“Remote keyboard application”

Prepared by

Arjun Anghan (16IT002)
Fenil Godhani (16IT023)

Under the supervision of

Prof. Pinal Shah

A Report Submitted to
Charotar University of Science and Technology
for Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology
in Information Technology

IT345 Software Group Project-II

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology

At: Changa, Dist: Anand – 388421

October,2018

CERTIFICATE

This is to certify that the report entitled “**Remote keyboard application”** is a bonafied work carried out by **Mr. Arjun Anghan(16IT002), Mr. Fenil K. Godhani(16IT023)** under the guidance and supervision of **Prof. Pinal Shah** for the subject **Software Group Project-II(IT345)** of **5th** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Pinal Shah
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

ABSTRACT

“Remote Keyboard Application” is Android application. This application is use to write a data in to your computer using your mobile. This application contain keyboard and voice recognizer. Using keyboard you can write in notepad and using voice recognizer whatever you speak in to the microphone it well display on monitor. Application are created in to the Android Studio. You are also use this application for your computer keyboard. This functionality are created by Java Robot class, Server Client Model, and Android Java. Application are created in to the Android Studio. You are also use this application for your computer keyboard.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without mentioning the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. I consider myself privileged to express gratitude and respect towards all those who has guided through the completion of project.

I am grateful to my guide Mr.Pinal Shah and Mr.Amit Paramar, for giving me the support and encouragement that was necessary for the completion of this project. I very thank full Amit Paramar sir to guide the way of Project Implementation. I would like to thank all the faculty members for their patience, understanding and guidance that gave me strength and will power to work through the long tedious hours for developing a project and preparing the report.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 : INTRODUCTION

1.1 **Project Overview :**

"Remote Keyboard Application" is android application. This application having two module.

1. Remote Keyboard
2. Voice Recognizer

First of all conform your computer and your mobile must in same network. After this enter your computer's IP address and Connection port address in application and connect computer to specific mobile. After connection your mobile to the computer whatever you write in your application it is also write in your computer notepad.

Using Voice Recognizer you speak in to microphone it is recognize by your application and convert into the message and when you click on send button at that time message send to your computer and this message display on your JFrame and Notepad.

Client side application are making using Android Java Programming Language and server client model. Client application are create using Android Studio and server side programing are making using Java Robot Class, Java Server Client Program ,Java J Frame. Server side is created in to NetBeans.

1.2 **Scope:**

For commutation between Client application and Server computer are must be in same network. And also your computer having mobile hotspot or wifi driver. If both are not in same network it can not possible to communicate. If both are connected then you can transmit your data Application to your computer.

In Voice recognizer we can't transmit message instantly first all speech are recognizer in your application and after that you transmit your whole message to computer JFrame. It can't type in to notepad but we well copy the data of JFrame and Paste in whatever you want.

1.3 Objective:

"Remote Keyboard Application" is Android application. This application is use to write a data in to your computer using your mobile. This application contain keyboard and voice recognizer. Using keyboard you can write in notepad and using voice recognizer whatever you speak in to the microphone it well display on monitor. Application are created in to the Android Studio. You are also use this application for your computer keyboard. This functionality are created by Java Robot class, Server Client Model, and Android Java. Application are created in to the Android Studio. You are also use this application for your computer keyboard.

We can use this application as keyboard. We can write our Blog, Note, Information using this keyboard. If your client and server are in same network then client type in any place of network area. Voice recognizer are also send message to your server and display in to JFrame.

Chapter 2: PROBLEM STATEMENT

First of all conform your computer and your mobile must in same network. After this enter your computer's IP address and Connection port address in application and connect computer to specific mobile. After connection your mobile to the computer whatever you write in your application it is also write in your computer notepad. Using Voice Recognizer you speak in to microphone it is recognize by your application and convert into the message and when you click on send button at that time message send to your computer and this message display on your JFrame and Notepad.

Client side application are making using Android Java Programming Language and server client model. Client application are create using Android Studio and server side programing are making using Java Robot Class, Java Server Client Program ,Java J Frame. Server side is created in to NetBeans.

We can use this application as keyboard. We can write our Blog, Note, Information using this keyboard. If your client and server are in same network then client type in any place of network area. Voice recognizer are also send message to your server and display in to JFrame.

Chapter 3 : SOLUTION

3.1 About Android :-

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touch screen mobile devices such as smart phones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

App Components :-

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.

There are four different types of app components:

- Activities
- Services
- Broadcast receivers
- Content providers

Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. The following sections describe the four types of app components.

1)**Activities:-**

The Activity class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.

2)<u>Activity-lifecycle:-</u>

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

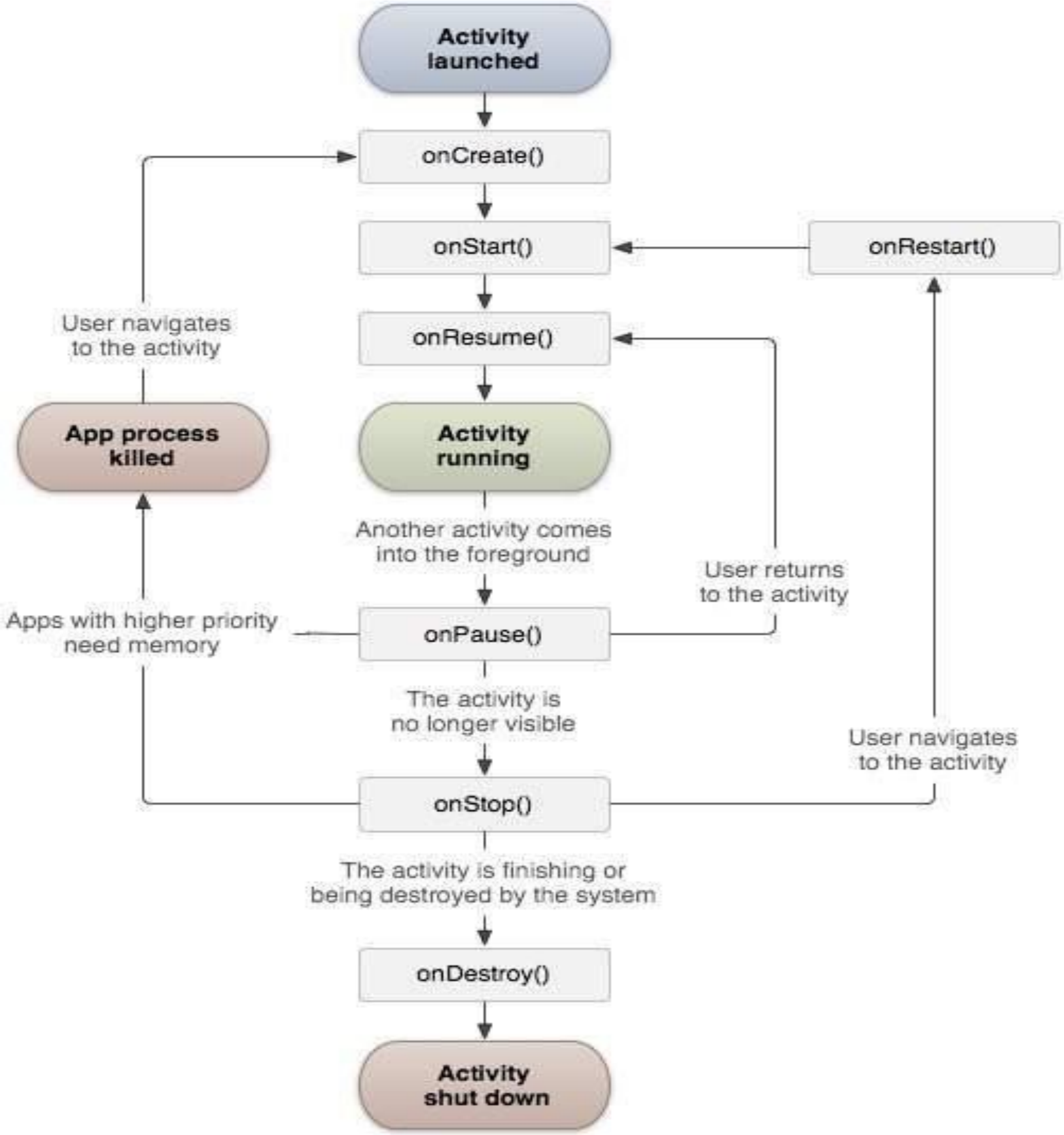

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Figure 3-1 Activity Life cycle

3)Intents and Intent Filters:-

Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

i) Starting an activity:-

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an intent to startActivity(). The Intent describes the activity to start and carries any necessary data.

If you want to receive a result from the activity when it finishes, call startActivityForResult(). Your activity receives the result as a separate Intent object in your activity's onActivityResult() callback. For more information, see the Activities guide.

ii) Starting a service:-

A Service is a component that performs operations in the background without a user interface. With Android 5.0 (API level 21) and later, you can start a service with JobScheduler. For more information about JobScheduler, see its API-reference documentation.

For versions earlier than Android 5.0 (API level 21), you can start a service by using methods of theService class. You can start a service to perform a one-time operation (such as downloading a file) by passing an Intent to startService(). The Intent describes the service to start and carries any necessary data.If the service is designed with a client-server interface, you can bind to the service from another component by passing an Intent to bindService(). For more information, see the Services guide.

4) Network Connection

Before you perform any network operations, you must first check that are you connected to that network or internet e.t.c. For this android provides **ConnectivityManager** class. You need to instantiate an object of this class by calling **getSystemService()** method. Its syntax is given below.ConnectivityManager check = (ConnectivityManager) ;

this.context.getSystemService(Context.CONNECTIVITY_SERVICE);

Once you instantiate the object of ConnectivityManager class, you can use **getAllNetworkInfo** method to get the information of all the networks. This method returns an array of **NetworkInfo**.

3.2 Java Robot Class:

The Robot class in the Java AWT package is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations. In simple terms, the class provides control over the mouse and keyboard devices.

List of event:

void	keyPress(int keycode)	Presses a given key.
void	keyRelease(int keycode)	Releases a given key.
void	mouseMove(int x, int y)	Moves mouse pointer to given screen coordinates.
void	mousePress(int buttons)	Presses one or more mouse buttons.
void	mouseRelease(int buttons)	Releases one or more mouse buttons.
void	mouseWheel(int wheelAmt)	Rotates the scroll wheel on wheel-equipped mice.

public Robot()

throws AWTException

➔Constructs a Robot object in the coordinate system of the primary screen.

Throws:

➔ AWTException - if the platform configuration does not allow low-level input control.

3.4 Java Server Client Model:

This article describes a very basic one-way Client and Server setup where a Client connects, sends messages to server and the server shows them using socket connection. There's a lot of low-level stuff that needs to happen for these things to work but the Java API networking package (java.net) takes care of all of that, making network programming very easy for programmers.

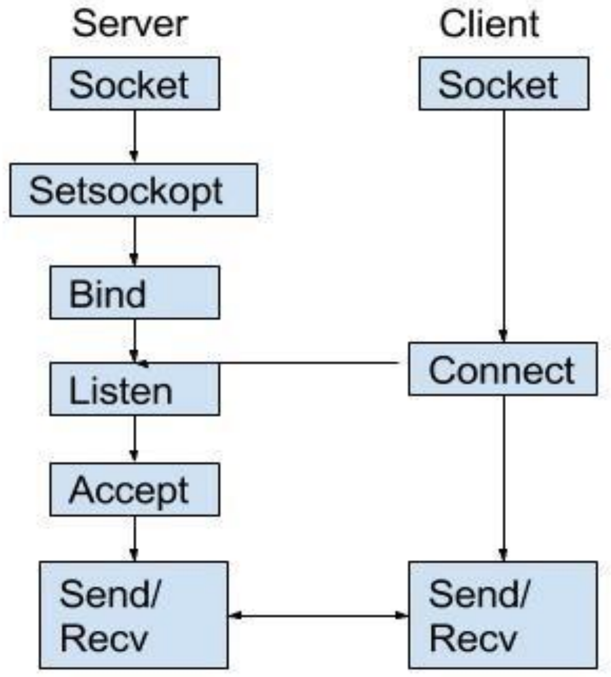


Figure 3.2 State diagram for server and client model

➔**Client Side Programming**:

To connect to other machine we need a socket connection. A socket connection means the two machines have information about each other's network location (IP Address) and TCP port.The java.net.Socket class represents a Socket. To open a socket:

→Socket socket = new Socket("127.0.0.1", 5000)

➔First argument – **IP address of Server**. (127.0.0.1 is the IP address of localhost, where code will run on single stand-alone machine).

→Second argument – **TCP Port**. (Just a number representing which application to run on a server. For example, HTTP runs on port 80. Port number can be from 0 to 65535)

➔**Server Programming**:

→Establish a Socket Connection To write a server application two sockets are needed.

- A ServerSocket which waits for the client requests (when a client makes a new Socket())
- A plain old Socket socket to use for communication with the client.

→**Communication**

- getOutputStream() method is used to send the output through the socket.

→**Close the Connection**

- After finishing, it is important to close the connection by closing the socket as well as input/output streams.

1.4 Tools & Technology:

➔Tools:

- →Android Mobile Devices
- →Computer with WIFI drivers

➔Software Requirement:

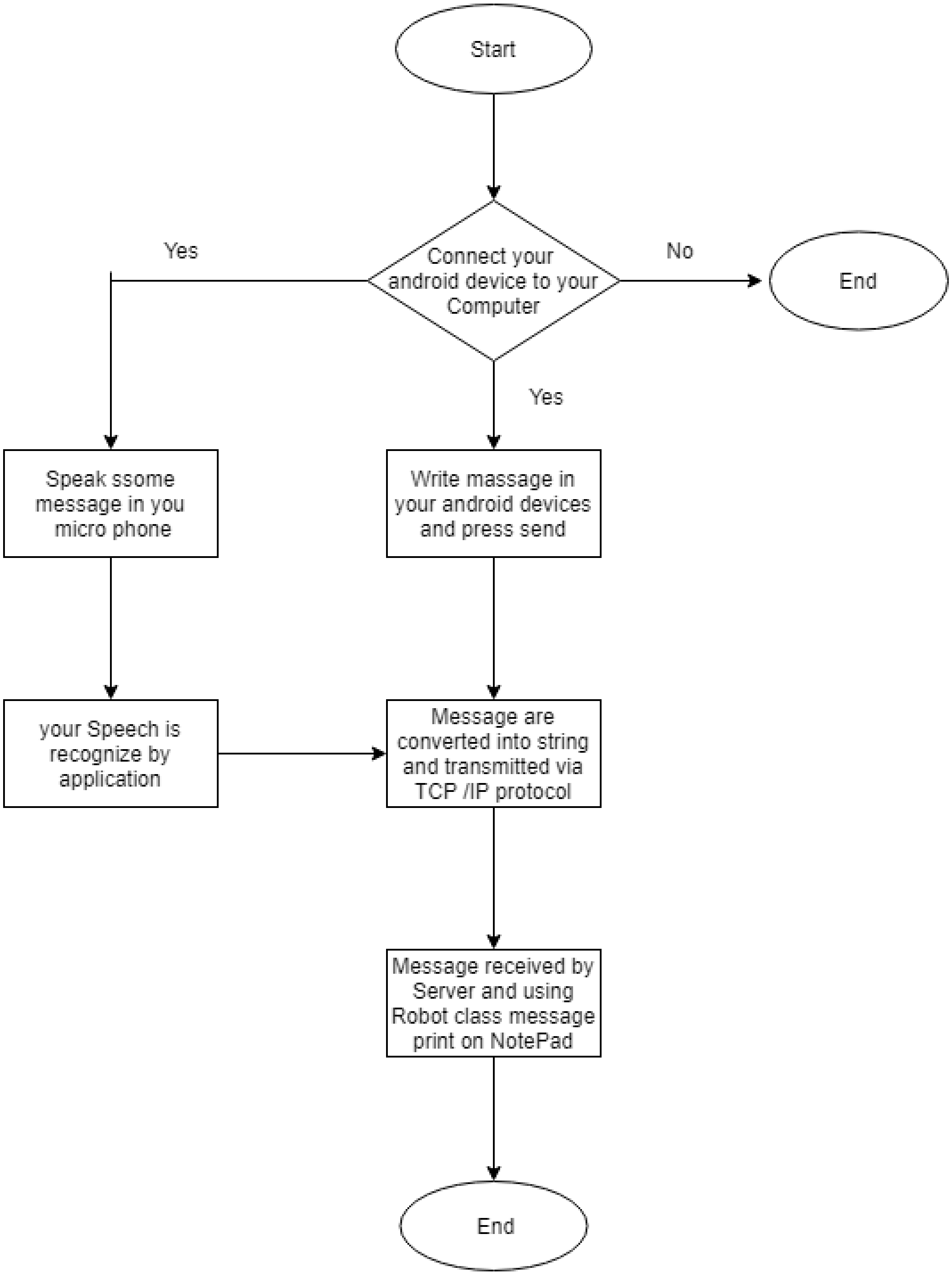
- →Android Studio
- →NetBeans

➔Technologys:

- →Android Java
- →Java Socket Programming

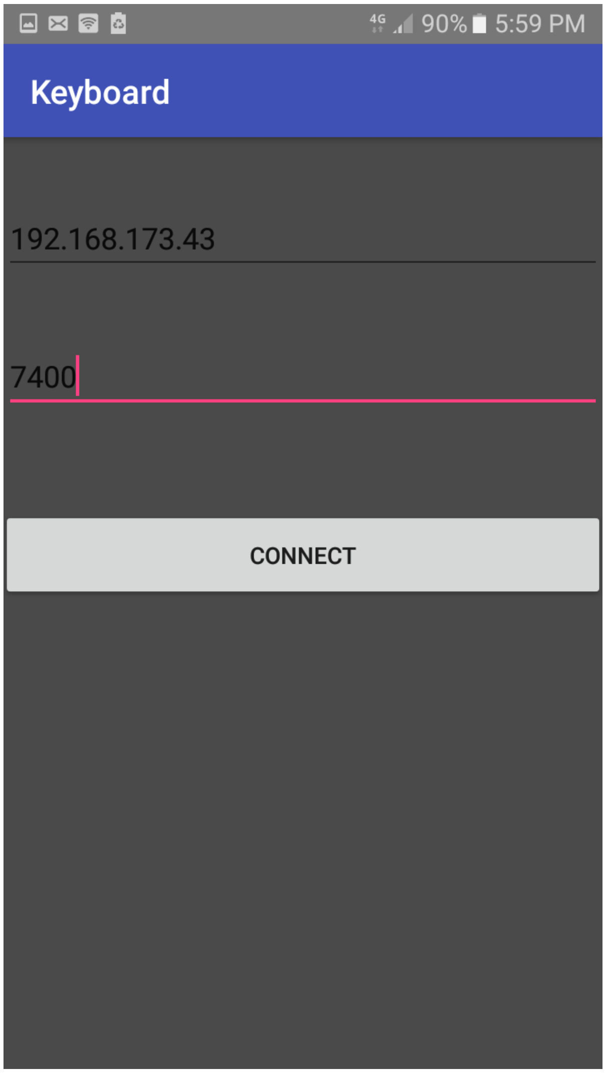
Chapter 4: IMPLEMENTATION

4.1 Implementation Flowchart:


Start
Connect your android device to your Computer
Yes
No
End
Yes
Speak ssome message in you micro phone
Write massage in your android devices and press send
your Speech is recognize by application
Message are converted into string and transmitted via TCP /IP protocol
Message received by Server and using Robot class message print on NotePad
End

4.2 **Snapshot:**

➔Remote Keyboard :

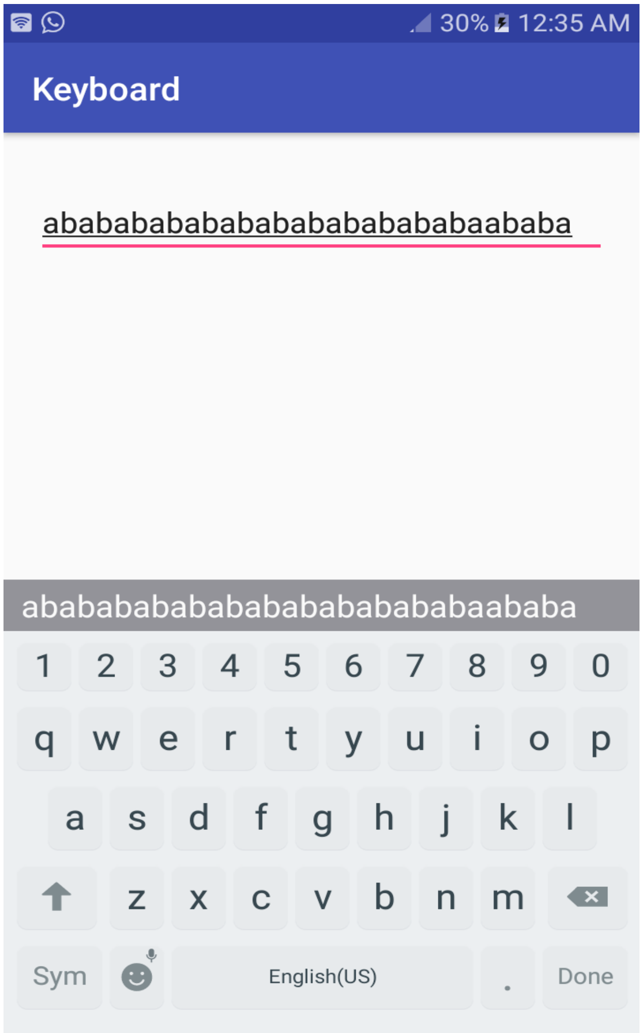


192.168.173.43

7400

Figure 4.1 Connection Activity

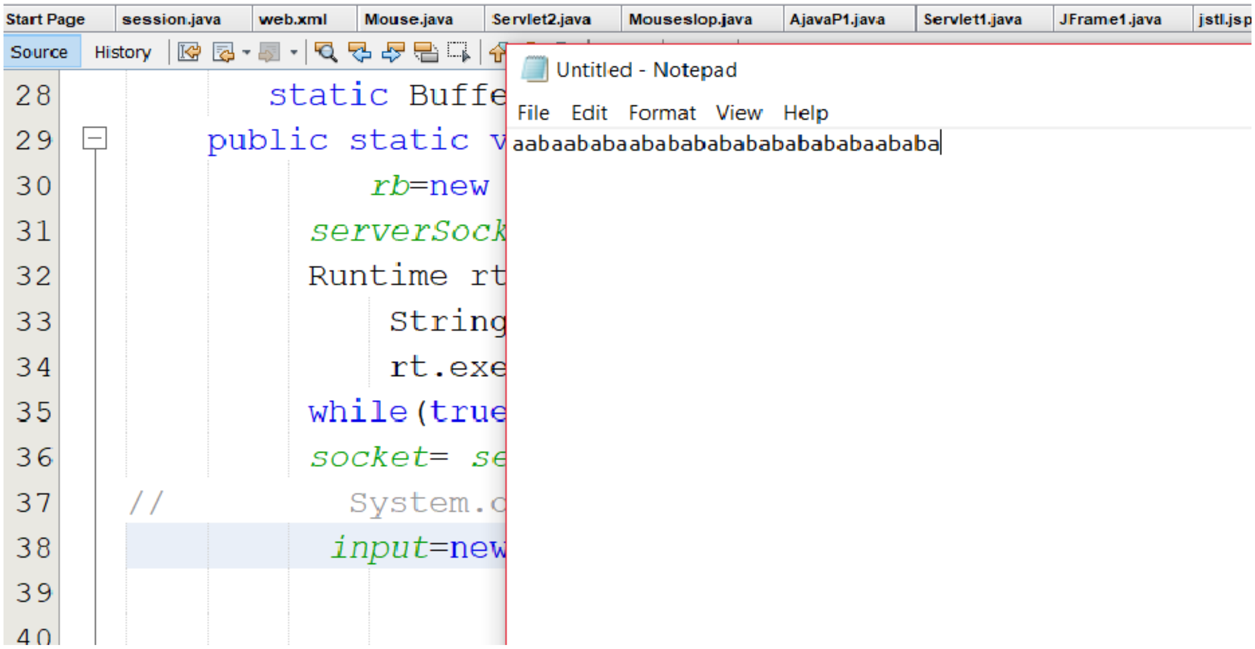
→ Massage are typing in to application:



ababababababababababababaababa

Figure 4.2 Typing in keyboard Activity

➔ Your massage are type on Notepad:

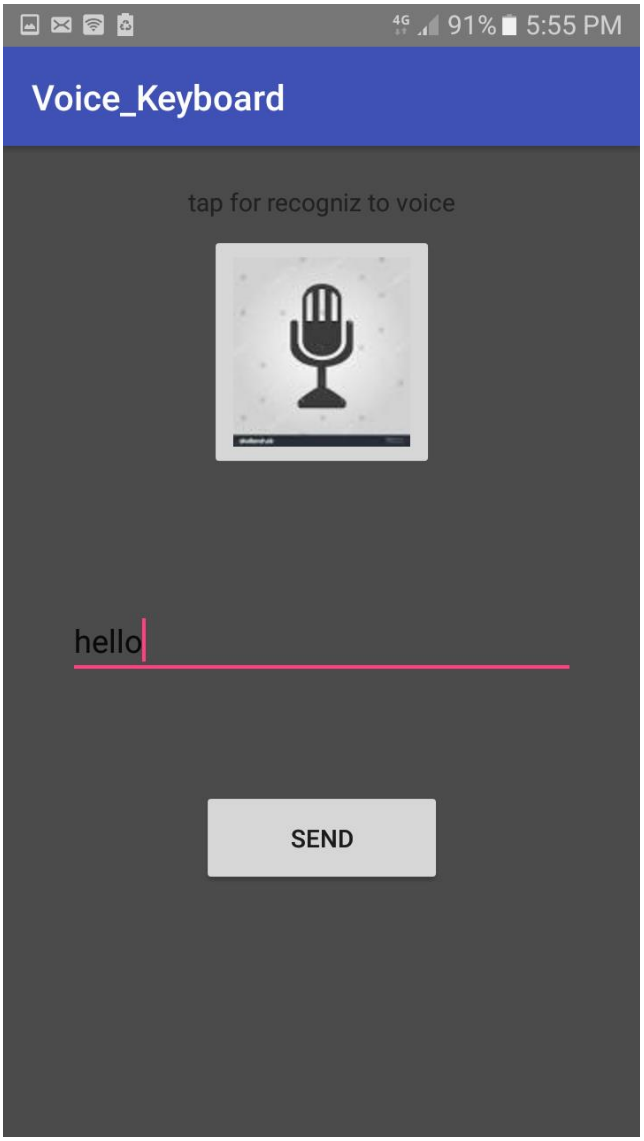


```
        static Buffe
    public static v
            rb=new
            serverSock
            Runtime rt
                String
                rt.exe
            while(true
            socket= se
//              System.c
             input=new
```

```
aabaababaabababababababababaababa
```

Figure 4.3 Typing in Notepad

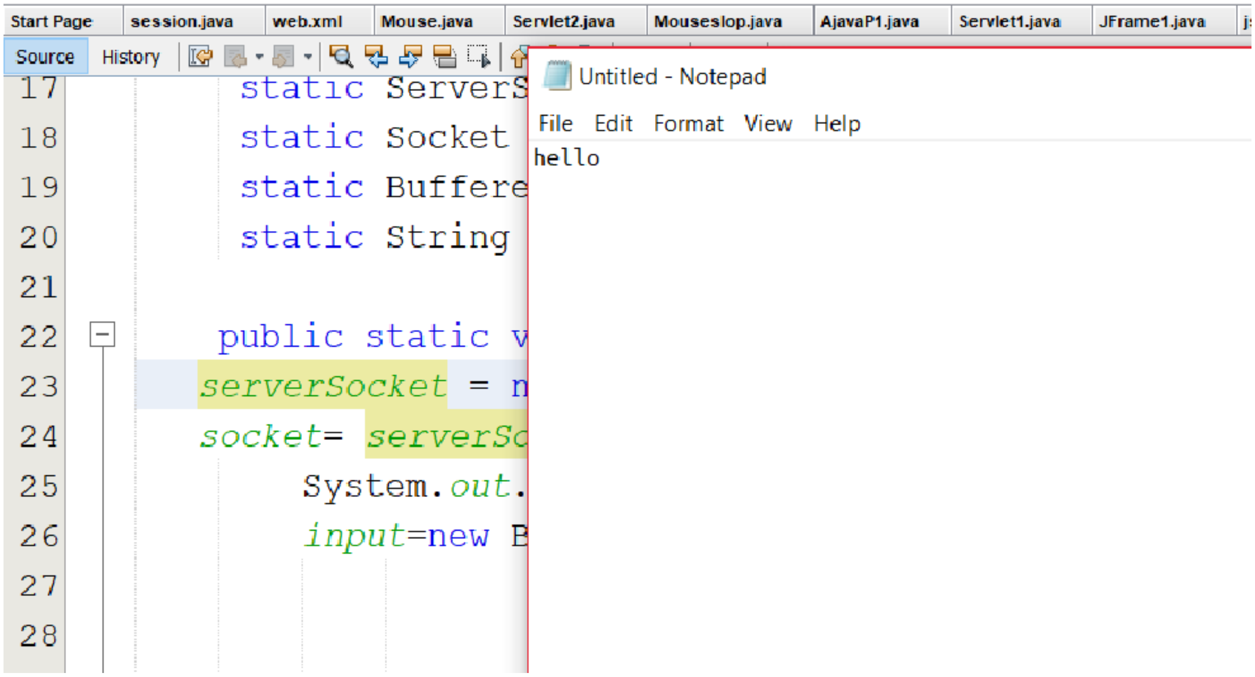
➔Voice Keyboard Activity:



hello

Figure 4.4 Speech recognize by application

➔massage is typing in your Notepad:



```
        static ServerS
        static Socket
        static Buffere
        static String

       public static v
      serverSocket = n
      socket= serverSc
          System.out.
          input=new B
```

```
hello
```

Figure 4.5 typing your voice massage in to Notepad

Chapter 5: CONCLUSION

. We really glad to making this Project and we succeeds in to making Remote Keyboard Application. We learn so many things to make this application("Remote Keyboard Application") for example Android studio, Java Robot class ,socket Programming . and we make this application for android users. User can use our application and connect with his/her computer and type in his/her computer without keyboard . using Voice Recognizer we can record long Speech and rapidly converted in to text and display on monitor. This application is very use full to type long without typing.

REFERENCES

- **https://developer.android.com/**
- **https://stackoverflow.com/**
- **https://www.tutorialspoint.com/**
- **https://www.geeksforgeeks.org/**