e-ISSN 2455–1392 Volume 3 Issue 3, March 2017 pp. 51 – 59
Scientific Journal Impact Factor : 3.468
http://www.ijcter.com

Design And Implementation of Android Based Voip System for Noise Pollution Control

Samuel Isaac[1], Airoboman Abel[2], Adewale Adeyinka[3], Ajaka Chiamaka[4]

[1,2,3,4]*Department of Electrical and Information Engineering, Covenant University*

Abstract—Noise pollution happens to be a menace that the world is currently fighting hence, a way of controlling such pollution is needed so that in the event of passing information to a particular individual or group such information will land on the table of the recipient without constituting a menace to those around. The application of Information and Communication Technology ICT is therefore envisaged in solving this problem especially in environment where ICT is the order of the day. This paper, therefore, proposes a technological approach to solving this problem; the design and implementation of a Voice over Inter-Protocol VoIP as a means of sending voice messages from one android application to another, over a wireless network to individual, groups etc. The android platform was chosen because of its merit in portability and been able to connect to the internet easily. The VoIP has been designed, implemented and tested within the halls of residence of Covenant University, Nigeria and the result was satisfactory hence, acceptability of this work will show a great reduction in noise pollution in tertiary institutions, places of worships, and every other open spaces, events where messages need to be sent but to the specific recipients.

Keywords—Activity, Admin, VoIP, Noise, Pollution, Voice

I. INTRODUCTION

Noise Pollution has been a major challenge in our society in recent time, the public address system used in place of worship and many other organizations and open places has been found to cause noise pollution which constitutes a nuisance to the unconcerned public. The ear drum may be able to deal with sounds of varying frequencies for the purpose of transmission to the brain however, it may not be able to deal with such sounds when it has converted to noise [1]. According to [1], his findings revealed that urban dwellers in Nigeria are exposed to high noise levels (above 70 decibels), with the attendant health implications[2]. Some of the announcements delivered on a public address system in schools, places of worship and other open events is not always directed to all within the hearing radius but many unconcerned are unduly disturbed. It is with respect to this that the Lagos State Government (Nigeria) considers it an imperative act to disturb the peace of the residents under the guise of the form of a gathering. The Lagos State Environment Protection Agency (LASEPA) and the Ministry of Environment kick against organization's that break the law by causing noise pollution [3]. In [4] assertion was made that the peak noise pollution in Delta State (Nigeria) at night overshoot the standard for the World Health Organization WHO citing generator noise as one of the major cause. Although this may be true but one can now begin to imagine the effect of an additional noises from places of worships and other ceremonies taking place during the night. Hence, proffering solution to curbing the menace caused by noise pollution is of great importance in our society today [1]. The first VoIP product was introduced in Israel by Volcatec and by 2005 major voice quality issues addressed by prioritizing voice over data traffic [4]. In simpler terms, VoIP converts the voice signal from your telephone into a digital signal that travels over the Internet [6]. The merit of VoIP technology includes cost saving, better productivity as well as a good user friendly interface. According to [6], VoIP requires a means for prospective communications partners to find each other and to signal the other party their desire to communicate. Furthermore, [8] asserts that Voice over Internet Protocol (VoIP) is a form of communication that allows you to make phone calls over a broadband internet connection instead of typical analog telephone lines. The aim of this

paper, therefore, is to design and implement a communication system based on Voice over Internet Protocol VoIP technology in an attempt to overcome the noise pollution posed by public address systems. VoIP is a new technology that enables the transmission of voice signals using Internet Protocol over the public internet or a private network. The voice signals from your telephone or from your smart device as is the case in this paper is converted to a digital signal that travels over the network. To transport voice on a data network, the human voice must be 'packetized' that is, broken into small pieces (packets) and sent through the network one-by-one. The process of packetization compresses the caller's voice signal transfers it over the network and it is then decompressed at the other end. [9]

II. METHODOLOGY

The android application used for the purpose of this research is made up of a combination of codes that interact together to create a secure communication system. The codes are written in Java Script and XML, the interface which the user interacts with is written in XML, while the back end codes that control the android application. The application is divided into different parts with different functions; the main architecture of the android application is shown in figure 1. The android device is used for the purpose of this research because of the following advantages: Android can be customized to suit each user, the settings are configurable to meet the preference of each user. The operating system can respond to real life actions such as swiping, tapping, pinching, and so on. This easy and convenient use of android has made it widely used. Android supports a wide range of audio formats, making it very suitable for voice messaging, which is the core essence of this project. Android also supports additional hardware; a device can be connected internally, provided this additional features are made available [3]. Android operating system has an open source framework, which makes it possible for users to build their own applications in accordance with their preferences.

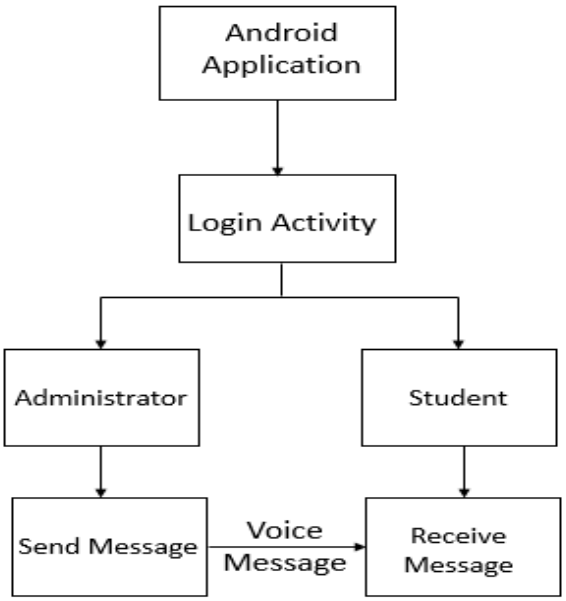

Figure 1: The Android Architecture Application

III. SYSTEM DESIGN

The Android application is designed using Android studio, the Android devices can receive or send voice messages with the aid of the embedded Wi-Fi module in each of them. A static IP address is assigned to each device on the network. Only one application is designed, but the application has the ability to both send and receive voice messages. The applications have two sections, available to two different kinds of users. As previously explained, one user (the Admin) has the ability to send, while

the other user (the recipient) receives. These two sections are separated with the aid of a Login Activity, the source code directs the user to the send Activity page if the details entered corresponds with the Admin details. When the details are those of the recipient, the user is directed to the receive Activity page. Any other detail entered would be rejected. The flow chart for the android application is shown in figure 2.

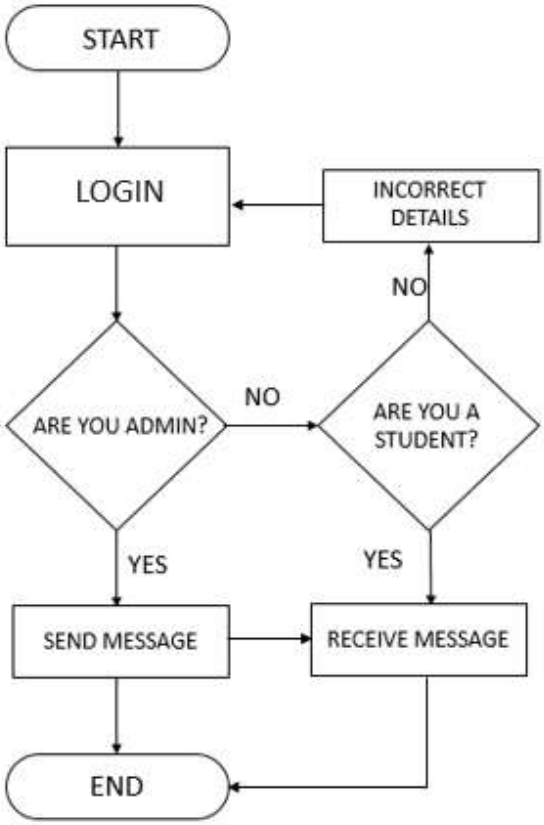

Figure 2: Android Application Flow Chart

The Android application starts with a Login activity, the Admin user can proceed to send a voice message, while the recipient would proceed to receive a message. If the detail entered is neither Admin nor recipient, the user is denied the permission to move on to the next activity, therefore that user cannot receive or send a voice message. The Use Case diagram shows the simplest interaction of the user with the android application, it explains the process of the android application. Highlight the key steps the user would encounter. Figure 3 shows the Use Case diagram of the android application which comprise two users, the Admin, and the user, each user starts with a Login activity, the Admin user inputs the Admin password and is directed to the Activity for sending a message, while the recipient would proceed to receive a message.

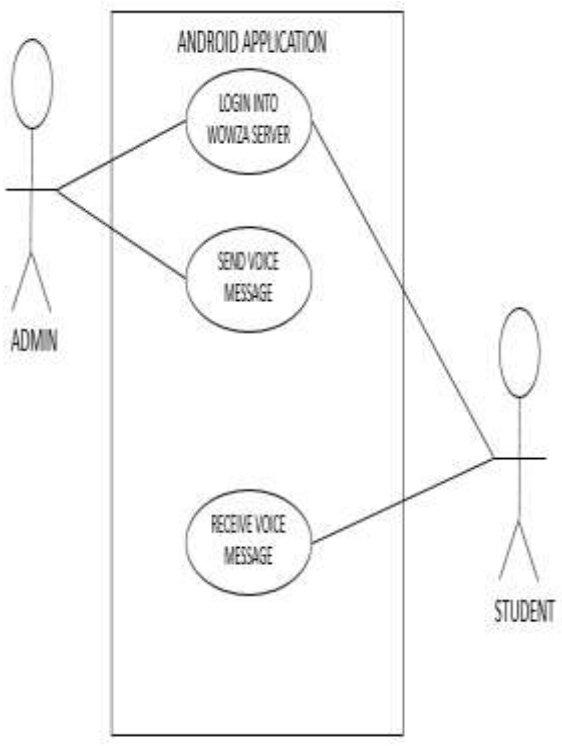

Figure 3. Use Case Diagram

There are two users, the Admin, and the recipient, each user starts with a Login activity, the Admin user inputs the Admin password and is directed to the Activity for sending a message, while the recipient would proceed to receive a message. If the details entered LOGIN 20 is neither Admin nor recipient, the user is denied the permission to move on to the next activity, therefore that user cannot receive or send a voice message.

3.1. Sequence Diagram

A sequence diagram is an interaction diagram that shows how the processes involved in the android application interact with each other. Figure 4 shows the detailed sequence diagram of this project.

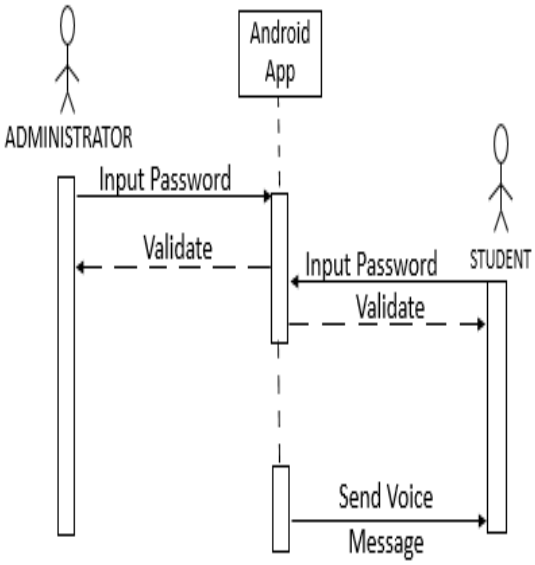

Figure 4. The Sequence Diagram

From the sequence diagram, one can understand the project fully, the first user to initiate a connection is the administrator, he has to log into the application before he can send a message to the recipient. The recipient also has to log in the application before the can receive the message. If the password provided by either the administrator or the recipient is wrong, the user cannot proceed to the next step. Some activities require a response before a new activity can start, if the password provided by either the Administrator or Recipient is not validated, the new activity won't start. The dotted lines represent the lifeline of the activity when an activity takes a pause before it's called up again, the dotted line would indicate that the activity is taking a break.

3.1.1. Activity Diagram.

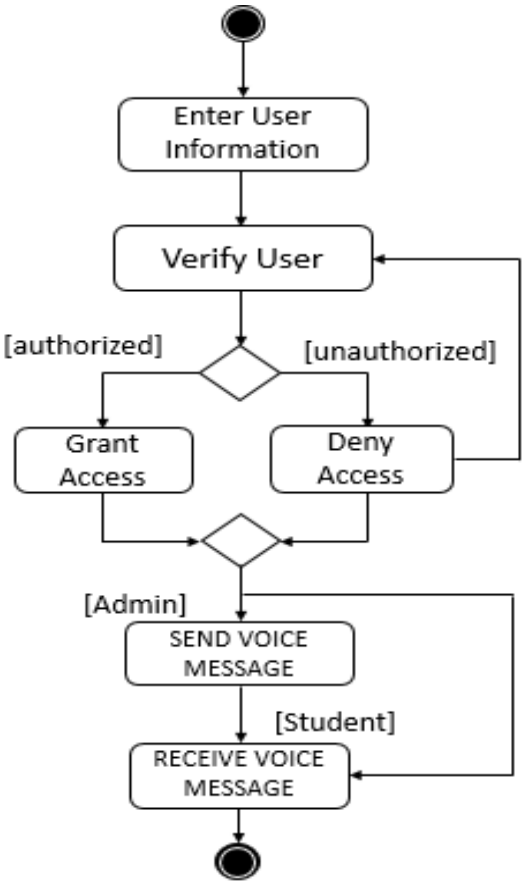

Figure 5. Activity Diagram

The user information provided determines whether the user is an Administrator or a Recipient, any wrong detail is rejected. The Administrator sends voice message after authorization has been granted, the Recipient likewise receives the voice message only after authorization. The successful sending and receiving of the voice message mark the end of the process.

The system has to be carefully designed for proper functionality before implementation can begin, the design has to be developed first. The design of the android application was developed to meet the requirements of the project, the login activity was introduced to merge the separate activities of sending and receiving into one application.

IV. IMPLEMENTATION

While building and running the codes, testing of each stage was performed to make troubleshooting easy. The Android Studio is able to run the application before it has reached its final stage, and errors that are detected. When you 'run' the application in the Studio, there is a panel to

show any errors that may exist and there is a preview panel to show what the project would turn out to be, figure 5 shows the overview of an android studio.

Figure 5. Android Studio

When there is an error in the system, the codes would be shown in red letters and the error would be stated in the message box. Clicking on the error would direct the user to the exact line of code within the error. Figure 6 shows one of the errors that was encountered, this particular error was because the identification used was a wrong one, instead of 'click button' the input was 'click'. An error can result from even the smallest mistake or it could be a more complicated error like missing permissions or missing import.

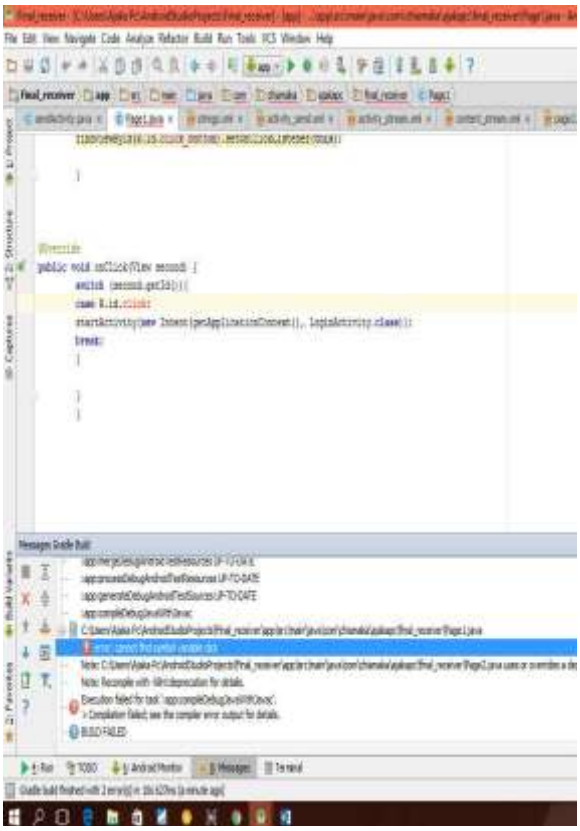

Figure 6. Detection of Error in Android Studio

Each Java class and Layout file is entitled a dedicated window to avoid confusion. It was convenient therefore to code different pages of the application, it was easier to detect errors and have a neat work too. The names of all the different Layout files and Java class was chosen according to preference. The different tabs available in the project are shown in figure 7. The Android studio also has an event log, this keeps track of the different times the app is being 'run' and the different errors that occur while the app runs. This log helps one to make reference to previous errors, figure 8 is one of the logs of this paper.

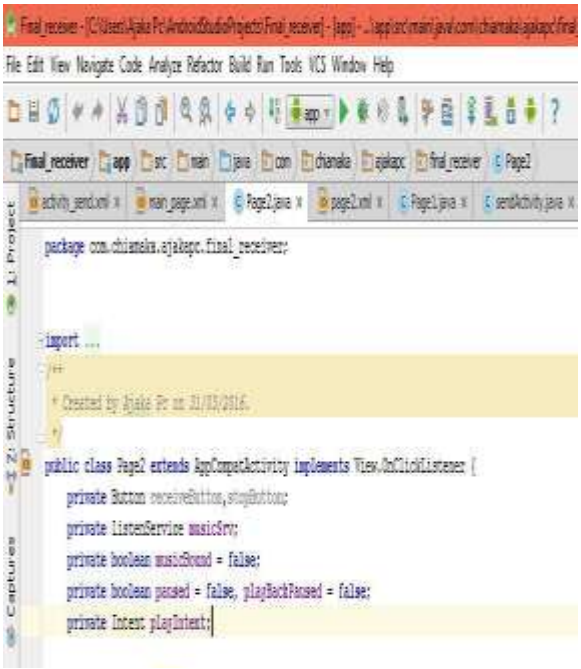

Figure7 Java Classes and Layout files are in separate windows

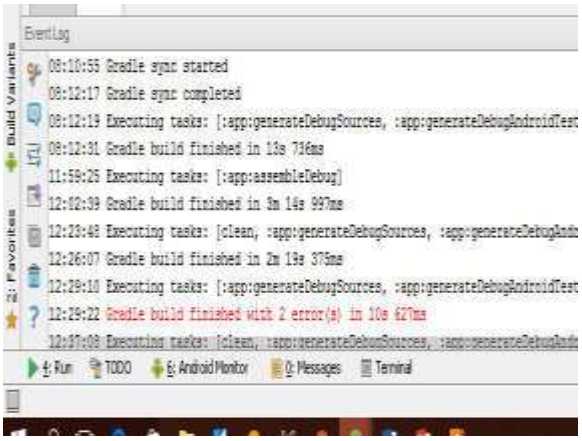

Figure 8 Event Log of an Android Studio

V. TESTING

The VLC media player was used to test it the application was capable of sending a voice message to any platform, android or not. The network stream of the VLC media player enabled the player to be assigned an IP address, then the computer in use and the android device would be connected and the same network. The android device then sends a voice message over to the VLC media player. The application combines the activities of sending and receiving voice messages at the

same time, it has five Layout files: The Welcome Page, The Hall Selection Page, The Login page, The Send Message Page and The Receive Message page. Let's take a look at the different Layout files and what each button on the file means.

The Welcome Page (Layout file - main_page.xml)
This page is the first the user sees once the application is opened, it is the Layout file labeled *main_page.xml*. One of the halls of residence (Lydia hall) in Covenant University is used as a case study. There's only one button on this page, the "CLICK TO ENTER" borderless button that takes the user to the next Layout file.

welcomePage.java
This is the Java class attached to the Layout file main_page.XML, it contains the set of codes that enable the welcome page function. Only one button on the welcome page performs a function, it enables the user moves over to the next Page. A function called '*set On Click Listener'* is used to make the button listen to the codes that direct it to the next page. The next Layout file to be opened here is the Login Page, the function '*get Application Context'* would call the next Java class. Note that the function didn't call the Layout file, it calls the Java class, but the Java class is attached to an appropriate Layout file.

The Hall Selection Page (Layout – hall.xml)
There are different halls in Covenant University, this page provides the user with the ability to select any hall of choice. The user has the option to choose which hall he is, if the user picks a wrong hall his or her details would not be available in the database of the hall. This layout leads to the next Login activity, the user then has to provide details to determine which action he wants to perform.

hallActivity.java
This Java class enables the user to select which hall he/she belongs to. It shows a list of available halls and the user gets to choose. If a user chooses a wrong hall, he would be denied access into the Login Activity. This page leads to the *activity_login.xml.*

The Login page (Layout file – activity_login.xml)
This is the page that enables a user to either log in as an Admin or as a recipient, it is the Layout file labelled *activity_login.xml*. When a user signs in as Admin, source code takes the user to the Send Message page, if the user signs in as a Recipient, the user is directed to the Receive Message page. A number of wrong tries are displayed to the user, to serve as a check. The login button has two functions, two direct the user to either the activity for sending a message or to receive the message. This makes the login button the actual merger in this
application, it combines the two very separate functions existing in this project.

loginActivity.java
The Layout file attached to this Java class is labeled *activity_login.xml*. This Java class enables the user to either sign in as an Admin or a Recipient, this class helps to solve the stress of building two different applications for sending and receiving. The Login Java class has made it possible call more than one Java class, unlike the *WelcomePage.java.* An '*Edit Text'* function which is used to enable the user input Text or Numerals into the application, is used to accept username and password. The Java class already has two categories of codes, it uses the If-else statement. **If** the details entered corresponds with the details of the Admin user, it would start '*send Activity class',* **else** if the details correspond the details of a recipient, it would start *Page2.class* (the Java class for receiving).

The Send Message Page (Layout file – activity_stream.xml)
After the user has signed into the android application, he is now able to send a message

for streaming. Once the "Start Streaming" button has been clicked, the Android application calls the recipient. After the recipient receives the call, the application real time streaming of the voice message between both users. As explained previously, Real Time Streaming Protocol is adopted in this project and it makes use of UDP to send the content to recipient first before using the reliable TCP to send the voice message.

sendActivity.java

This Java class enables the user to send a voice message to the receiver over the network, the Layout file connected to this Java class is the *activity_send.xml*. The RTSP client was imported into this class, the audio import is also present.

The RTSP client enables the efficient transmission of the voice message.

The Receive Message Page (Layout file – activity_page2.xml)

The Receive message page is opened when the user is a recipient. The user at the receiver end would click the receive call button once the voice message enters the application, then real-time communication between the both users would take place.

receiveActivity.java

This is the Java class for receiving the voice message, its Layout is *page2.xml*. It has imports for audio recording and playing out to the user. This Java class is linked to another Java class called *listenService.java*, this listens to get the voice stream from the RTSP server.

VI. CONCLUSION

This paper discussed the design and implementation of android based VoIP extensively as a new method of controlling noise pollution in our society. The test with VLC media player has proven that this project can be implemented in many other ways. The XML files in combination with the Java script have produced an android application capable of sending and receiving a voice message. However, future work can also be done by introducing short message service SMS to the design, which is expected to serve as a form of redundancy, so in the event that the recipient is away, the message would not be lost.

REFERENCES

[1] A, Abel Urban Noise Pollution in Nigerian Cities: Imperatives for Abatement British Journal of Applied Science & Technology. Vol. 10, No. 6 pp. 1-9. 2015

[2] Oyedepo Sunday Olayinka "Effective Noise Control Measures and Sustainable Development in Nigeria" World Journal of Environmental Engineering, 2013, Vol. 1, No. 1, 5-15, 2013

[3] E. B. a. L. C. Tunde Alao, "Controversy trails implementation of noise pollution law in Lagos," The guardian, 22 January 2015. [Online]. Available: at HTTP://guardian.ng/features/weekend/controversy-trails-implementation-of-noisepollution-law-in-lagos/. [Accessed 05 May 2016].

[4] O. Anomohanran**.** C, Iwegbue**,** O, Oghenerhoro **and** J, Egbai Investigation of Environmental Noise Pollutin Level of Abraka in Delta State, Nigeria. Vol.3, No. 4 pp.292-297, 2008

[5] F. Alejandro and J. Arash "Voice over IP (VoIP) Past, Present and Future". https://inst.eecs.berkeley.edu/~ee233/sp06/student_presentations/EE233_VoIP.pdf [Accessed October 20, 2016]

[6] J, Hallock, (2004) "A Brief History of VoIP" University of Washington p.1-17

[7] C, Vaishnav (2006) "Voice over Internet Protocol (VoIP): The Dynamics of Technology and Regulation" Massachusetts Institute of Technology. pp.1-166

[8] M, Desantis "Understanding Voice over Internet Protocol (VoIP)" US-CERT. pp.1-5, 2008

[9] W.-M. Lee, Beginning Android 4, Application Development, Indianapolis Indiana: John Wiley & Sons, Inc, 2012

