****

**A**

**Project Report**

**On**

**Audio to Text Conversion**

**Submitted By**

**Name** **Seat No**

**Ms. Prerana Raut 188983**

**Ms. Shrutika Mamdyal 188934**

**Ms. Aparna Mangalaram 188944**

**Ms. Pranjali More 189004**

**Under The Guidance Of**

**Mr.Sathe A.**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**GOVERNMENT POLYTECHNIC, SOLAPUR**

**2018-2019**

**Government Polytechnic, Solapur**

**Department of Computer Technology**

****

**CERTIFICATE**

It is to certify that

**Ms. More Pranjali**

**Class:-CM6G**

Has satisfactorily completed the project titled as

**“Audio to Text Conversion”**

As a partial fulfilment

For awarding the Diploma in Computer Technology

By Maharashtra State Board of Technical Education , Mumbai

For the year 2018-2019.

Project Guide: -Mr. Sathe A.

Principal Head Of Department

Internal ExaminerExternal Examiner

Date:-

Place:-Solapur

**ACKNOWLEDGEMENT**

It was very exciting for us to work on the project Audio to Text Conversion. During these work we have gained both the practical as well as theoretical knowledge of great significance.

We are greatly thankful to Mr. Abhay Sathe Sir and Mr. Ravi Gangundi sir for guiding us through our project. We also thankful to our project guide Mr. Abhay Sathe Sir for his valuable guidance, encouragement and provision of necessary facilities made these possible works. It has been a highly encouraging and knowledge gaining experience. He has been a source of inspiration throughout the project and helps us in all our problems.

We are absolutely grateful to all non teaching staff for their assistance which is key factor behind our success.

Finally we offer our great thanks and regards to our family for their support which helped us through the difficulty and hardships of life to earn this achievement.

**INDEX**

**Content Name**

1**.** Introduction

1.1 Planning

1.2 Feasibility Study

1.3 Scheduling and Resources

2. Analysis

2.1 Problem Statements

2.2 Objective of project

2.3 Software/hardware requirements

2.4 Input and output

3. Design

3.1 Use Case Diagram

3.2 Activity Diagram

4. Coding

4.1 Algorithm

4.2 Important modules information

4.3 Output

5. Testing

5.1 Test Plan

5.2 Test cases

6.Reference

7.Conclusion**1. Introduction**

Audio to Text Conversion is Java Project developed for deaf people , college students, anybody can use it. It simply takes audio file(.wav) and captures voice then converts it into Text. It Is totally offline. It also performs live speech recognition. We can use microphone to record our voice and then it it will convert spoken words into text offline.

**Purpose:-**

This project has number of applications in different areas and provides potential benefits. Nearly 20% people of the world are suffering form various disabilities; many of them are blind or unable to use their hands effectively.The speech recognition system in those cases provide a significant help to them. So that they can share information with people by operating computer through voice input.

**1.2 Feasibility Study:**

Feasibility study is carried out whenever there is a complex problem or opportunity. It is undertaken to decide the existing system or probability of either improving the existing system or developing completely new system. It helps to get rough assessment of whether a feasible solution exists. This is essential to avoid committing large resources to a project and then resend on it latter. It is required to answer following queries-

* Whether a new system is to installed or improving the existing system?
* What should be embedded in the new system?
* Define the potential of the existing system.
* Avoid costly repair at later stage.

**1.3 Economically Feasibility:-**

The system being developed is economic with respect to blind or deaf people point of view. It is cost effective in the sense that has eliminated the paper work completely.

The system is also time effective because we can write report by using voice recognition. It also saves physical work to write-up any document.

The result obtained contains minimum errors and are highly accurate as the data is required.

**Technical feasibility:**

The technical requirement for the system is economic and it does not use any other additional Hardware except (Microphone is optional )and software.

**Behavioural Feasibility:**

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.

**2.Analysis**

**2.1 Problem Statement:**

This project is aimed to convert the audio to text in java using API. This will help the peoples to generate the text of their audios. It also aimed to help the peoples those who are blind and deaf. This project is capable of recognizing the speech and covert the audio to text.

**2.2 Objectives of Project:**

* To convert that audio file into text

**2.3 Software and Hardware Requirements:**

* **Software Requirement:**

The minimum software requirements used for this software are

1. JDK 1.6 onwards

2. Netbeans 8.1/2

3. Sphinx4 core and data dependency.

4. Maven code builder.

5. Swing

* **Operating System:**

Windows(32-bit or 64-bit) .[window series(win xp onwarrds)]

* **Hardware Requirement:**

RAM -260 MB.

Disk- 1TB

Processor-intel series

Microphone

Soundcard

Working laptop /pc with above specification

**2.4 Input and Output Format**

**Input Format:**

You have to input wav audio file having sample rate 16bit ,mono 16000Hz or 8000Hz. (Must in .wav file)

RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 16000 Hz

or

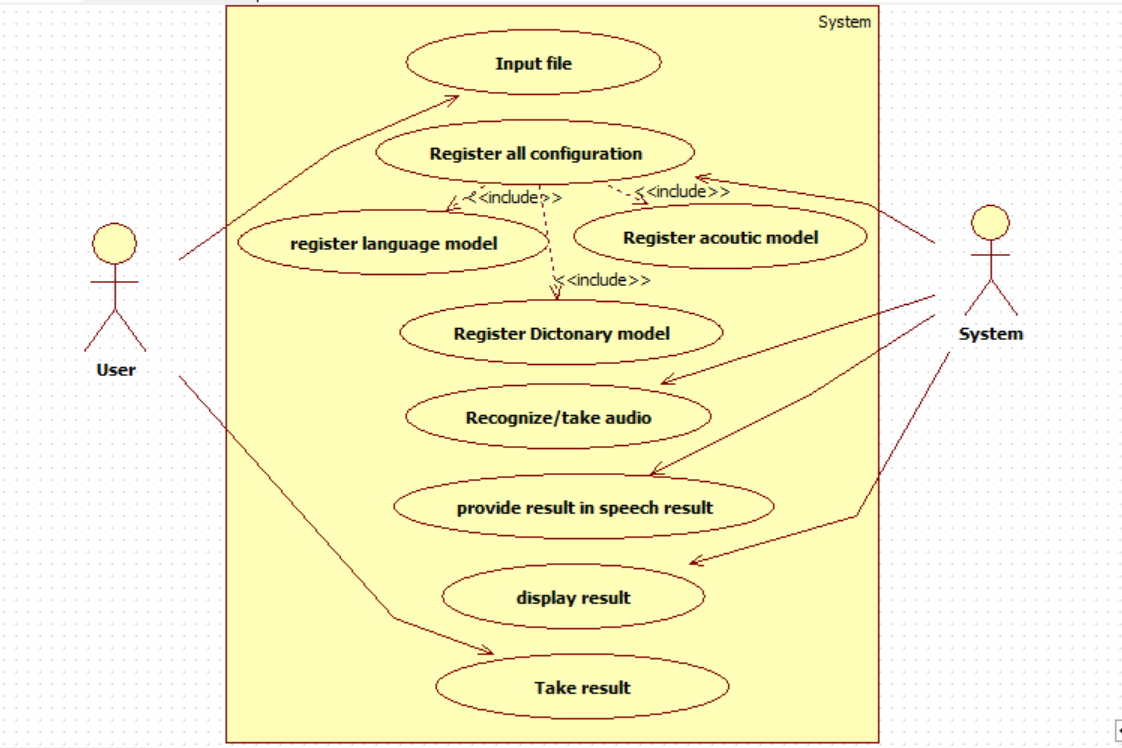
RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 8000 Hz

**Output Format:**

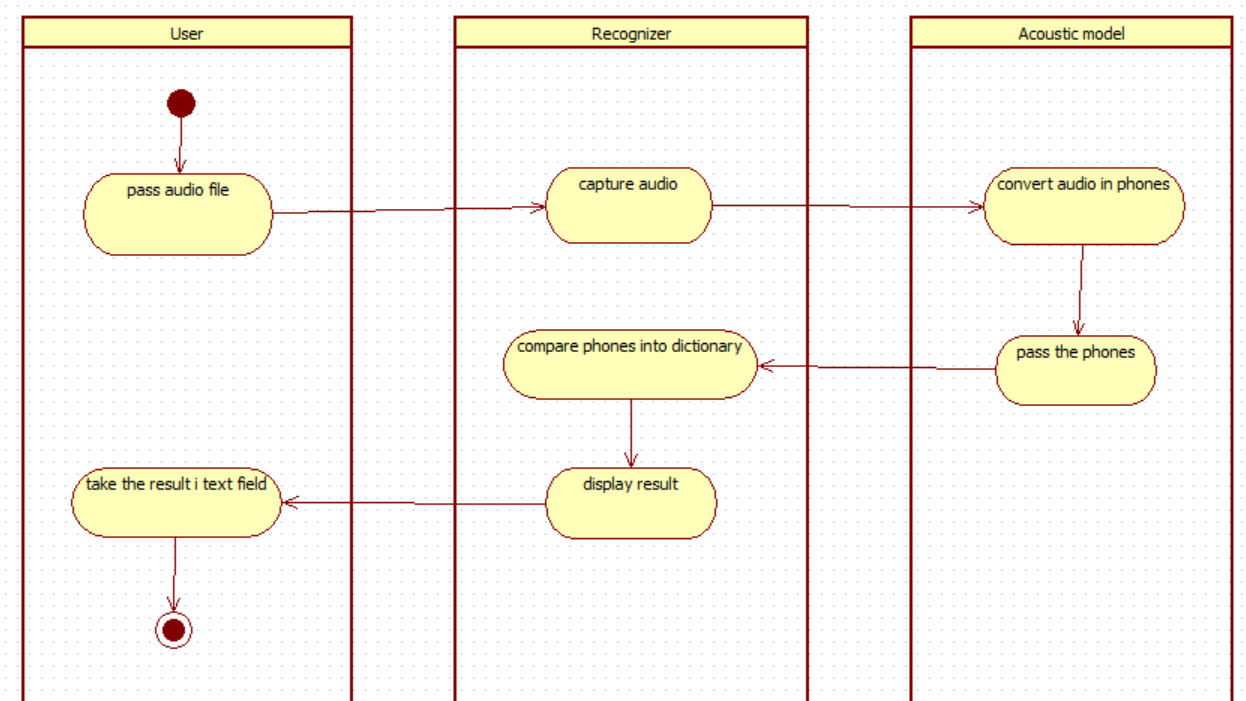
A SpeechResult provides access to various parts of the recognition result, such as the recognized utterance, a list of words with timestamps, the recognition lattice etc.

**3 Design**

**3.1 Use Case Diagram**



**3.2 Activity Diagram**

****

**4.Coding**

**4.1 Algorithm**

Step 1: Start

Step 2: Initialize Configuration object

Step 3: accept audio

Step 4: recognise file

Step 5: align speech

Step 6: convert into phonemes using acoustic model

Step 7: examine related phonemes

Step 8: select best match

Step 9: Display result in text field

Step 10: Close configuration object

**4.2 Information about modules**

Sphinx4 is a pure Java speech recognition library. It provides a quick and easy API to convert the speech recordings into text with the help of CMUSphinx acoustic models. It can be used on servers and in desktop applications. Besides speech recognition, Sphinx4 helps to identify speakers, to adapt models, to align existing transcription to audio for timestamping and more.

**For most of the speech recognition jobs high-level interfaces should be sufficient. Basically, you will only have to setup four attributes:**

* Acoustic model
* Dictionary
* Grammar/Language model
* Source of speech

**Configuration**

A Configuration is used to supply the required and optional attributes to the recognizer.

* Configuration configuration = new Configuration();

*// Set path to acoustic model.*

* configuration.s etAcousticModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us");

*// Set path to dictionary.*

* configuration.setDictionaryPath("resource:/edu/cmu/sphinx/models/en-us/cmudict-en-us.dict");

*// Set language model.*

* configuration.setLanguageModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us.lm.bin");

A SpeechResult provides access to various parts of the recognition result, such as the recognized utterance, a list of words with timestamps, the recognition lattice, etc.:

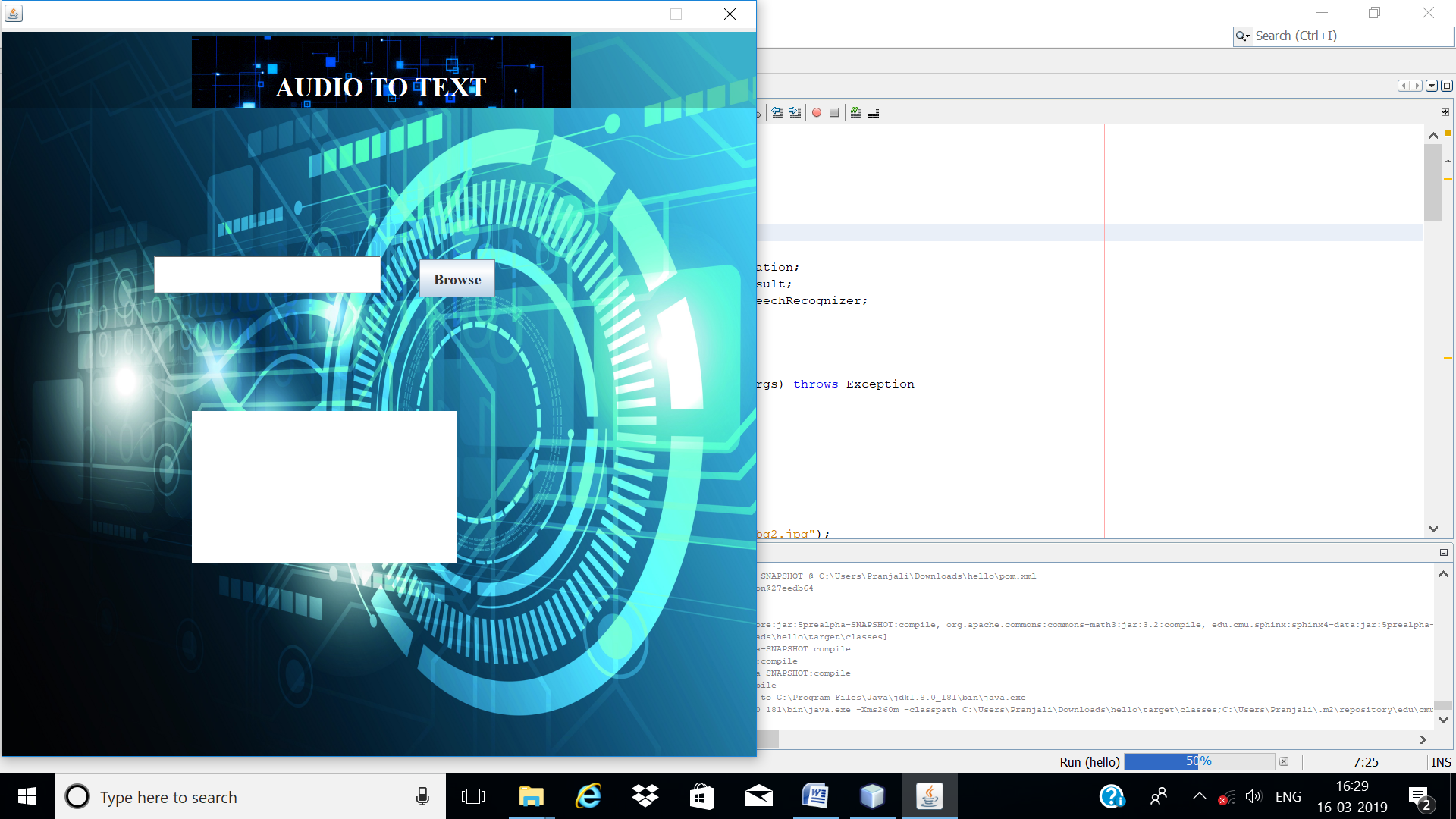
* *// Print utterance string without filler words.* System.out.println(result.getHypothesis());

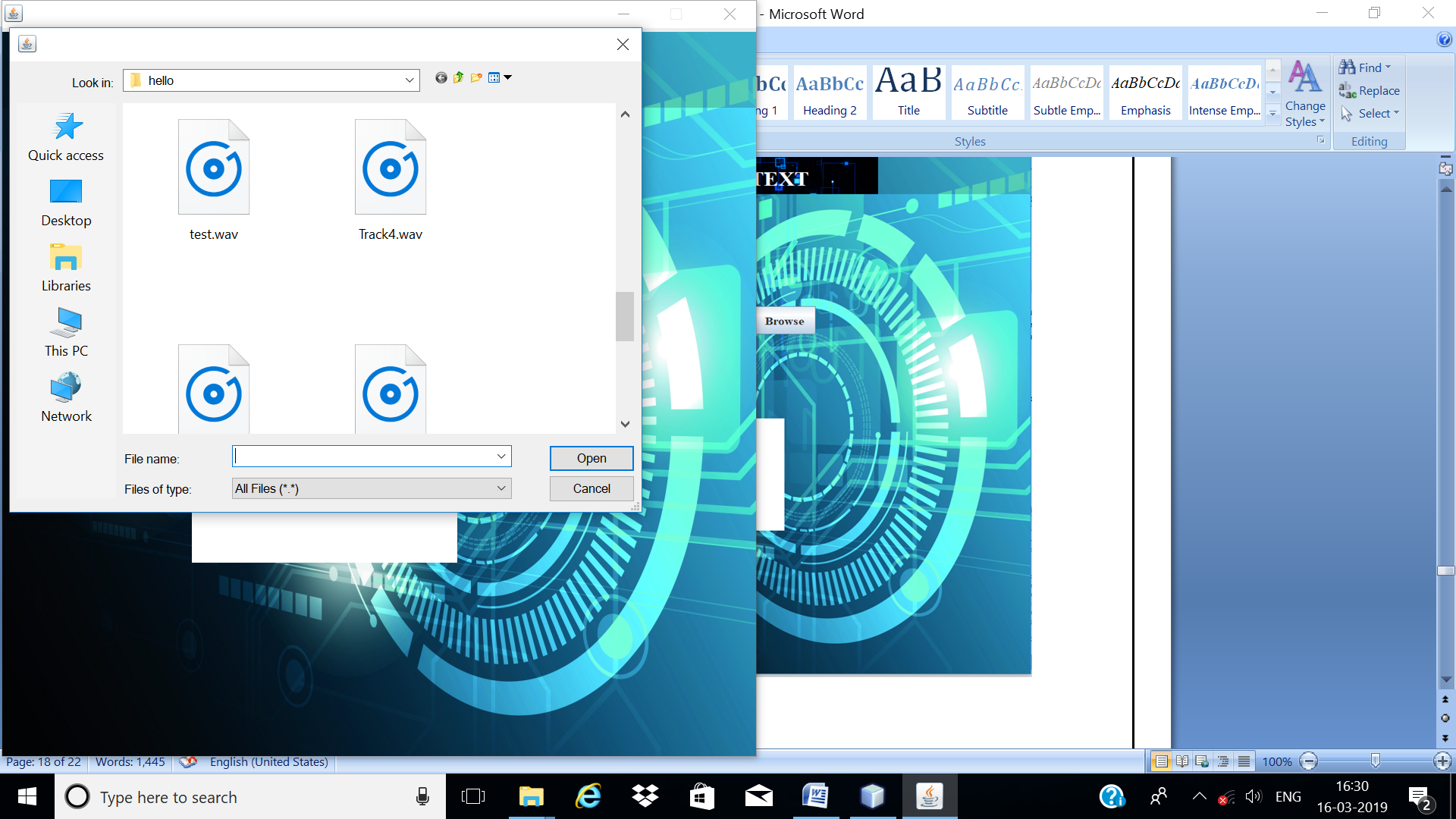
Note that the audio for this decoding must have one of the following formats:

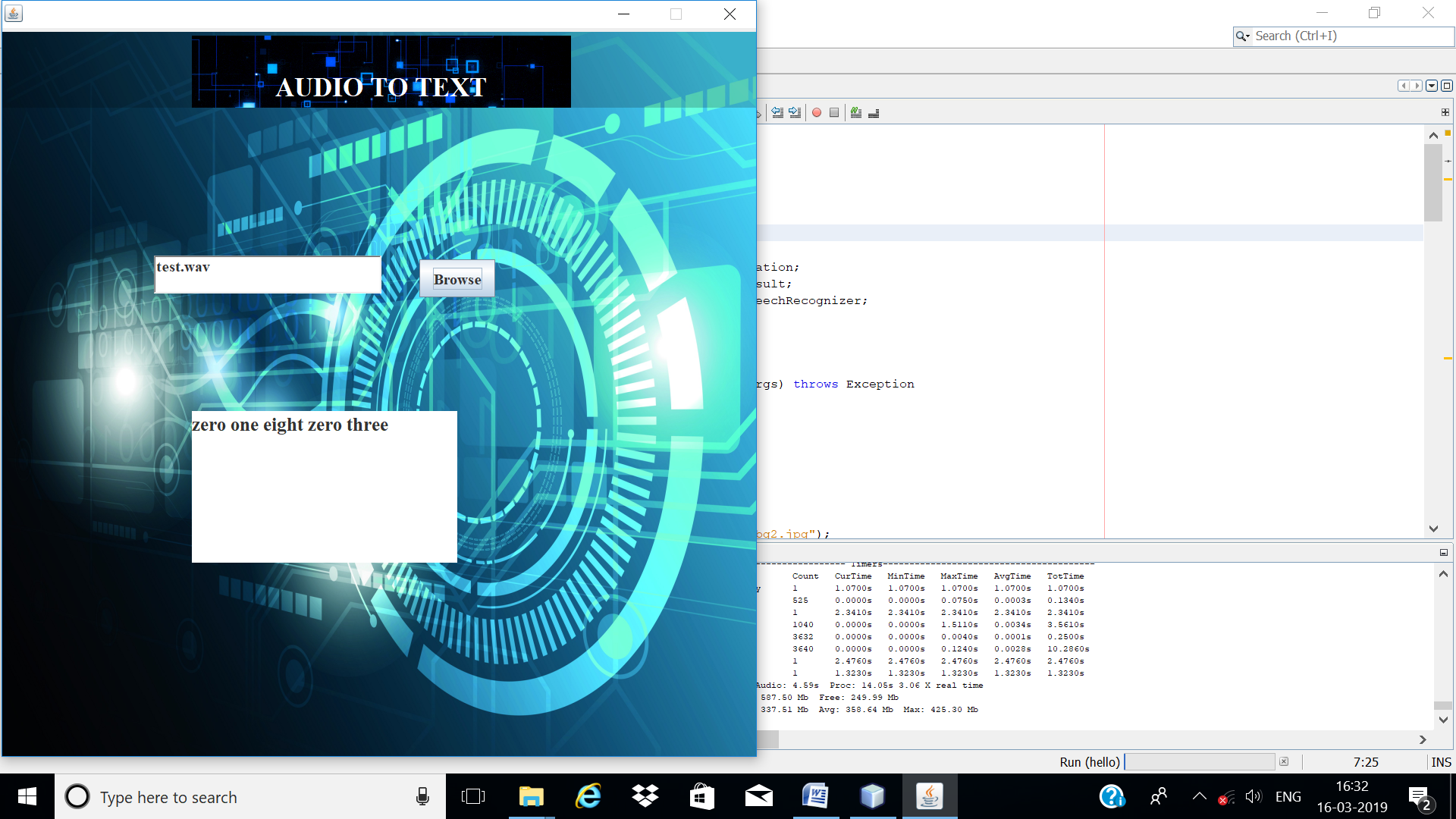
RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 16000 Hz.

(The decoder does not support other formats. If the audio format does not match, you will not get any results. you need to convert your audio to a proper format before decoding.)

**4.3 Output**

****





**5. Testing**

**5.1 Test Plan**

After completing any project it is necessary to test that particular project. The need of testing the project comes because of verification, that’s the software, is working properly or not. Whether it is gives the expected output as per the requirement.

**The Testing includes following types of testing:**

1. Unit Testing
2. Integration Testing
3. High Order Test
4. Validation Testing
5. Unit Testing:

Initially, Test focuses on each component individually, ensuring that it functions properly as unity. Unit testing focuses verification effort on the smallest unit of software design the software component of modules.

1. Integration Testing :

Integration Testing addresses the issues associated with the dual problem of verification and program construction. It is the systematic technique for the constructing the software architecture. While at the same time

conducting test to uncover errors associated with interfacing.

1. High Order Test:

After has been integrated (constructed), set of high order tests are conducted. After the test cases, the high order tests are conducted for testing the project. When we link different forms together this test removes the errors.

1. Validation Testing:

It provides the final assurance that software meets all functional, behavioural and performance requirements. Validation refers to the different set of tasks which ensures that the software that has been built is traceable to customer requirements.

**In this project we are implementing two types of testing:**

1. Unit Testing:

In the unit testing the module interface is tested to ensure that, if wepass .wav audio file then only it will displays correct result.

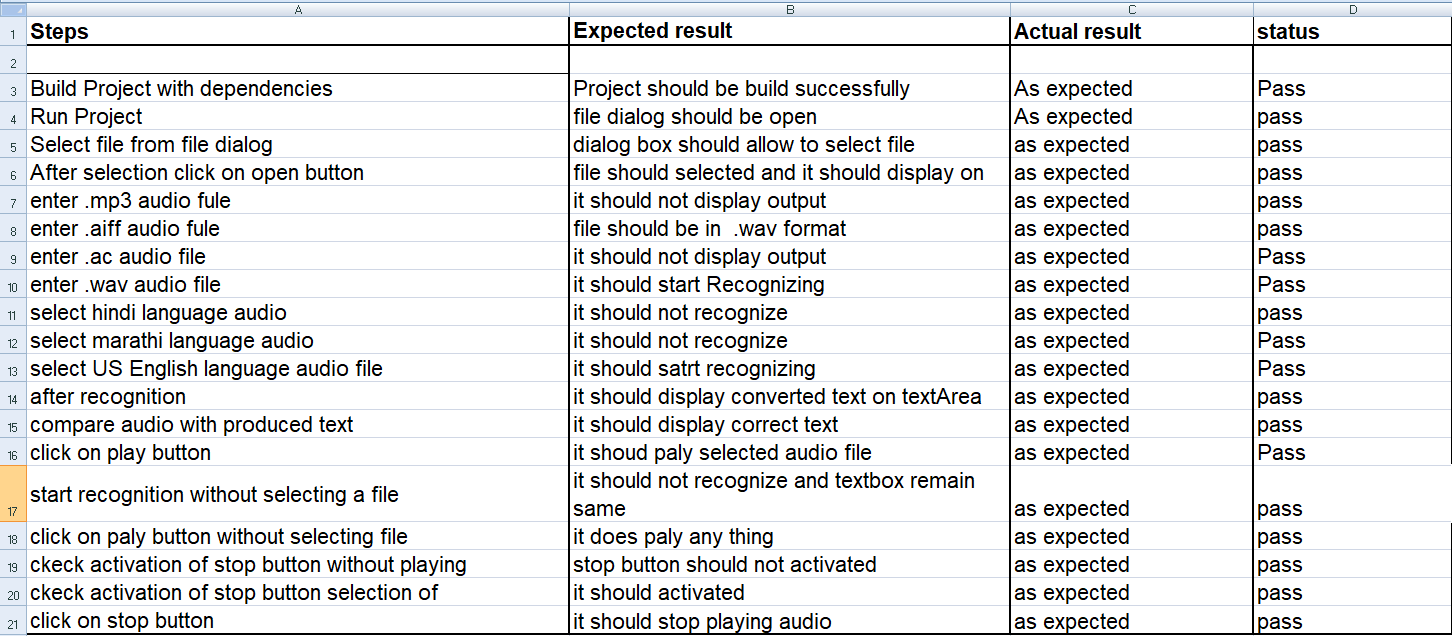
According the unit testing we tested each and every forms individually and resurrected errors that where encountered.

1. Validation Testing:

In this testing, we are validated whether software giving correct result / text corresponding to the given audio.

In this testing we tested the whole project according to itsexpected behaviour, function and performance requirements.

**5.2 Test cases**



**6.References**

**During the project work we have used the following References: -**

1. www.developers.com

2. www.puneetk.com

3. www.cmusphinx.com

4. www.stackoverflow.com

5. [www.oracle.com](http://www.oracle.com)

6. [www.github.com](http://www.github.com)

7. [www.uniphore.com](http://www.uniphore.com).

8.https://www.uniphore.com/blog/2012/08/how-speech-recognition-works

9.https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition2.html

10.http://my.fit.edu/~vkepuska/ece5527/Projects/Fall2011/Burgos,%20Wilson/sphinx41.0beta6/sphinx41.0beta6/doc/ProgrammersGuide.html#helloCodeWalk

11.https://library.vuforia.com/content/vuforialibrary/en/reference/unity/classVuforia\_1\_1WordResult.html#details

12. <https://github.com/goxr3plus/java-google-speech-api>

13. <https://github.com/eracle/speech-transcription2srt>

14. <https://github.com/saurabhshri/CCAligner>

15. <https://www.programcreek.com/java-api-examples/?code=dnbn%2Fsubmerge%2Fsubmerge-master%2Fsubmerge-web%2Fsrc%2Fmain%2Fjava%2Fcom%2Fgithub%2Fdnbn%2Fsubmerge%2Fweb%2Fmodel%2Fentity%2FDualSubtitleConfig.java>

16. <https://github.com/saurabhshri/CCAligner>

**Conclusion**

This project has number of applications in different areas and provides potential benefits. Nearly 20% people of the world are suffering from various disabilities; many of them are blind or unable to use their hands effectively. The speech recognition system in those cases provides a significant help to them. So that they can share information with people by operating computer through voice input.