support Wi-Fi ad-hoc mode.

[5].

**3.3.1 IEEE 802.11**

IEEE 802.11 is a set of media access control and physical layer specifications for

implementing wireless local area network computer communication in the 900 MHz and 2.4, 3.6, 5, and 60 GHz frequency bands. They are the world's most widely used wireless computer networking standards, used in most home and office networks to allow laptops, printers, and smartphones to talk to each other and access the Internet without connecting wires. They are created and maintained by the Institute of Electrical and Electronics Engineers (IEEE) LAN Standards Committee (IEEE 802). The base version of the standard was released in 1997 and has had subsequent amendments. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand. While each amendment is officially revoked when it is incorporated in the latest version of the standard, the corporate world tends to market to the revisions because they concisely denote the capabilities of their products. As a result, in the marketplace, each revision tends to become its own standard [6].

**3.3.2 Security**

Terminology behind Wi-Fi Direct is that it makes simple tasks with a simple connection. Such as printing an image from a mobile device or laptop to a wireless printer within a couple of clicks. For performing such tasks, we need all is the device with Wi-Fi Direct support and general knowledge of Wi-Fi Direct, and we do not need a working internet connection. It is already a common feature in all mobile devices, so we do not need to worry about installing third-party software or driver to make it function.

It is a secure way of making a connection because it uses Wi-Fi Protected Setup and WPA2 to prevent from making an unauthorized connection and helps to make a safe and secure connection within a group.

Android devices with Android 4.0 Ice Cream Sandwich and above supports Wi-Fi Direct function. To make a connection from one android device to another all we need to do is go to Setting function, then Wifi tab and search for Wi-Fi Direct option. Once we are in Wi-Fi Direct tab, we can see devise those support Wi-Fi Direct listed. From a list of devices, we can invite the desired device and make a secure connection. In such a connection we can share data at a speed of 250mbps which is ten times faster than Bluetooth 4.0.

**CHAPTER 4: IMPLEMENTATION**

**4.1 Software development process model**

The Prototype Model is a System Development Methodology (SDM) within which a

paradigm output (or an early approximation of a final system or product) is constructed, tested, and then reworked. It is done until an appropriate paradigm is achieved to help develop the entire system or product. This model works best in situations when all the details or requirements are not known well in advance. It is majorly a trial-and-error a process that works in an iterative fashion.

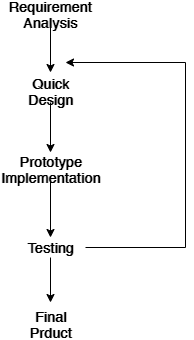


Figure 4.1 Prototype Process Model

* Requirement analysis

Our prime requirements for the application to be fully functional as per the

objectives within its regards are allowing the app to have a connection between users via Wi-Fi Direct allowing the users to have the messaging ability, file sharing between all the users in the same network. The most important analysis is that the app had to have a share option, on the opening of which would let the user open the shared file and save or delete it.

* Quick design

As per the requirements of the system, I designed the application keeping

primarily the messaging feature in mind and developed a simpler version of the

app. Then again designed a simple UI along with some graphical improvement and added file sharing feature that allowed file sharing between all the users that are in the same P2P connection group. Then developed prototype was tested for errors and then designed a proper design for the users to have saved or deleted the shared file was made along with giving a proper design for the list of people connected to be seen and how the chat between users would look like.

* Prototype implementation

Since application required various functionality like chatting, file sharing and drawing, there was the need for using a prototype model for application to be built. First, messaging code was written and simple buttons were added with the simpler user interface and a prototype was developed.

* Testing and the final product

The first prototype only allowed the chatting feature. For the next step, file sharing was added and the graphical user interface was made more in detail and the second porotype was developed. Then checked it for any errors and flaws and then again added more non-functional requirements like audio recording, video message recording, and drawing within the app and sharing it on the group. Thus, that porotype was later checked for more errors and after full functionality of the app, the app was decided to settle for that prototype as the final product.

**4.2 Algorithm**

Algorithm for the overall development process of MessengerLAN android application.

Steps

1. Start.

2. Create class MainActivity() to set getSetChatName() to take username for user, a button for connecting Wi-Fi Direct, set goToChat() to open chat group, set getmReceiver() to receive message from others users and set onPause() and onResume() to pause and play audio respectively.

3. Create class WebViewActivity() to display the username and exit button to quit an application.

4. Create class MessageService() to set enable communication port for the purpose of receiving a message from the other users.

5. Create class ChatActivity() to to set sendMessage() to take text input and send, set deleteMesage() function to delete message, and set downloadFile() to download file from group.

2. Create class FilePickerActivty() to set choose file() function to choose a file from the directory, set file directory() to interact with default file manager function of the system and set on back pressed() to allow a user to return back to the application.

6. Create class ViewImageActivty() to open the image message sent by another client.

7. Create class RecordAudioActivty() to allow a user to record an audio message and send it to another client in a group.

8. Create class PlayVideoActivity() to allow to play video message shared by another client in a group.

9. Create class DrawingActivity to set paintClicked() function to allow a user to draw on the screen with the full support of tools.

This algorithm shows the surface part of the MessengerLAN application and shows basic and general function and implementation.

**4.3 Block diagram**

****

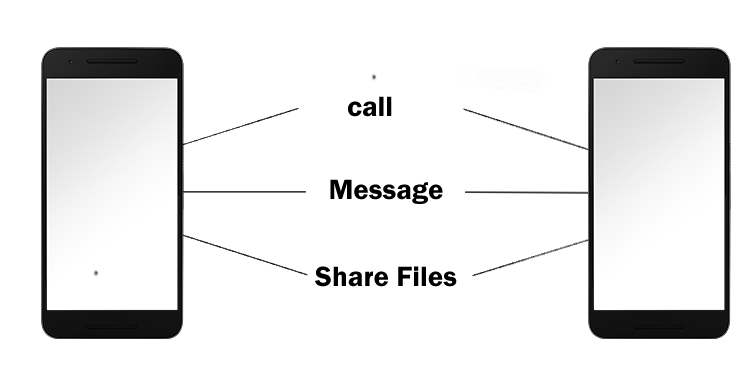


Figure 4.3 Block diagram of MessengerLAN system

**4.4 Flow chart**

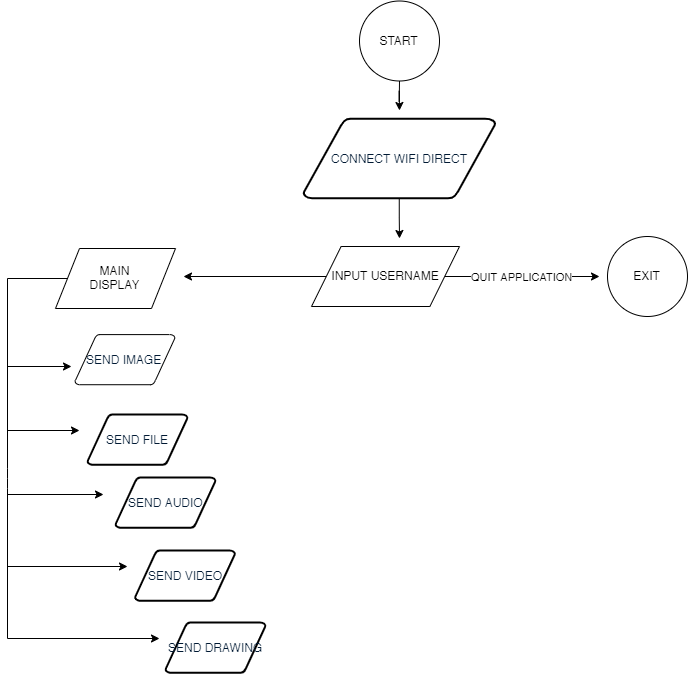
****

Figure 4.4 Flow Chart

The above figure shows the flow chart diagram for the application. From top to bottom, the first step is to START application. Making a connection comes in the second step. After making a successful connection via Wi-Fi Direct user is allowed to set a username or exit application by choosing EXIT. After application gets a username, user gets in MAIN DISPLAY where SEND MESSAGE, SEND IMAGE, SEND FILE, SEND AUDIO, SEND VIDEO AND SEND DRAWING function are available.

**4.5 Use case diagram**

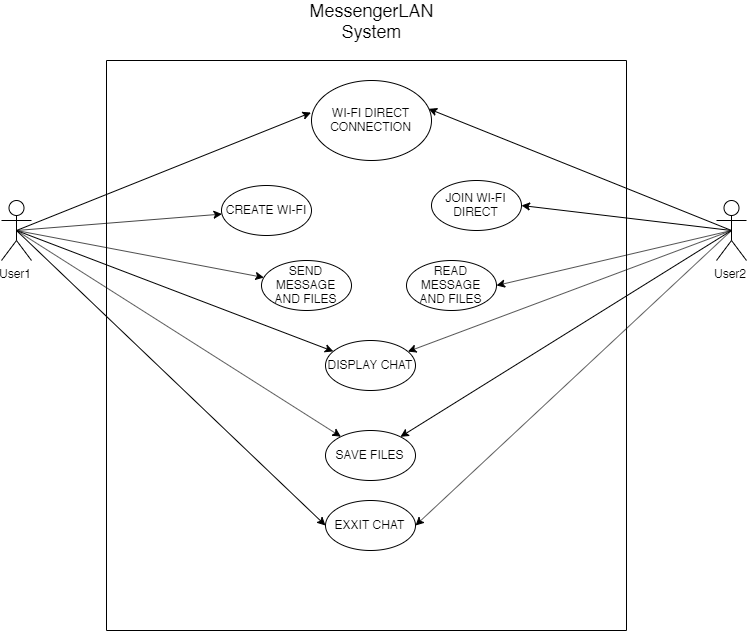
****

Figure 4.5 Use case diagram

The above figure shows the use case diagram for the application. In a simple case, there are two actors, both of them being users in which one is a host and another is the receiver or simply user. The host can create a Wi-Fi Direct network to allow another user to connect to each other via P2P. All other users can now send messages or files as per their need and save the respective files and at the end of the time period, a host can break the connection that breaks the connection between users or simply users can leave the chat group.

**4.6 Class diagram**

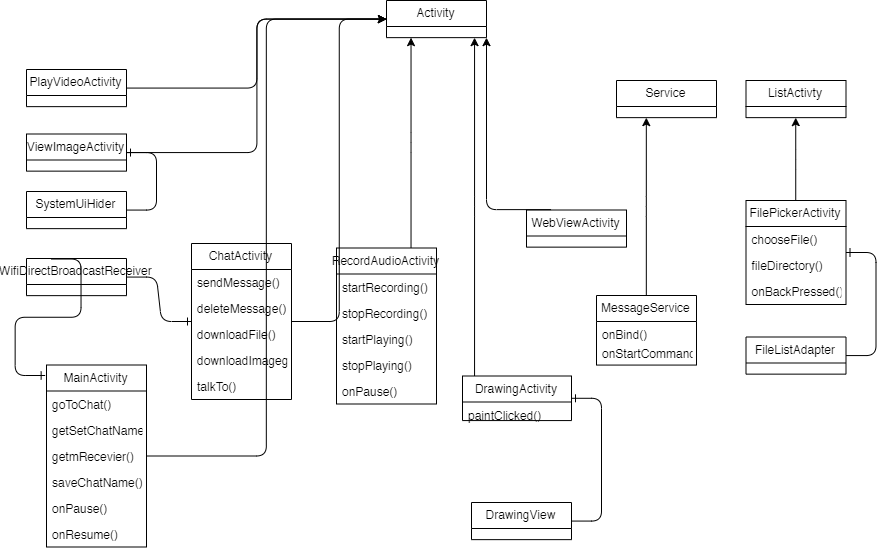
****

Figure 4.6 Class diagram

The above diagram is a graphical representation of a static view of MessengerLAN system that shows different functions and aspects of the application. It also illustrates the operations and attributes of the classes.

**4.7 Sequence Diagram**

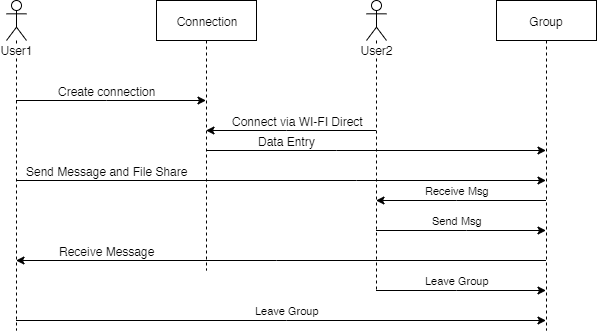
****

Figure 4.5 Sequence Diagram

The above sequence diagram shows the simple form of connection that can be done between two users where one is host and other is receiver showing all the available\*-

functions that can be done between two like messaging, file sharing, etc.

**CHAPTER 5: FUTURE WORK**

**5.1 Conclusion**

As per my objective, I have developed my project with the main objective to create a small group using Wi-Fi Direct and allowing files shared by one user to be visible with all the users in the same P2P connection network. Also, a secondary objective that was to be able to record an audio or video message and share it with a group has been completed with additional availability of usage of the given application. Groups of people limited in number are now able to share files and communicate at one instance while files being shared with all of them using this application. We have used JAVA and Android Studio for the making of this application.

**5.2 Future Enhancements**

 This app further can be made the chatting system a common system i.e. all the connected users can view the same message and give their inputs making it a group chat. Private calling between two users from group features can be added. All those messaging and file-sharing functions can be made private between only two users. This app can have its own lock system which requires a password to open the app.

**5.3 Gantt chart**

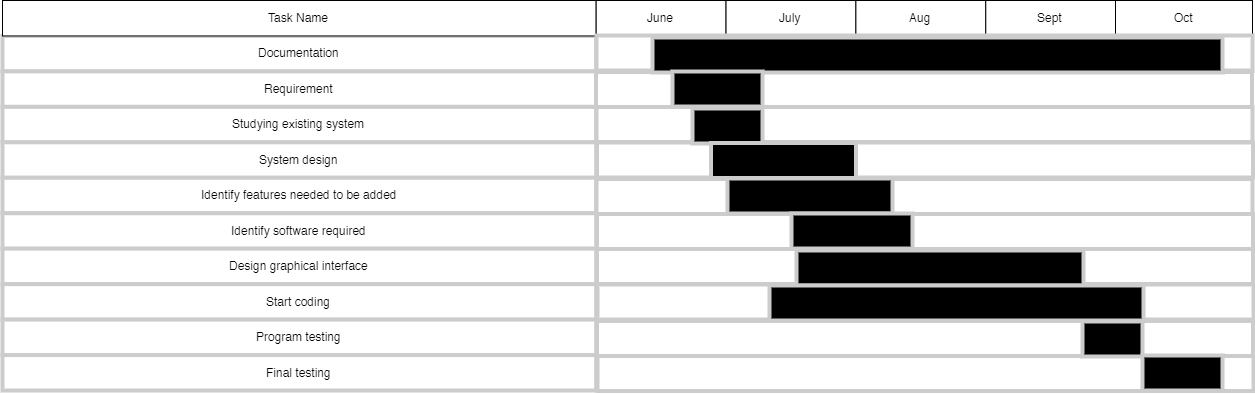
****

Figure 5.3 Gantt chart

This Gantt chart is based on the estimated date and duration of the events

involved in project development. The submission date and duration can be changed according to the demand.

**REFERENCES**

[1] "wifi direct" [Online]. Available: https://en.wikipedia.org/wiki/Wi-Fi\_Direct

[Accessed on June 2019]

[2] “how wifi direct works” [Online] Available:

https://www.netspotapp.com/what-is-wifi-direct.html [Accessed on June 2019]

[3] “wi-fi direct connections happen anywhere, anytime” [Online] Available:

https://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[Accessed on July 2019].

[4] “android studio development tools [Online]. Available:

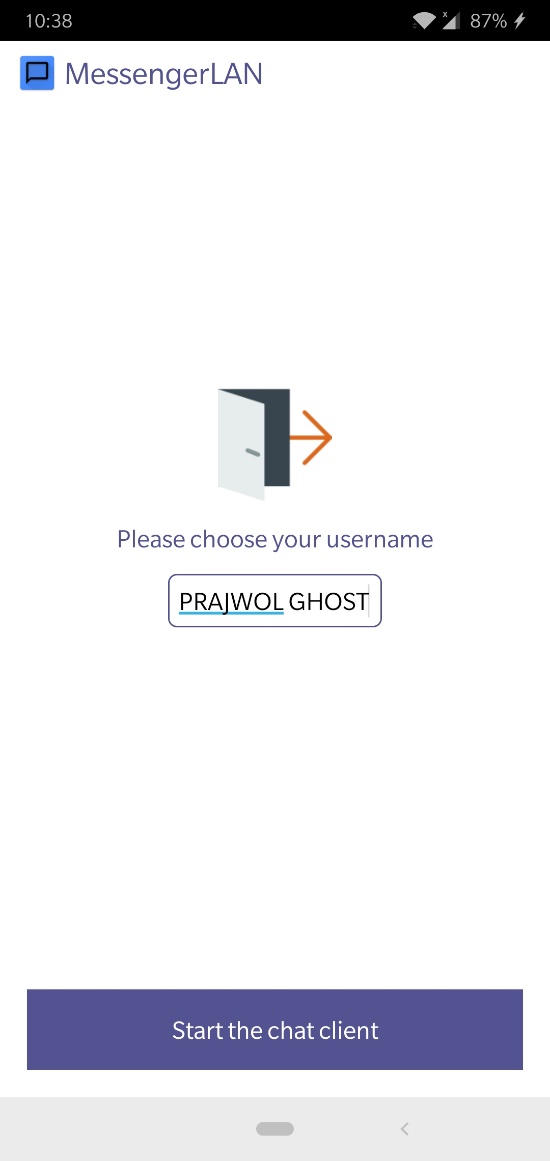
https://en.wikipedia.org/wiki/Android\_Studio [Accessed in July 2019].

[5] “create P2P connections with wifi direct” [Online]. Available: https://developer.android.com/training/connect-devices-wirelessly/wifi-direct

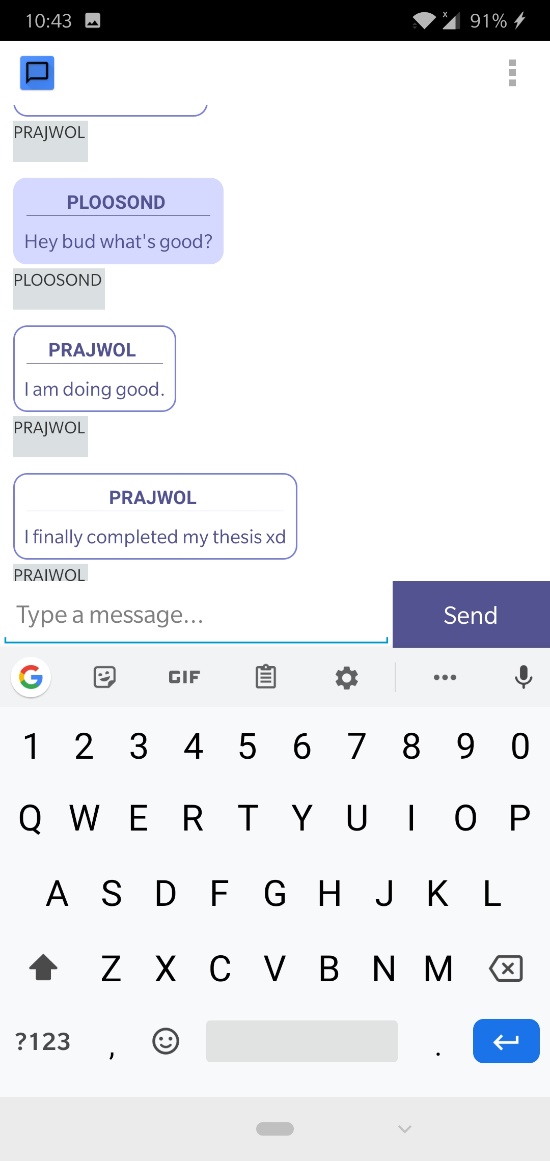
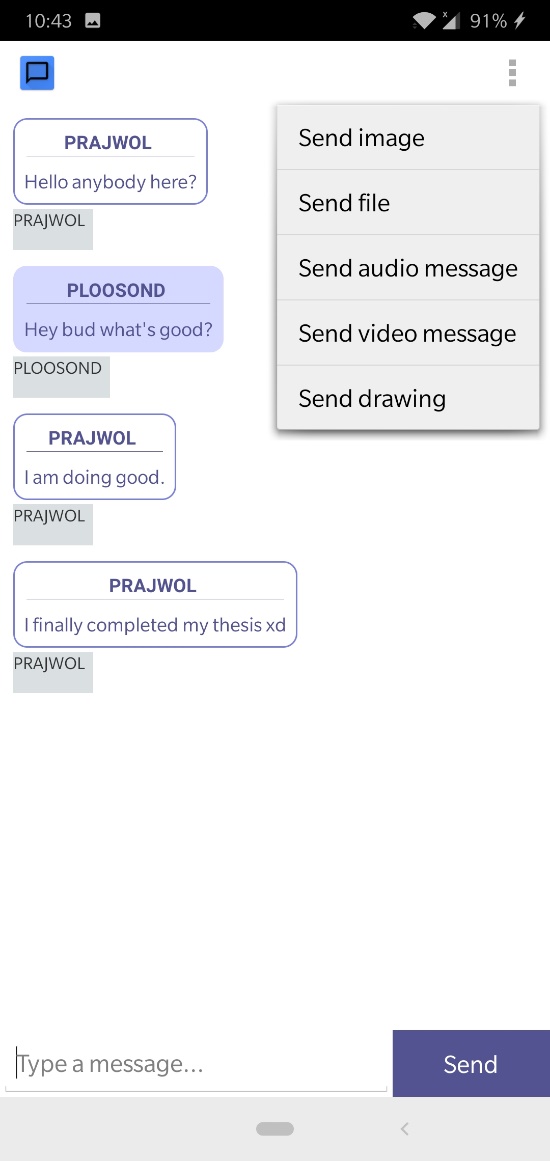
[6] “media access control”, [Online] Available:

https://en.wikipedia.org/wiki/IEEE\_802.11 [Accessed in July 2019].

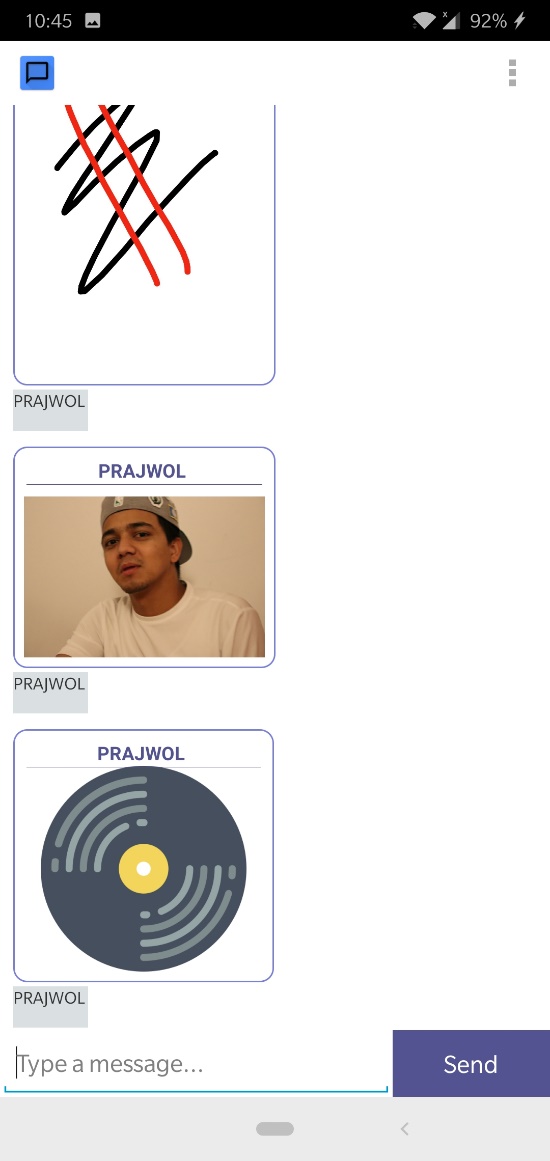
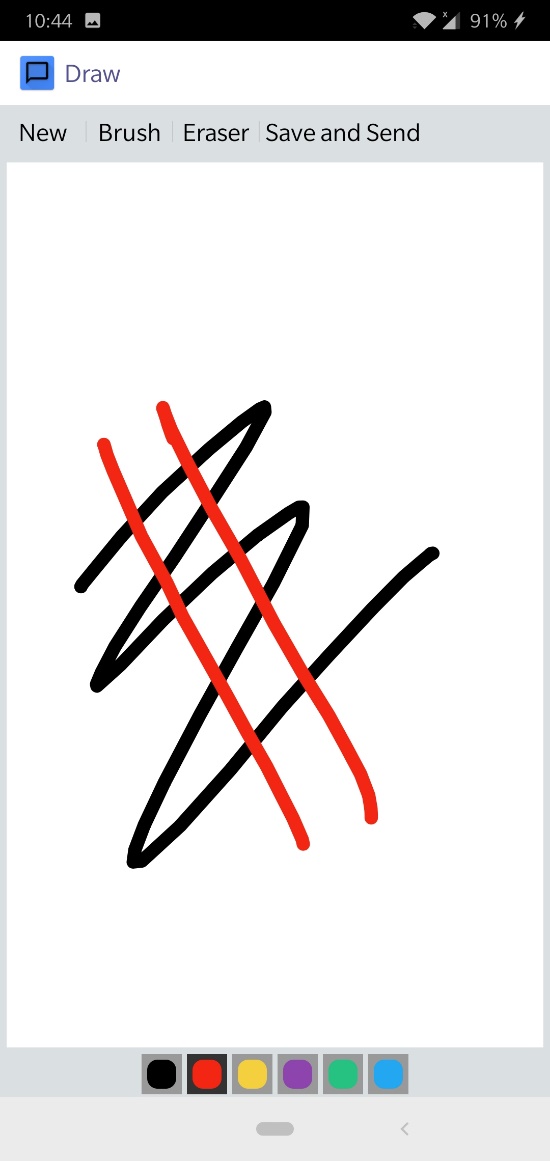
**APPENDIX:**

** **

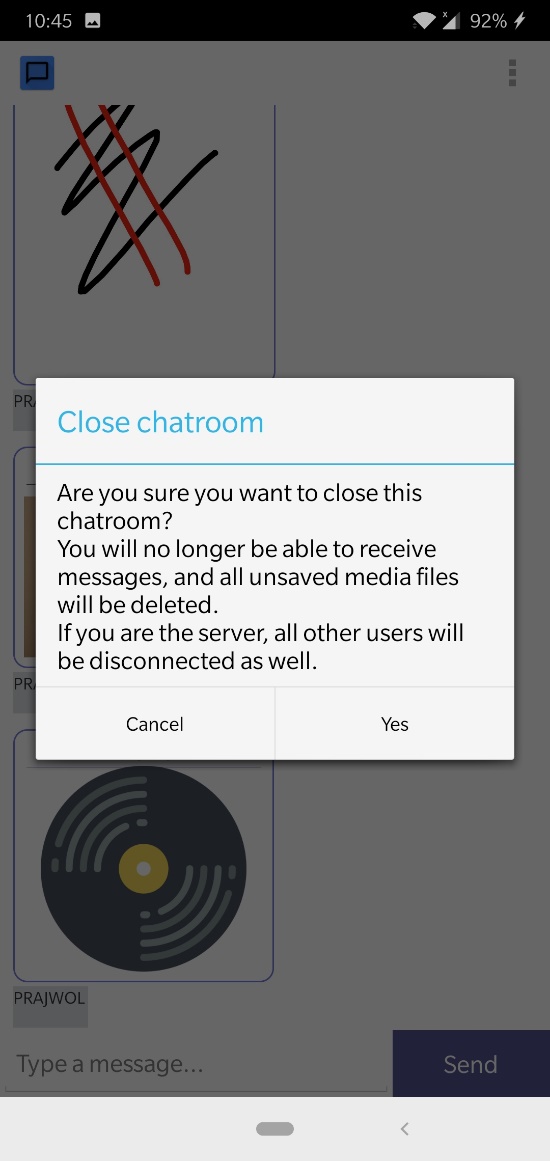
Wi-Fi Direct connection functionality Choose username and start group chat

Conversatation layout Overall functions of MessengerLAN app

Shared files layout Drawing functionality layout

Audio recording layout Functionality for exiting app