A PROJECT REPORT

ON

INTELLIGENT HANDS-FREE SPEECH BASED SMS SYSTEM ON ANDROID

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING
BY

INAMDAR MOHSIN	**(15DCO58)**
KHAN ARSHIYA BANO RAQUEEB	**(15DCO46)**
SHAIKH SHAHEDA FAIYAZ	**(15DCO75)**

UNDER THE GUIDANCE OF
PROF.ANSARI MUKHTAR

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2017-2018
AFFILIATED TO
UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

INTELLIGENT HANDS-FREE SPEECH BASED SMS SYSTEM ON ANDROID

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

INAMDAR MOHSIN (15DCO58)
KHAN ARSHIYA BANO RAQUEEB (15DCO46)
SHAIKH SHAHEDA FAIYAZ (15DCO75)

UNDER THE GUIDANCE OF
PROF. ANSARI MUKHTAR

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2017-2018
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-I-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

Intelligent Hands-Free Speech Based SMS System on Android

submitted by

INAMDAR MOHSIN	**(15DCO58)**
KHAN ARSHIYA BANO RAQUEEB	**(15DCO46)**
SHAIKH SHAHEDA FAIYAZ	**(15DCO75)**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2017-2018, under our guidance.

Date: / /

Prof.ANSARI MUKHTAR
Project Supervisor

Prof.KALPANA BODKE
Project Coordinator

Prof. TABREZ KHAN
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express our sincere thanks to our guide **Prof.ANSARI MUKHTAR**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

We are grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. TABREZ KHAN**, Head of Department of Computer Engineering and **Prof. KALPANA BODKE**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Inamdar Mohsin Haru

Khan Arshiya Raqueeb

Shaikh shaheda faiyaz

Project II Approval for Bachelor of Engineering

This project entitled ***Ïntelligence Hands-Free Speech Based System¨*** by ***Inamdar Mohsin Harun (Roll No: 14DC058), Khan Arshiya Raqueeb (Roll No: 15DC046), Shaikh Shaheda Faiyaz (Roll No: 15DC075)*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

INAMDAR MOHSIN
Roll Number: 14DCO58

KHAN ARSHIYA BANO RAQUEEB
Roll Number: 15DCO46

SHAIKH SHAHEDA FAIYAZ
Roll Number: 15DCO75

ABSTRACT

SMS and Texting is an important feature of using Mobile phone and we also know that the mobile phone usage is spreading over the World rapidly and has gone through the number of features due to new techniques and Developers. This paper is based on creating an application that works on Google libraries and API's for conversion of Text-To-Speech and Speech-To-Text converter. It also works for Searching Contact with the Alphabets and Numeric read. Mainly the goal of the project is, it is for those who not be in the position of using mobile phones for texting ,surfing on web and dialing calls such kind of the communicating features ,so we called it as an Application that is useful for society.

The Application converts your text into the speech, speech into text, search a contacs manually from contact list or can be selected by taking name of a person which is voice based.We can select a multiple contacts for sending a message to multiple people at a time. Previous speech recognition system was difficult to use and it was having a lots of drawbacks, with leads in new technologies and techniques it is possible to generate a desire speech recognition system. This comes with lots of features by using an algorithm i.e Hidden Markov Model(HMM),which makes it possible to get a desire output. Another technologies are Android System, SR (speech recognition)libraries i.e. speech API's which is used in this paper

Keywords: Speech-to-Text, Text-to- speech converter (both side),contacts selection with numeric and Alphabets, Multiple contacts selections.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Now a day's Android try to make an applications more attractive for each and every categories of people. It tries to cover every people in our society. Likewise, android try to improve a speech recognition system for comfort of people who are physically disabled, people who are having a less knowledge about language and to prevent a people from accidents. In this application user is able to access the services of smart phone with their SR (speech recognition) Command. This application is also developed for making a conversation in a very short period of a time. Speech can be processed faster than a text. Sender can send a message from their contact list as well as from a speaker, which automatically select a contact from a user list. This application contains a different services and functionality: speech to text, text to speech, and making a selection of contacts by using numbers, manually and by using a name of a recipient. Speech is a natural way of communication; conversations which are voice based are very clear and understanding. In message system their may be a misunderstanding between people just because accents are hidden from receivers. Sender should speak in a clear manner so that it can be understandable by a system. System uses different HMM models for every word of sentence. There are lots of HMM models are using for making a conversation possible.HMM models keeps every state of words different from each other, so that it can make a correct sentence.

1.1 Purpose

In the era of digital world people uses cell phones for their general purpose uses such as calling messaging ,surfing etc. our application helpful for person who are illiterate such as laymen , old age , servants etc. Our main purpose to develop this application is provide a better GUI as compare to the previous system. This project is mainly aims to EASE for laymen or old age people. For those who they are unable to handover mobile phones. For those who are disabled by physical structure.Our system betterment for those peoples who can't write those who can't type sentences or messages with their spells.

1.2 Project Scope

To perform some basic operations using voice command.It perform speech-to-text and text-to-speech conversion.To develop an Android application which will interact user with voice command to perform some emergency option.User can sends the message to multiple contacts.User can easily send a message to the recipient available in their contact list as well as to the mobile number by voice command. In this Project physically disable person or the person having less knowledge about smart phone or how to access the smart phone can easily access the phone with their voice or speech command.

1.3 Project Goals and Objectives

1.3.1 Goals

Our main goals is to developing a speech based system that provide a speech system to those peoples who are illiterate,people with physical disabilities.It is used for making a conversation faster than previous methods.Speech based system is used to saves a time of users by providing a faster processing.It aims to providing contact selection with speech of users. Speech recognition and understanding systems enabled to have high quality speech recognition, without the need for access to information by keyboard or touch-tone button pushes.

1.3.2 Objectives

Mobile phone users are using short message service (SMS) instead of making voice calls. To satisfy the demands of users, mobile phone manufacturers are constantly adapting and innovating to ensure that they can survive in this competitive market. The important innovation in SMS technology is the speech recognition technology that can convert voice messages into text messages.the main purpose of the system is to support disabled persons who unable to write or read the messages, the old age persons ,laymen and uneducated peoples.To handle incoming as well as outgoing SMS using voice command.To perform some basic operations using voice commands.

1.4 Organization of Report

In Chapter 1, we have considered Project overview under which we have explained various important terminologies like introduction of the project, motivation (what exactly motivates us to create travel guide application),goals of this project.

In chapter 2, We have discussed about various papers that we have referred for our project, we have mentioned the description, pros and cons, and how to overcome the problems of each paper. 3 papers have been referred before the development of this project.

In chapter 3, We have done with the project planning in which every members' capabilities and responsibilities have been detailed. Assumptions and constraints have been discussed and project management approach has been given and also the ground rules for the project has been detailed.

In Chapter 4, We have discussed about the requirement analysis, under which we have discussed about platform requirement(supporting OS for the software), Software requirement and hardware requirement along with feasibility study.

In Chapter 5, We can see the system design and its architecture, various diagrams can be seen in this chapter which represent the software, diagrams included are System architecture, class diagram, sequence diagram, component diagram and deployment diagram.

In Chapter 6, We discussed about the implementation details of the system. This part contains details of the implementation of various modules. In short we describe how the system works.

In Chapter 7, We have shown the test cases and results along with analytical discussion. This part contains the results of the output of our project.

In Chapter 8, We have shown various screenshots of the project.

In Chapter 9,We have concluded the whole project and future scope along with the limitations. Followed by references and appendix.

Chapter 2

Literature Survey

2.1 Intelligent Hands Free Speech based SMS System on Android

Over the years speech recognition has taken the market. The speech input can be used in varying domains such as automatic reader and for inputting data to the system. Speech recognition can minimize the use of text and other types of input, at the same time minimizing the calculation needed for the recognition was difficult to use in any system, but with elevation in technology leading to new algorithms, techniques and advanced tools. Now it is possible to generate the desired speech recognition output. One such method is the hidden markov models which is used in this paper. Voice or signaled input is inserted through any speech device such as microphone,then speech can be processed and convert it to text hence able to send SMS, also Phone number can be entering either by voice or you may select it from contact list. Voice has opened up data input for a variety of user's such as illiterate, handicapped, as if the person cannot write then the speech input is a boon and other's too which can lead to better usage of the application.

2.1.1 Advantages of Paper

-to-text and text-to-speech conversionmultiple contact selection. Phone number can be entering either by voice or you can select it from contact list.

2.1.2 Disadvantages of Paper

- If user speaks his name for contact, it will be displayed as invalid contact.
- Multiple contact selection manually.

2.1.3 How to overcome the problems mentioned in Paper

- User can select contacts with names.

- Multiple contact selection with names.

2.2 Intelligent Hands Free Speech based SMS System on Android

Over the years speech recognition has taken the market. The speech input can be used in varying domains such as automatic reader and for inputting data to the system. Speech recognition can minimize the use of text and other types of input, at the same time minimizing the calculation needed for the process. A decade back speech recognition was difficult to use in any system, but with elevation in technology leading to new algorithms, techniques and advanced tools. Now it is possible to generate the desired speech recognition output. One such method is the hidden markov models which is used in this paper. Voice or signaled input is inserted through any speech device such as microphone, then speech can be processed and convert it to text hence able to send SMS, also Phone number can be entering either by voice or you may select it from contact list. Voice has opened up data input for a variety of user's such as illiterate, handicapped, as if the person cannot write then the speech input is a boon and other's too which can lead to better usage of the application.

2.2.1 Advantages of Paper

- Developed Speech recognizer system tested for a SMS sending application and found that it recognizes the speech to an accuracy of more than 90
- Enter phone number by speech or select contact from contact list. As user presses select contact here by selecting name of person it gives all phone numbers of that person in phone contact list box. Now it is possible to send sms to all numbers of same person on one click which results in reducing time of searching each number.

2.2.2 Disadvantages of Paper

- If any user try to insert any other character into the information an error would be displayed e.g. if user speaks his name for contact, it will be displayed as invalid contact.
- Multiple language selection can not be perform.

2.2.3 How to overcome the problems mentioned in Paper

- Contact selection with names.
- Multiple language selection for speech input and output.

2.3 Technical Review

Our system will be an andriod application.The technologies that we are using in our project are as follows:-1.java, 2.Android , 3.Google API

2.3.1 Advantages of Technology

- Our system is based on andriod,most of the people uses andriod phones.Thus our system can target large amount of users.
- we will be using Android ,for Whole project code is very advantageos as the data is stored in databse of android libararies of users phones.
- Due to google API speech to text and text to speech is become more easiest and frequent in nature.

2.3.2 Reasons to use this Technology

- Android creates an App in Efficient way which is easy to use.
- Jva provides and support to the system.
- Google API is used for conversion.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Inamdar mohsin haroon	Database design , UI Design
2	Shaikh shaheda faiyaz	Module integration
3	Khan arsiya rakeeb	System testing

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Inamdar mohsin haroon	Team Leader	UI Design
2	Shaikh shaheda faiyaz	Team member	System integration
3	Khan arsiya rakeeb	Team member	System testing

3.3 Assumptions and Constraints

1. Assumptions
   The team member should know the andriod coding.
   To deveople a system better available in the market.
   No significant changes in technology to change our system.

2. Constraints
   The project should be completed before the deadline.
   The module which is to be added should be known in advance.
   The user should be able to understand how the system works

3.4 Project Management Approach

We have use Agile methadology for the development of this project.The Agile Project Management Process is a value-centered methods of project management that allows projects to get processed in small phases or cycles. The methodology is one that is extremely flexible and projects that exhibit dynamic traits would benefit from this process as you would find that project managers working in this environment treat milestones the goal being to continuously adapt to abrupt changes from our project guide feedback.

3.5 Ground Rules for the Project

1. We treat each other with respect.
2. We intend to develop personal relationships to enhance trust and open communication.
3. We value constructive feedback. We will avoid being defensive and give feedback in a constructive manner.
4. As team members, we will pitch in to help where necessary to help solve problems and catch-up on behind schedule work.
5. Additional meetings can be scheduled to discuss critical issues or tabled items upon discussion and agreement with the team leader.
6. One person talks at a time; there are no side discussions
7. When we pose an issue or a problem, we will also try to present a solution.

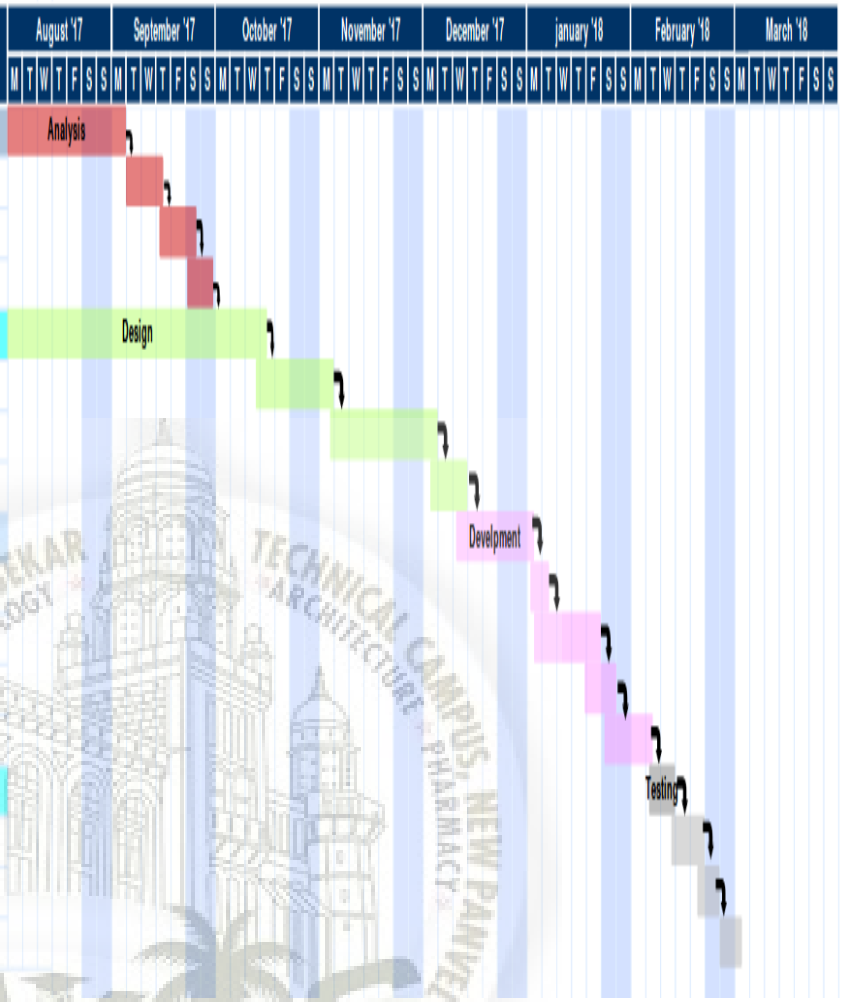
3.6 Project Budget

The budget for this project is very low as most of the tools we have use are open source.Following are the budget for the project

1. Operating System:linux mint (Open Source).
2. IDE:Andriod Studio (Open Source).
3. API:Google SR API. (Open Source)

3.7 Project Timeline

	Task Name	Duration	Start	ETA
1	**Analysis**	35 days	08.08.17	11.09.17
2	Meetings with faculties	5 days	12.09.17	16.09.17
3	Discussions with team members	2 day	17.09.12	19.09.17
4	Analysis complete	19 days	20.09.17	08.10.17
5	**Design**	60 day	22.08.12	21.10.12
6	Design software	24 days	22.10.12	14.11.17
7	Design interace	27 days	15.11.17	11.12.17
8	Create design specification	5 day	12.12.17	16.12.17
9	**Development**	20 days	17.12.17	05.01.18
10	Development system modules	08 days	06.01.18	13.01.18
11	System modules	15 days	07.01.18	24.04.18
12	Perform initial testing	5 days	22.01.18	26.01.18
13	Development complete	5 days	27.01.18	03.02.18
14	**Testing**	4 days	04.02.19	08.02.18
15	Performance system testing	10 days	09.02.18	19.02.18
16	Document issues found	7 days	20.02.18	27.02.18
17	Correct issues found	5 days	28.02.18	01.03.18



Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

he application here will use the SR with Google server which uses HMM method. The description of how the speech recognized are as follows. Initially a speech inputted and sound fluctuates which can be represented by set of signals. Signals which are generates application is depends on quality of sound .If sound quality is high then signal level increases at a high level. Speech recorded a recorder. After a recording a done, speech divided into set of frames or words and every words and phrases works as independently Additional sounds comes with speech is filtered by a MFCC model, so that it can be easily understood by a system .Background voice and low quality voice all should be filter to convert it into desired text. Then algorithm is used for making a conversion from speech to text at sender site. These converted texts send to receivers. As above processes Text-to-Speech is also done.

4.1.2 Product Features

- Speech to text conversion:- Speech Recognition stands majorly on five pillars that are, feature extraction, acoustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. The inputs data i.e. voice are first converted to digital signal and are sampled on time and amplitude axis.
- SMS:-The user will be having 2 ways to send SMS in this project. He can send directly by telling the mobile number and the message. Here the user will tap on the mic and will tell the number it will be displayed in the edit text. Only numbers are allowed to be spelled in the number edit text. He needs to do the same for the message edit text also.
- View and Add contacts:-Here instead of sending message directly the user was also allowed to add and view contacts. He will add contacts by spelling and all

the contacts will be displayed in the list view. By clicking on the list the text to speech conversion takes place and it will be spelled to the user.

4.1.3 User Classes and Characteristics

- Text-to-speech:conversion is to be done on the basis of message.
- Speech-to-text:conversion is to be done on the basis of voice quality.
- Multiple contact selection: Multiple contact can be select by manually or by speech.

4.1.4 Operating Environment

Software Requirements
For Implementation:

1. Programming Language: Java Platform
2. IDE: Android Studio
3. Operating system : Windows 7
4. Tool Kit : Android 2.3 ABOVE

For Deployment : Android Platform 2.3.3.

Hardware Requirements
For Implementation

1. Processor : Core i5
2. 1280*800 minimum Screen Resolution
3. Hard Disk : 40 GB.
4. Ram : 3GB ; plus 1 GB for Android Emulator

For Deployment:

1. MOBILE : ANDROID

4.1.5 Design and Implementation Constraints

The visually impaired segment of the population, the inability to read has a substantive negative impact on their quality of life. Printed text (books, magazines, menus, labels, etc.) still represents a sizable portion of the information this group needs to have unrestricted access. Over the years speech recognition has taken the market. The speech input can be used in varying domains such as automatic reader and for inputting data to the system.

4.2 System Features

The major features of our system it gives message services and phone number can be selected manually or by using a voice.multiple contacts can be selected at a time by voice command and can by manually.

4.2.1 System Feature

1. Text-to-speech.
2. Speech-to-text.
3. Multiple contact selection.

Description and Priority

1. Working of Speech-To-Text Recognition :

   First the speech is taken as the input, now it analyzed by the speech analysis with the help of speech dictionary or speech to text conversion database and then it further checking by the vocabulary database database by the selection of words ,phrases according to the sound and ascent of the user then it finally converts all the speech into the text and can send this speech by the text message.

   Text-To-Speech Recognition: First the input is taken as the text it analyzed by text analysis with the help of text dictionary or text to Speech converter and it sends to the speech database which selects the units of words spoken on the mike now it further sends speech generation module and on the basis of this process text is converted into the text.

Stimulus/Response Sequences

1. Speech-to-Text conversion.
2. Contact selection with voice command.
3. Multiple Contact selection.
4. Notification.

Functional Requirements

1. The user should be able to handle the system.
2. The system should able to get conversion of Text.
3. The system should able to get conversion of Speech.
4. Multiple contact will be selected by voice and manual.

4.3 External Interface Requirements

4.3.1 User Interfaces

Speech is a very natural and basic way in human-to-human communication. For communicating purpose in this digital world peoples uses their smart phones with messages and calling. In this system if a user unable to type or read the message data coming from sender side can be easily read by mic on commanding Like ”Speak” and can write easily by tap on mic it automatically converts speech to text and commanding like send it send to selected contact.

4.3.2 Hardware Interfaces

In this System mobile or smart phones are used as hardware for communicating purpose. Firstly in our smart phones we have to install application after installing we have to first login into the system after login system is ready to communicate over the available contacts.for sending and receiving smart phone is used as a hardware communication channel .

4.3.3 Software Interfaces

For sending message, voice command is provided to open application to send message .Once application is open, it will ask for contact of receiver, then it will ask for the message to be sent, then it will speak that message to check, after conforming the message it will send it to corresponding receiver. Every time the application asks anything, through voice and user also provides response with voice commands that are told by guide.As part of sending message application is responsible for voice to text transmission to convert message told by user into text,text to voice to check message, and for interaction through voice.

4.3.4 Communications Interfaces

1. The major communication for location purposes will be done by google api,the data is accessed by the google by using the google apis.

2. The interface between the android SR algorithm and the system will be done by using http protocol

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

Speech based solutions have taken center stage with growth in the services industry where there is a need to cater to a very large number of people from all strata of the society. While natural language speech interfaces are the talk in the research community, yet in practice, menu based speech solutions thrive. Typically in a menu based speech solution the user is required to respond by speaking from a closed set of words when prompted by the system. A sequence of human speech response to the IVR prompts results in the completion of a transaction.

4.4.2 Safety Requirements

If there is any damage to the large amount of the data in the database than the whole system will go down.The database should be periodically maintained and have to keep upon it.The data which is updated by the user should be commited in the database.

4.4.3 Security Requirements

The major security requirements for the system will be the safeguarding of the user data from any kind of exploit. Inorder to protect the user data the data is not stored in local databases we will be storing in the cloud for better security.

Chapter 5

System Design

5.1 System Requirements Definition

System requirement definitions specify what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement.

5.1.1 Functional requirements

1. Speech to Text : Feature is useful to elderly people who have bad hearing issues.Speech Recognition in Android SDK 1.5 Our software listens to speech and writes them into text.
2. Text to speech: For elderly that have problems seeing our software will insure that every menu and options will be read out load. When an option is pressed, the name of the option is read out load.

Our proposed system using speech recognition technologies of mobile devices , speech to text ,text to speech and wireless communication technologies was implemented by Java programming language. The proposed application was designed and developed on Android as well.

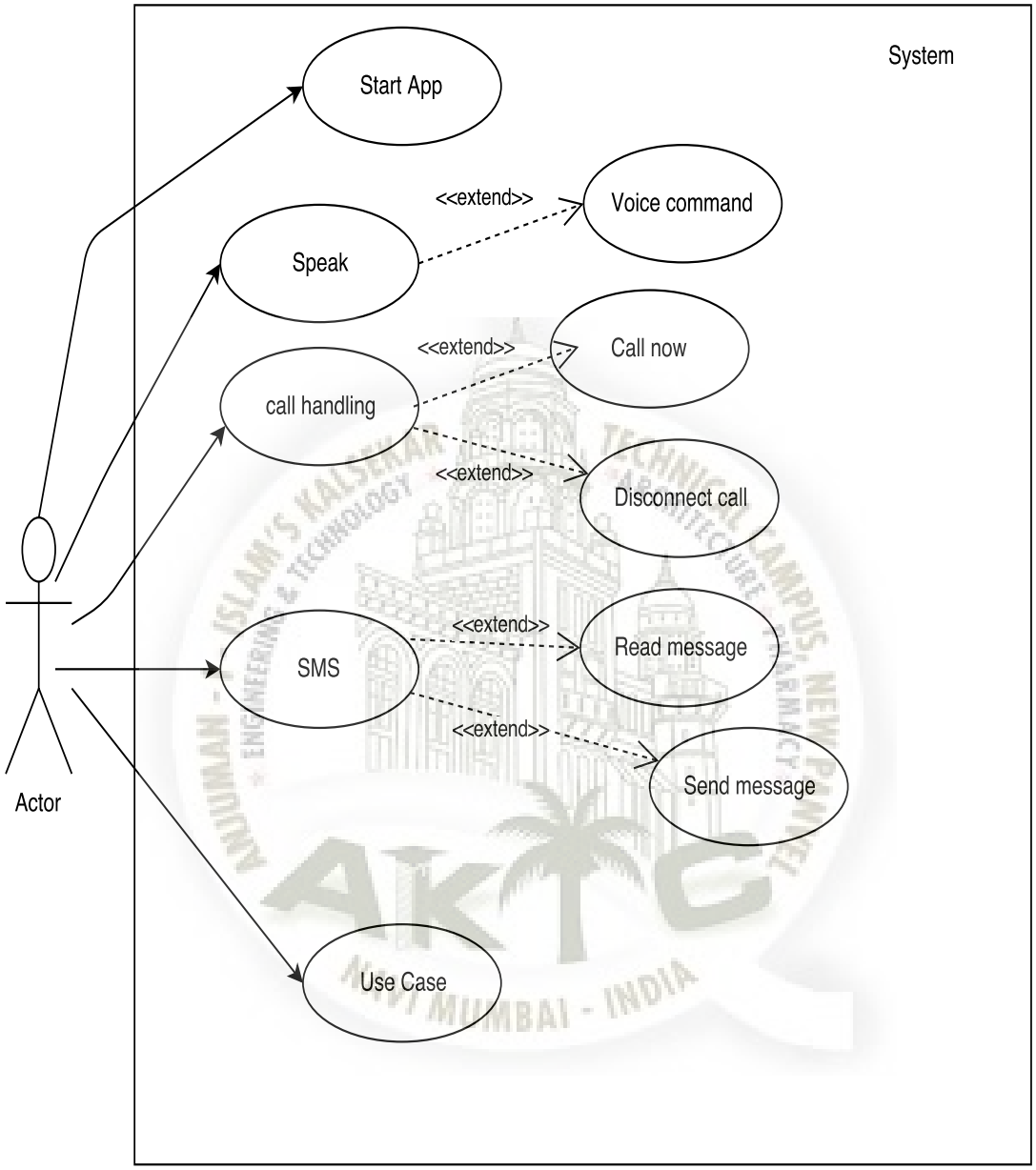


Figure 5.1: Usecase Diagram

Data-flow Diagram

DFD level 0

Speech Recognition System will take input from one application and convert it by using Speech recognition(SR) and output will represent in an another application.It performs speech-to-text and vice versa.

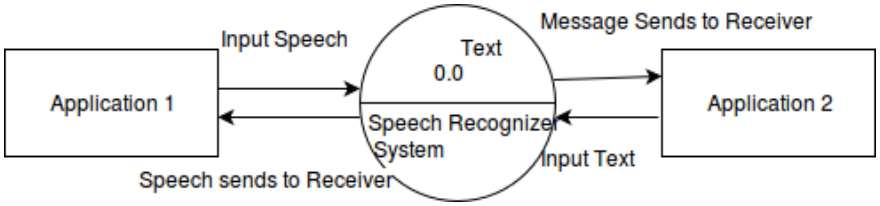


Figure 5.2: DFD level 0

DFD level 1

Application is get started and we can further search contact can be select at the same for this purpose we can use speech or voice command too select particular task .for the purpose of text to speech voice can be recognized and n-datagram is use to convert. Level 1 DFD's aim to give an overview of the full system. They look at the system in more detail. Major processes are broken down into sub-processes. Level 1 DFD's also identifies data stores that are used by the major processes. When constructing a Level 1 DFD, we must start by examining the Context Level DFD. We must break up the single process into its sub-processes. We must then pick out the data stores from the text we are given and include them in our DFD. Like the Context Level DFD's, all entities, data stores and processes must be labelled. We must also state any assumptions made from the text.

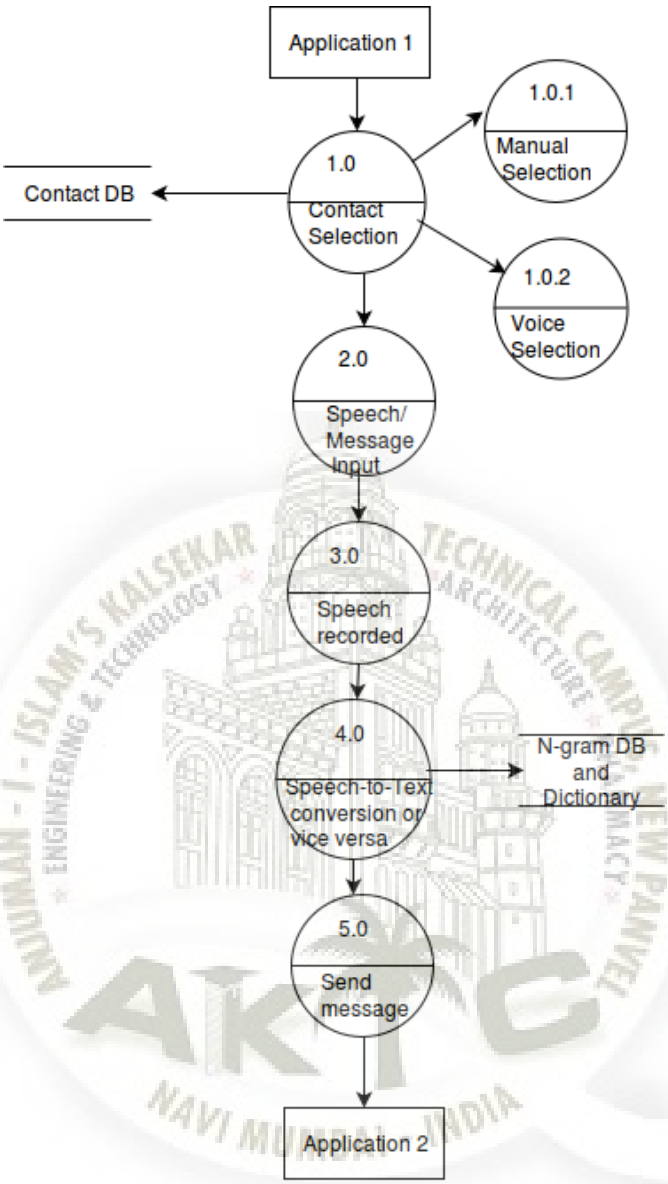


Figure 5.3: DFD level 1

5.1.2 System requirements (non-functional requirements)

1. Usability requirement-A user interface for updating the information for travel that would allow the system to better adapt the overall system.
2. Efficiency requirement-The application should be able to response quickly the user's request.
3. Performance requirement-The application should be able to response the queries submitted by the user without delay.
4. Reliability requirement-The application should work under all conditions and performed the required functionality.

Database Schema/ E-R Diagram

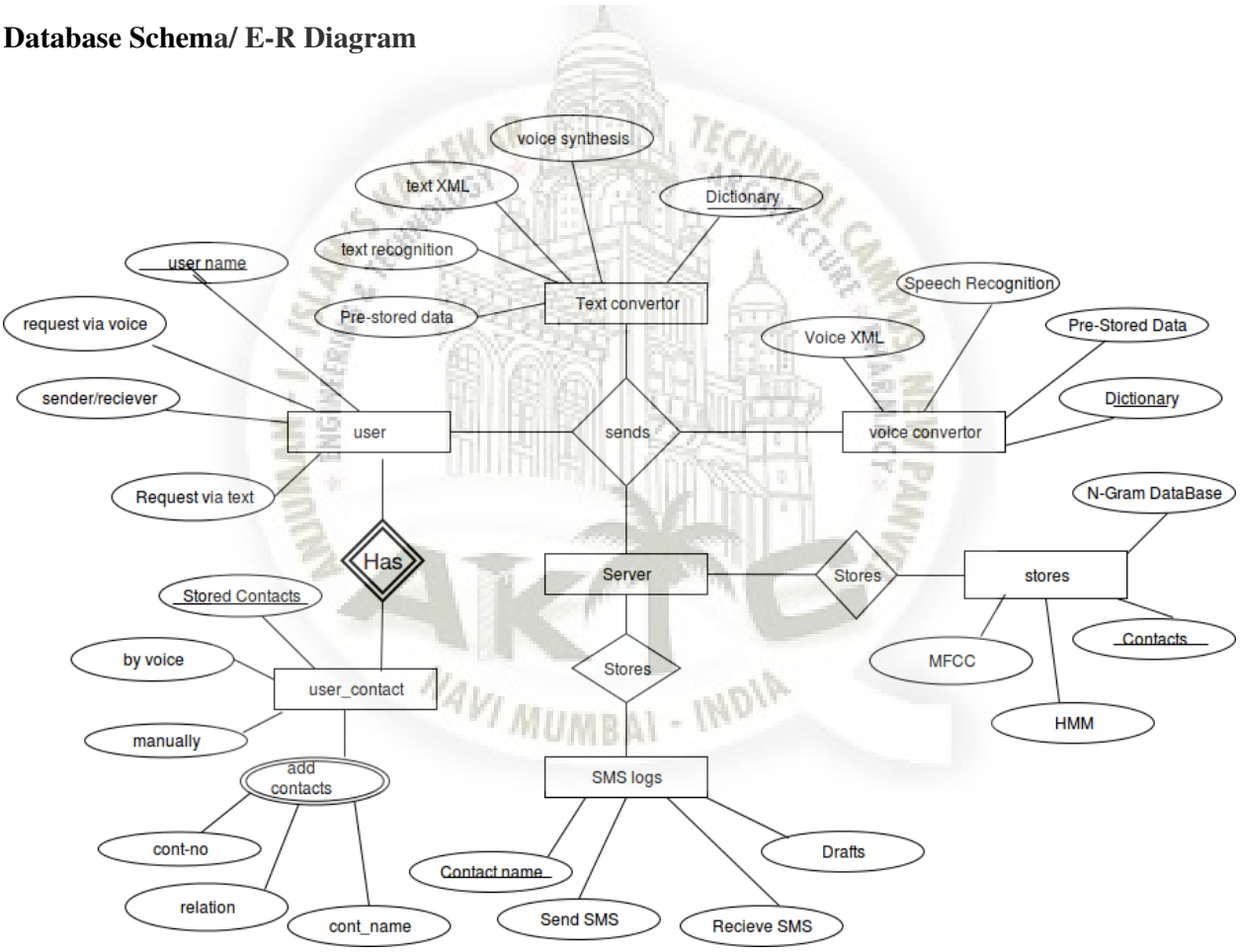


Figure 5.4: E-R Diagram

This is the ER diagram of the system in which the modules which will be there after the deployment are shown

5.2 System Architecture Design

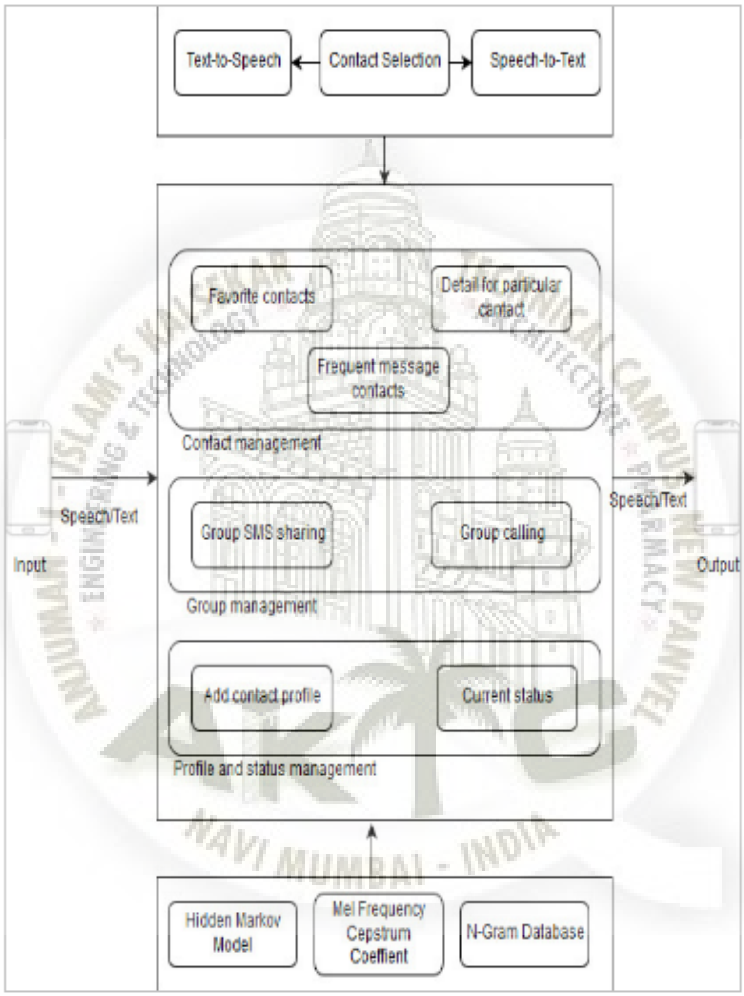


Figure 5.5: System Architecture

In the below System Architecture,the application here will use the SR with Google server which uses HMM method. The description of how the speech recognized are as follows. Initially a speech inputted and sound fluctuates which can be represented by set of signals. Signals which are generates application is depends on quality of sound .If sound quality is high then signal level increases at a high level. Speech recorded a recorder .After a recording a done, speech divided into set of frames or words and every words and phrases works as independently. Additional sounds comes with speech is filtered by a MFCC model, so that it can be easily understood by a system.Background voice and low quality voice all should be filter to convert it into desired text. Then algorithm is used for making a conversion from speech to text at sender site.These converted texts send to receivers.

5.3 Sub-system Development

Application will always be in running state at the background once it is started.The application is built on top of SMS, so that once application is installed on mobile,all SMS related activities are by default performed by application. With respect to user perspective,application working is divided in two ways–One application is used for sending messages and other when application is used to read received messages. As part of sending message application is responsible for voice-to-text transmission to convert message told by user into text, text-to-voice to check message, and for interaction through voice. Modules :

1. Voice Recognition.
2. Text Recognition.
3. Multiple Contact Selection.

5.3.1 Speech recognition

First the speech is taken as the input, now it analyzed by the speech analysis with the help of speech dictionary or speech to text conversion database and then it further checking by the vocabulary database database by the selection of words, phrases according to the sound and ascent of the user then it finally converts all the speech into the text and can send this speech by the text message.

Speech Recognition Flow Diagram

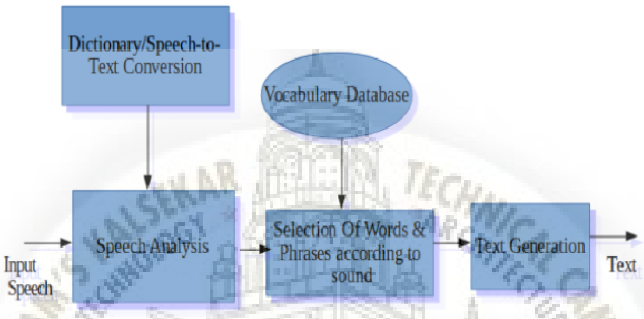


Figure 5.6: Speech Recognition

5.3.2 Text Recognition.

First the input is taken as the text it analyzed by text analysis with the help of text dictionary or text to Speech converter and it sends to the speech database which selects the units of words spoken on the mike now it further sends speech generation module and on the basis of this process text is converted into the text.

Text Recognition Flow Diagram

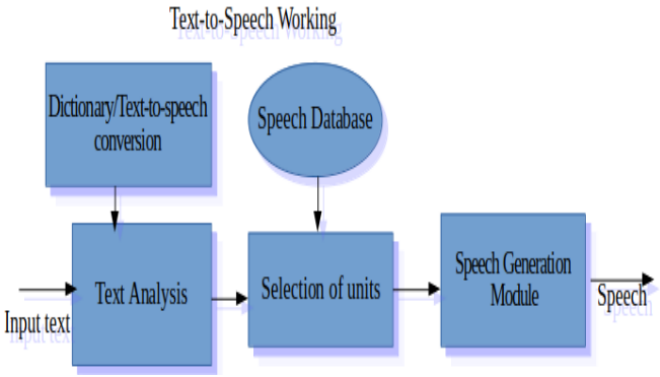


Figure 5.7: Text Recognition

5.3.3 Multiple Contact Selection

multiple contact can be selected at a time by the user it can be done by manually or can be done by speech command. Now the message can send to selected contacts or it can be typed by manually or by speech.

Multiple Contact Selection Flow Diagram

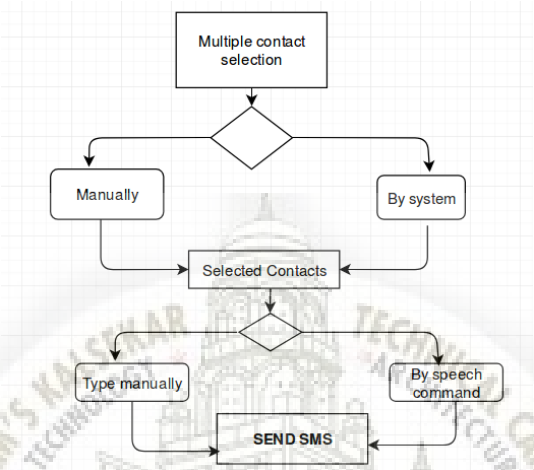


Figure 5.8: Multiple Contact Selection

5.4 Systems Integration

The speech input can be used in varying domains such as automatic reader and for inputting data to the system. Speech recognition can minimize the use of text and other types of input, at the same time minimizing the calculation needed for the process. A decade back speech recognition was difficult to use in any system, but with elevation in technology leading to new algorithms, techniques and advanced tools. Now it is possible to generate the desired speech recognition output. One such method is the hidden Markov models which is used in this paper. Voice or signaled input is inserted through any speech device such as microphone, then speech can be processed and convert it to text hence able to send SMS, also Phone number can be entering either by voice or you may select it from contact list. Voice has opened up data input for a variety of user's such as illiterate, handicapped, as if the person cannot write then the speech input is a boon and other's too which can lead to better usage of the application.

5.4.1 Class Diagram

The interaction between objects arranged in time sequence is described using a task event diagram.In other words,this diagram is used to describe how tasks respond to each of their input events or messages. The order in which messages are passed between tasks can be used to help engineers in implementing the system tasks more efficiently.

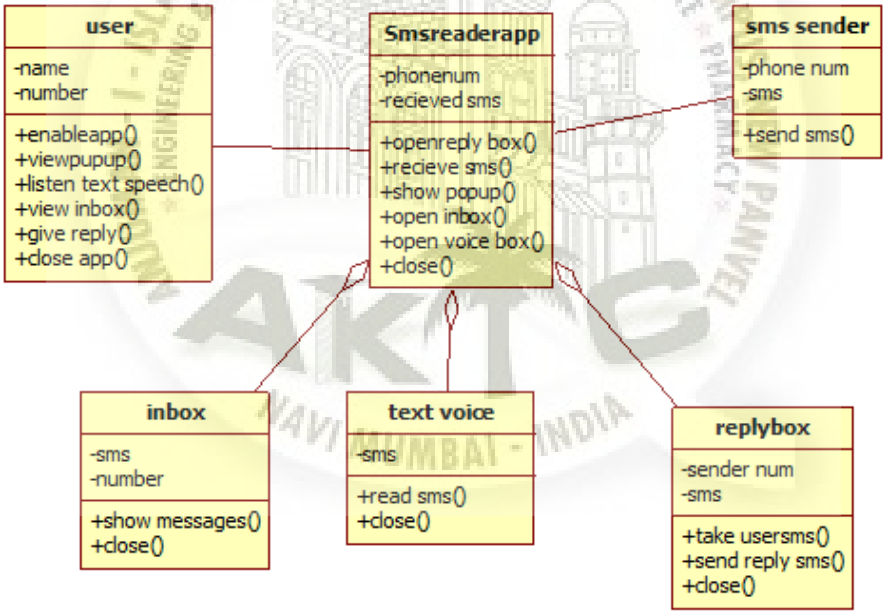


Figure 5.9: Class Diagram

5.4.2 Sequence Diagram

This is the Sequence Diagram for our system which shows the sequential flow of our system when particular user searches or visit a new location this are explained in the below figure:

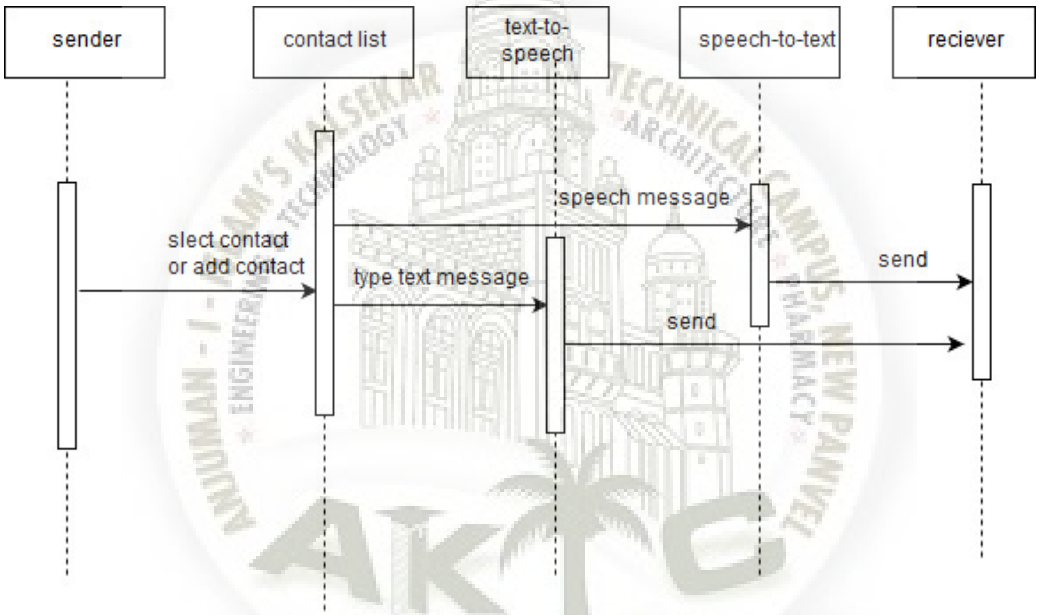


Figure 5.10: Sequence Diagram

5.4.3 Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities .This diagram of our system shows the components which are included in our system :

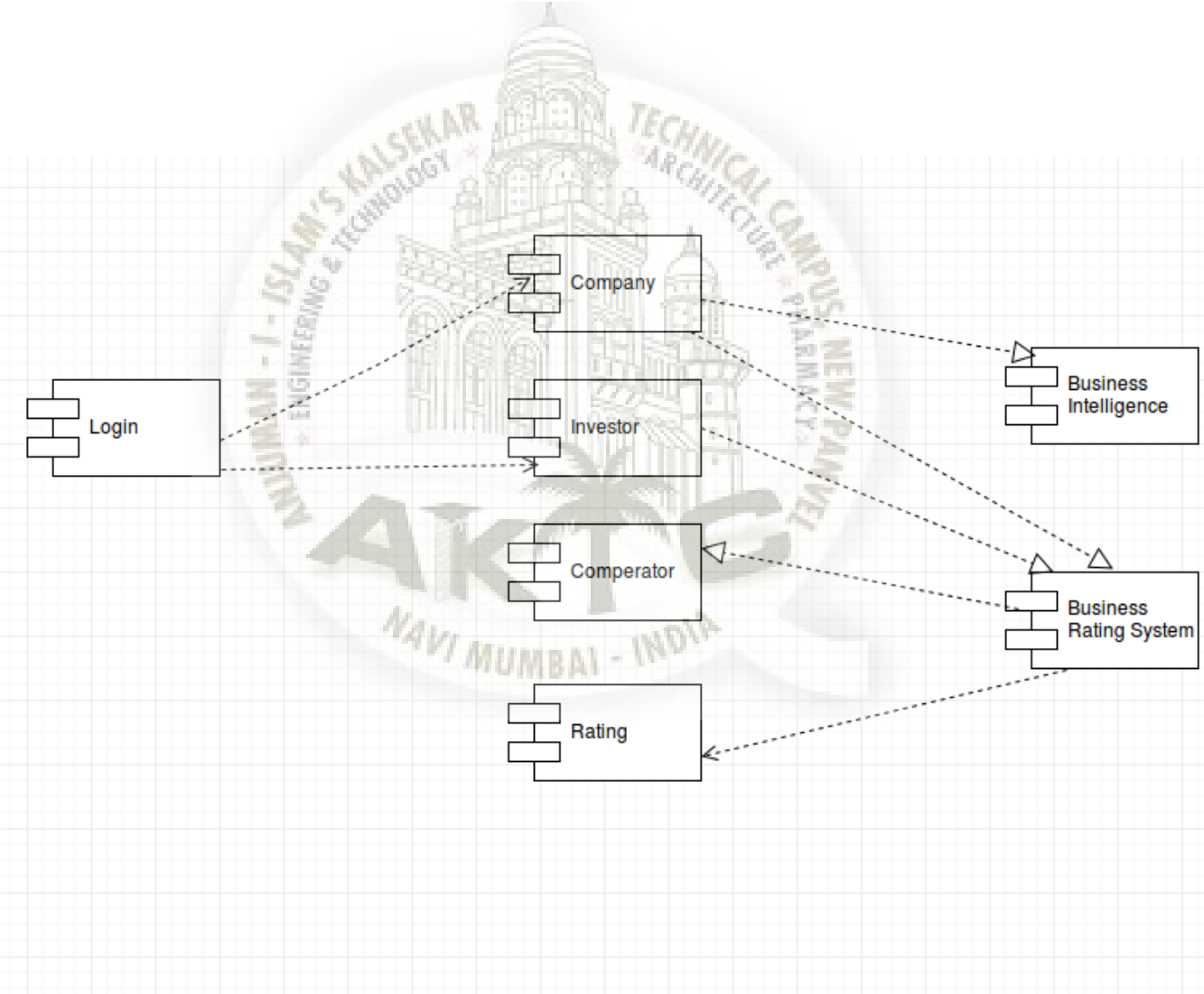


Figure 5.11: Component Diagram

5.4.4 Deployment Diagram

This diagram of our system shows the deployment stages of our system which all together makes the whole system:

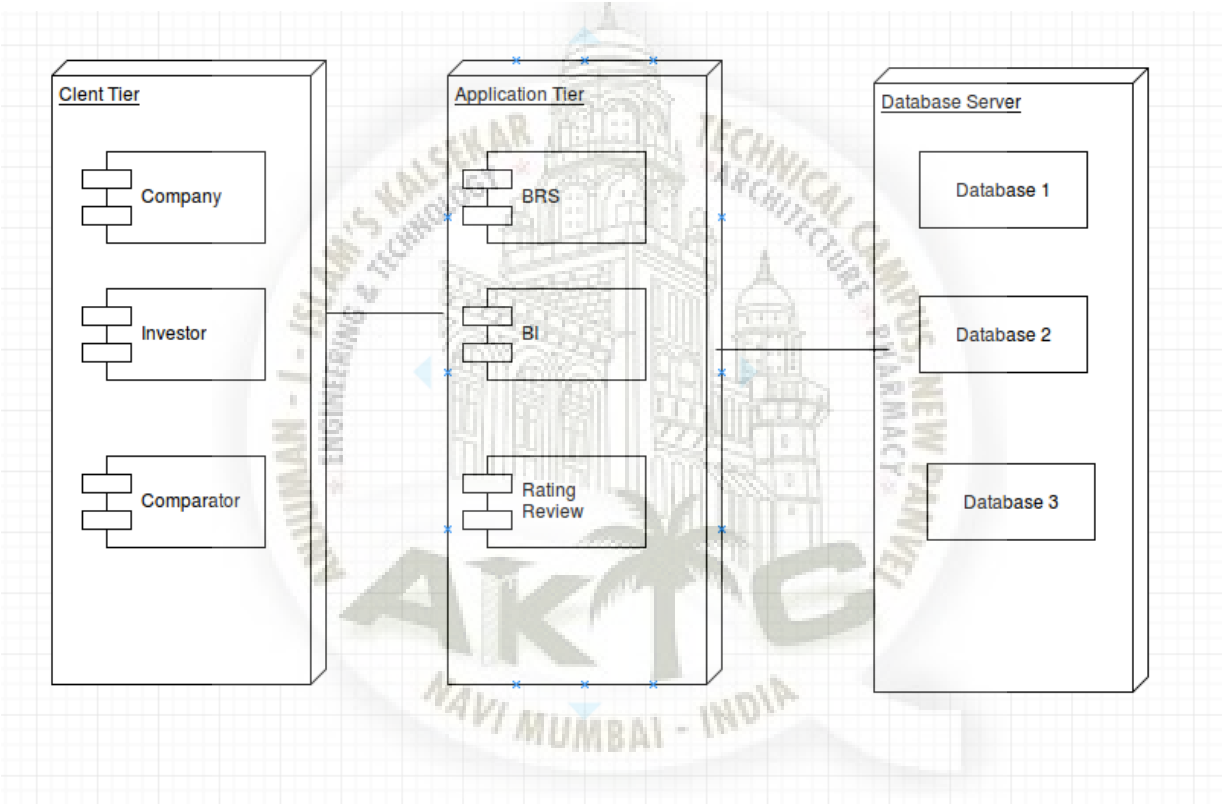


Figure 5.12: Deployment Diagram

Chapter 6

Implementation

6.1 Message

This module consist of feature such as:create message and read message.It consist of message,date of message,name of contact.User can perform both i.e speech-to-text and text-to-speech.This module consist of actual conversion of speech-to-text and text-to-speech.

6.1.1 Message list

User can read a message from a message list.Message list contains message,contact number and date of message .User can read this message or they can also use a speech feature.When user click on message speech will generate which speaks a message,date and contact of message.

Figure 6.1: Massage List

6.1.2 Create speech-to-text message

User can create a new message from a create message module.Create message module consist of two fields such as:number field and message field.User has to fill these information in a fields to send a new message.User has to speech a number and message in a mice then these speech get convert into text and then send to a receiver

Figure 6.2: Speech-to-text

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.mohsin.vbs.MainActivity">

    <ImageButton
        android:id="@+id/micButton"
        android:layout_width="89dp"
        android:layout_height="89dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/background_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.445"
        app:srcCompat="@drawable/micc" />

    <ImageButton
        android:id="@+id/contButton"
        android:layout_width="89dp"
```

```
        android:layout_height="89dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/background_light"
        android:onClick="conAdClick"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.799"
        app:srcCompat="@drawable/con" />

    <ImageButton
        android:id="@+id/msgButton"
        android:layout_width="89dp"
        android:layout_height="89dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/background_light"
        android:onClick="msgClick"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.093"
        app:srcCompat="@drawable/msg" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="31dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Message"
        android:textColor="@android:color/holo_blue_dark"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.273" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="31dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Tap To Speak"
```

```
        android:textColor="@android:color/holo_blue_dark"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.573" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="31dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Contacts"
        android:textColor="@android:color/holo_blue_dark"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.89" />

</android.support.constraint.ConstraintLayout>

package com.example.mohsin.vbs;

import java.util.ArrayList;
import java.util.Locale;
import android.annotation.SuppressLint;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.media.Image;
import android.speech.RecognizerIntent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity
{
    // SendMsg sm=new SendMsg();
     ImageButton msgBtn,micBtn,conBtn;
     private final int REQ_CODE_SPEECH_INPUT=100;

     @Override
     protected void onCreate(Bundle savedInstanceState)
     {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.activity_main);
         msgBtn=(ImageButton) findViewById(R.id.msgButton);
         conBtn=(ImageButton) findViewById(R.id.contButton);
```

```
151         micBtn=(ImageButton) findViewById(R.id.micButton);
152
153
154
155        micBtn.setOnClickListener(new View.OnClickListener()
156         {
157             public void onClick(View view)
158              {
159                  pspchInput();
160
161              }
162
163         });
164     }
165
166     public void msgClick(View view)
167     {
168         Intent intent=new Intent(MainActivity.this,MsgActivity.class);
169         startActivity(intent);
170     }
171
172     public void conAdClick(View view)
173     {
174
175         Intent in=new Intent(this,contactActivity.class);
176         startActivity(in);
177     }
178
179
180
181
182     public void pspchInput()
183     {
184         Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
185         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
186                 RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
187         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
188         intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak The Message");
189         try {
190             startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
191
192            // Intent inte = new Intent(this,SendMsg.class);
193             //startActivity(inte);
194         } catch (ActivityNotFoundException a) {
195
196             Toast.makeText(getApplicationContext(),
197                     "Sorry! Speech recognition is not supported in this device."
                            ,
198                     Toast.LENGTH_SHORT).show();
199
200         }
201     }
202
203     protected void onActivityResult(int requestCode, int resultCode, Intent data
        )
204     {
205         super.onActivityResult(requestCode, resultCode, data);
206
207         switch (requestCode) {
208             case REQ_CODE_SPEECH_INPUT: {
209                 if (resultCode == RESULT_OK && null != data) {
```

```
210
211                     ArrayList<String> result = data.getStringArrayListExtra(
                            RecognizerIntent.EXTRA_RESULTS);
212                     String text=result.get(0);
213
214                     //Intent inte =new Intent(this,SendMsg.class);
215                     //startActivity(inte);
216                     //sm.msg.setText(text);
217                     Toast.makeText(this,text,Toast.LENGTH_SHORT).show();
218
219
220                     if(text.equals("open messages")||text.equals("open message")
                            )
221                     {
222
223                         Intent intent=new Intent(MainActivity.this,MsgActivity.
                                class);
224                         startActivity(intent);
225                     }
226                     else if(text.equals("create message")||text.equals("create
                            messages"))
227                     {
228                         Intent intent = new Intent(this, SendMsg.class);
229                         startActivity(intent);
230
231
232                     }
233                     else if(text.equals("open contact")||text.equals("open
                            contacts"))
234                     {
235                         Intent intent = new Intent(this, contactActivity.class);
236                         startActivity(intent);
237                     }
238                     else if(text.equals("add contact")||text.equals("add
                            contacts"))
239                     {
240                         Intent intent = new Intent(this, addContacts.class);
241                         startActivity(intent);
242                     }
243
244                 }
245                 break;
246             }
247
248         }
249     }
250
251 }
252 <?xml version="1.0" encoding="utf-8"?>
253 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
        android.com/apk/res/android"
254     xmlns:app="http://schemas.android.com/apk/res-auto"
255     xmlns:tools="http://schemas.android.com/tools"
256     android:layout_width="match_parent"
257     android:layout_height="match_parent"
258     tools:context="com.example.mohsin.vbs.MsgActivity">
259
260     <android.support.design.widget.FloatingActionButton
261         android:id="@+id/idCreateMsg"
262         android:layout_width="90dp"
263         android:layout_height="91dp"
```

```
264          android:layout_marginBottom="8dp"
265          android:layout_marginEnd="8dp"
266          android:layout_marginStart="8dp"
267          android:layout_marginTop="8dp"
268         android:onClick="sendClick"
269          android:tint="@android:color/transparent"
270          app:backgroundTint="@android:color/transparent"
271          app:elevation="0.1sp"
272          app:layout_constraintBottom_toBottomOf="parent"
273          app:layout_constraintEnd_toEndOf="parent"
274          app:layout_constraintHorizontal_bias="0.909"
275          app:layout_constraintStart_toStartOf="parent"
276          app:layout_constraintTop_toTopOf="parent"
277          app:layout_constraintVertical_bias="0.98"
278          app:rippleColor="@android:color/transparent"
279          app:srcCompat="@drawable/smsg" />
280
281      <ListView
282          android:id="@+id/idList"
283          android:layout_width="0dp"
284          android:layout_height="0dp"
285          android:layout_marginBottom="8dp"
286          android:layout_marginEnd="8dp"
287          android:layout_marginStart="8dp"
288          android:layout_marginTop="8dp"
289          android:background="#E3F2FD"
290          android:divider="#FFFFFF"
291          android:dividerHeight="10sp"
292          android:fadingEdge="vertical"
293          android:fadingEdgeLength="5sp"
294          android:fastScrollAlwaysVisible="true"
295          android:fitsSystemWindows="true"
296          android:foregroundGravity="top|bottom"
297          android:headerDividersEnabled="false"
298          android:scrollbars="vertical"
299
300          app:layout_constraintBottom_toBottomOf="parent"
301          app:layout_constraintEnd_toEndOf="parent"
302          app:layout_constraintHorizontal_bias="1.0"
303          app:layout_constraintStart_toStartOf="parent"
304          app:layout_constraintTop_toTopOf="parent"
305          app:layout_constraintVertical_bias="1.0" />
306
307 </android.support.constraint.ConstraintLayout>
308
309
310 package com.example.mohsin.vbs;
311
312 import android.content.Context;
313 import android.media.AudioManager;
314 import android.os.Build;
315 import android.os.Bundle;
316 import android.app.Activity;
317 import android.content.ContentResolver;
318 import android.content.Intent;
319 import android.database.Cursor;
320 import android.net.Uri;
321 import android.os.Bundle;
322 import android.speech.tts.TextToSpeech;
323 import android.support.annotation.RequiresApi;
324 import android.support.design.widget.FloatingActionButton;
```

```
325 import android.util.Log;
326 import android.view.View;
327 import android.widget.AdapterView;
328 import android.widget.AdapterView.OnItemClickListener;
329 import android.widget.ArrayAdapter;
330 import android.widget.ListView;
331 import android.widget.Toast;
332 import java.text.SimpleDateFormat;
333 import java.util.ArrayList;
334 import java.util.Date;
335 import java.util.Locale;
336
337 public class MsgActivity extends Activity implements OnItemClickListener
338 {
339    // contactActivity ca=new contactActivity();
340
341     TextToSpeech tts;
342     String smsMessageStr;
343     int result;
344     private static MsgActivity inst;
345     ArrayList<String> smsMessagesList = new ArrayList<String>();
346     ListView smsListView;
347     ArrayAdapter arrayAdapter;
348     FloatingActionButton  createMsg;
349
350     public static MsgActivity instance()
351     {
352         return inst;
353     }
354
355     @Override
356     public void onStart() {
357         super.onStart();
358         inst = this;
359     }
360
361     @Override
362     protected void onCreate(Bundle savedInstanceState)
363     {
364         super.onCreate(savedInstanceState);
365         createMsg=(FloatingActionButton)findViewById(R.id.idCreateMsg) ;
366         setContentView(R.layout.activity_msg);
367         smsListView = (ListView) findViewById(R.id.idList);
368         arrayAdapter = new ArrayAdapter<String>(this, android.R.layout.
              simple_list_item_1, smsMessagesList);
369         smsListView.setAdapter(arrayAdapter);
370         smsListView.setOnItemClickListener(this);
371         refreshSmsInbox();
372         tts=new TextToSpeech(getApplicationContext(), new TextToSpeech.
              OnInitListener()
373         {
374             @Override
375             public void onInit(int status)
376             {
377                 if(status !=TextToSpeech.ERROR)
378                 {
379                     tts.setLanguage(Locale.US);
380
381                 }
382
383             }
```

```
384         });
385
386     }
387
388     public void refreshSmsInbox()
389     {
390         ContentResolver contentResolver = getContentResolver();
391         Cursor smsInboxCursor = contentResolver.query(Uri.parse("content://sms/
                inbox"), null, null, null, null);
392         int indexBody = smsInboxCursor.getColumnIndex("body");
393         int indexAddress = smsInboxCursor.getColumnIndex("address");
394         long timeMillis = smsInboxCursor.getColumnIndex("date");
395         Date date = new Date(timeMillis);
396         SimpleDateFormat format = new SimpleDateFormat("dd/MM/yy");
397         String dateText = format.format(date);
398
399         if (indexBody < 0 || !smsInboxCursor.moveToFirst()) return;
400         arrayAdapter.clear();
401         do {
402             String str = smsInboxCursor.getString(indexAddress) +" at "+dateText
                    +
403                     "\n" + smsInboxCursor.getString(indexBody);
404             arrayAdapter.add(str);
405         } while (smsInboxCursor.moveToNext());
406     }
407
408     public void updateList(final String smsMessage)
409     {
410         arrayAdapter.insert(smsMessage, 0);
411         arrayAdapter.notifyDataSetChanged();
412     }
413
414
415     @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
416     public void onItemClick(AdapterView<?> parent, View view, int pos, long id)
417     {
418         try {
419
420             String[] smsMessages = smsMessagesList.get(pos).split("\n");
421             String address = smsMessages[0];
422             String smsMessage = "";
423             for (int i = 1; i < smsMessages.length; ++i) {
424                 smsMessage += smsMessages[i];
425             }
426
427              smsMessageStr = address + "\n";
428             smsMessageStr += smsMessage;
429
430
431             String toSpeech=smsMessageStr.toString();
432             Toast.makeText(MsgActivity.this, toSpeech, Toast.LENGTH_SHORT).show
                    ();
433             //Toast.makeText(this, smsMessageStr, Toast.LENGTH_SHORT).show();
434             tts.speak("message from"+toSpeech,TextToSpeech.QUEUE_FLUSH,null);
435
436
437         } catch (Exception e) {
438             e.printStackTrace();
439         }
440
441
```

```
    }



    @Override
    protected void onDestroy()
    {

        if(tts!=null)
        {
            tts.stop();
            tts.shutdown();

        }
        super.onDestroy();
    }


    public void sendClick(View view)
    {
       Intent intent=new Intent(MsgActivity.this,SendMsg.class);
       startActivity(intent);
    }
}


<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.mohsin.vbs.SendMsg">

    <EditText
        android:id="@+id/idNumber"
        android:layout_width="334dp"
        android:layout_height="57dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="#B2EBF2"
        android:cursorVisible="true"
        android:defaultFocusHighlightEnabled="true"
        android:ems="10"
        android:fadeScrollbars="false"
        android:hint="Enter Number"
        android:inputType="phone"
        android:textAlignment="center"
        android:textAllCaps="false"
        android:textColor="@android:color/black"
        android:textColorLink="@android:color/black"
        android:textSize="24sp"
        android:textStyle="normal|italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.175" />

    <EditText
        android:id="@+id/idMsgtext"
        android:layout_width="348dp"
        android:layout_height="142dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="#B2EBF2"
        android:bufferType="normal"
        android:cursorVisible="true"
        android:ems="10"
        android:fadeScrollbars="true"
        android:hint="Enter Message Here"
        android:inputType="textMultiLine"
        android:isScrollContainer="false"
        android:keyboardNavigationCluster="true"
        android:linksClickable="true"
        android:requiresFadingEdge="vertical"
        android:singleLine="true"
        android:textColor="@android:color/black"
        android:textColorLink="@android:color/black"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.098"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.98" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/send"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:clickable="true"
        android:onClick="sendBTnClick"
        app:backgroundTint="@android:color/background_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.974"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.958"
        app:srcCompat="@drawable/sm" />

    <ImageButton
        android:id="@+id/speakNum"
        android:layout_width="60dp"
        android:layout_height="56dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
```

```
563         android:background="#B2EBF2"
564         android:clickable="true"
565         android:hapticFeedbackEnabled="true"
566         android:onClick="micNumClick"
567         app:layout_constraintBottom_toBottomOf="parent"
568         app:layout_constraintEnd_toEndOf="parent"
569         app:layout_constraintHorizontal_bias="0.926"
570         app:layout_constraintStart_toStartOf="parent"
571         app:layout_constraintTop_toTopOf="parent"
572         app:layout_constraintVertical_bias="0.175"
573         app:srcCompat="@drawable/smallmic" />
574
575     <ImageButton
576         android:id="@+id/speakMsg"
577         android:layout_width="60dp"
578         android:layout_height="56dp"
579         android:layout_marginBottom="8dp"
580         android:layout_marginEnd="8dp"
581         android:layout_marginStart="8dp"
582         android:layout_marginTop="8dp"
583         android:background="#B2EBF2"
584         android:onClick="micMsgClick"
585         android:clickable="true"
586         android:hapticFeedbackEnabled="true"
587         app:layout_constraintBottom_toBottomOf="parent"
588         app:layout_constraintEnd_toEndOf="parent"
589         app:layout_constraintHorizontal_bias="0.971"
590         app:layout_constraintStart_toStartOf="parent"
591         app:layout_constraintTop_toTopOf="parent"
592         app:layout_constraintVertical_bias="0.822"
593         app:srcCompat="@drawable/smallmic" />
594
595 </android.support.constraint.ConstraintLayout>
596 package com.example.mohsin.vbs;
597
598 import android.Manifest;
599 import android.content.ActivityNotFoundException;
600 import android.content.Context;
601 import android.content.Intent;
602 import android.content.pm.PackageManager;
603 import android.speech.RecognizerIntent;
604 import android.support.design.widget.FloatingActionButton;
605 import android.support.v4.app.ActivityCompat;
606 import android.support.v4.content.ContextCompat;
607 import android.support.v7.app.AppCompatActivity;
608 import android.os.Bundle;
609 import android.telephony.SmsManager;
610 import android.text.TextUtils;
611 import android.util.Log;
612 import android.view.View;
613 import android.widget.EditText;
614 import android.widget.ImageButton;
615 import android.widget.Toast;
616
617
618 import java.util.ArrayList;
619 import java.util.Locale;
620
621 public class SendMsg extends AppCompatActivity
622 {
623     String text;
```

```
624     EditText number,msg;
625     FloatingActionButton Send;
626     ImageButton micNum,micMsg;
627     boolean Numclicked=false;
628     boolean Msgclicked=false;
629     private static final int MY_PERMISSION_REQUEST_SEND_SMS=0;
630     private final int REQ_CODE_SPEECH_INPUT=100;
631     Context context;
632     ArrayList<String> arrayName;
633     ArrayList<String> arrayNum;
634     @Override
635     protected void onCreate(Bundle savedInstanceState)
636     {
637         super.onCreate(savedInstanceState);
638         setContentView(R.layout.activity_send_msg);
639         Send = (FloatingActionButton)findViewById(R.id.send);
640         number=(EditText)findViewById(R.id.idNumber);
641         msg=(EditText)findViewById(R.id.idMsgtext);
642         micNum=(ImageButton)findViewById(R.id.speakNum);
643         micMsg=(ImageButton)findViewById(R.id.speakMsg);
644
645
646         Bundle bundle = getIntent().getExtras();
647         Bundle bundle2 = getIntent().getExtras();
648         try{
649         arrayNum = (ArrayList<String>) bundle.getStringArrayList("numbers");
650         arrayName = (ArrayList<String>) bundle2.getStringArrayList("names");
651         }catch (NullPointerException e){
652
653         }
654 //
655 //          if(getIntent().getStringArrayListExtra("numbers") != null)
656 //          {
657 //
658 //              //number.setText(getIntent().getStringExtra("name"));
659 //              Toast.makeText(this, getIntent().getStringArrayListExtra("numbers
    ").toString(), Toast.LENGTH_SHORT).show();
660 //              Log.d("btn click","numbers"+getIntent().getStringArrayListExtra("
    numbers"));
661 //          }
662         if(arrayName!=null)
663         {
664             for(int i=0; i < arrayName.size(); i++){
665
666                 number.setText(number.getText() + arrayName.get(i) + " , ");
667             }
668
669
670         }
671
672
673         micMsg.setOnClickListener(new View.OnClickListener()
674         {
675             public void onClick(View view)
676             {
677                 Msgclicked=true;
678                 pspchInput();
679
680             }
681
682         });
```

```
683         micNum.setOnClickListener(new View.OnClickListener()
684         {
685             public void onClick(View view)
686             {
687                 Numclicked=true;
688                 pspchInput();
689
690             }
691
692         });
693
694     }
695
696     public void sendBTnClick(View view)
697     {
698         int permissionCheck= ContextCompat.checkSelfPermission(this, Manifest.
                permission.SEND_SMS);
699         if(permissionCheck== PackageManager.PERMISSION_GRANTED)
700         {
701             MyMessage();
702         }
703         else
704         {
705             ActivityCompat.requestPermissions(this,new String[]{Manifest.
                    permission.SEND_SMS},MY_PERMISSION_REQUEST_SEND_SMS);
706         }
707
708     }
709
710     private void MyMessage()
711     {
712         String myNumber =   number.getText().toString().trim();
713         String myMsg=msg.getText().toString().trim();
714
715         if(myNumber == null || myNumber.equals("")|| myMsg == null || myMsg.
                equals(""))
716         {
717             Toast.makeText(this, "Filed Can't be Empty", Toast.LENGTH_SHORT).
                    show();
718         }
719         else
720         {
721             if(arrayNum!=null)
722             {
723                 //Toast.makeText(this,getIntent().getStringExtra("names"), Toast.
                        LENGTH_SHORT).show();
724                 for(int i=0; i < arrayNum.size(); i++)
725                 {
726                     myNumber = arrayNum.get(i);
727                     SmsManager smsManager = SmsManager.getDefault();
728                     smsManager.sendTextMessage(myNumber, null, myMsg, null, null)
                            ;
729                     Toast.makeText(this, "Message Sent ", Toast.LENGTH_SHORT).
                            show();
730                 }
731
732             }
733             else {
734                 myNumber = number.getText().toString().trim();
735                 // if(TextUtils.isDigitsOnly(myNumber))
736                 //{
```

```
737                 SmsManager smsManager = SmsManager.getDefault();
738                 smsManager.sendTextMessage(myNumber, null, myMsg, null, null);
739                 Toast.makeText(this, "Message Sent", Toast.LENGTH_SHORT).show();
740             }
741
742
743             //}
744 //              else
745 //              {
746 //                  Toast.makeText(this, "Enter Integers Only", Toast.LENGTH_SHORT
    ).show();
747 //              }
748         }
749     }
750
751     public void onRequestPermissionResult(int requestCode,String[] permissions,
        int[] grantResults)
752     {
753         super.onRequestPermissionsResult(requestCode,permissions,grantResults);
754         switch (requestCode)
755         {
756             case MY_PERMISSION_REQUEST_SEND_SMS:
757             {
758                 if(grantResults.length>=0 && grantResults[0]==PackageManager.
                    PERMISSION_GRANTED)
759                 {
760                     MyMessage();
761                 }
762                 else
763                 {
764                     Toast.makeText(this, "You Dont Have Required Permissions",
                        Toast.LENGTH_SHORT).show();
765                 }
766             }
767         }
768     }
769
770
771
772     public void pspchInput()
773     {
774
775         Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
776         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
777                 RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
778         intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
779         intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak The Message");
780         try {
781             startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
782
783             // Intent inte = new Intent(this,SendMsg.class);
784             //startActivity(inte);
785         } catch (ActivityNotFoundException a) {
786
787             Toast.makeText(getApplicationContext(),
788                     "Sorry! Speech recognition is not supported in this device."
                        ,
789                     Toast.LENGTH_SHORT).show();
790
791         }
792     }
```

```
793
794
795
796     protected void onActivityResult(int requestCode, int resultCode, Intent data
            )
797     {
798         super.onActivityResult(requestCode, resultCode, data);
799         switch (requestCode)
800         {
801             case REQ_CODE_SPEECH_INPUT:
802             {
803                 if (resultCode == RESULT_OK && null != data)
804                 {
805
806                     ArrayList<String> result = data.getStringArrayListExtra(
                            RecognizerIntent.EXTRA_RESULTS);
807                     text=result.get(0);
808
809                     //Intent inte =new Intent(this,SendMsg.class);
810                     //startActivity(inte);
811                     //sm.msg.setText(text);
812                     Toast.makeText(this, text, Toast.LENGTH_SHORT).show();
813
814                     if(Numclicked==true)
815                     {
816 //                          Integer myNum = Integer.parseInt(text);
817 //                         number.setText(myNum);
818                         number.setText(text);
819                         Numclicked=false;
820
821                     }
822
823                     else if(Msgclicked==true)
824                     {
825                         msg.setText(text);
826                     }
827
828
829                 }
830                 break;
831             }
832
833         }
834     }
835
836
837
838 }
839
840
841 package com.example.mohsin.vbs;
842
843
844 import android.content.BroadcastReceiver;
845 import android.content.Context;
846 import android.content.Intent;
847 import android.os.Bundle;
848 import android.telephony.SmsMessage;
849 import android.widget.Toast;
850
851 import java.text.SimpleDateFormat;
```

```
852 import java.util.Date;
853
854 public class SmsBroadcastReceiver extends BroadcastReceiver
855 {
856
857     public static final String SMS_BUNDLE = "pdus";
858
859     public void onReceive(Context context, Intent intent)
860     {
861         Bundle intentExtras = intent.getExtras();
862         if (intentExtras != null)
863         {
864             Object[] sms = (Object[]) intentExtras.get(SMS_BUNDLE);
865             String smsMessageStr = "";
866             for (int i = 0; i < sms.length; ++i)
867             {
868                 SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) sms[i
                        ]);
869
870                 String smsBody = smsMessage.getMessageBody().toString();
871                 String address = smsMessage.getOriginatingAddress();
872                 long timeMillis = smsMessage.getTimestampMillis();
873
874                 Date date = new Date(timeMillis);
875                 SimpleDateFormat format = new SimpleDateFormat("dd/MM/yy");
876                 String dateText = format.format(date);
877
878                 smsMessageStr += address +" at "+"\t"+ dateText + "\n";
879                 smsMessageStr += smsBody + "\n";
880
881             }
882             Toast.makeText(context, smsMessageStr, Toast.LENGTH_SHORT).show();
883
884             //this will update the UI with message
885             MsgActivity inst = MsgActivity.instance();
886             inst.updateList(smsMessageStr);
887         }
888     }
889 }
890
891
892
893
894 }
```

6.2 Contact

Contact module is developed for sending a message to a required receiver.Contact module is used for sending a message without remembering a number. It has a multiple sub modules such as:contact list and multiple contact selection.Contact has a contact number and a users name.

6.2.1 Contact list

Contact list is created for having a number of multiple users on a single screen.This contact list consist of user name and number.When user want to send a message,he/she will select a number direct from a contact list.This contact list will display in a ascending order,which is based on users name.

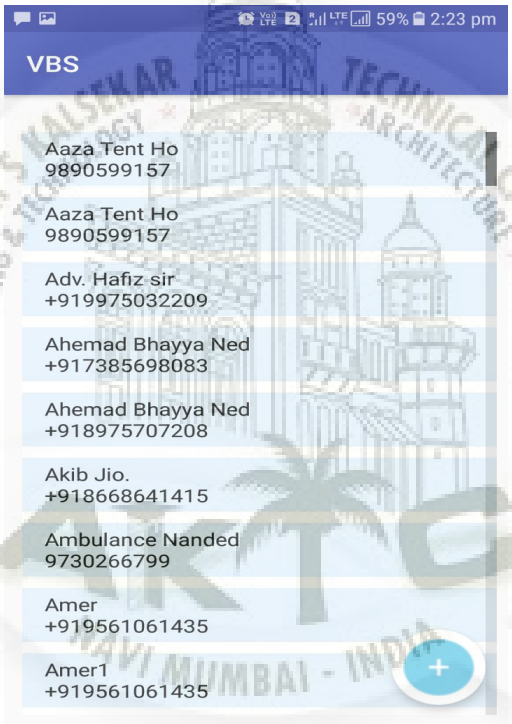


Figure 6.3: Contact List

6.2.2 Multiple Contact

Multiple contact is available for sending message to multiple users.When user want to send a same message to more than one user ,user will select more than one contact from list at a time.User just has to open a contact list and at a time numbers of receiver can be selected,which is like a broadcasting of message.

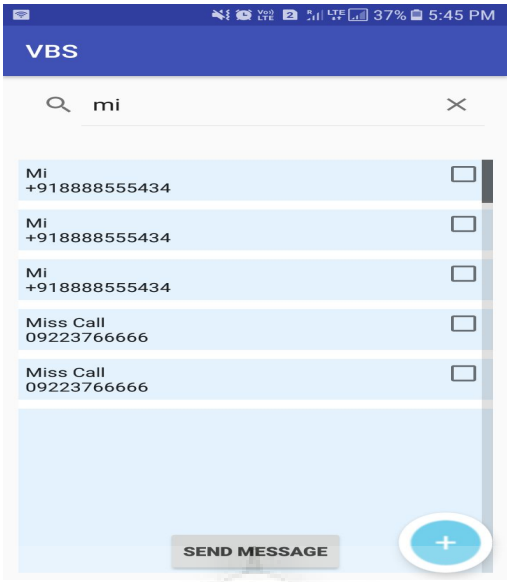


Figure 6.4: Multiple Contact

6.2.3 Add Contact

When user wants to add a number of new user,user can add it from add contact page.This page has two fields which are user name and contact information.When user fill these fields they can add new contact in a contact list.

Figure 6.5: Add Contact

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.mohsin.vbs.contactActivity">
```

```
8
9
10      <SearchView
11          android:id="@+id/idsearch"
12          android:layout_width="0dp"
13          android:layout_height="48dp"
14          android:layout_marginBottom="428dp"
15          android:layout_marginEnd="8dp"
16          android:layout_marginLeft="8dp"
17          android:layout_marginRight="8dp"
18          android:layout_marginStart="8dp"
19          android:layout_marginTop="8dp"
20          android:gravity="top"
21          android:hapticFeedbackEnabled="true"
22          android:iconifiedByDefault="false"
23
24          android:imeOptions="actionSearch"
25          app:layout_constraintBottom_toBottomOf="parent"
26          app:layout_constraintEnd_toEndOf="parent"
27
28          app:layout_constraintHorizontal_bias="0.38"
29          app:layout_constraintStart_toStartOf="parent"
30          app:layout_constraintTop_toTopOf="parent"
31          app:layout_constraintVertical_bias="0.0" />
32          <requestFocus/>
33
34
35
36
37      <android.support.design.widget.FloatingActionButton
38          android:id="@+id/idConAdd"
39          android:layout_width="80dp"
40          android:layout_height="67dp"
41          android:layout_marginBottom="8dp"
42          android:layout_marginEnd="8dp"
43          android:layout_marginStart="8dp"
44          android:layout_marginTop="8dp"
45          android:clickable="true"
46          android:foregroundGravity="bottom"
47          android:onClick="conAdClick"
48          android:scrollbarAlwaysDrawHorizontalTrack="false"
49          android:scrollbarAlwaysDrawVerticalTrack="true"
50          app:backgroundTint="@android:color/background_light"
51          app:fabSize="mini"
52          app:layout_constraintBottom_toBottomOf="parent"
53          app:layout_constraintEnd_toEndOf="parent"
54          app:layout_constraintHorizontal_bias="0.992"
55          app:layout_constraintStart_toStartOf="parent"
56          app:layout_constraintTop_toTopOf="parent"
57          app:layout_constraintVertical_bias="1.0"
58          app:srcCompat="@drawable/add" />
59
60      <Button
61          android:id="@+id/idsendBtn"
62          android:layout_width="wrap_content"
63          android:layout_height="48dp"
64          android:layout_marginBottom="8dp"
65          android:layout_marginEnd="8dp"
66          android:layout_marginLeft="8dp"
67          android:layout_marginRight="8dp"
68          android:layout_marginStart="8dp"
```

```
        android:layout_marginTop="8dp"
        android:gravity="right|fill_vertical"
        android:onClick="sendMessage"
        android:text="Send  Message"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/idConList"
        app:layout_constraintVertical_bias="1.0" />


    <ListView
        android:id="@+id/idConList"
        android:layout_width="339dp"
        android:layout_height="459dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="#E3F2FD"
        android:divider="#FFFFFF"
        android:dividerHeight="10sp"
        android:drawableLeft="?android:attr/listChoiceIndicatorMultiple"
        android:fadingEdge="vertical"
        android:fadingEdgeLength="5sp"
        android:fastScrollAlwaysVisible="true"
        android:fitsSystemWindows="true"
        android:foregroundGravity="top|bottom"
        android:headerDividersEnabled="false"
        android:scrollbars="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.6"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />
</android.support.constraint.ConstraintLayout>



```

```
package com.example.mohsin.vbs;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.provider.ContactsContract;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
```

```
130 import android.widget.Button;
131 import android.widget.ListView;
132 import android.widget.SearchView;
133 import android.widget.TextView;
134 import android.widget.Toast;
135
136 import java.util.ArrayList;
137
138 public class contactActivity extends AppCompatActivity implements AdapterView.
    OnItemClickListener,SearchView.OnQueryTextListener
139 {
140     ListView listV;
141     ArrayAdapter<String> adapter;
142     FloatingActionButton conButn;
143     Cursor c;
144     SearchView sv;
145     ArrayList<String> contacts;
146     ArrayList<String> selectedItemNums;
147     ArrayList<String>selectedItemNames;
148     private Context context;
149
150
151     private static final int PERMISSION_REQUEST_READ_CONTACTS = 100;
152
153     @Override
154     protected void onCreate(Bundle savedInstanceState)
155     {
156         super.onCreate(savedInstanceState);
157         setContentView(R.layout.activity_contact);
158
159 //          LayoutInflater inflater = (LayoutInflater) context//
160 //                  .getSystemService(Context.LAYOUT_INFLATER_SERVICE);//
161 //
162 //          View gridView;//
163 //          gridView = inflater.inflate(R.layout.activity_contact, null);//
164         selectedItemNames=new ArrayList<String>();
165         selectedItemNums=new ArrayList<String>();
166
167
168         sv=(SearchView)findViewById(R.id.idsearch);
169
170         listV=(ListView) findViewById(R.id.idConList);
171         listV.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
172         Button bs=(Button)findViewById(R.id.idsendBtn);
173         conButn=(FloatingActionButton)findViewById(R.id.idConAdd);
174         int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.
            permission.READ_CONTACTS);
175
176         if(permissionCheck == PackageManager.PERMISSION_GRANTED)
177         {
178             showContacts();
179
180         }
181         else
182         {
183             ActivityCompat.requestPermissions(this,new String[]{Manifest.
                permission.READ_CONTACTS},PERMISSION_REQUEST_READ_CONTACTS);
184
185         }
186
187         adapter=new ArrayAdapter<String>(this,R.layout.rowlayout,R.id.txt_lan,
```

```
            contacts);
188         listV.setAdapter(adapter);
189         listV.setOnItemClickListener(this);
190         sv.setOnQueryTextListener(this);
191
192
193     }
194
195     public boolean onQueryTextSubmit(String query)
196     {
197         sv.clearFocus();
198         return false;
199     }
200     public boolean onQueryTextChange(String newText)
201     {
202         String text=newText;
203         adapter.getFilter().filter(newText);
204         Toast.makeText(this, "query is"+text, Toast.LENGTH_SHORT).show();
205         return false;
206     }
207
208     public void showContacts()
209     {
210
211         c=getContentResolver().query(ContactsContract.CommonDataKinds.Phone.
                CONTENT_URI, null, null, null, ContactsContract.Contacts.DISPLAY_NAME);
212         contacts=new ArrayList<String>();
213         while(c.moveToNext())
214         {
215             String contactName = c.getString(c.getColumnIndexOrThrow(
                    ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
216             String phnNum = c.getString((c.getColumnIndexOrThrow(
                    ContactsContract.CommonDataKinds.Phone.NUMBER)));
217             contacts.add(contactName + "\n" + phnNum);
218         }
219
220         c.close();
221     }
222
223
224     public void conAdClick(View view)
225     {
226         Toast.makeText(this, "button Clicked ", Toast.LENGTH_SHORT).show();
227         Intent intent=new Intent(contactActivity.this,addContacts.class);
228         startActivity(intent);
229     }
230
231     @Override
232     public void onItemClick(AdapterView<?> parent, View view, int position, long
            id)
233     {
234         String[] cont=contacts.get(position).split("\n");
235         //Intent in=new Intent(this,SendMsg.class);
236        // Log.d("list_click","number "+cont[1]);
237         //Toast.makeText(this,cont[1], Toast.LENGTH_SHORT).show();
238
239
240         String selectedName = cont[0];
241         String selectedContact = cont[1];
242         if (selectedItemNums.contains(selectedContact) && selectedItemNames.
                contains(selectedName)) {
```

```
            selectedItemNums.remove(selectedContact);
            selectedItemNames.remove(selectedName);
        } else {
            selectedItemNums.add(selectedContact);
            selectedItemNames.add(selectedName);
        }


//        String items="";
//
//        for(String item:selectedItemNums)
//        {
//            items+="-"+item+"\n";
//
//        }
      //  Log.d("list_click","name"+selectedName+ "\nlist "+selectedItemNums);

       //Toast.makeText(this, "u have selected \n"+items, Toast.LENGTH_SHORT).
           show();
            // in.putExtra("number",cont[1]);
             //in.putExtra("name",cont[0]);
//                startActivity(in);
    }

    public void sendMessage(View view)
    {
        Intent in=new Intent(this,SendMsg.class);
        in.putExtra("numbers",selectedItemNums);
        in.putExtra("names",selectedItemNames);
        startActivity(in);
    }
}
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
    android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".addContacts">

    <EditText
        android:id="@+id/conName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="Enter Name"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.28" />

    <EditText
```

```
302         android:id="@+id/conNum"
303         android:layout_width="wrap_content"
304         android:layout_height="wrap_content"
305         android:layout_marginBottom="8dp"
306         android:layout_marginEnd="8dp"
307         android:layout_marginLeft="8dp"
308         android:layout_marginRight="8dp"
309         android:layout_marginStart="8dp"
310         android:layout_marginTop="8dp"
311         android:ems="10"
312         android:hint="Enter Number"
313         android:inputType="number"
314         app:layout_constraintBottom_toBottomOf="parent"
315         app:layout_constraintEnd_toEndOf="parent"
316         app:layout_constraintStart_toStartOf="parent"
317         app:layout_constraintTop_toTopOf="parent" />
318
319     <Button
320         android:id="@+id/button"
321         android:layout_width="wrap_content"
322         android:layout_height="wrap_content"
323         android:layout_marginBottom="8dp"
324         android:layout_marginEnd="8dp"
325         android:layout_marginLeft="8dp"
326         android:layout_marginRight="8dp"
327         android:layout_marginStart="8dp"
328         android:layout_marginTop="8dp"
329         android:text="Add Contact"
330         android:onClick="addCon"
331         app:layout_constraintBottom_toBottomOf="parent"
332         app:layout_constraintEnd_toEndOf="parent"
333         app:layout_constraintStart_toStartOf="parent"
334         app:layout_constraintTop_toTopOf="parent"
335         app:layout_constraintVertical_bias="0.684" />
336
337 </android.support.constraint.ConstraintLayout>
338
339
340
341 package com.example.mohsin.vbs;
342
343 import android.content.ContentProviderOperation;
344 import android.content.ContentProviderResult;
345 import android.content.Intent;
346 import android.content.OperationApplicationException;
347 import android.net.Uri;
348 import android.os.RemoteException;
349 import android.provider.ContactsContract;
350 import android.provider.ContactsContract.*;
351 import android.support.v7.app.AppCompatActivity;
352 import android.os.Bundle;
353 import android.util.Log;
354 import android.view.View;
355 import android.widget.Button;
356 import android.widget.EditText;
357 import android.widget.Toast;
358
359 import java.util.ArrayList;
360
361
362 public class addContacts extends AppCompatActivity
```

```
{
    EditText number;
    EditText name;
    EditText nam;
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_contacts);
        name=(EditText)findViewById(R.id.conName);
        number=(EditText)findViewById(R.id.conNum);
        btn=(Button)findViewById(R.id.button);
        //nam=(EditText)findViewById(R.id.editText3);

    }

    public void addCon(View view)
    {
        Add_RawContact();
        Toast.makeText(this, "Contact Added", Toast.LENGTH_SHORT).show();
        Intent intent=new Intent(this,contactActivity.class);
        startActivity(intent);
    }
    public void Add_RawContact()
    {
        String Nem = String.valueOf(name.getText());
        String num =String.valueOf(number.getText());


        ArrayList<ContentProviderOperation> ops = new ArrayList<
            ContentProviderOperation>();
        int rawContact_NewID= ops.size();

        try {
            ops.add(ContentProviderOperation.newInsert(RawContacts.CONTENT_URI)
                    .withValue(RawContacts.ACCOUNT_TYPE, null)
                    .withValue(RawContacts.ACCOUNT_NAME, null)
                    .build());
        }
        catch(Exception e)
        {
            Log.e("Add","could not find account_type null");
            return;
        }




        ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.
            CONTENT_URI)
                .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID,
                    rawContact_NewID)
                .withValue(ContactsContract.Data.MIMETYPE, ContactsContract.
                    CommonDataKinds.StructuredName.CONTENT_ITEM_TYPE)
                .withValue(ContactsContract.CommonDataKinds.StructuredName.
                    DISPLAY_NAME,Nem)
                .withValue(ContactsContract.CommonDataKinds.StructuredName.
                    FAMILY_NAME, null)
                .withValue(ContactsContract.CommonDataKinds.StructuredName.
```

```
                    GIVEN_NAME, null)
418                 .build());
419
420
421         ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.
                CONTENT_URI)
422                 .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID,
                        rawContact_NewID)
423                 .withValue(ContactsContract.Data.MIMETYPE, ContactsContract.
                        CommonDataKinds.Phone.CONTENT_ITEM_TYPE)
424                 .withValue(ContactsContract.CommonDataKinds.Phone.NUMBER, num)
425                 .withValue(ContactsContract.CommonDataKinds.Phone.TYPE,
                        CommonDataKinds.Phone.TYPE_MOBILE )
426                 .build());
427
428         ContentProviderResult[] res = new ContentProviderResult[0];
429         try {
430             res = getContentResolver().applyBatch(ContactsContract.AUTHORITY,
                    ops);
431         } catch (RemoteException e) {
432
433             Log.e("getContentResolver()",e.getMessage());
434         } catch (OperationApplicationException e) {
435
436             Log.e("getContentResolver()",e.toString());
437         } catch (Exception e) {
438
439             Log.e("getContentResolver()",e.toString());
440         }
441
442
443         if (res != null && res[0] != null) {
444
445             Uri newContactUri = res[0].uri;
446             Log.d("AddContact", "URI added contact:" + newContactUri);
447
448         }
449         else {
450
451         }
452
453     }
454 }
455
456
457
458 <?xml version="1.0" encoding="utf-8"?>
459
460 <CheckedTextView   xmlns:android="http://schemas.android.com/apk/res/android"
461         android:id="@+id/txt_lan"
462         android:layout_width="match_parent"
463         android:layout_height="wrap_content"
464         android:gravity="left"
465         android:checkMark="?android:attr/listChoiceIndicatorMultiple"
466         android:padding="5dp"
467
468         />
469
470
471         <?xml version="1.0" encoding="utf-8"?>
472 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    package="com.example.mohsin.vbs">

    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MsgActivity" />
        <activity android:name=".SendMsg" />

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />

        <receiver
            android:name=".SmsBroadcastReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter android:priority="999">
                <action android:name="android.provider.Telephony.SMS_RECEIVED"
                    />
            </intent-filter>
        </receiver>

        <activity android:name=".contactActivity"
            android:windowSoftInputMode="adjustPan" />
        <activity android:name=".addContacts"></activity>
    </application>

</manifest>
```

Chapter 7

System Testing

The System is tested in every environment,we have done unit testing and make them intergrated together for intergration testing.We have done Beta testing by the person apart from our group.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Testing contact Information	Is it displaying whole information?	Display only contact name	Display contact number with name
T02	Text-to-speech	User clicks on message ,Speech has to perform	Message display on another text field	Perform speech

7.2 Sample of a Test Case

Title: Contact information – Display contact information.
Description: When user selects a contact from list ,it has to show a contact name and number on message page.
Precondition: Open to a contact module and select contact.
Assumption: Our system must be installed in the user mobile phone .

Test Steps:

1. Go to menu
2. Select a contact
3. Select any contact in contact list
4. Now it will display a contact information on message list

Expected Result:A page displaying the contact of a user with it's name for making it more convenient

Actual Result:

When user select a contacts from a contact list ,application shifts to another page where it displays a contact of a recipient.expected Contact is to display a contact of a recipient with name and number but it displays a contact information with only name.

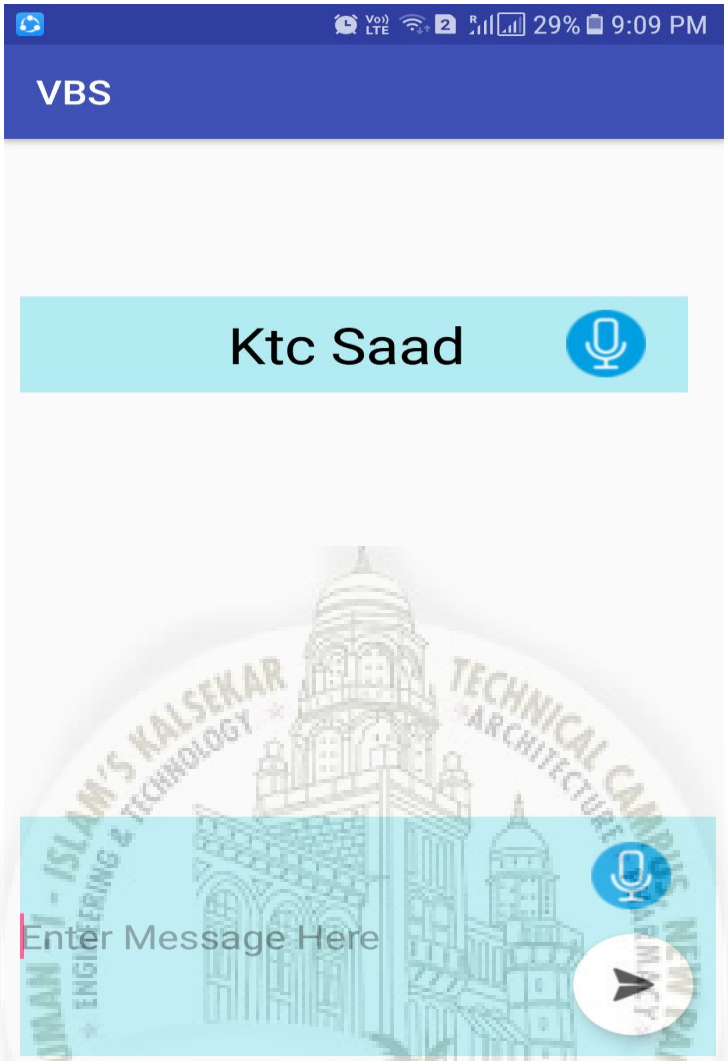


Figure 7.1: Add Contact Scrteen shot

7.3 Sample of a Test Case

Title: Text-to-speech –To perform speech-to-text conversion.
Description: When user clicks on a message,to has to perform text-to-speech conversion.
Precondition: Select a message module and rclick on message.
Assumption: Our system must be installed in the user mobile phone .

Test Steps:

1. Go to menu
2. Select a message
3. Select any message to read.
4. Now it will convert it into speech.

Expected Result: When user select a message module,it performs text-to-speech conversion.

Actual Result:

User select a message module,it display ba list of messages.when usre click to convert these messages into speech,it only show a same message on screen rather than converting it into speech.

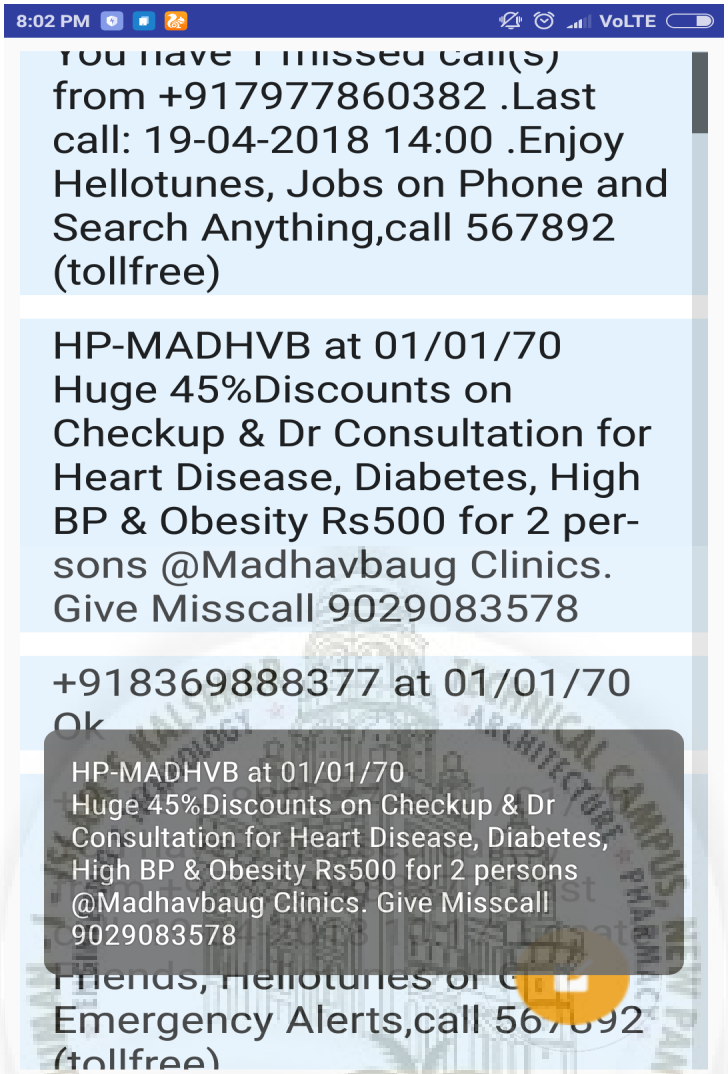


Figure 7.2: Add Contact Scrteen shot

7.3.1 Software Quality Attributes

1. AVAILABILITY: The system should not be down,whenever the user use the system the specific data should be available to the user.
2. CORRECTNESS: As per the user search the correct data should be shown to the user like at time for searching the near by place the system should show only the places around the user.
3. MAINTAINABILITY: The administrators of the system will maintain the system with effective updates though on air update if needed.

4. EXTENSIBILITY: The system is capable to be modified by changing some modules or by adding some features to the existing system

Chapter 8

Screenshots of Project

8.1 Home Screen

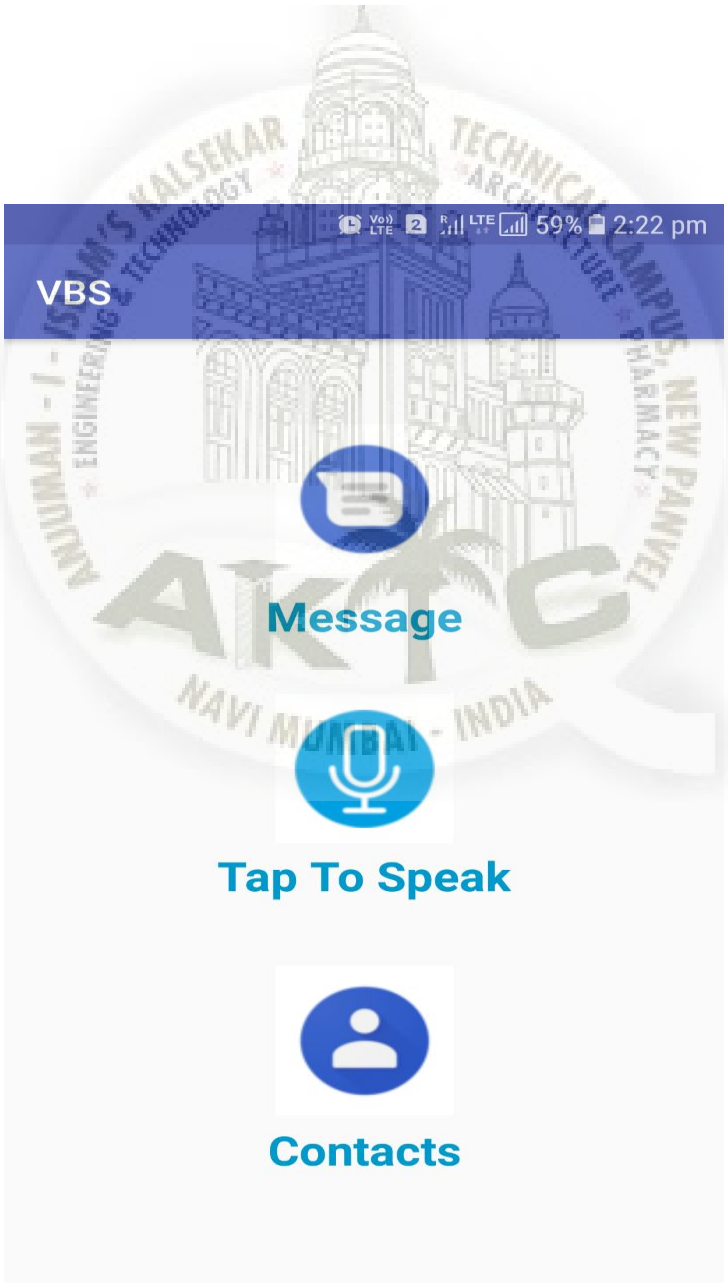


Figure 8.1: Home Screen Screen shot

8.2 Module selection

Figure 8.2: Module Selection Screen shot

8.3 Messages

Figure 8.3: Messages Screen shot

8.4 Send message

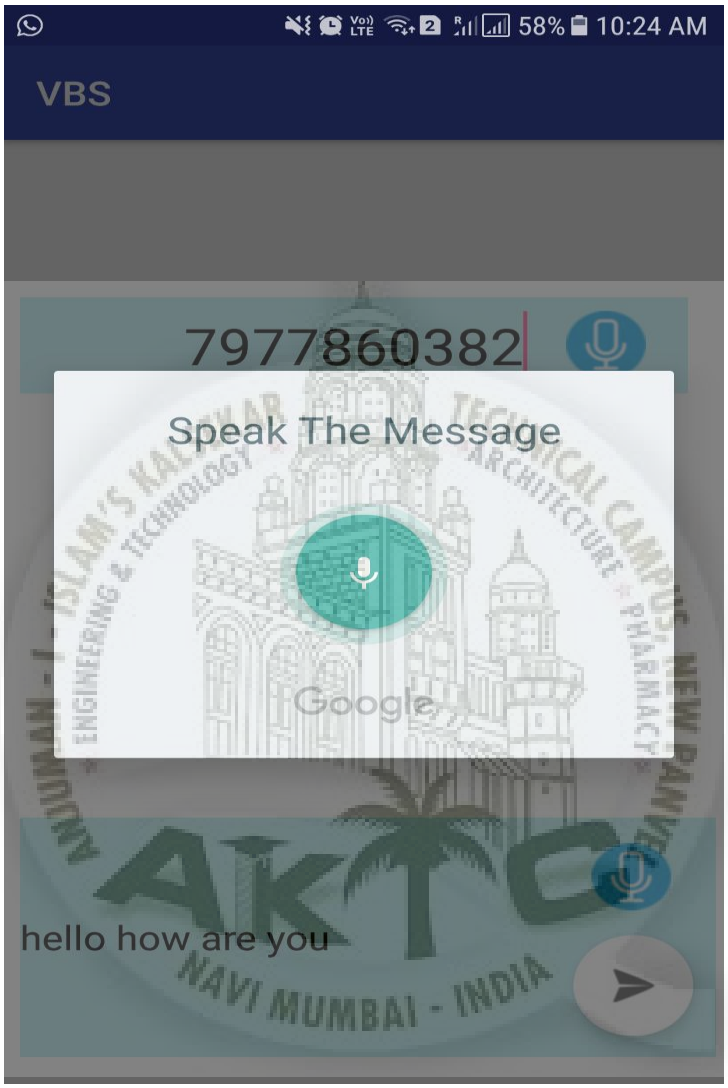


Figure 8.4: Send Message Screen shot

8.5 Contact list

Figure 8.5: Contact list Screen shot

8.6 Multiple Contact

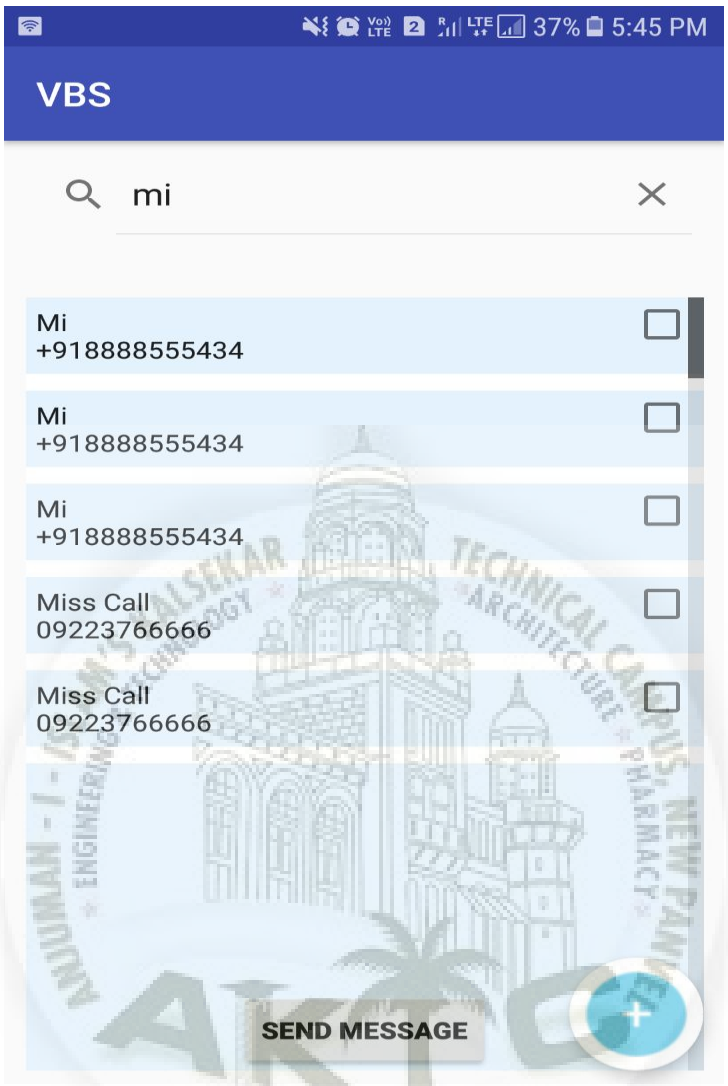


Figure 8.6: Multiple Contact Screen shot

8.7 Add Contact

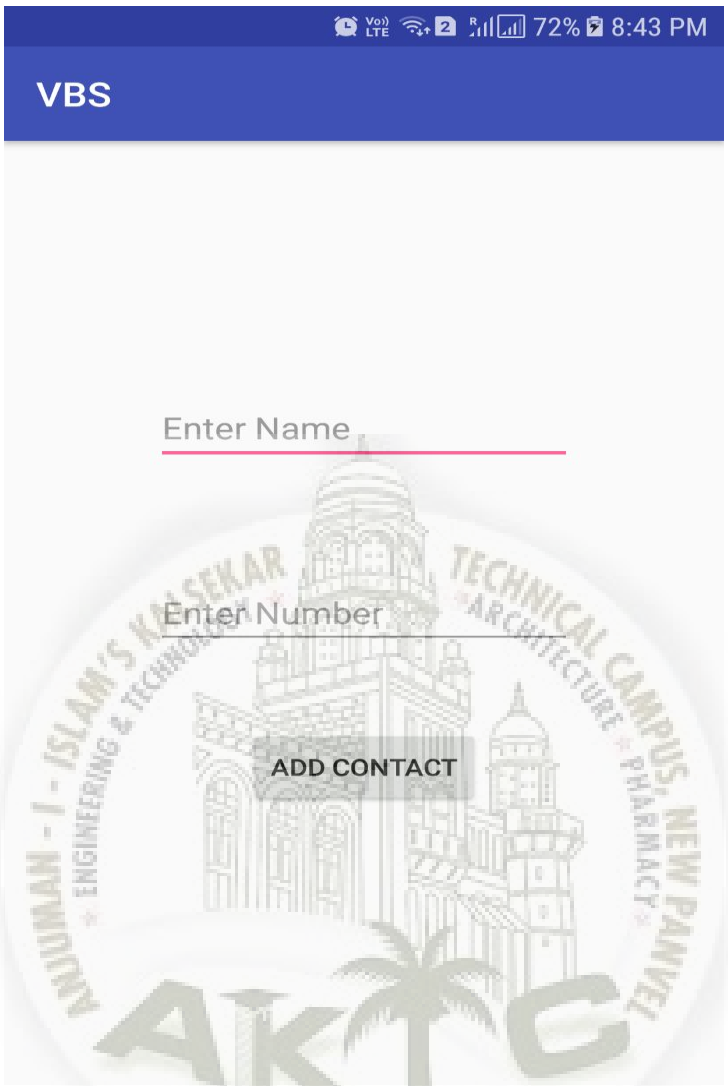


Figure 8.7: Add Contact Scrteen shot

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The speech to text conversion may seem effective and efficient to its users if it produces natural speech and by making several modifications to it. This system is useful for deaf and dumb people to Interact with the other peoples from society. Speech to Text synthesis is a critical research and application area in the field of multimedia interfaces. In this project we gathers important references to literature related to the endogenous variations of the speech signal and their importance in automatic speech recognition. A database has been created from the various domain words and syllables. The desired speech is produced by the Concatenating speech synthesis approach. Speech synthesis is advantageous for people who are visually handicapped. This project made a clear and simple overview of working of speech to text system (STT) in step by step process. The system gives the input data from mice in the form of voice, then processed that data converted into text format displayed on mobile. The user types the input string and the system reads it from the database or data store where the words, phones, diaphones, triphone are stored. In this project, we presented the development of existing STT system by adding spellchecker module to it for different language. There are many speeches to text systems (STT) available in the market and also much improvisation is going on in the research area to make the speech more effective.

9.2 Future Scope

An application of speech synthesis have been developed for android platform. The developed application is reliable and user friendly and performed an impressive communication. This system can be a problem solution for the people and specially targeting to those with audible problems as it would help them to listen to their problems that they faced regarding their mobile operators. The application work has been done for English language. In future, this work can also be done for rest of the regional languages like Gujarati, Tamil, Telugu, etc .

- This application can be implemented for multiple languages selection.
- Scanner can also be added as a feature which can scan a written text paper and can convert it into a message form.
- Suggestions can also be added for both i.e message and contact.Based on a words type on screen it can predict a next word.It can also predict a contact number based on frequent use.

References

[1] *Hands Free Speech based SMS System on Android*; Gulbakshee Dharmale,Dr. Vilas Thakare , Dr. Dipti D. Patil,International Conference on Advances in Human Machine Interaction, March 03-05-2016.

[2] *Android text messaging application for visually impaired people*; Siddhesh R.Baravkar,Mohit R.Borde, Mahendra K. Nivangune,IRACST – Engineering Science and Technology, February 2013

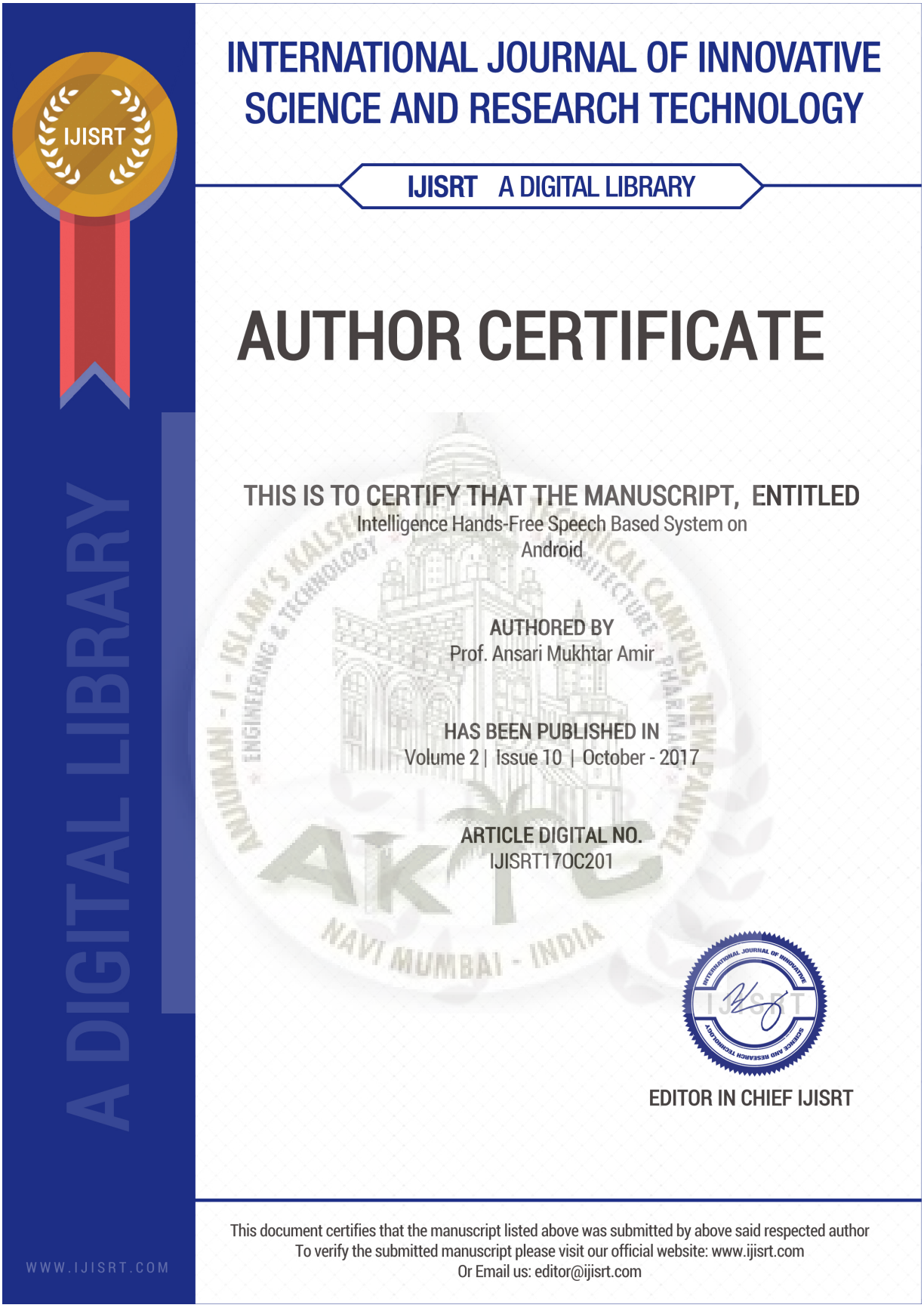
[3] *Smart Voice Assistant: a universal voice control solution for non-visual access to the Android operating system*;Aditi Bhalerao,Samira Bhilare, Anagha Bondade, Monal Shingade ,IRJET International Research Journal of Engineering and Technology, Jan -2017

[4] *A REVIEW ON SPEECH TO TEXT CONVERSION METHODS*;Miss.Prachi Khilari,Prof. Bhope V. P,International Journal of Advanced Research in Computer Engineering Technology (IJARCET).7, July 2015

[5] *www.scribd.com*;Shaikh Shaheda Faiyaz, Khan Arshiya Rakeeb ,Inamdar Mohsin Harun,`https://www.scribd.com`//Intelligence-Hands-Free-Speech-Based-System-on-Android, Volume 2, Issue 10, October–2017

Achievements

1. Publications
   (a) *INTELLIGENCE HANDS-FREE SPEECH BASED SMS SYSTEM ON ANDROID*; INMADAR MOHSIN HARUN,KHAN ARSHIYA BANO,SHAIKH SHAHEDA FAIYAZ,ijisrt,October 2017 (http://www.ijisrt.com)
   (b) *INTELLIGENCE HANDS-FREE SPEECH BASED SMS SYSTEM ON ANDROID*; INMADAR MOHSIN HARUN,KHAN ARSHIYA BANO,SHAIKH SHAHEDA FAIYAZ,ijisrt, april 2018(http://www.ijisrt.com)



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands-Free Speech Based System on Android

AUTHORED BY

Prof. Ansari Mukhtar Amir

HAS BEEN PUBLISHED IN

Volume 2 | Issue 10 | October - 2017

ARTICLE DIGITAL NO.

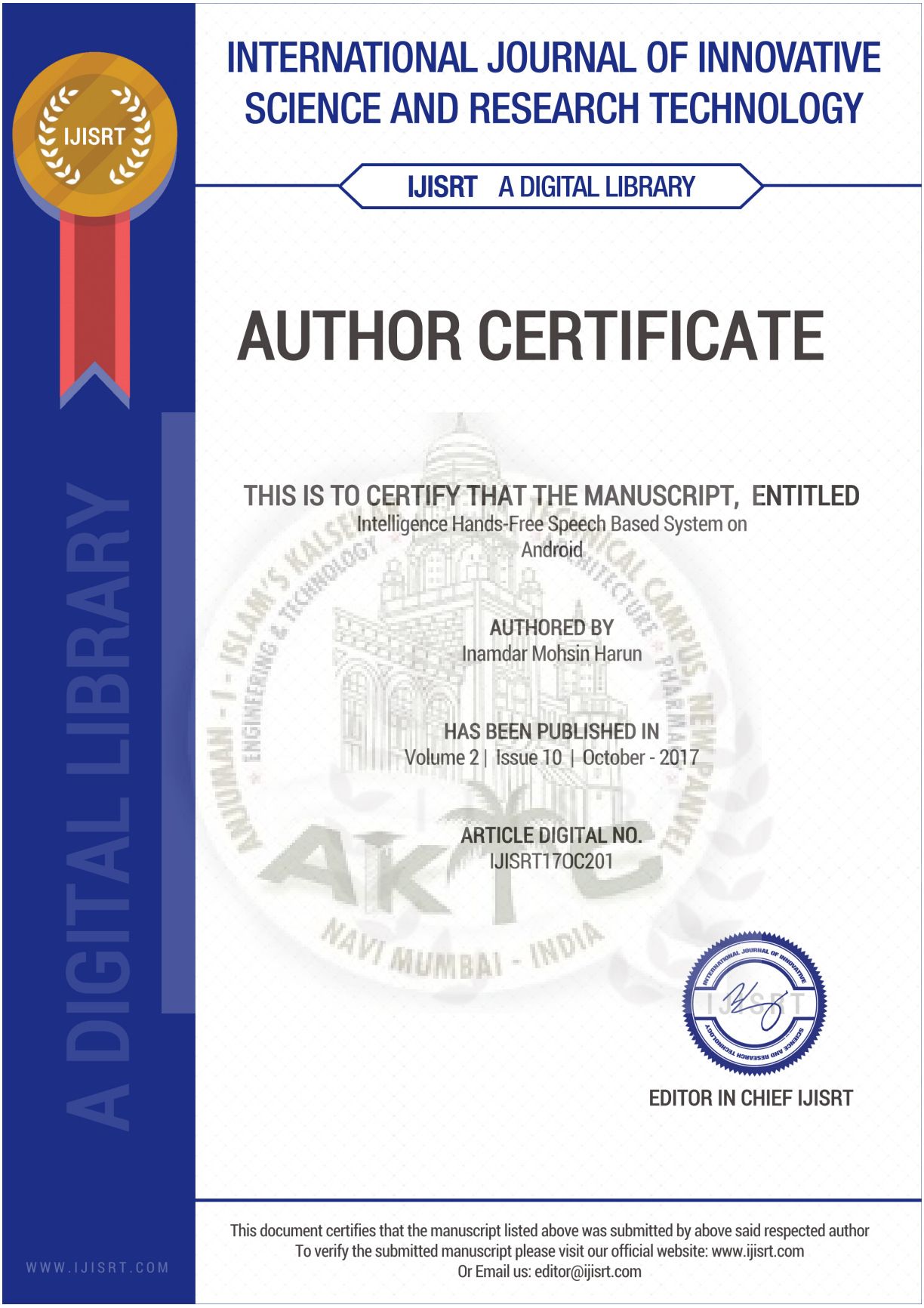
IJISRT17OC201

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

A DIGITAL LIBRARY

WWW.IJISRT.COM



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands-Free Speech Based System on Android

AUTHORED BY

Inamdar Mohsin Harun

HAS BEEN PUBLISHED IN

Volume 2 | Issue 10 | October - 2017

ARTICLE DIGITAL NO.

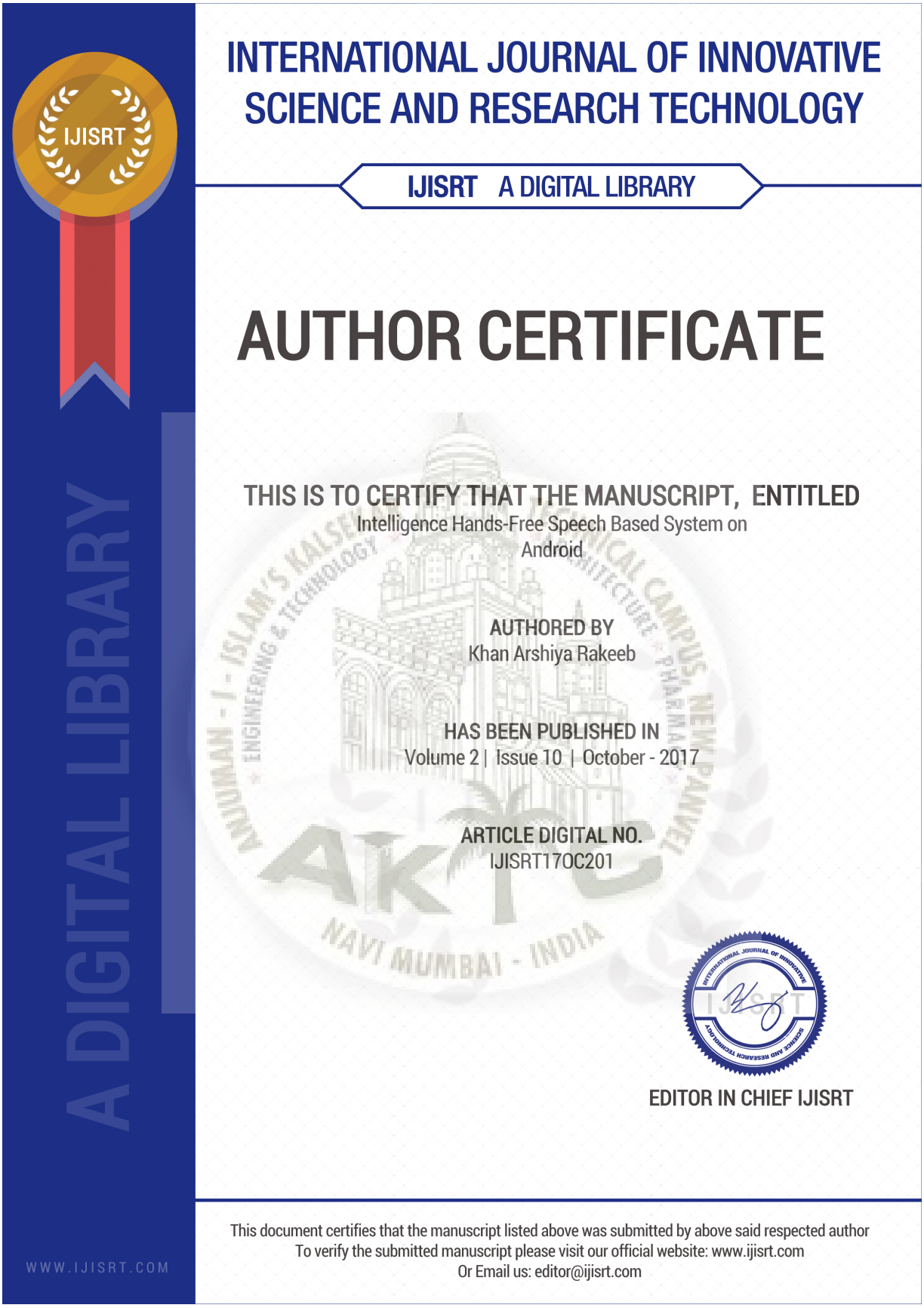
IJISRT17OC201

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

A DIGITAL LIBRARY

WWW.IJISRT.COM



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands-Free Speech Based System on Android

AUTHORED BY

Khan Arshiya Rakeeb

HAS BEEN PUBLISHED IN

Volume 2 | Issue 10 | October - 2017

ARTICLE DIGITAL NO.

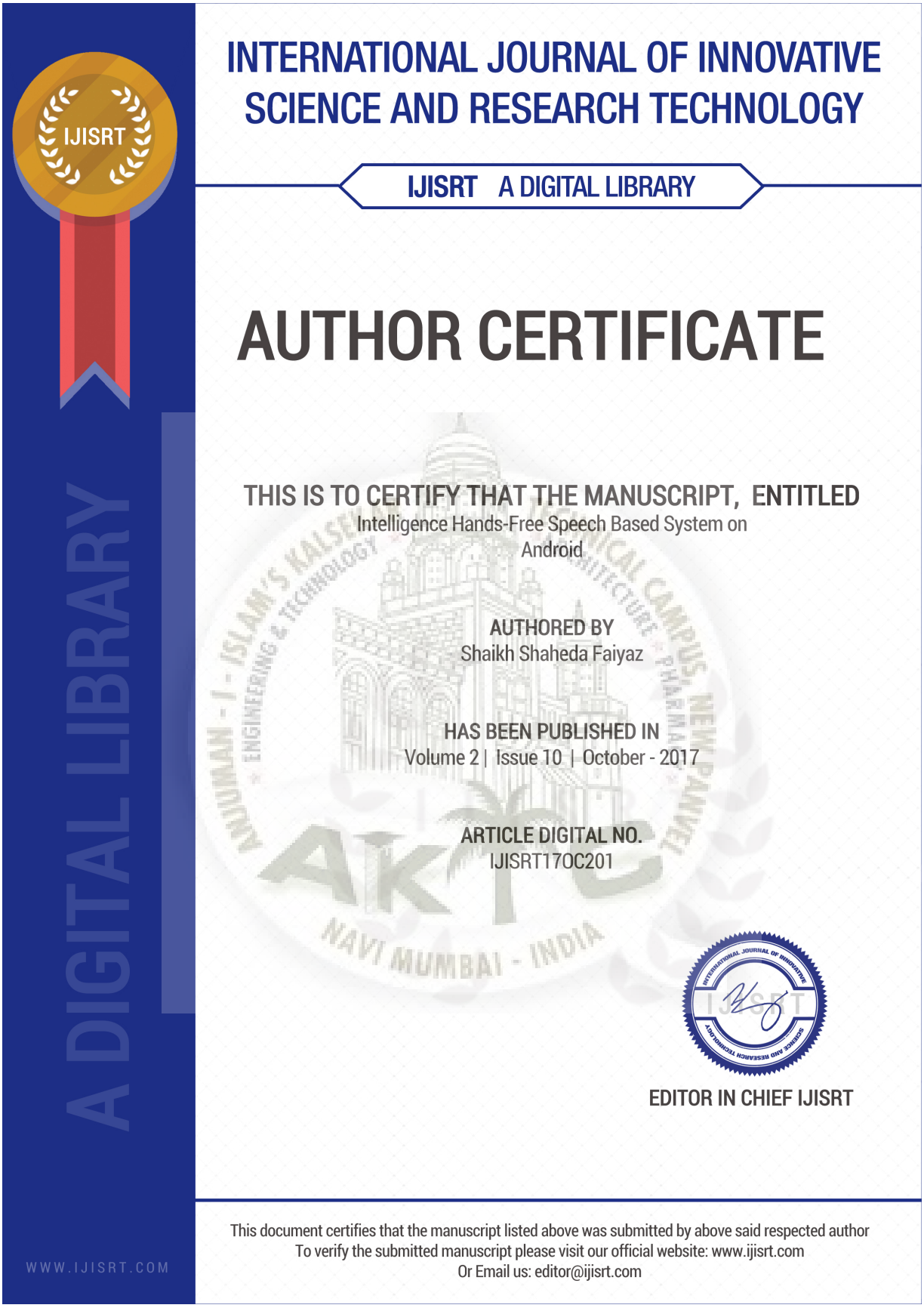
IJISRT17OC201

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

A DIGITAL LIBRARY

WWW.IJISRT.COM



INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands-Free Speech Based System on Android

AUTHORED BY

Shaikh Shaheda Faiyaz

HAS BEEN PUBLISHED IN

Volume 2 | Issue 10 | October - 2017

ARTICLE DIGITAL NO.

IJISRT17OC201

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

A DIGITAL LIBRARY

WWW.IJISRT.COM

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands Free Speech Based SMS System on Android

AUTHORED BY

Ansari Mukhtar Amir

HAS BEEN PUBLISHED IN

Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.

IJISRT18AP173

EDITOR IN CHIEF IJISRT

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands Free Speech Based SMS System on Android

AUTHORED BY

Inamdar Mohsin Harun

HAS BEEN PUBLISHED IN

Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.

IJISRT18AP173

EDITOR IN CHIEF IJISRT

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands Free Speech Based SMS System on Android

AUTHORED BY

Khan Arshiya Raueeb

HAS BEEN PUBLISHED IN

Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.

IJISRT18AP173

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

A DIGITAL LIBRARY

WWW.IJISRT.COM

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED

Intelligence Hands Free Speech Based SMS System on Android

AUTHORED BY

Shaikh Shaheda Faiyaz

HAS BEEN PUBLISHED IN

Volume 3 | Issue 4 | April - 2018

ARTICLE DIGITAL NO.

IJISRT18AP173

EDITOR IN CHIEF IJISRT