DEVELOPMENT OF AD-HOC VOICE NETWORK FOR RAPID DEPLOYMENT

FINAL YEAR PROJECT

FINAL INTERIM REPORT

GROUP MEMBERS:

Capt Ibrahim

CaptSumair

Capt Adeel

Capt Bilal

CaptZahir

BETE 51 A

Military College of Signals

Supervised By

Lt Col Dr Hasnat Khurshid

DATED: ____ June 2018

_____

(Lt Col Dr Hasnat Khurshid)

TABLE OF CONTENTS

1. Introduction

The project will develop an ad-hoc communication solution using android as front end and 802.11 as underlying protocol. The mode of communication will be voice. The project involves in-depth understanding of networking protocols as well as development of android as front end. Product development for use in calamity hit areas with no internet or other communication infrastructure. The aim of this document is to present detailed description of the AD-HOC VOICE NETWORK application keeping cost-effectiveness as a major concern.

1.1 **Purpose**

This archive covers the prerequisite for venture improvement of convention for voice establish for ad-hoc actions. The possibility of the undertaking is to generate an ad-hoc based two path correspondence over voice utilizing the 802.11 as under laying convention. This archive is intended to layout the highlights and prerequisites of venture to fill in as a manual for the engineers on one hand and a product approval report for the forthcoming customer on the other. The motivation behind this report is to introduce a point by point portrayal of the AD-HOC VOICE NETWORK application. It clarifies the reason and highlights of the application, the interfaces of the application, what the application do, the requirements under which it must work. This archive is expected for both the partners and the engineers of the framework and be proposed to the Evaluation Panel for its endorsement.

1.2 Product Vision

Name	AD-HOC VOICE NETWORK
Is	Two way communication application full duplex.
For	This application is being developed primarily for adhoc arrangements, but can be used commercially as well as in domestic.
What	This application is used by the android supported smartphone.
That	The application is intended to provide full duplex voice communication over IEEE 802.11 as underlying protocol.

1.3 References

All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document

1.3.1 Android studio

- **https://developer.android.com/studio/**
- https://en.wikipedia.org/wiki/Android_Studio

1.3.2 Adobe Photoshop

- http://www.adobe.com/products/photoshopfamily.html
- http://www.photoshop.com

1.3.3 **Links to web pages**

All links have been provided with underlined font, the title of the web page or e-book is written at the top of the link and the title may be searched on google to pinpoint to the exact address.

2. Overall Description

2.1 **Product Perspective**

The project is basically being built to help the host to use a rapid ad-hoc based network using IEEE 802.11 protocol for smooth two-way voice communication in calamity hit areas mountainous terrain and in forest or where there is no mean of communication.

2.2 **Product Features**

The main features of this application are highlighted below:

- Full duplex voice calling

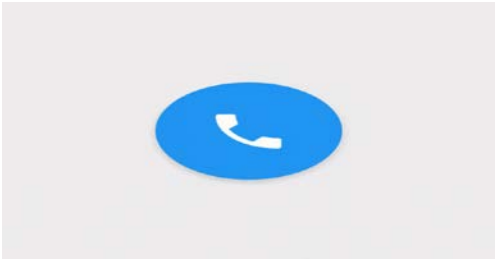
- Auto Node discovery

- Distributed directory

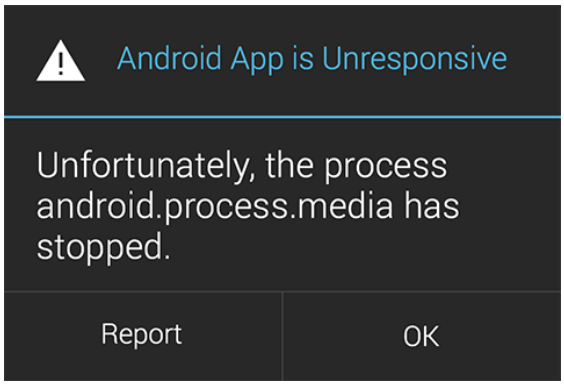
- Multiple hosts

- Device Compatibility

- Call alert



- Robust



- Ad-hoc (infrastructure less)

2.3 **Operating Environment**

2.3.1 **Hardware**

The application operates, directly with the following:

- **Android Compatible Mobile**: smart phone that support the android features with microphone,speaker,wifi network working condition for better result if integrated with noise canceller and audio booster sensors

- **WIFI Router**: The data (voice) will be transferred from hosts using an IEEE 802.11 protocol for a rapid based communication without the involvement of server/without internet.

2.3.2 **Software**

- Android operating system 5.0 (lolipop) or above.
- Android studio 3.0.1
    - **<u>Reasons to use the software:</u>**
        - A fast and feature-rich software compatibility support
        - A unified environment where you can develop for all Android devices
        - Moment Run to push changes to your running application without building another APK.
        - Extensive testing tools and frameworks
        - Build up devices to get execution, ease of use, adaptation similarity.

2.4 **Design and Implementation Constraints**

Ad-Hoc communication requires a gist of highly advanced set of tools. Keeping in view the aforesaid, few shortcomings are:

Constraint	**Explanation**
Technical	• Nevertheless, simultaneous transmission and reception in dense deployment scenarios increases the inter- cell interference compared to a traditional communication, due to a larger number of nodes simultaneously transmitting. • A standout amongst the most surely understood limitations in RF that it is by and large difficult to transmit and get in the meantime on similar frequencies in light of the fact that the demonstration of transmission makes a huge measure of impedance for the recipient, keeping the beneficiary from "hearing" the coveted flag originating from nature.
OS Required	Application can only run on android phones having OS version lollipop (5.0) and above.

- The **basic concepts of Networking** and acquaintance with NW devices
- Understanding of **802.11 protocol** details related to project design. (security, delay)
- Design and Implementation of **Auto node discovery algorithm**
- Mechanism for Maintaining a **distributed directory**
- Android development
- Efficiently **Accessing Hardware resources**(mic/speaker) in Android platforms (i.e without slowing down the device and min delay)
- IP **Communication** using android devices
- **Voice transfer**with minimum delay on network
- Understanding and implementing **Noise cancellation**technique
- **Performance analysis** (deterioration) with inclusive number of nodes
- Design, test, trial and troubleshooting

3. External Interface Requirements

3.1 **User Interfaces**

- Register on network by entering a name.
- Search for available users in real-time to connect with over the network.
- Call interface.

3.2 **Software Interfaces**

- Adobe Photoshop be used to design graphics of 2D UI elements that can be imported in application as images/textures.
- Once the application has been fully designed, it can be compiled to .(dot)apk file by providing location of android SDK which in our case is located on the local machine.

3.3 **Hardware Interfaces**

- Wi-Fi communication router.
- Hot-spot functionality in smartphone.

3.4 **Communication Interfaces**

- Communication between smartphone running the application be done using Wi-Fi or mobile hot-spot.

4. System Features

Following are the system features of AD-HOC VOICE NETWORK with detailed descriptions.

- Full duplex voice calling.
- Auto Node discovery.
- Distributed directory.
- Multiple hosts.

- Device Compatibility.
- Call alert.
- Robust.
- Ad-hoc (infrastructure less)

4.1 **Full duplex voice calling**

This application will provide full duplextwo-way communication which in telephony and data communications, full duplex is the ability for both ends of a communication to simultaneously send and get data without debasing the quality or clarity of the substance. Full duplex gadgets, consequently, can impart forward and backward all the while.

Duplex frameworks are utilized in numerous interchanges systems, either to take into account a correspondence "two-route road" between two associated parties or to give an "invert way" for the observing and remote alteration of hardware in the field.

There are a couple of favourable circumstances to full-duplex. Right off, time isn't wasted, since no ends should be retransmitted, as there are no impacts. Besides, the full information limit is accessible in the two bearings because the sending and receiving capacities are isolated. Thirdly, stations (or hubs) don't need to hold up until the point when others finish their transmission, since there is just a single transmitter for host.

4.2 **Auto Node discovery**

Programmed and decentralized revelation and checking of hubs occurrences with worked in help for a variable number of ace procedures, benefit promoting and channel informing so you have an entire cluster of hub forms running however you include no chance to get inside each procedure to figure out where alternate procedures are or what they can do. This application aims to make discovery of new processes as simple as possible.

Auto Node Discovery designs the IP locations of the hubs in a group, consequently includes new hubs, and updates the bunch hub list.

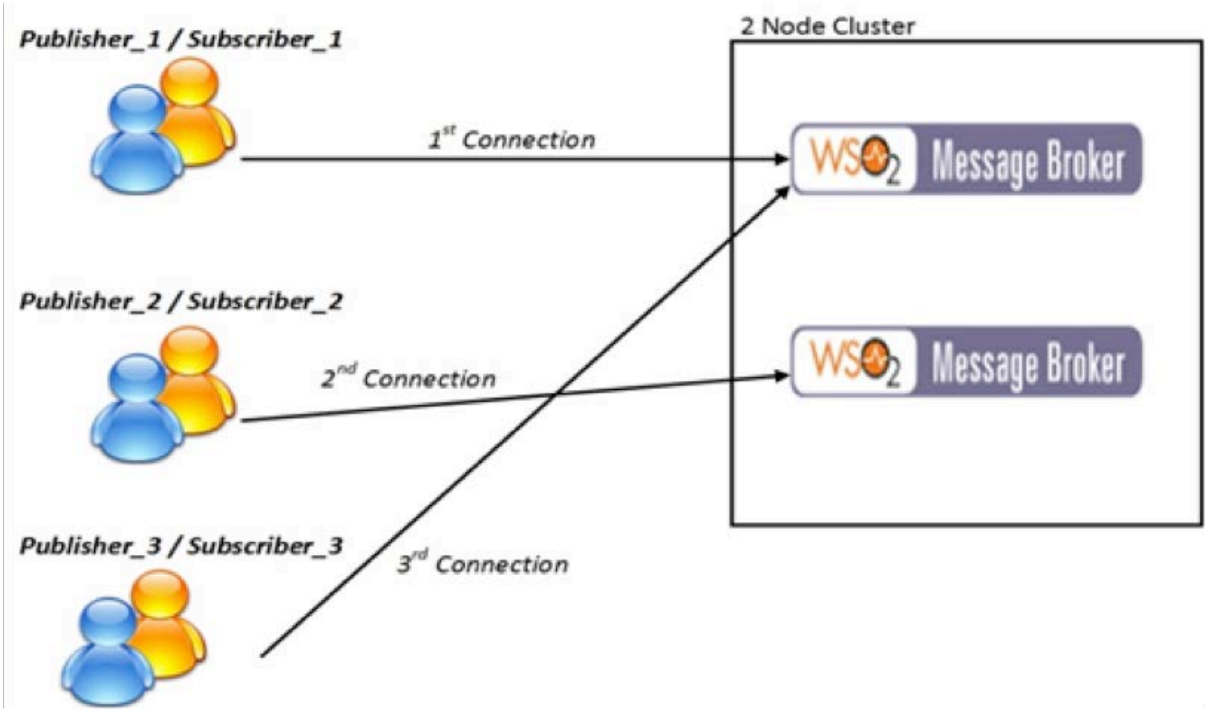


Figure 1: Auto Node Discovery

4.3 **Distributed directory**

This application provides a distributed directory in which data is partitioned across multiple directory application. To make the distributed directory appear as a single directory to client applications, one or more substitution application is provided which have knowledge of all the application and the data they hold.

Substitution application distribute incoming requests to the proper application and gather the results to return a unified response to the client. A set of backend application hold their portions of the distributed directory.

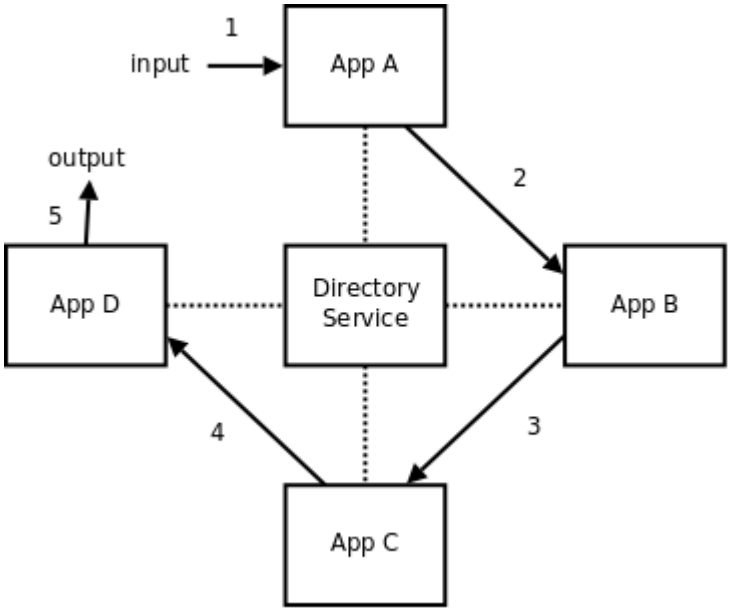


Figure 2: Concept of Distributed Directory

4.4 **Multiple hosts**

The host in this is basically the persons who want to have an ad-hoc based communication using an IEEE 802.11 protocol. Host should be able to

successfully communicate with other hosts in the range of the WIFI router for a rapid based communication system.

4.5 **Device Compatibility**

Duplex frameworks are utilized in numerous interchanges systems, either to take into account a correspondence "two-route road" between two associated parties or to give an "invert way" for the observing and remote alteration of hardware in the field.

4.6 **Call alert**

The cell phone can alarm you of occasions, for example, an approaching call or an approaching message in various ways. The two most prevalent ones are vibration and ringing. While vibration is basically obvious, the ringing cautions can be of a few sorts relying upon the cell phone.

In the past cell phones used to enable just monophonic tones to be set as ringing cautions. With the progression of innovation, polyphonic ringtones likewise ended up bolstered.

Later on as cell phones additionally developed, they began utilizing MP3 tones for different alarm purposes. As progressively (probably prevalent) sound configurations ended up accessible, (for example, AAC), they were additionally added to the cell phones' ready framework.

As of late, cell phones have even begun to utilize short video cuts as call cautions (not to be mistaken for video calls).

Regardless of what sort of ringing tones the cell phone utilizes, clients have dependably appreciated modifying their ringtones by growing the present ones provided by the maker.

This application is good with the default ring tone of the gadget utilizing and it will caution the client of the approaching call.

4.7 **Robust**

A powerful item can be one that doesn't break effectively. In this manner, a working framework in which any individual application can come up short without irritating the working framework or different applications can be said to be robust. Android studio adaptation had made it conceivable further Robust is an Android Hotfix arrangement with high similarity and high dependability. Vigorous can settle bugs instantly without distributing apk. It is used to check the different bugs and made it pure as it can be to facilitate the user to the utmost.

4.8 **Ad-hoc (infrastructure less)**

This application gives an adhoc based system which in advanced cell specially appointed or PDA impromptu systems (SPANs) develop from the hidden idea, engineering and innovation behind a remote impromptu system. An advanced mobile phone is a gadget. Once installed with specially appointed systems administration innovation, an advanced mobile phone can make impromptu systems among different gadgets. PDA impromptu systems use the current equipment (fundamentally Bluetooth and Wi-Fi) in monetarily accessible cell phones to make shared systems without depending on cell transporter systems, remote passageways, or conventional system foundation. Traverses utilize the system behind Wi-Fi impromptu mode, which enables telephones to talk specifically among each other, through a straightforward neighbor and course revelation component.

Ranges contrast from conventional center and talked systems, for example, Wi-Fi Direct, in that they bolster multi-bounce steering (specially appointed directing) and transfers and there is no idea of a gathering pioneer, so companions can join and leave freely without crushing the system.

Equipped for going off-matrix and empowering shared correspondences without depending on cell bearer systems, remote passages, or customary system foundation. Discretionary Internet access through entryway gadgets, for example, portable hotspots in the work. Discretionary stationary or compact foundations, for example, switches, work extenders, or other non-telephone equipment. Use the gadgets that individuals bear on their individual and utilize each day. Essentially utilize Bluetooth and Wi-Fi since the cell range

is authorized and controlled by cell suppliers and FCC directions. Set up and tear down are on-request. Join and leave voluntarily. Directing convention might be executed at the Network Layer or Link Layer.

Figure 3: Concept of Adhoc Network

4.9 **Proposed working mechanism**

- To install a Wi-Fi router.
- Make compatibility with the mobile network to show the available network.
- Connect with the network.
- Open the application.
- To add the name of the contact to begin.
- The application doesn’t work until you enter your name on the network.
- Enter your name and register to the network.
- Similarly, all on the network will update their names on the network.
- Their names will be displayed on the network and broadcasted to all on network.
- Now the host can make individual call to other host by clicking the contact.
- Host will get an option to make a call.
- After the call, the next host B will receive an incoming call from the host A.
- Host B will receive an incoming call alert with a ringtone either accept or reject the call.

- After accepting he will be able to make a full duplex smooth two-way communication.
- When host from the network exit his name will be deleted from the network.

Figure 4: AVN Adhoc Voice Network cloud

5. LAYOUT

Following is the general layout of our project:

5.1 **Block Diagram:**

The following block diagram shows the general idea of our project.

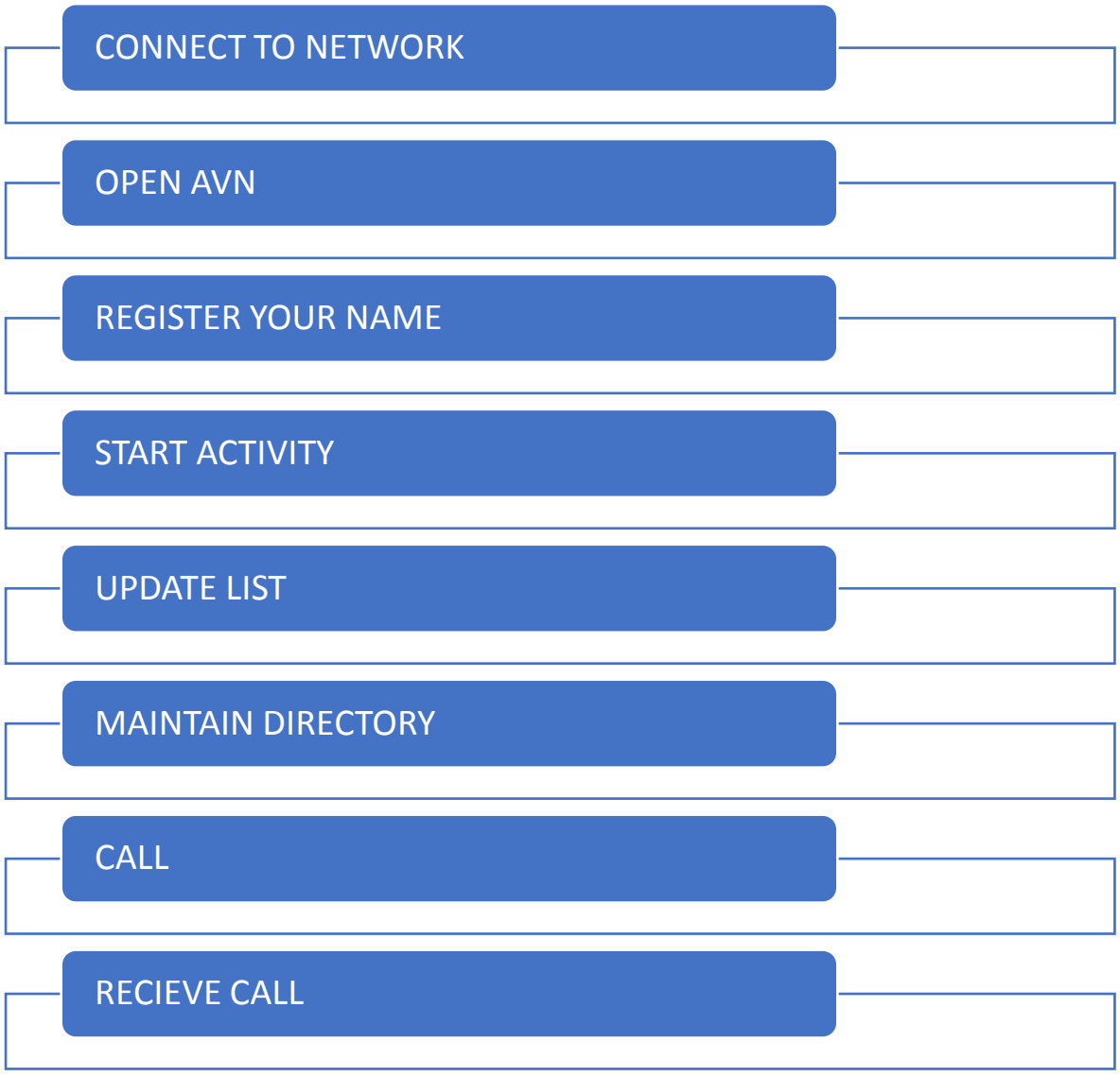


Figure 5:General Idea Of Our Project

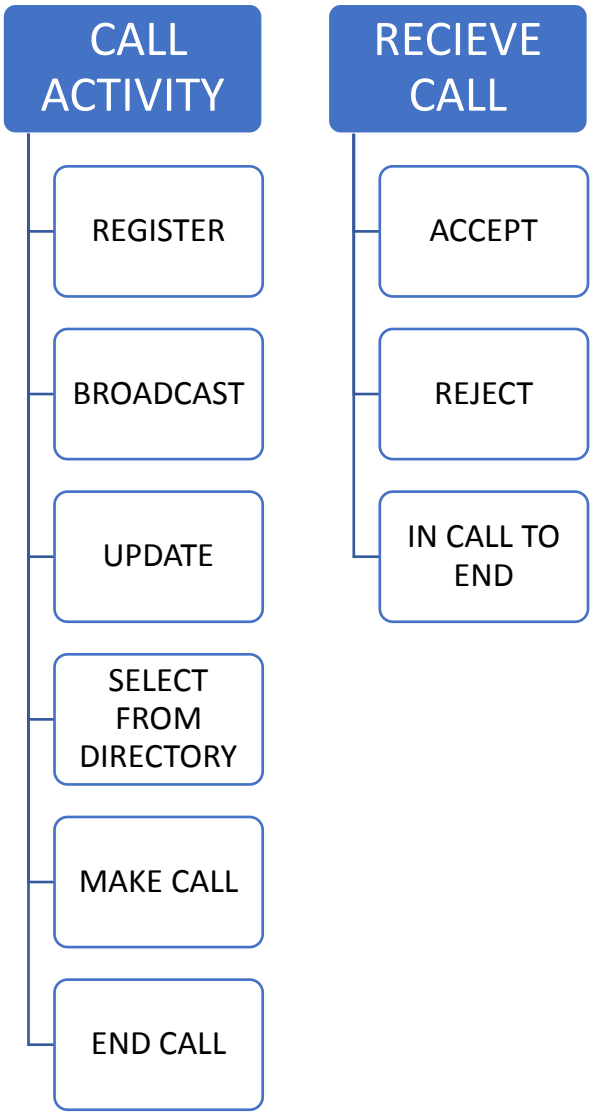


Figure 6: General Layout of Make and Receive Call Activity

6. Class Description

Name	Description
DatagramPacket	This class speaks to a datagram parcel. Datagram packets are utilized to execute a connectionless bundle conveyance benefit. Each message is directed starting with one machine then onto the next construct exclusively in light of data contained inside that bundle. Various bundles sent starting with one machine then onto the next may be directed in an unexpected way, and might land in any request. Bundle conveyance isn't ensured.
DatagramSocket	This class speaks to an attachment for sending and getting datagram packets. A datagram attachment is the sending or getting point for a parcel conveyance benefit. Every parcel sent or got on a datagram attachment is independently tended to and directed. Various bundles sent starting with one machine then onto the next might be directed in an unexpected way, and may land in any request. Where conceivable, a recently developed {@code Datagram Socket}

	has the {@link SocketOptions#SO_BROADCAST SO_BROADCAST} attachment choice empowered in order to permit the transmission of communicate datagrams. So as to get communicate packets a Datagram Socket ought to be bound to the trump card address. In a few usage, communicate bundles may likewise be gotten when a Datagram Socket is bound to a more particular address.
Inet address	This class speaks to an Internet Protocol (IP) address. An IP address is either a 32-bit or 128-piece unsigned number utilized by IP, a lower-level convention on which conventions like UDP and TCP are fabricated. A case of an InetAddress comprises of an IP address and potentially its relating host name (contingent upon whether it is developed with a host name or whether it has officially done turn around have name determination. Host name-to-IP address is expert using a blend of neighborhood machine setup data and system naming administrations, for example, the Domain Name System (DNS) and Network Information Service(NIS). The specific naming services(s) being utilized is as a matter of course the neighborhood machine arranged one. For any host name, its comparing IP address is returned.
AudioFormat	The class is utilized to get to various sound organization and channel design constants. They are for example utilized as a part of { AudioTrack} and AudioRecord}.

AudioManager	AudioManager gives access to volume and ringer mode control.
AudioRecord	The AudioRecord class deals with the sound assets for Java applications to record sound from the sound information equipment of the stage. This is accomplished by "pulling" (perusing) the information from the AudioRecordobjectThedecision of which strategy to utilize will be founded on the sound information stockpiling group that is the most advantageous for the client of AudioRecord.Upon creation, an AudioRecord protest introduces its related sound cushion that it will load with the new sound information. The extent of this cushion, indicated amid the development, decides to what extent an AudioRecord can record before "finished running" information that has not been perused yet. Information ought to be perused from the sound equipment in pieces of sizes substandard compared to the aggregate account cushion estimate.
AudioTrack	The AudioTrack class oversees and plays a solitary sound asset for Java applications.It permits gushing of PCM sound cushions to the sound sink for playback. An AudioTrack case can work under two modes: static or spilling. In Streaming mode, the application composes a ceaseless stream of information to the AudioTrack. These are blocking and return when the information has been exchanged from the Java layer to the local layer and lined for playback. The gushing mode is most helpful when playing squares of sound information that for

	example are too huge to fit in memory in light of the length of the sound to play too enormous to fit in memory due to the attributes of the sound information (high testing rate, bits per test got or produced while beforehand lined sound is playing.The static mode ought to be picked when managing short sounds that fit in memory and that should be played with the littlest idleness conceivable. The static mode will accordingly be favored for UI and amusement sounds that are played frequently, and with the littlest overhead possible. Upon creation, an AudioTrack question instates its related sound cushion. The extent of this cushion, indicated amid the development, decides to what extent an AudioTrack can play before coming up short on data.For an AudioTrack utilizing the static mode, this size is the greatest size of the sound that can be played from it.For the gushing mode, information will be composed to the sound sink in pieces of sizes not exactly or equivalent to the aggregate cradle estimate.
MediaRecorder	Used to record sound and video. The chronicle control depends on a straightforward state machine.
Wifi manager	This class gives the essential API to dealing with all parts of Wi-Fi network. Get an occurrence of this class by calling.
Wifi info	Defines the state of any Wi-Fi connection that is dynamic or is in the process of being established.

7. User Interface

The application graphical user interface is as follows which we achieved:

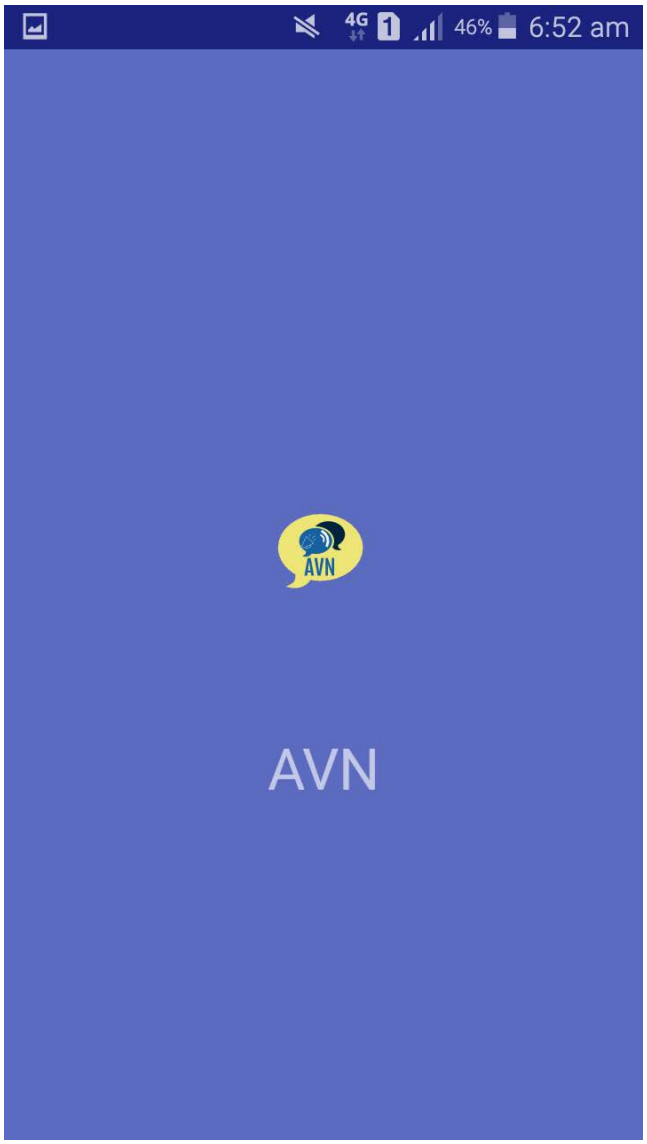


Figure 7: Starting Page

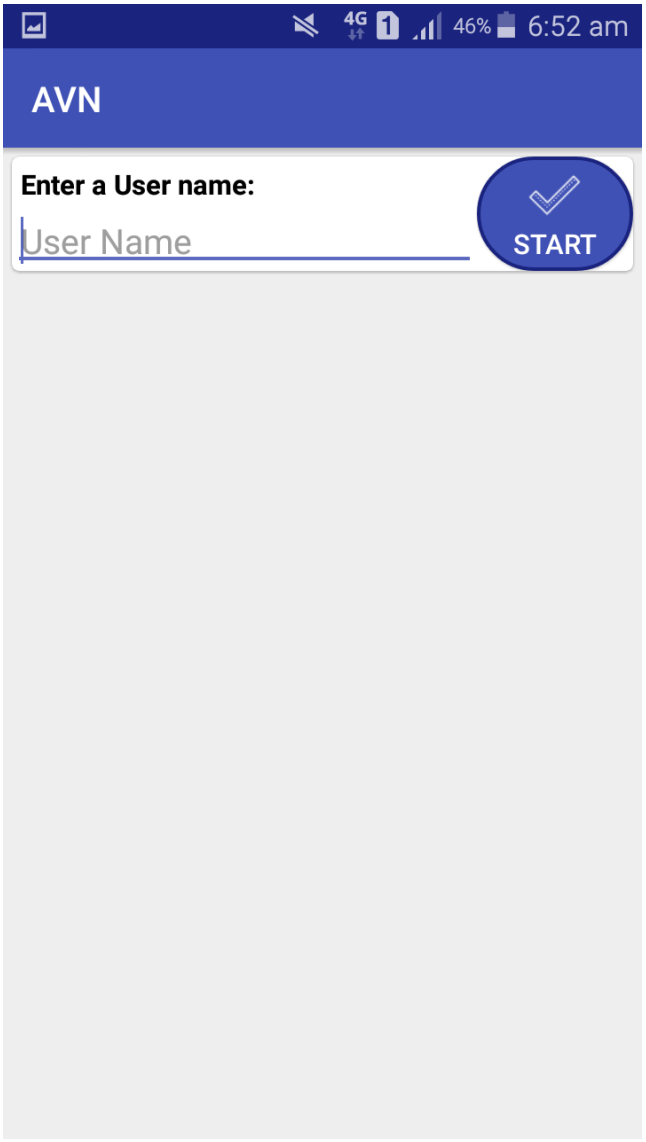


Figure 8: Enter User Name

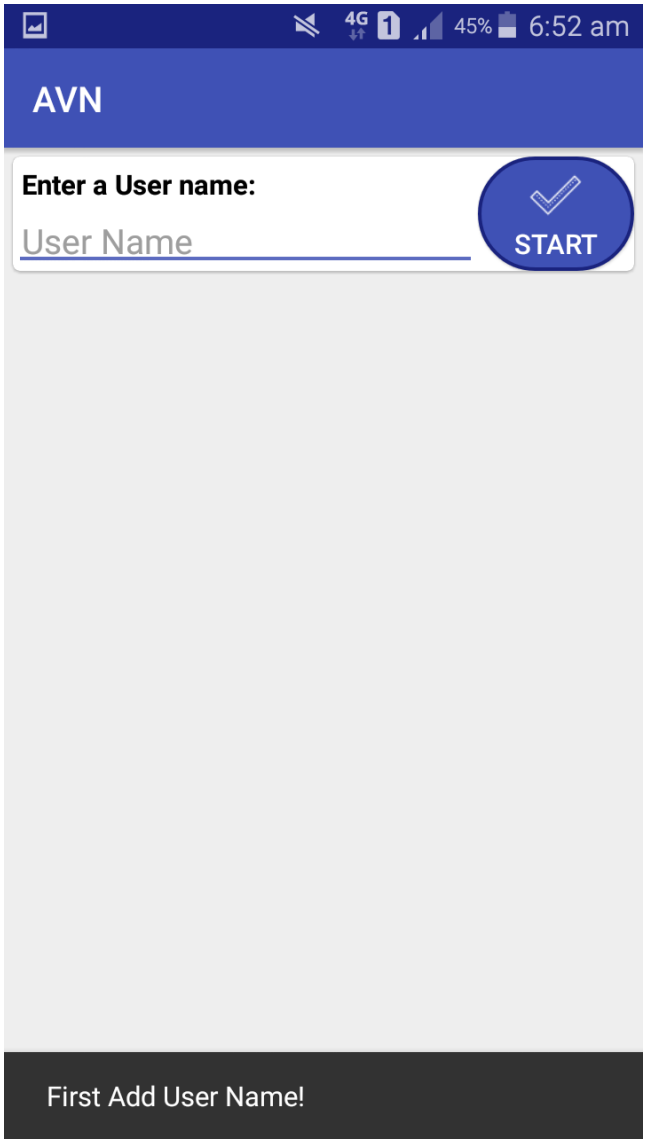


Figure 9: Add User Name To Proceed

Figure 10: Typing Name To Register

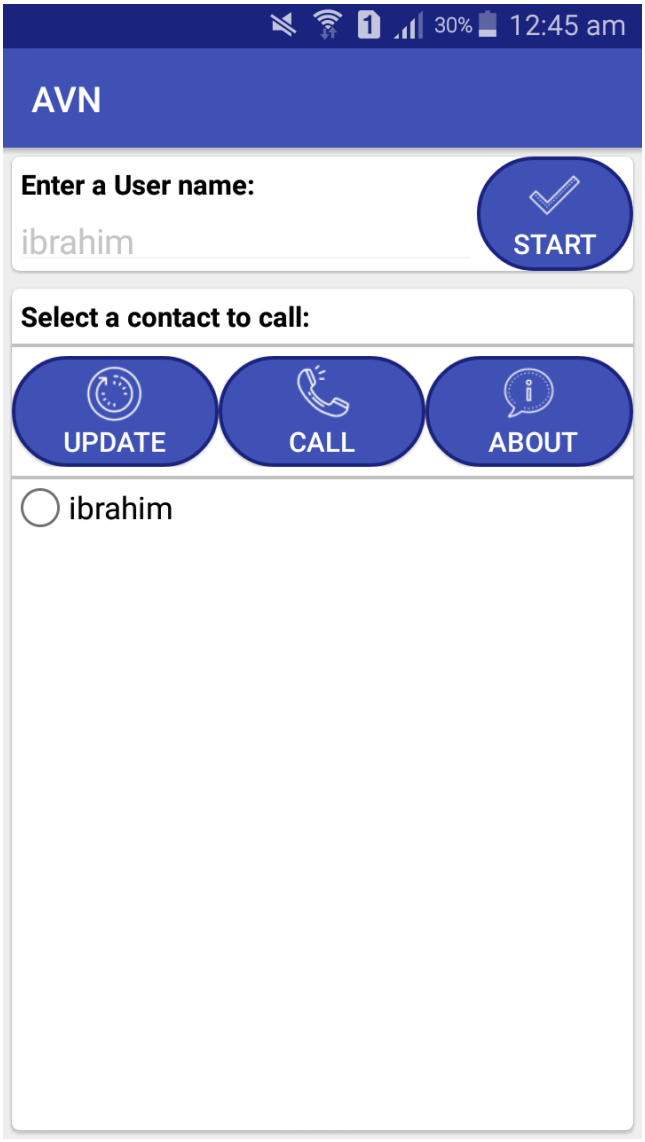


Figure 11: Press Start And Update The Directory

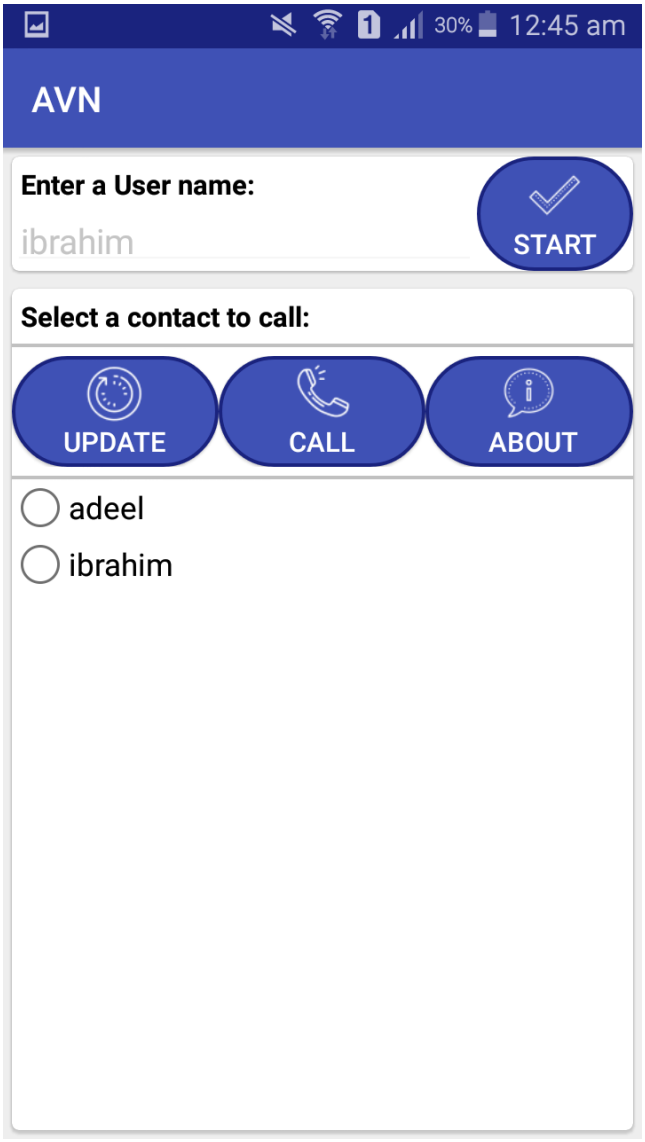
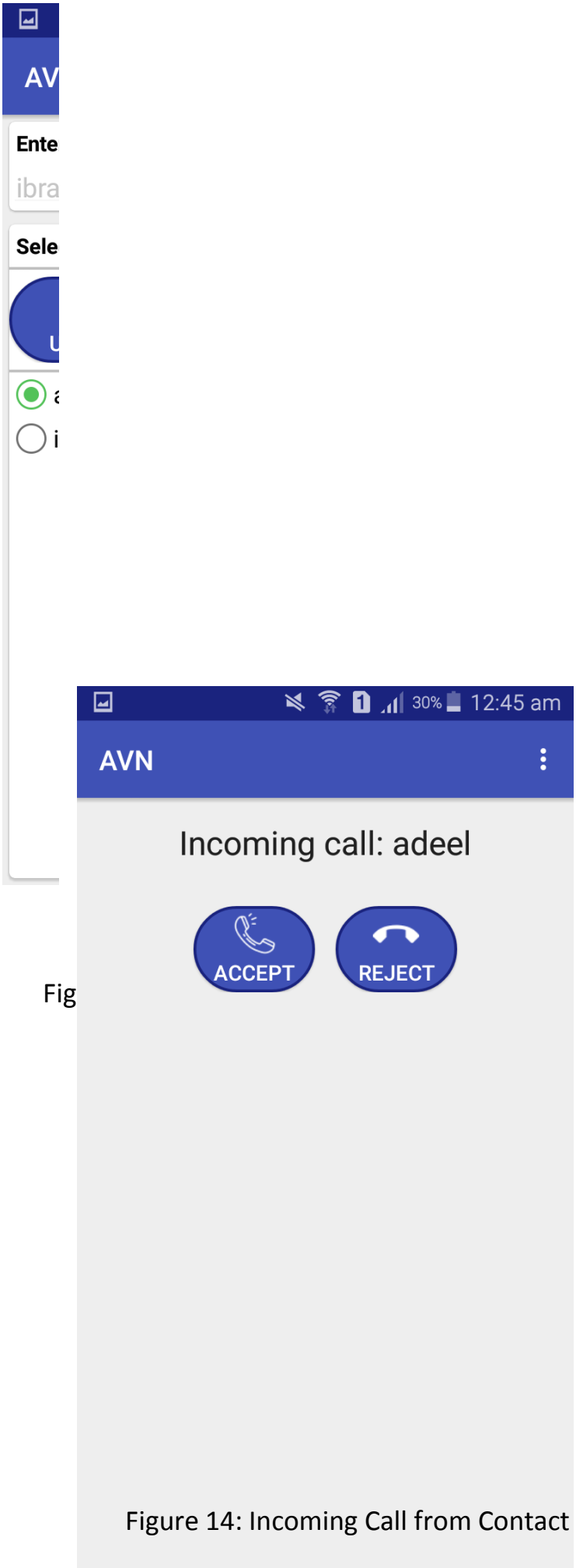


Figure 12: Updating Directoty To Add Users



Fig

Figure 14: Incoming Call from Contact

Figure 15: Disconnect Call

8. Detailed Description of Components

8.1 **Enter User Name**

Identification	**Name**: Enter User Name **Location**: Application layer
Type	Component
Purpose	• The purpose of this component is to receive user input. • The input through touch screen of smart phone will be processed by this component so the name is further registered in the directory.
Function	**What the component does:** Detect input in the form of typing from the user
Subordinates	**Constituents of the component:** The component has no sub-components. **Functional Requirements:** **Requirement 1:**The user will be able to view the name in the box where he has registered himself.
Dependencies	**Components using this component:** The sub component input events of the unity component will get its input from this module. This input will be further processed by input events component.

Interfaces	The external hardware interfaces interacting with this component will be: • Android smartphone **Error messages:** Enter username first.
Resources	• The resources used by this component are touch screen (for touch inputs). • RAM
Processing	The processing required for this component is receiving the users input and giving this input to the input event module application.
Data	Float values, integer values, strings

8.2 **Start**

Identification	**Name:** Start **Location:** Application layer
Type	Component
Purpose	To display the name in the directory.
Function	**What the component does:** • It processes the name to all the applications and maintain it in the directory.
Subordinates	**Constituents of the component:** The component has no sub-components.

	Functional Requirements: **Requirement 1:**The user will be able to view the name in the directory which is being maintained below.
Dependencies	Start is dependent on user input. By pressing he will be able to move forward.
Interfaces	**External interface requirement** An output display screen which it will be using to display icon created by the application i.e.update, call and about.
Resources	• Android smartphone screen
Processing	The processing required of this component is to respond and display input of the user.
Data	User input (touch)

8.3 **update**

Identification	**Name:** update **Location:** Processing layer
Type	Module
Purpose	Update will process and broadcast the name to all the user connected over the same network and similarly receive so the directory is maintained who all are present in the network.
Function	**What the component does:**

	The main functions of this module are as follows: • Handle directory. • Broadcast. • Update the list.
Subordinates	**Constituents of the component:** The other components using this component are: • Network Manager **Functional requirements:** **Requirement 1:** The user is able to connect devices. **Requirement 2**: The system will share the data over wireless to all the connected phone in the network. **Requirement 3:** The application allows the user to update directory.
Dependencies	Update is dependent on user input. By pressing he will be able to move forward.
Interfaces	The external hardware interfaces interacting with this component will be: • Android smartphone
Resources	The resources used by this module are

	RAM Network manager
Processing	The processing done by this module is that it: • Initiates directory • Manages memory requirements
Data	Float values, integer values, strings

8.4 **Call**

Identification	**Name:** call **Location:** Processing layer
Type	Module
Purpose	Call will process will be able to direct call to the host selected to make the call.
Function	**What the component does:** The main functions of this module are as follows: • Make two-way full duplex communication with the host selected. • Broadcast the voice. • audio call
Subordinates	**Constituents of the component:**

	The other components using this component are: • Network Manager **Functional requirements:** **Requirement 1:** The user is able to connect devices. **Requirement 2**: The system will share the data audio call over wireless to all the connected phone in the network. **Requirement 3:** The application allows the user to have full duplex communication.
Dependencies	call is dependent on user input. By pressing he will be able to move forward.
Interfaces	The external hardware interfaces interacting with this component will be: • Android smartphone **Error messages:** • Select user first
Resources	The resources used by this module are • RAM • Network manager • Speakers • Microphone • Audio player

	• Storage
Processing	The processing done by this module is that it: • Initiates call • Manages memory requirements • Manages speakers • Manages microphone • Manages audio player • Manages network
Data	User input (touch)

8.5 **Accept**

Identification	**Name:** accept **Location:** Processing layer
Type	Module
Purpose	Accept will process the call from the incoming user and will be able to connect to the host who is calling.
Function	**What the component does:** The main functions of this module are as follows: • Make two-way full duplex communication with the host selected. • Broadcast the voice. • audio call

Subordinates	**Constituents of the component:** The other components using this component are: • Network Manager **Functional requirements:** **Requirement 1:** The user is able to connect devices. **Requirement 2**: The system will share the data audio call over wireless to all the connected phone in the network. **Requirement 3:** The application allows the user to have full duplex communication.
Dependencies	accept is dependent on user input. By pressing he will be able to move forward.
Interfaces	The external hardware interfaces interacting with this component will be: • Android smartphone
Resources	The resources used by this module are • RAM • Network manager • Speakers • Microphone • Audio player

	• Storage
Processing	The processing done by this module is that it: • Initiates call • Manages memory requirements • Manages speakers • Manages microphone • Manages audio player • Manages network
Data	User input (touch)

8.6 **Reject**

Identification	**Name:**Reject **Location:** Processing layer
Type	Module
Purpose	Reject will end the connection and reject the incoming call request.
Function	**What the component does:** The main functions of this module are as follows: • To stop the incoming call.
Subordinates	**Constituents of the component:** The other components using this component are: • Network Manager

	Functional requirements: **Requirement 1:** The user is able to reject the call **Requirement 2**: The system will share the data audio call over wireless to all the connected phone in the network.
Dependencies	reject is dependent on user input. By pressing he will be able to move forward.
Interfaces	The external hardware interfaces interacting with this component will be: • Android smartphone
Resources	The resources used by this module are • Network manager
Processing	The processing done by this module is that it: • terminates call • Manages network
Data	User input (touch)

Adhoc Voice Network is project which provides full duplex voice communication over IEEE 802.11 and hotspot without the use of any external server. However, materials already available on internet is being used to aid the modelling, enhance the graphics/performance of the application.

We have used a client to client architecturewith IEEE 802.11 as underlying protocol for this application. The reasons are as follows.

8.7 **Two devices:**

The application uses two android phones. The communication between these phones is done via Wi-Fi because it is wireless and is faster than Bluetooth. Hence, we have chosen a clientarchitecture with IEEE 802.11 as underlying protocol to implement this networked communication voice calling.

8.8 **Central storage:**

The display phone provides the user with a view of the contacts/users and hence all the application users are on the display phone. The smart phone is being used as an input/output device to fetch data and process it and to make call and receive activity.

8.9 **Multiple users:**

This project is based on multiple users and keeping in view this we implemented our applicationwith client architecture with IEEE 802.11 as underlying protocol.

9. Project Test and Evaluation

This document proposes the plan used for testing the application. The test plan proposed will ensure that the AVN will be working in a manner which was intended.

The purpose of this document is to describe the tests that will be conducted on the implementation of AVN. The aim is to check that the modules (system features) such as

- Displaying the AVN
- Connectivity configuration between devices
- Voice Communication between the devices
- Voice quality
- Standard AVN rules

9.1 **Test Items**

The test items selected for testing include the following

- Performance
- Interface
- User control

9.2 **Features to be Tested**

The features of our AVN include the functionality mentioned in the use cases. Following features are to be tested keeping in view the test items and system features aforementioned

- Selection of App mode
- Connection between the phones
- Distributed directory
- Call alert
- Full duplex voice communication
- Robust

9.3 **Approach**

Functional Testing will focus on each use case that is included in the version currently being worked on. Testing will mainly consist of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions.

The testing strategy for AVN will be alpha testing. Our project is in modules so we will start the testing phase by testing the modules separately and then step by step integrating modules to test them with each other i.e. integration testing and then the complete application is tested as a whole in system testing.

9.4 **Pass/Fail Criteria for Test Items**

The criteria are as follows

- The pre-conditions are met.
- Inputs are carried out as specified.
- Test case will pass if it produces the desired output for a specified input.
- Test will fail otherwise.

9.5 **Test Suspension Criteria**

Testing procedure will be suspended whenever a defect is found that restricts further testing. A corrective measure will be applied depending upon the criticality of the defect and testing will be resumed.

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling is applied more deeply to cover all these issues but unforeseen circumstances may happen.

9.5.1 **Testing tasks**

- Develop Test Cases.
- Execute tests based on the test cases developed.
- Report defects during tests if any.
- Manage the changes made after testing.

10. Assumptions and Dependencies

Normal weather conditions have been assumed for our project. Weather conditions can have a huge impact on wireless signal integrity. Lightning, for example, can cause electrical interference.

11. Timeline followed

OCTOBER 2017	NOV,DEC 2017 JAN 2018	FEB,MARCH,APRIL 2018	MAY,JUNE,JULY 2018
• Project defense	• The *basic concepts of Networking* and acquaintance with NW devices • Understanding of *802.11 protocol* details related to project design. (security, delay) • *Design and Implementation of Auto node discovery algorithm*	• *Mechanism for Maintaining a distributed directory* • Android development • Efficiently *Accessing Hardware resources* (mic/speaker) in Android platforms (i.e without slowing down the device and min delay) • IP (ver 4.0)	• *Voice transfer* with min delay on network • Understanding and implementing *Noise cancellation* technique • *Performance analysis* (deterioration) with inc number of nodes • Design, test, trial and trouble shooting

		Communication **using android devices**	

12. References

https://www.techwalla.com/articles/the-advantages-and-disadvantages-of-full-duplex

https://developer.android.com/studio/intro/index.html

https://developer.android.com/reference/android

http://www.instructables.com

https://www.wikipedia.org/

https://www.google.com.

http://developer.android.com/guide/index.html.

http://developer.android.com/reference/packages.html

http://docs.oracle.com/javase/6/docs/api/

http://developer.android.com/guide/components/fundamentals.html

http://docs.oracle.com/javase/tutorial/

http://developer.android.com/guide/topics/ui/index.html

http://developer.android.com/guide/topics/ui/declaring-layout.html

http://developer.android.com/guide/appendix/faq/commontasks.html

http://code.google.com/android/add-ons/google-apis/maps-overview.html

http://developer.android.com/guide/practices/ui_guidelines/icon_design.html

https://www.google.com.pk/search?q=Auto+Node+discovery+means&rlz=1C1CHZL_enPK795PK795&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjg8J6Yqt7bAhXUh6YKHen6C9kQ_AUICigB&biw=1366&bih=642#imgrc=FW6387Nvj86fPM

http://github.com/

http://developer.android.com/resources/samples/get.html

http://developer.android.com/training/index.html.

http://android-developers.blogspot.com/

http://developer.android.com/resources/faq/

http://developer.android.com/resources/community-groups.html

http://groups.google.com/group/android-developers?lnk=