English Accent Trainer For Non-Native Speakers

Usama Abbas

Shabeer Ahmad

DEPARTMENT OF COMPUTER SCIENCES
COMSATS UNIVERSITY ISLAMABAD,
ATTOCK CAMPUS - PAKISTAN

SESSION 2017-2021

English Accent Trainer For Non-Native Speakers

Undertaken By:

Usama Abbas
CIIT/SP17-BCS-037/ATK
Shabeer Ahmad
CIIT/SP17-BCS-032/ATK

Supervised By:

Mr. Najam Dar

A PROJECT SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCES
COMSATS UNIVERSITY ISLAMABAD,
ATTOCK CAMPUS - PAKISTAN

SESSION 2017-2021

UNDERTAKEN

We certify that this is my/our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged. If this statement is untrue, we acknowledge that we will have committed an assessment offence and shall be liable to punishable action under the plagiarism rules of HEC.

_______________	______________________
Usama Abbas	Shabeer Ahmad
SP17-BCS-037	SP17-BCS-032
Dated: _________	Dated: _________

FINAL APPROVAL

Certified that we have read this project report submitted by Mr. (Usama Abbas and Shabeer Ahmad) and it is, in our judgment, of sufficient standard to warrant its acceptance by Department of Computer Science, COMSATS University Islamabad, Attock Campus, for the BSCS degree.

Committee:

1. External Examiner ______________________________

(Examiner Name)
Designation
University Name

2. Supervisor ______________________________

(Mr. Najam Dar)
Lecturer

3. Head of the Department ______________________________

(Dr. Khalid Awan)

DEDICATION

This Project is dedicated to my parents and teachers.

For their endless, love, support and encouragement.

Acknowledgment

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Najam Dar”. Without his supervision, advice, and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Shabeer Ahmad	Usama Abbas
_____________	____________

Project Brief

PROJECT NAME	ENGLISH ACCENT TRAINER FOR NON NATIVE SPEAKERS
OBJECTIVE	Providing application for English Accent Detection and learning accents Accurately detect the accent of speaker. Detecting Noise, performing noise removal. Detection of silence and their removal. Extracting MFCC features. A Machine Learning Model for Accent Classification
UNDERTAKEN BY	USAMA ABBAS SHABEER AHMAD
SUPERVISED BY	MR. NAJAM DAR LECTURER COMPUTER SCIENCE COMSATS UNIVERSITY ISLAMABAD, Attock
STARTED ON	01/10/2019
COMPLETED ON	20/11/2020

COMPUTER USED	HAIER Y11-C
SOURCE LANGUAGE	PYTHON,JAVA,XML
OPERATING SYSTEM	WINDOWS 10
TOOLS USED	ANDROID STUDIO, JUPYTER Note Book, MS Office, PyCharm

ABSTRACT

The English language is considered as International language and is spoken by most of the people and used as primary source of communication. People from all around the world comes with different accents which is due to the fact they belongs to different regions, represents different culture and all of them have their own distinct accents. Accent provides information about educational background also.originating from various geographical locations. Accent identification is an important problem in technology today.

The work in the field of Accent Classification has increased despite the progress of automatic speech recognition (ASR) systems that have led to assistants like Alexa and Siri, accent is still an issue in developing robust ASR systems that deliver high performance across diverse user groups.ASR systems like Google Now and Siri are usually trained on and perform best for these accents.

We are developing a Machine Learning Model that will used for accent classification which we used than to develop an android application which will perform "Similarity Test"- this test is performed by taking audio of user and performing noise removal, pause detection and removal, extracting MFCC features, standardizing the features and then feeding these features to our CNN model which will classify the accent of user and tells the results as "Similarity Index"- it is relevance of user's accent to the selected accent. In this user will have a model which will give similarity test with high accuracy and then will provide practice exercises. Practice Exercises contains audio clips of native peoples and user will improve their accents.

Accent trainer is an android application that will be developed by using NLP (Natural Language Processing) and Machine Learning. Good communication is the primary need of a person to carry out the daily tasks but when an accent is not native to you and you are not good in a specific accent then it becomes difficult for you to convey your message to others. So the primary goal of this application is to train a non-native English speaker to speak English in an accent similar to native speakers.

Our machine learning model can be used in Automatic Speech Recognition field to improve the recognition of the words pronounced by user so that it can help in the smart applications like Siri, Alexa and many other home/office automation gadgets.

TABLE OF CONTENTS

List of Figures

List of Tables:

CHAPTER 1

INTRODUCTION

1. Introduction:

The accent of a person tells about the background of the person where he is from. It also tells about the social and educational background. People from the different region have different accents but some of the accents have become a standard and does not belong to a particular region. Received Pronunciation (RP) is such an accent that is not related to a particular region. The main fact is that nowadays the majority of English speakers are non-native speakers. Everyone has its accent which tells about their civilization so how someone accents is an issue? Why do we need to develop this application?

Nowadays the English as treated as an international language so the main problem is that how we can communicate with someone who has either native accent or non-native accent. In keeping in mind the need for a solution to this problem we are developing an android application that will tell the user about their accent relevancy to a particular native accent. After knowing the result of the relevancy user can take exercises to improve their accent by listening and practicing the passage provided by the user, the application will provide audio so that the user can listen to it and repeat it.

As we know there is very progress in automatic speech recognition (ASR) systems that have helped us in develop the applications like Alexa and Siri, but still the problem in developing the independent and robust ASR projects and system that can give overcome the problems we face now a days. Studies and analysis based on data has revealed that gender and accent are very crucial factors involved in a speaker voice which effects the effectiveness of ASR systems ASR systems like Google Now and Siri are developed and being trained on the American and British Accents and also give best results for mentioned accents, and it can be seen by personal experience that by not giving audio input in these accents speech recognition systems usually not give very healthy results. To make it work we try to speak an accent which is not our own so that we can get the ASR to identify and recognize the speech and our said words correctly, which is very unnatural way and is a kind of evidence that ASR systems still needs to be improved. So we came to a result by our discussion that accent is very crucial and important in speech recognition, we were motivated to develop a model based on machine learning for accent classification and by which we will develop our “English Accent Trainer”. Other possible usage of this model and its applications include immigration screening.

1.1. Brief

Accent trainer is an android application that will be developed by using NLP (Natural Language Processing) and Machine Learning. Good communication is the primary need of a person to carry out the daily tasks but when an accent is not native to you and you are not good in a specific accent then it becomes difficult for you to convey your message to others. So the primary goal of this application is to train a non-native English speaker to speak English in an accent similar to native speakers. It will helpful for the students as well as other people because with the help of it they can learn an accent of their interest. By using this application user can test his/her accent that how much it relates to a specific accent. The application will first test the accent similarity index of the user. This step is the most important and core of this application because it is the base of this application. In order to learn an accent user must know with time to time that how much his/her accent relates to the desired accent. So to achieve this we need a test, which is being performed here i.e “Similarity Index Test”. Here accent similarity index means that how much a person’s accent relates to a set of English accents so basically its a comparison test of user’s accent to an accent from a given set of accents. After this test application will display result of this test in term of percentage and will also mention to which accent he relates, this is known as similarity index. After these two steps now another important step comes, here user is provided with a set of practice exercises which includes of text passage and audio clips in that specific accent so by listening and practicing it user can learn an accent. This application is very helpful for those who want to give linguistic tests like IELTS etc and for those who are applying for the visa and it is not limited to these benefits, its use is diverse. It will be very helpful for people who want to learn different accents and in education sector.

1.2. Relevance to Course Modules

1.2.1. **Mobile Application Development:** In this course, we have studied android development using an android studio and this will help us in developing the interface of our application. Our system will have an interface in android so that user can interact with easily and with comfort. In this course we have studied android development.

1.2.2. **Machine Learning:** In this course, we have studied different algorithms which are used to train models to perfume task automatically after getting the training data.

1.2.3. **Report Writing Skills:** This course is about learning how to write reports and other formal documentation and in our project we need to write our documentation so this course is helping a lot in this task.

1.2.4. **Human-Computer Interaction:** An interactive system is easy and comfortable for the user to use the system and understand it easily and this course is all about designing an interacting system following standard rules.

1.3. Project Background

The main idea behind this project is that every year there is a large number of students who want to carry on their studies Abroad and many other people who want to do jobs. The main restriction or barrier for students are mostly the linguistic test. Understanding the accent of others is very important for smooth communication and it is not possible until you are not able to understand an accent. People who provide training in English speaking charge heavy fees. So there is a need for such a system which is comfortable to use and is completely free to use. This application will provide users to test their accent similarity. Testing accent similarity is the key feature in this application because there are many other application in market which provides you the facility of practice exercises in different accents but there is no such application which provides you the facility to test your accent relevancy to a specific accent of your accent. This accent similarity test also helps user to monitor his/her accent improvement because he/she can test his/her accent relevancy at any time.

After this important similarity test now the user is aware about the accent relevancy level so now he/she needs training in his/her desired accent. It is covered by providing practice exercises to practice to improve their accent.

The applications which already exist are not a complete package and following problems exists:

- **Similarity Index Test :** The feature that makes it better application than other application is this test. This test is the key, because without knowing the improvement in accent it will be a blind path to follow. So there should be criteria through which user can evaluate his/her improvement.
- They do not have a friendly interface which is a major problem because if the user does not find it comfortable then they will not use it.
- They are very restrictive in terms of accent variations. They do not cover a large number of accents.

1.4. Literature Review

Here we are going to discuss and analyze already existing accent training applications and how our application covers these remaining features. Our application will provide an interactive user interface that is very easy to use. In this way, users will easily able to check their accent proficiency and will be able to improve accent by audio exercises.

The application with which we are comparing our application is British Accent Training. The main problem with this application is that it has a very ugly user interface which is very annoying for a user. If the interface of a system is not friendly and comfortable then it will restrict the user to further use it. It does not provide any mechanism for testing the similarity of the user's accent.

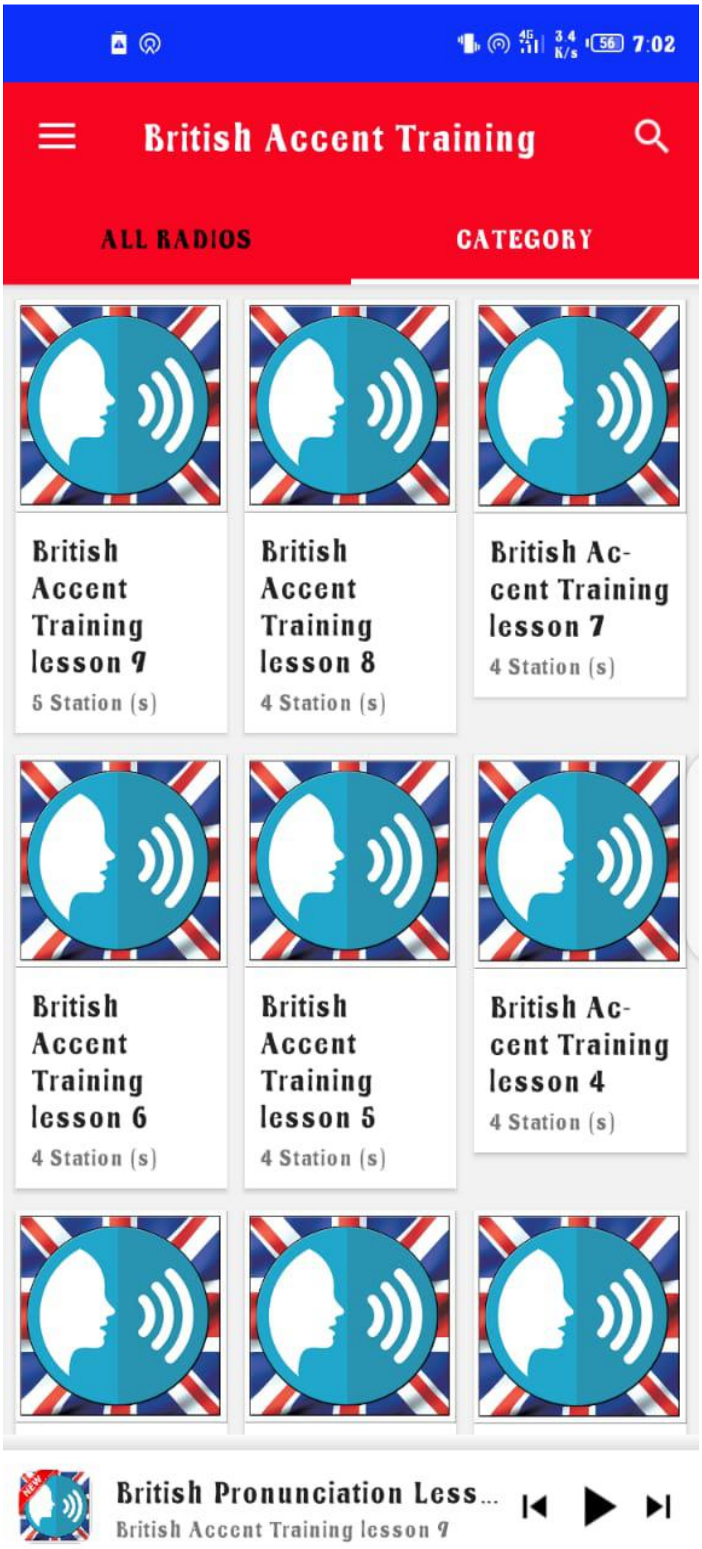

7:02
British Accent Training
ALL RADIOS
CATEGORY
British Accent Training lesson 9
5 Station (s)
British Accent Training lesson 8
4 Station (s)
British Accent Training lesson 7
4 Station (s)
British Accent Training lesson 6
4 Station (s)
British Accent Training lesson 5
4 Station (s)
British Accent Training lesson 4
4 Station (s)
British Pronunciation Less...
British Accent Training lesson 9

Figure 1.1 British Accent Training

Another application for comparison is “Speak English”. The problem with this application is that it is very complex and it has a lot of unrelated things and formalities that disturbed a lot during the usage. The application provides a simple mechanism so that users can easily use it without any hurdle because there will be many users who will get frustrated from this long procedure, the reason is that not everyone is technology literate. So we need to provide them simple mechanisms so that they feel comfortable during the learning process.

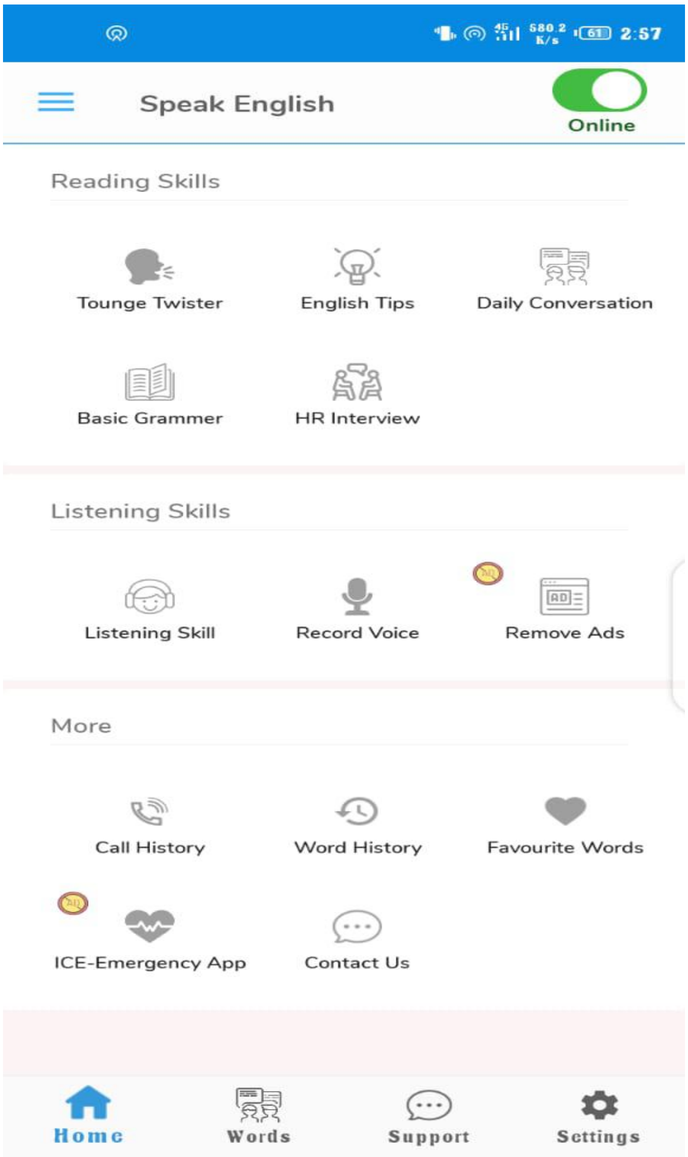


Figure 1.2 Speak English

There are many other applications which have many problems and we have done a complete study on these issues and try to cover most of them and specifically the most important ones.

1.5. Analysis of Literature Review

This application will provide easy assistance to those who don’t want to waste the time and money in taking coaching classes and also don’t want to get stuck in difficult mechanisms provided by some of the applications available.

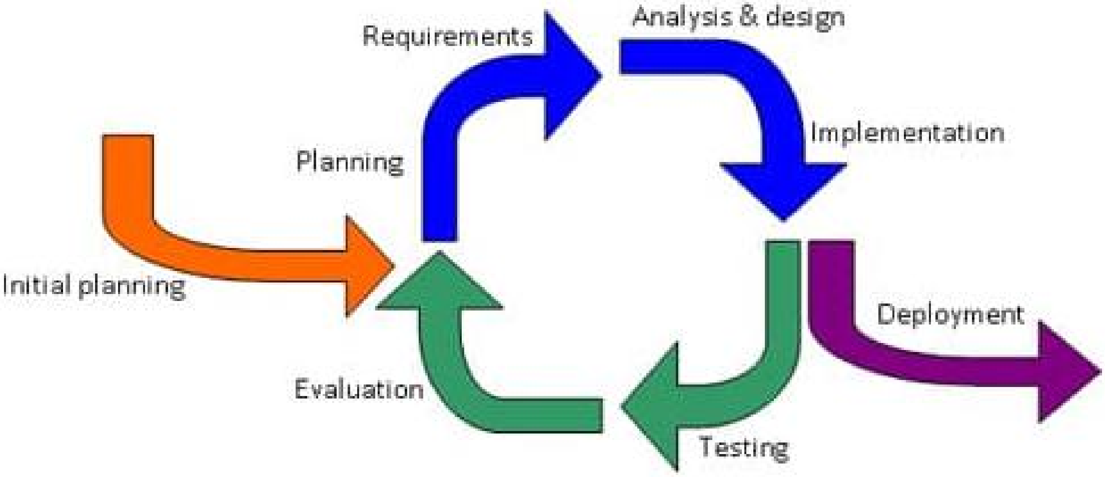
Comparison Matrix:

Features	English Accent Trainer	Speak English	British Accent Trainer
Similarity Test	✓	✕	✕
Easy Mechanism	✓	✕	✓
Practice Exercises	✓	✓	✓
Records	✓	✕	✕
Friendly User Interface	✓	✕	✕

Table 1 Comparison Matrix

1.6. Methodology and Software Life cycle of Project

We use the iterative approach to develop this application. The iterative method is a way to breakdown the development process into small patches so that instead of developing the whole software at once (which is difficult to develop).



Model 1: Typical iterative development process

iterative-model

Figure 1.3 Model

1.6.1. Rationale behind selected methodology

Building a complete project in one go is a difficult task due to several reasons because there are some requirements about which we not aware at the beginning of the project but came across it in mid of development process and building a complete thing in one cycle is it self a difficult task because you have to complete the whole project than you can test its core functionality. The iterative process is the way to breakdown the whole process to develop complete software not in one go but into smaller chunks. The reason behind this approach is that it is easily measurable and small chunks are easy to test and debug as compare to developing the whole software and then testing it, at this point debugging becomes difficult.

CHAPTER 2

PROBLEM DEFINITION

2. Problem Definition

2.1. Problem Statement

English is recognized as an International language and people of different language communicate to each through the English language. Moreover, International universities, as well as local universities, use the English language as their means of education. Companies, where international employees are working, use English for communication. The good part is that those employees and students know English but when it comes to accent then the problem arises because there are a lot of accents that are used in different European countries. A person who knows how to talk English but it's obvious that his/her accent varies because accent depends on background and culture. So here is the problem, people face difficulty in understanding a different accent. Our application will not just tell users about their accent similarity to a specific accent, will also provide practice exercises to improve their accent.

2.2. Deliverable and Development Requirements

Following is the list of deliverable of our project:

- User profile
- Accent Similarity Test
- Practice exercises
- Number of different accents

Development requirements include the following software requirements:

- IDE : Android Studio, PyCharm
- Database: SQLite
- Programming Languages: Java, Python, XML

CHAPTER 3

REQUIREMENT ANALYSIS

3. Requirement Analysis

Software Requirement Analysis (SRS) is not an ordinary document because it provides a basic understanding of functional as well as non-functional requirements. We can consider it as the starting point of the project because it serves as a written contract between the client and organization about the features and the functionalities of the project. With the help of SRS, both clients and organizations make clear to each other about the deliverable project.

3.1. Use case diagrams

This type of diagram is a visual representation of the interaction of user with the system. It is used to describe the use cases (set of different actions) that can be performed by the actors (external users) by using the system. With the help of the use case, you can visualize the behavior of the system when the user uses it.

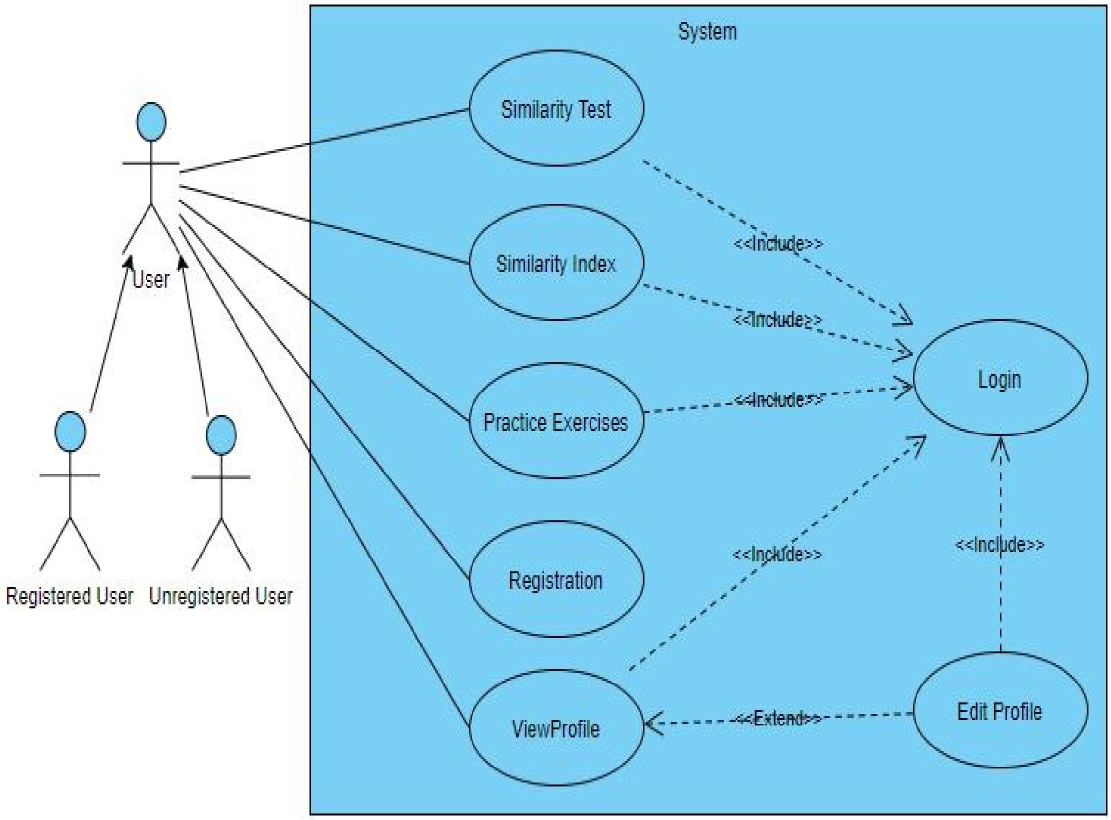


Figure 3.1 Use-Case Diagram

This diagram here describes the who are our actors and their roles and the what are use-cases. By use-cases we meant that what are the functionality that can be performed by our system. Our system has only one actor which is our user and then we have generalized our user so that we can define the type of users. Basically user has only one type but it varies in status i.e, Registered User and unregistered user.

To use this application one must register itself so that his/her profile can be maintained and it is necessary to get benefit of this application that one must be registered so he/she can maintain the profile and can tract the improvement in the accent.

The second unregistered user is every user who comes for the first time and needs to register himself so he can use the application.

After registration, user have following use cases :

Similarity Test

Similarity Index

Practice Exercises

We have following relationship between different use-cases:

There is an <<include>> relationship between Similarity Test use-case and Login use-case.

There is an <<include>> relationship between Similarity Index use-case and Login use-case.

There is an <<include>> relationship between Practice Exercises use-case and Login use-case.

There is an <<include>> relationship between View Profile and Login use-case.

There is an <<include>> relationship between Edit Profile and Login use-case.

Above mentioned all are <<include>> relationships.

There is only one <<Extend>> relationship between Edit Profile Login use-case.

3.2. Detailed Use Case

Use-case name: Registration

Actor: User

Description: After providing personal information users will be able to maintain their records.

Pre-condition: User must be using the application for the first time because registration needs to be done the only first time.

Post-condition: The user should get his panel.

Use-case name: Login

Actor: User

Description: Users provide their login credentials to use this application so that he can maintain his record.

Pre-condition: To log in to a system user must be registered and should be provided a way to log in.

Post-condition: After giving the login information (if correct) user will go into the application.

Use-case name: Similarity Test

Actor: User

Description: User will always have an accent, so they need to test to which English accent their accent match. In this module, the user will speak into the application and the application will perform the test and will give the test result. In this way, the user will know to which accent, his/her accent matches.

Pre-condition: Users must use a device with a microphone.

Post-condition: After this, the user must be provided with the test result and the name of the accent with which his/her accent matches.

Use-case name: Similarity Index

Actor: User

Description: After performing the similarity test, the user must be provided with the result of the test which we call similarity index. This index will show how much a user's accent matches to a particular accent.

Pre-condition: To see the similarity index, the user must perform a similarity test.

Post-condition: Now the user must be provided a way to go to practice exercises.

Use-case name: Practice Exercises

Actor: User

Description: To improve a particular accent, the user needs to do practice by listening to the audio of a particular accent. These audios will be provided to users and the user can listen to them and practice them to improve by listening to the audio.

Pre-condition: Should take the test so that the user can choose correct practice exercise.

Post-condition: Users should be able to test again the accent.

3.3. Functional Requirements

Functional requirements are those functionalities that software must offer. It describes the basic behavior of the software. Functional requirements should include the details of functionality or operations provided by each screen. The complete workflow of the software should be made clear. Login and Registration are also functional requirements but these two will not be described here because they are conventional, we will discuss following two

3.3.1. Accent Similarity Test:

Accent Similarity is basically to provide the relevance of the user accent to any of the accents covered by this application. The user reads some passage provided by the application which application uses to process to identify the accent of the user. Based on this analysis application compares it with the accents in the dataset and the result of the analysis is given to the user in terms of similarity index.

3.3.2. Practice Exercises:

After knowing the similarity index of the accent, now users need to improve their accent so they need exercises of their desired accent. This application should provide audio exercises to the users so that they can listen to the audio and can practice the accent by listening to the passages. The application must provide the practice exercises because the main goal of the application is to train the user for a particular accent. So first application performs the similarity test to know the current accent of the user and tells the user how much his accent relates to a particular accent.

3.4. Non-functional Requirements

Non-functional requirements show that how the system should perform certain functionality. These requirements generally deal with the quality of the software. Following are the non-functional requirements of this project:

3.4.1. Performance:

Performance defines how efficient is the application. Our project will be good in a performance like it will provide test results in quick time and will manage the profile of the user accordingly each test.

3.4.2. Portability:

Portability means when using application user is not restricted to use it by sitting on a fixed place or it does not need a different system to operate. It will be on your android mobile phone which you always carry with your self.

3.4.3. Usability :

By usability of software, you mean that it should be easy to learn. The user should not feel any difficulty in using this system. In this project we have included this in our priorities that this application should be user friendly, Moreover, it should not be complex in its working mechanism.

3.4.4. Compatibility :

This application is compatible with the android operating system, it is being designed for specifically android OS. Users can use it like other android applications for login, voice recording, and practice exercises.

3.4.5. Maintainability :

The requirement of maintainability is easily achieved in this application because first, it is providing the functionality of user profile maintainability.

CHAPTER 4

DESIGN AND ARCHITECTURE

4. Design And Architecture

4.1. System Architecture

To explain the design and architecture of our project English Accent trainer, here we used Software Design Description (SDD) to precisely describe its architecture along with its design. We use three tier system architecture because our system is divided in such a way that we need three tier architecture to implement our system.

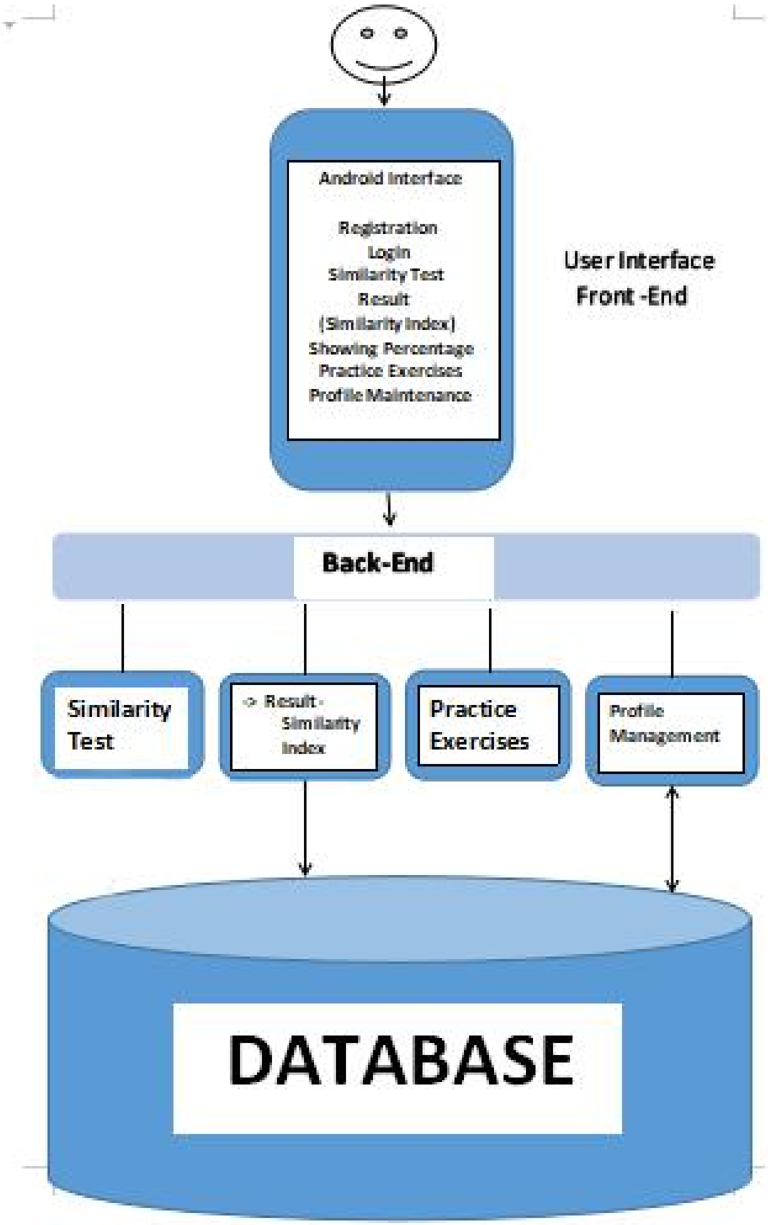


Figure 4.1 System Architecture

4.1.1. Description:

This diagram explains how the overall interaction between user and application takes place. The user interacts using an android interface which is used to control the whole process and background processing. The user inputs its voice and application in background performs similarity testing and on the other hand, we have a dataset of voices of different accents that are used for similarity index.

4.2. Data Representation

4.2.1. Context Diagram:

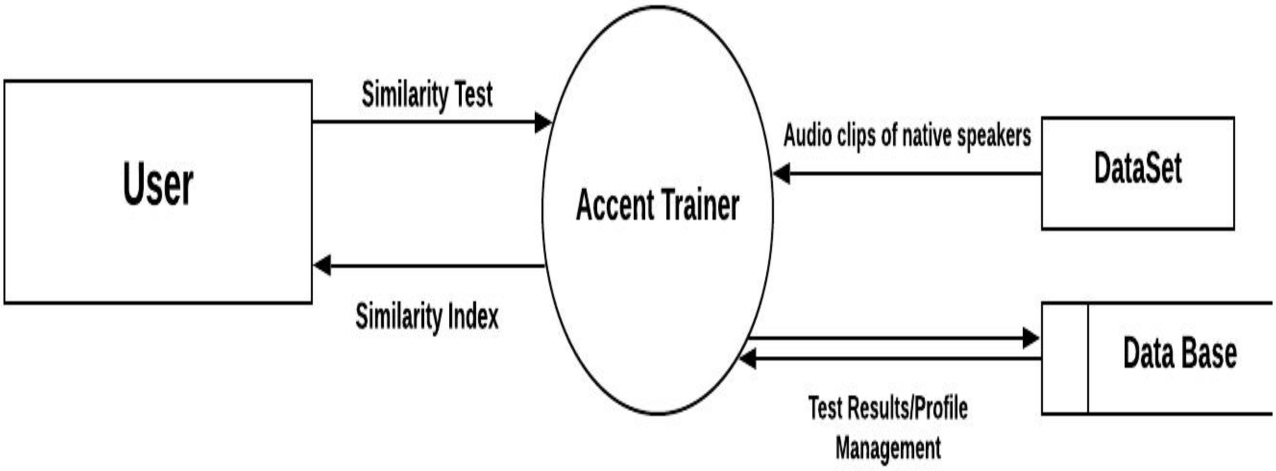


Figure 6 Context Diagram

This diagram explains the whole system as single process because context diagram describes a system in abstract view at highest level and defines its boundaries and shows its relation with external entities. Here user interacts with system for similarity test and system provides similarity index in return which is result of test. System stores user’s profile information and tests history in database to management profile and results. System has interacted with data-set to retrieve audios of native speakers to train machine learning model for accent classification.

4.2.2. Level Zero Diagram:

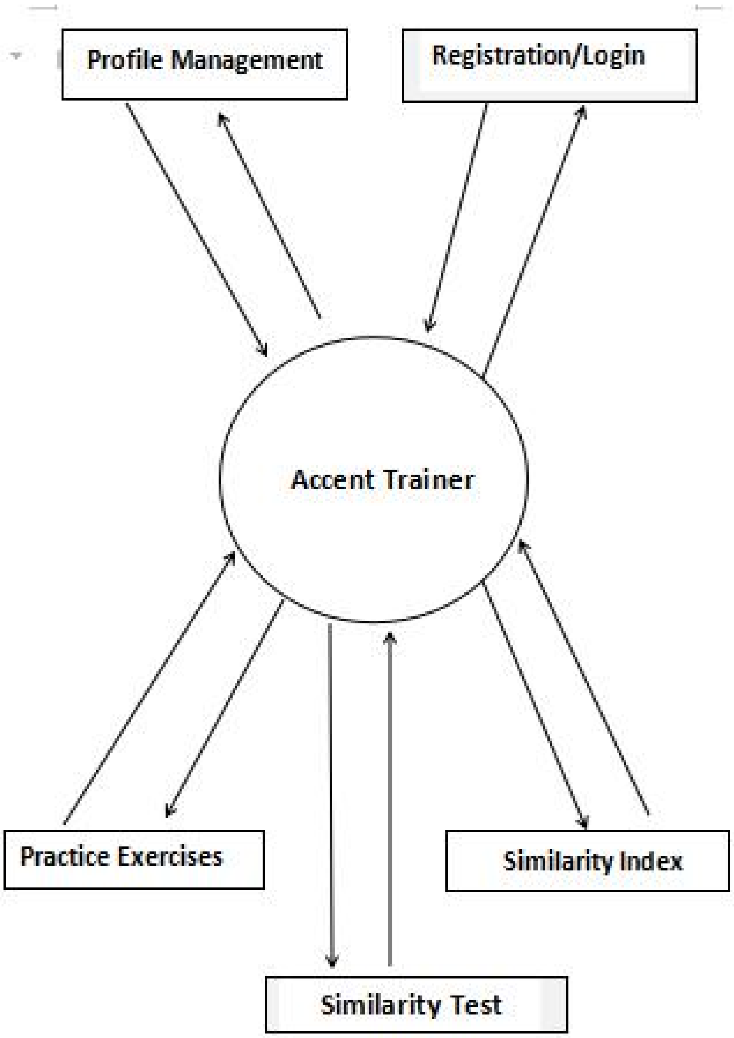


Figure 4.3 Level Zero Diagram

4.2.3. Description:

This diagram explain the working of our system in abstract view that enlist the number of modules that are present in our system and now here we will enlist the main functions of each module:

Profile management module consists of record of the user and their progress so far.

Registration/Login module consists the function of registration of a new user and authentication of a existing user.

Practice exercise module consists of the voice samples of native speakers.

Similarity Test Module perform the voice recording and testing of the accent of the user.

Similarity Index Module consists of the results of the similarity test.

4.2.4. Level 1 Diagram

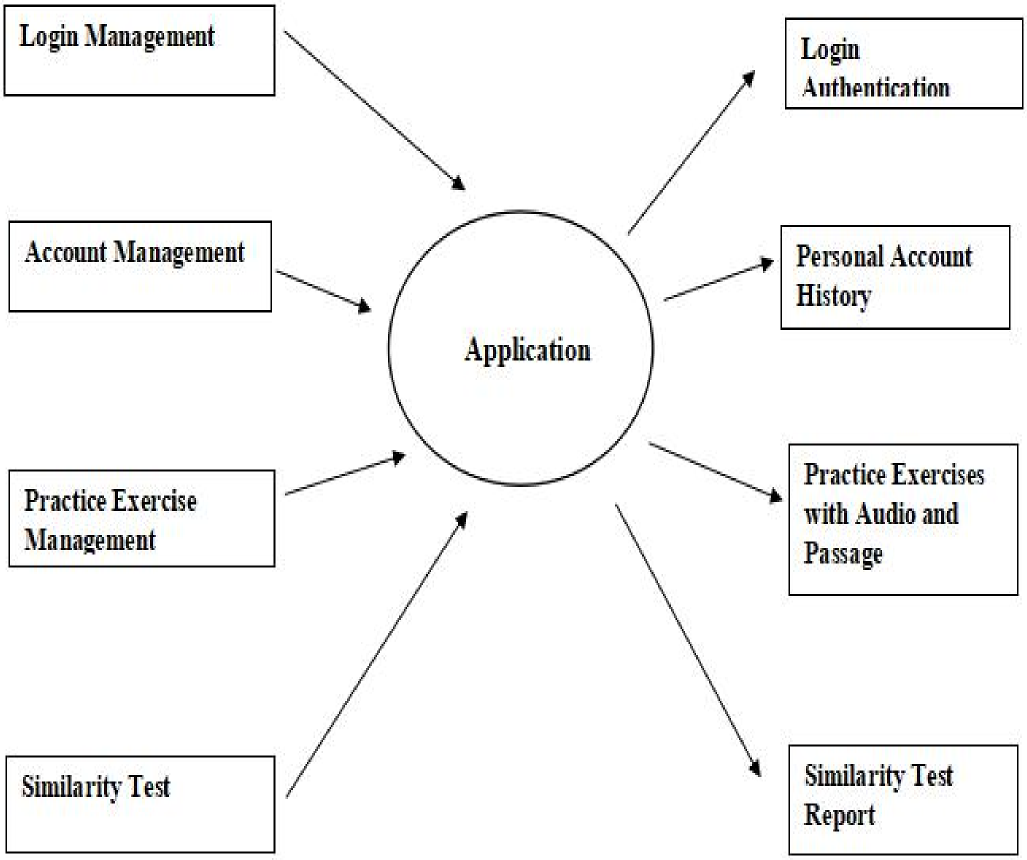


Figure 4.4 Level 1 Diagram

4.3. Process Flow Representation

4.3.1. Activity Diagram

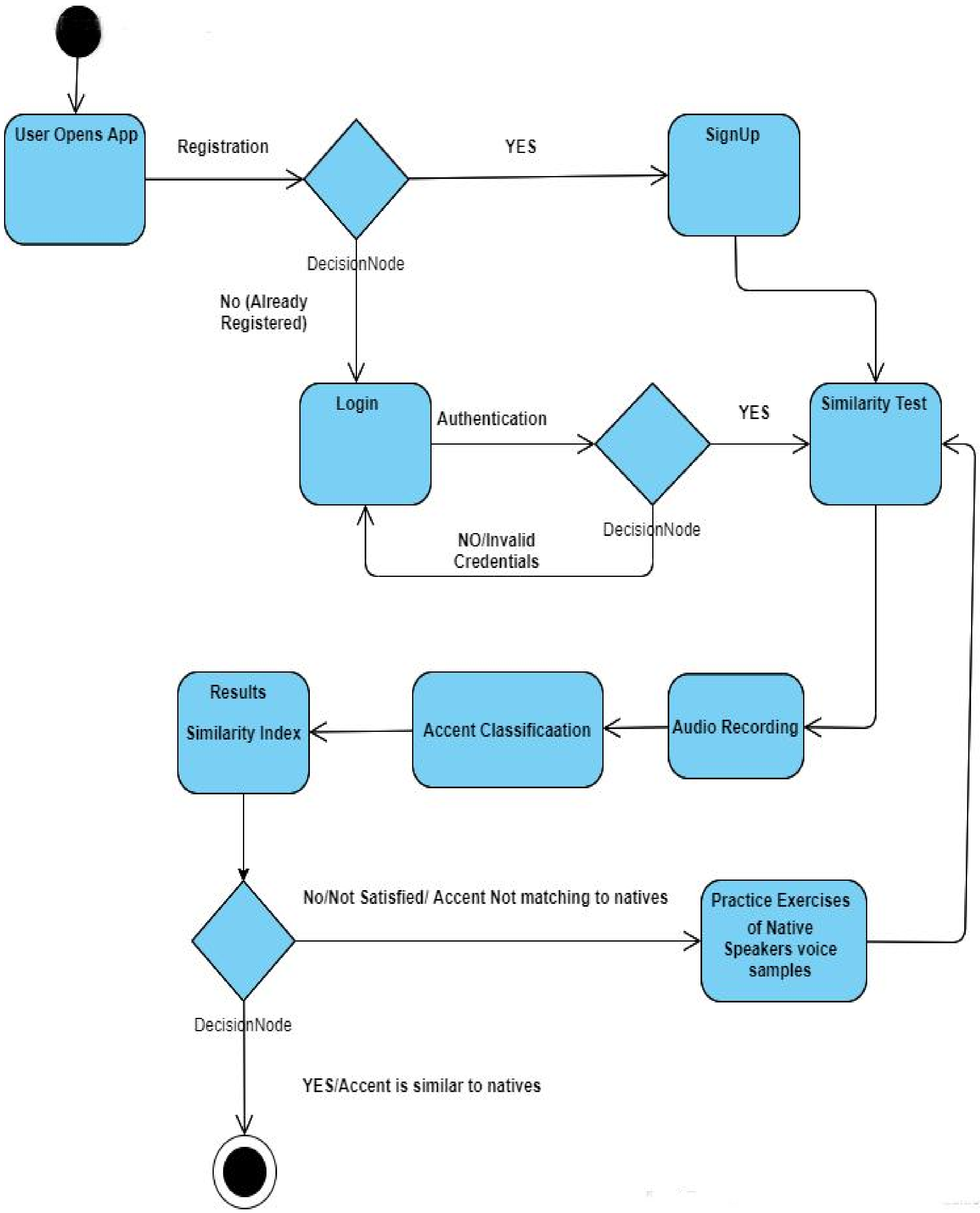


Figure 4.5 Activity Diagram

4.4. Design Model

4.4.1. Sequence Diagram

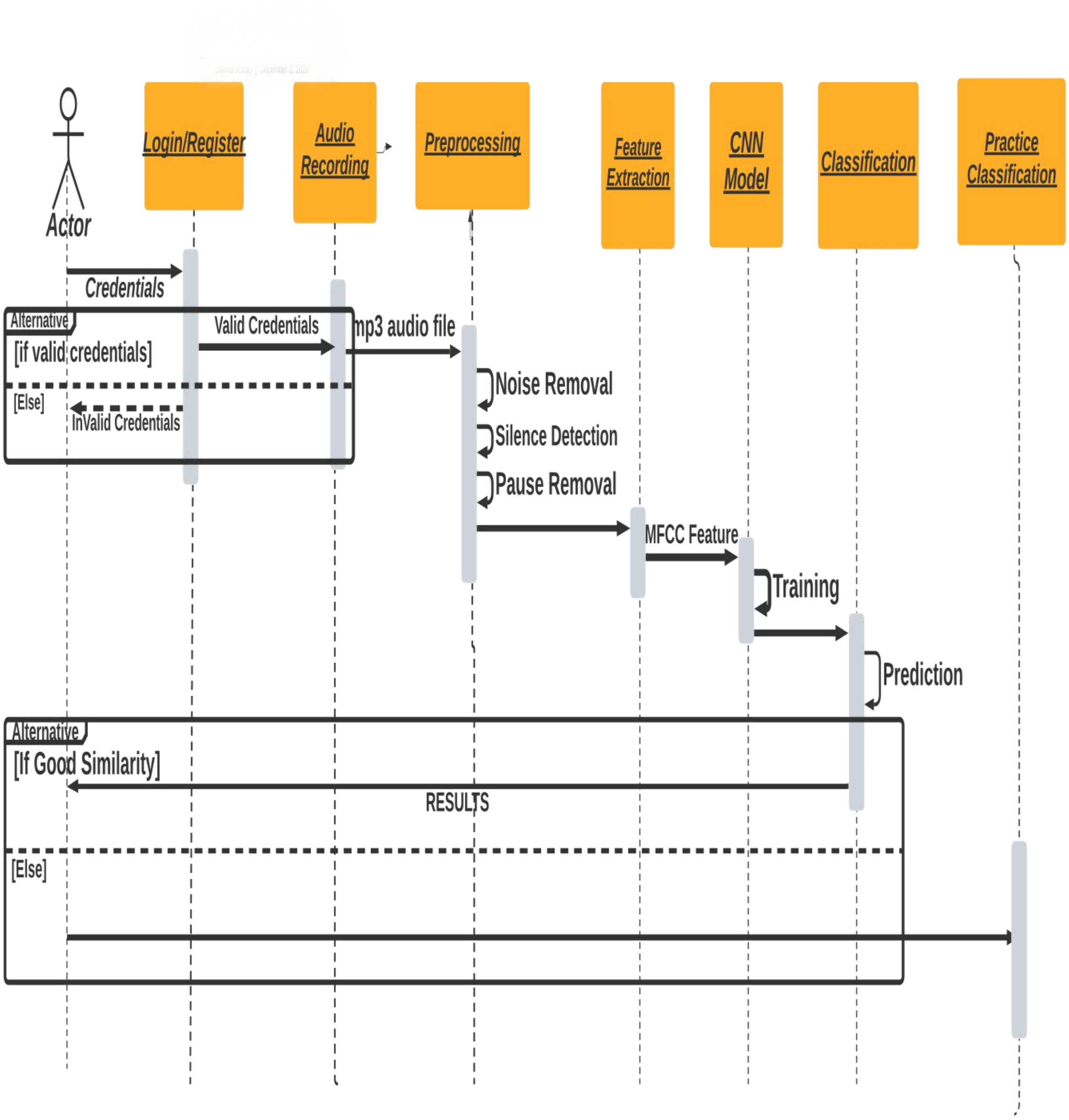


Figure 4.6 Sequence Diagram

CHAPTER 5

IMPLEMENTATION

5. Implementation

In this chapter the whole implementation of the idea of this project is explained. This is the most important portion because this portion explains that how we will convert our idea into real world project. Here we will explain the technologies that we used and moreover how we used them to implement this idea.

Here it is briefly outlined what steps did we took in abstraction and then in next part of algorithms, complete details are given.

For this project we worked with Mozilla common voice data-set. All the audio clips were in mp3 form so we took 5000 samples for each class and converted them into wave file. After converting them on wav form. We extracted MFCC features of each wav file and stored all of them in a list. After extracting MFCC features we normalized those features by taking standard deviation of each feature vector. After normalizing we split those normalized feature in training and testing data and fed this data to CNN classifier and got the identical training and testing accuracy of 88% after 200 epochs. While working with real time data we applied different checks on voice clip like noise removal from audio clip and removing long chunks of silence from audio.

5.1. Algorithms

Here we used following algorithms, here the used algorithms are listed and then each algorithm will be precisely described in context of this project.

- MFCC
- CNN
- Spectral Gatting
- Pydub

Last two algorithms are for noise removal and silence detection and removal. They are used in the pre-processing of audio. We will discuss first two in detail because they are involved in model training.

MFCC :

Whenever we talk about automatic speech recognition system, the first step that comes in place is “Feature Extraction”. Extracting the features from audio is the way of identifying the components of a audio signal that are helpful in identifying content related to linguistic context and with help of it we can discard all unnecessary things which have data like background noise, emotion etc.

When studying about speech then the most important thing is that “ How the speech (to be precise sounds) is generated by human?”

The answer to which is the sound that is produced by human is filtered out and depends on shape of vocal tract of that human which includes teeth and tongue etc and through this shape it is determined what sound will come out. Now if we are able recognize the shape of vocal tract and we are accurate in determining it then we can accurately get the representation of the “ phoneme” that is generated by the vocal tract. Now the question arises that how we can express or represent this vocal tract so that we can work on it?

The answer is that “ Vocal tract has a shape and it manifests its shape in the package of short time power spectrum” and now the envelope is represented by MFCC. So that’s why these MFCC features are used in this project.

Mel Frequency Cepstral Coefficents also commonly known as MFCCs are those features which are widely used for automatic recognition of speech and speaker. Before MFFC feature, LPCs feature were used for Automatic Speech Recognition and were used with HMM classifiers.

Following is the high level intro to the implementation steps :

- First of all this audio signal is framed into short frames.
- Now calculate periodogram estimate of the power spectrum for every frame.
- Now first apply the mel filterbank to the power spectra.
- After above step sum the energy in each filter.
- Now take the logarithm of all filterbank energies.
- Now take DCT of the log filterbank of energies.

- Now keep the DCT coefficients 2 - 13 and others will be discarded.

Now each step will be described individually.

When we talk about audio signal it is continuously changing, so to simplify it we consider that on short time scales, this audio signal does not change much so this is the reason the signals are framed on 20 - 40 ms frames. If we make it more shorter then we are unable to get a reliable spectral estimate and if its longer than signal changes throughout the frame.

The next step is motivated by an organ of human hearing system i.e cochlea.Our ear contains this organ which vibrates at different spots when sounds of different frequency comes. Now on the basis of the portion that vibrates in cochlea different nerves fires information to brain that following frequencies are present. Here this task is done by the “Periodogram Estimate” and identifies what frequencies are present in the frame. The information that we get from periodogram estimate still contains extra details that we don’t need for Automatic Speech Recognition. Cochlea can not differentiate between the frequencies which are closely spaced. So for this reason we take clumps of periodogram bins and by this we get an idea how much energy exists in various frequency region by getting the sum of the clumps and this task is performed by Mel Filterbank. After getting the filterbank energies, logarithm of them is calculated. This is motivated by human hearing ability: human ear can’t hear loudness linearly (on linear scale). In order to double the percieved volume of the sound , 8 times more energy is put into it. This means that large variations in energy may not sound all that different if the sound is loud to begin with. This compression operation makes our features match more closely what humans actually hear.

The final step in following calculations is DCT computation of the log filterbank energies. We have following two reasons to perform this step. Filterbanks are all overlapping, the filterbank energies that we have calculated are quite correlated with one another. The DCT actually decorrelates these energies which means in order to model the features diagonal convariance matrices ca be used e.g. a HMM classifier. While performing this keep in mind that only 12 of the 26 DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so we get a small improvement by dropping them.

What is the Mel scale?

The Mel scale relates actual measured frequency of a pure tone to its perceived frequency, or pitch. Humans are much better at judging tiny changes in pitch at lower frequencies than they are at higher frequencies. To match features more near to what ears hear we incorporate this scale.

Conversion to Mel scale from frequency, use following formula:

$$M(f) = 1125 \ln(1 + f/700) \qquad (1)$$

Conversion to frequency from mEL BACK:

$$M^{-1}(m) = 700(\exp(m/1125) - 1) \qquad (2)$$

Implementation Steps:
Assume that we sampled a speech signal at 16kHz.

1. First of all frame the signals into 20-40 ms frames. 25ms frame is standard. Now calculate frame length for a 16kHz signal is 0.025*16000 = 400 samples. Frame step is usually like round about 10ms (160 samples), due to which it allows some overlap to frames. Our first 400 sample frame will start from sample 0, the next 400 sample frame will start from sample 160 etc. until we reach the end of the speech file. Pad the file with zeros if the speech file does not divide into an even number of frames.

Now MFCC coefficients are extracted for every frame.

It is known as Periodogram estimate of the power spectrum. Now we will square the result after taking absolute vaue of complex fourier transform. Generally a 512 point FFT is performed and only the first 257 coefficents are kept.

3. Now the Mel-spaced filterbank is computed is a set of 20-40 (Standard = 26) triangular filters that will be applied to the periodogram power spectral estimate from above step. Here this filterbank will come in

form of 26 vectors of length 257 (FFT settings are assumed fom step 2). The vectors are mostly zeros, but it will be non-zero for a section of tspectrum. Each filterbank is multiplied with the power spectrum to calculate filterbank energies, then we will add up coefficents. 26 numbers are left that gives an indication of that how much energy was in each filterbank, once it is performed. See below for a detailed explanation to calculate the filterbanks. This plot will clear up the things:

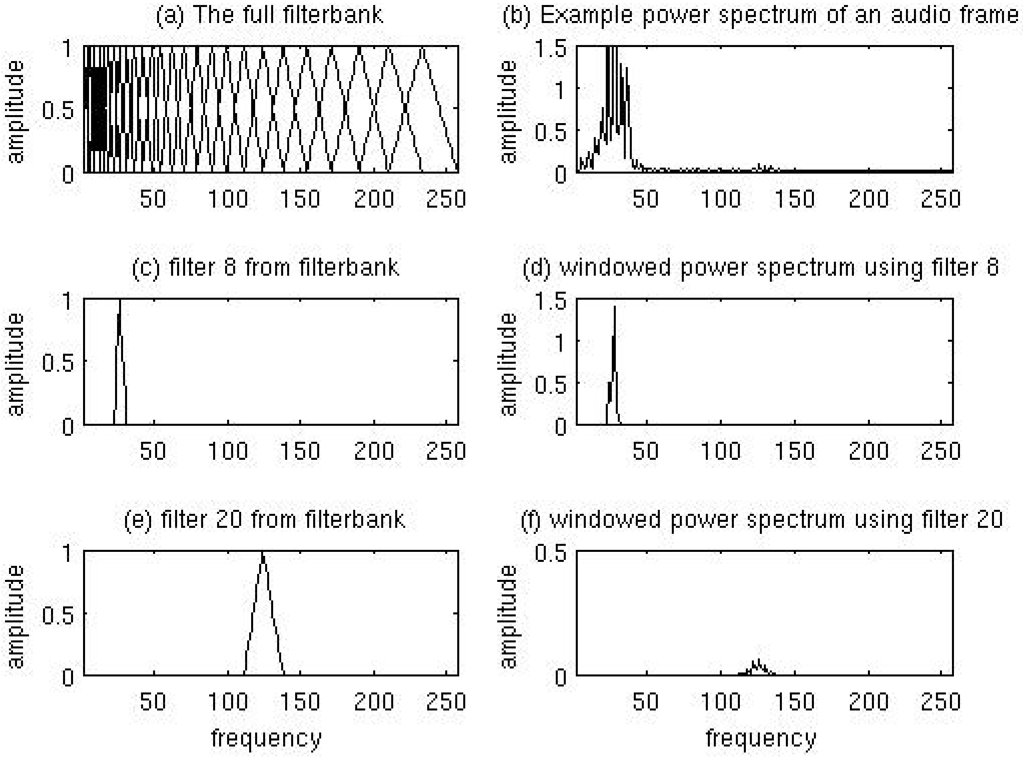


Figure 5.1 Plot of Mel Filterbank and windowed power spectrum

4. We will take the log of each of the 26 energies from step 3. In result of which we will get 26 log filterbank energies.

5. Now take the Discrete Cosine Transform (DCT) of these 26 log filterbank energies, in result of which we get 26 cepstral coefficents. Out of these 26 cofficients only lower 12-13 are kept for ASR.

So we get our Mel Frequency Cepstral Cofficients which are the resulting features (12 numbers for each frame).

Computing the Mel filterbank

In actual 26-40 filterbanks should be used but here we will use only 10 filterbanks because it will be easy to display. Use lower and upper frequency so you can get the filterbanks that are shown in figure 1(a). Good ranges for the values of lower and upper frequency are 300Hz and 8000Hz respectively. Our upper frequency will be limited to 4000Hz if speech is sampled at 8000Hz. Then follow these steps:

Rephrasing:

By equation 1, upper and lower frequencies are converted to Mels. Here in this case 300Hz = 401.25 Mels and 8000Hz = 2834.99 Mels.

Here we will do 10 filterbanks for this example and we will need 12 points for it. So here we will need 10 points additionally which will be spaced linearly in between 401.25 and 2834.99. This will result in:

m(i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74,

1949.99, 2171.24, 2392.49, 2613.74, 2834.99

With the help of equation 2, we will convert them back to Hertz:

h(i) = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33,

3261.62, 4122.63, 5170.76, 6446.70, 8000

The start and end points are at the frequencies we wanted.

In order to put filters at the exact points calculated above we don't require the frequency resolution, so we will round those frequencies to their nearest FFT bin. The accuracy of features is not affected by this process process. We should know the FFT size and the sample rate in order to covert the frequencies into fft bin numbers,

f(i) = floor((nfft+1)*h(i)/samplerate)

We get following sequence in result of it:

f(i) = 9, 16, 25, 35, 47, 63, 81, 104, 132, 165, 206, 256

The final filterbank finishes at bin 256 and it corresponds to 8kHz with a 512 point FFT size.

From here we will create our filterbanks. The starting point for first filterbank will be at the first point, at the second point it will reach its peak, then at 3rd point it will return to zero. At the second point, the second filterbank will start and at 3rd point it will reach its max, then at 4th it will be zero etc. To calculate these, use following formula:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

here M represents the number of filters that we want, and the list of M+2 Mel-spaced frequencies is represented by $f()$.

Following is the plot of all the 10 filters overlayed on each other is:

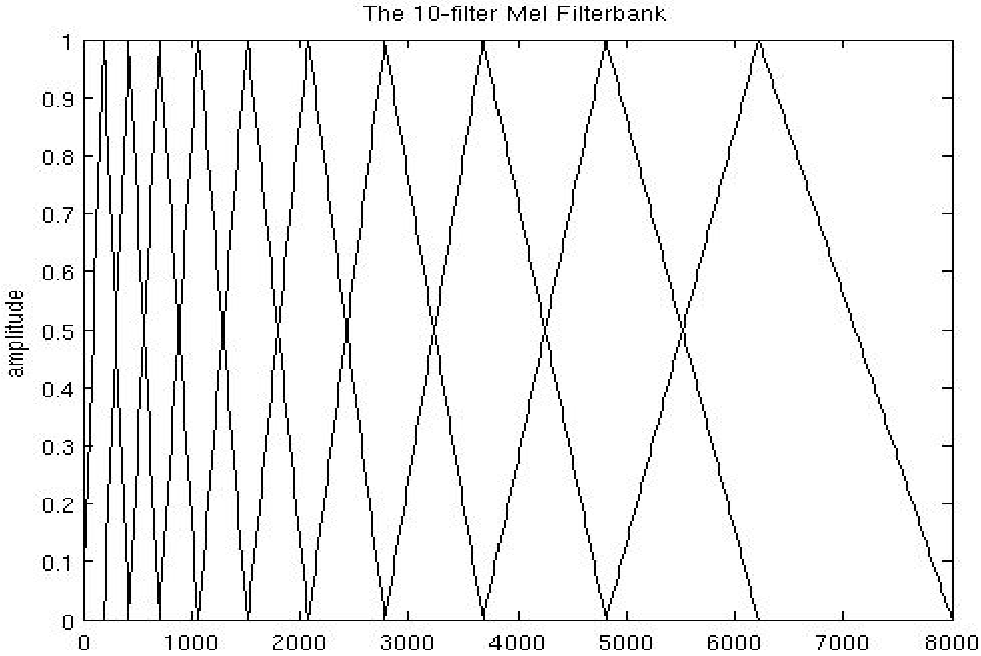


Figure 5.2 Filters Overlayed

A Mel-filterbank containing 10 filters. This starting point of this filterbank starts at 0Hz and end of this filterbank is at 8000Hz.

Deltas and Delta-Deltas

The second known term that is used for it is differential and acceleration coefficients. Actually the thing that is described by MFCC feature vector is the power spectral envelope of a frame, but the speech would also have information in other dynamics i.e.when we talk about MFCC coefficients over time then what are the trajectories. So by calculating the MFCC trajectories and adding or appending them to the original feature vector helps in increasing ASR performance.

Delta coffiecient is calculated by following formula

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

here d_t is delta coefficient,which is from frame t which is calculated in terms of the static coefficients c_{t+N} to c_{t-N}. N has a value of 2. Similarly Delta-Delta (Acceleration) coefficients are calculated, the only difference is that they are not calculated from the static coefficients but from deltas.

CNN:

The   MFFC features that we extracted were then used to train CNN.

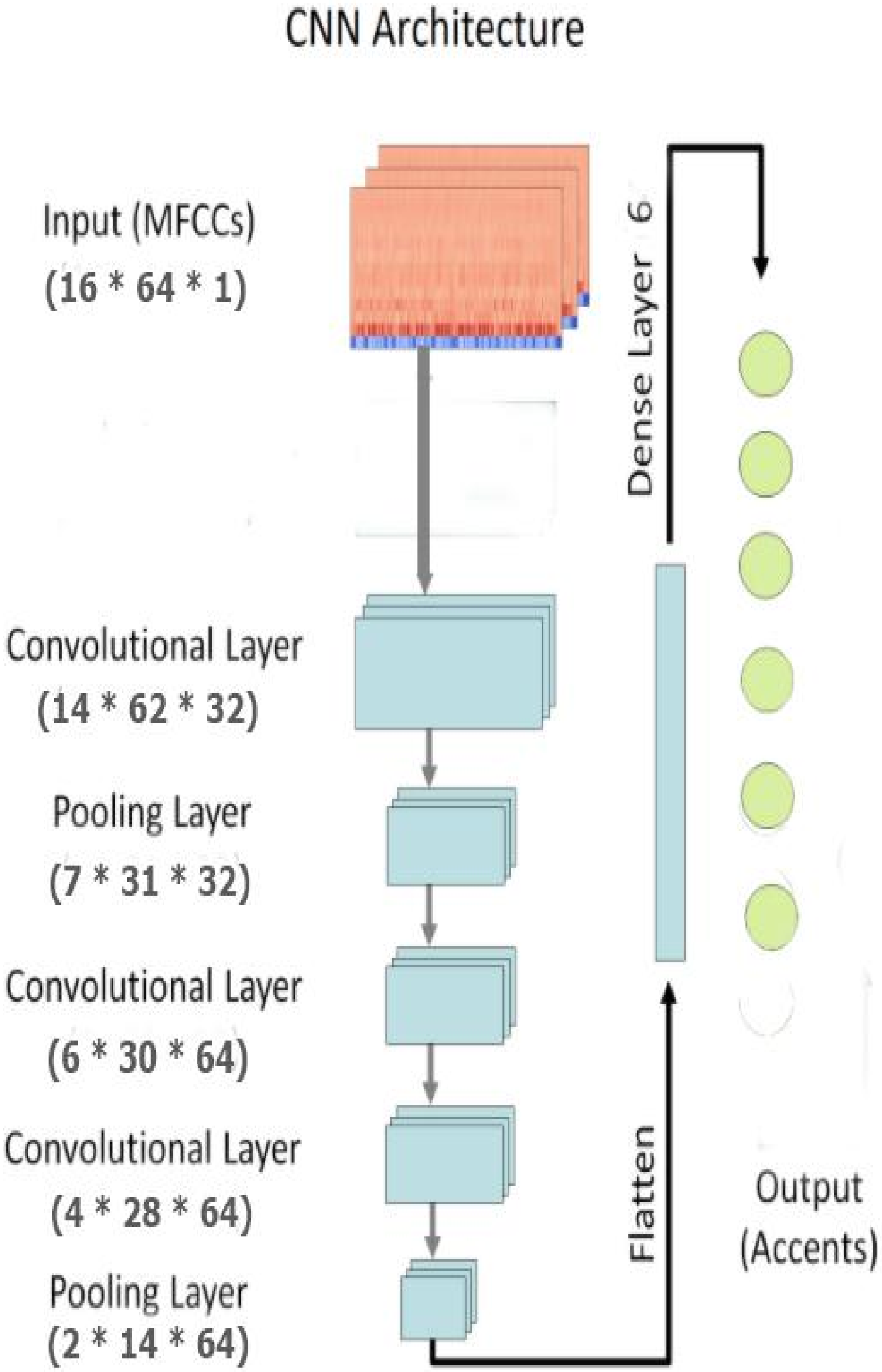


Figure 5.3 CNN Architecture

Our accuracy started from 82% and after using 150 epochs, training accuracy reaches to 90 % and testing accuracy reaches to 88%.

Speech Accent Archive Data

Test Accuracy on Holdout Data: 73%

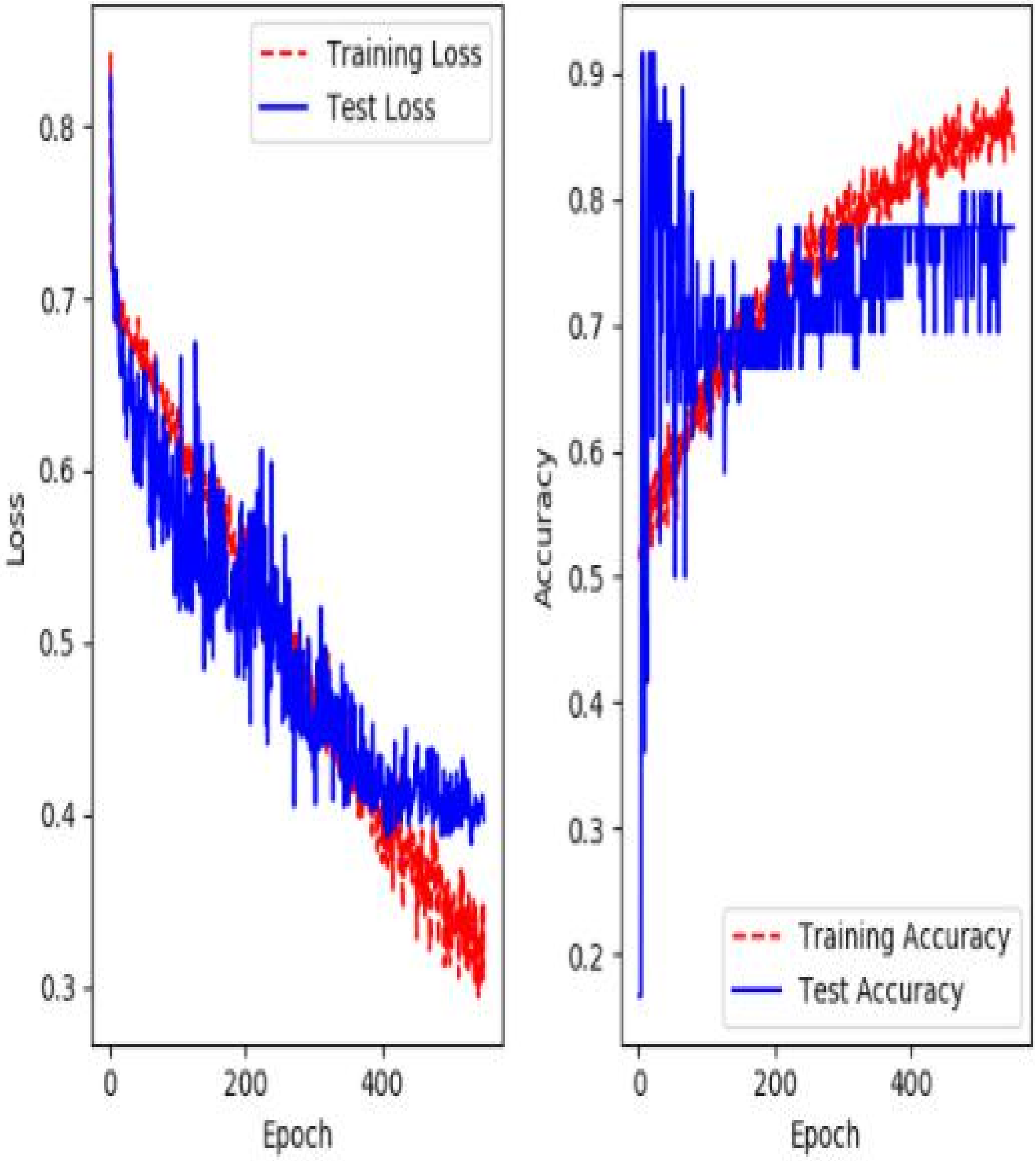


Figure 5.4 Three Accent Classification

Mozilla Voice Data:

Test Accuracy on Holdout Data: 90%

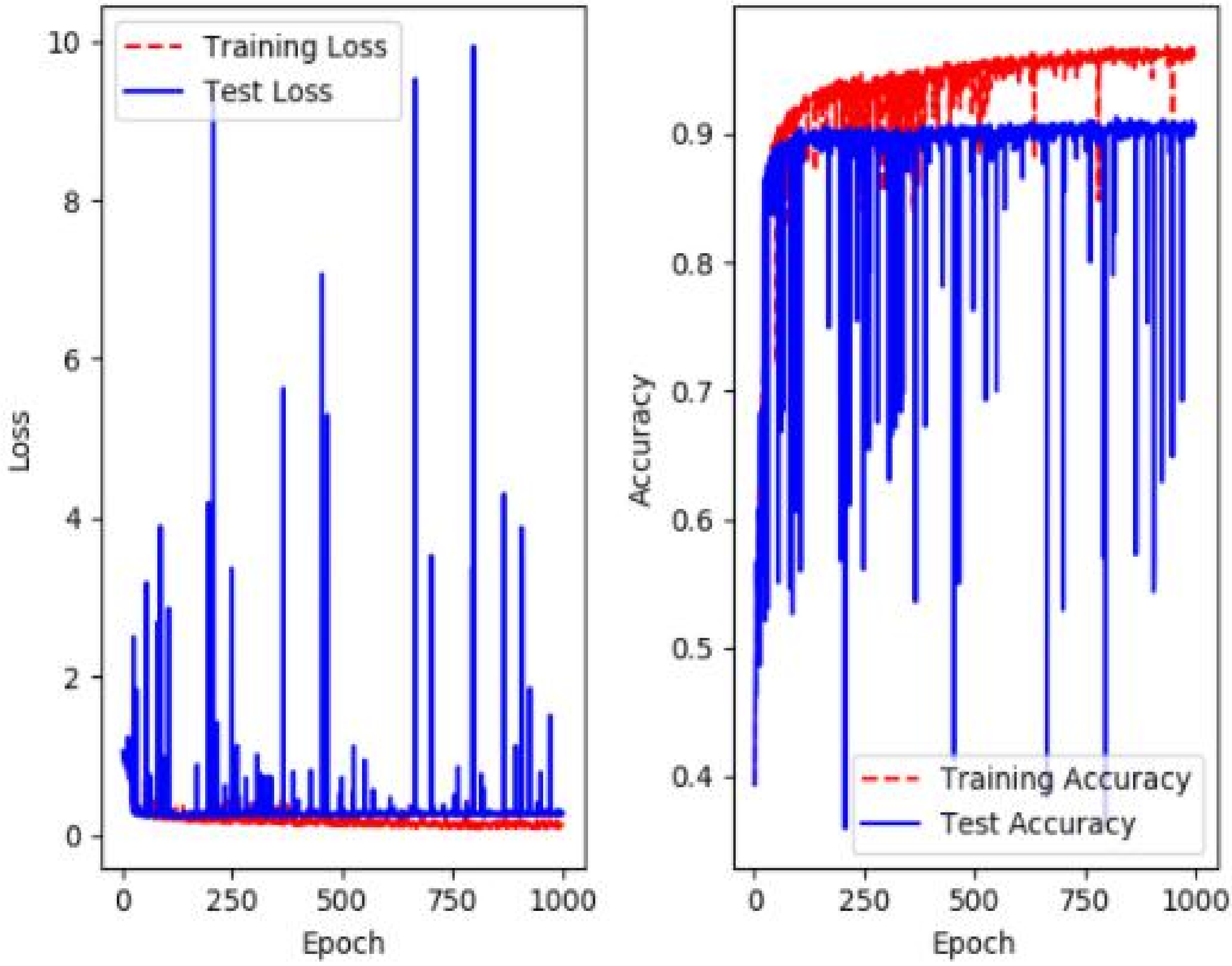
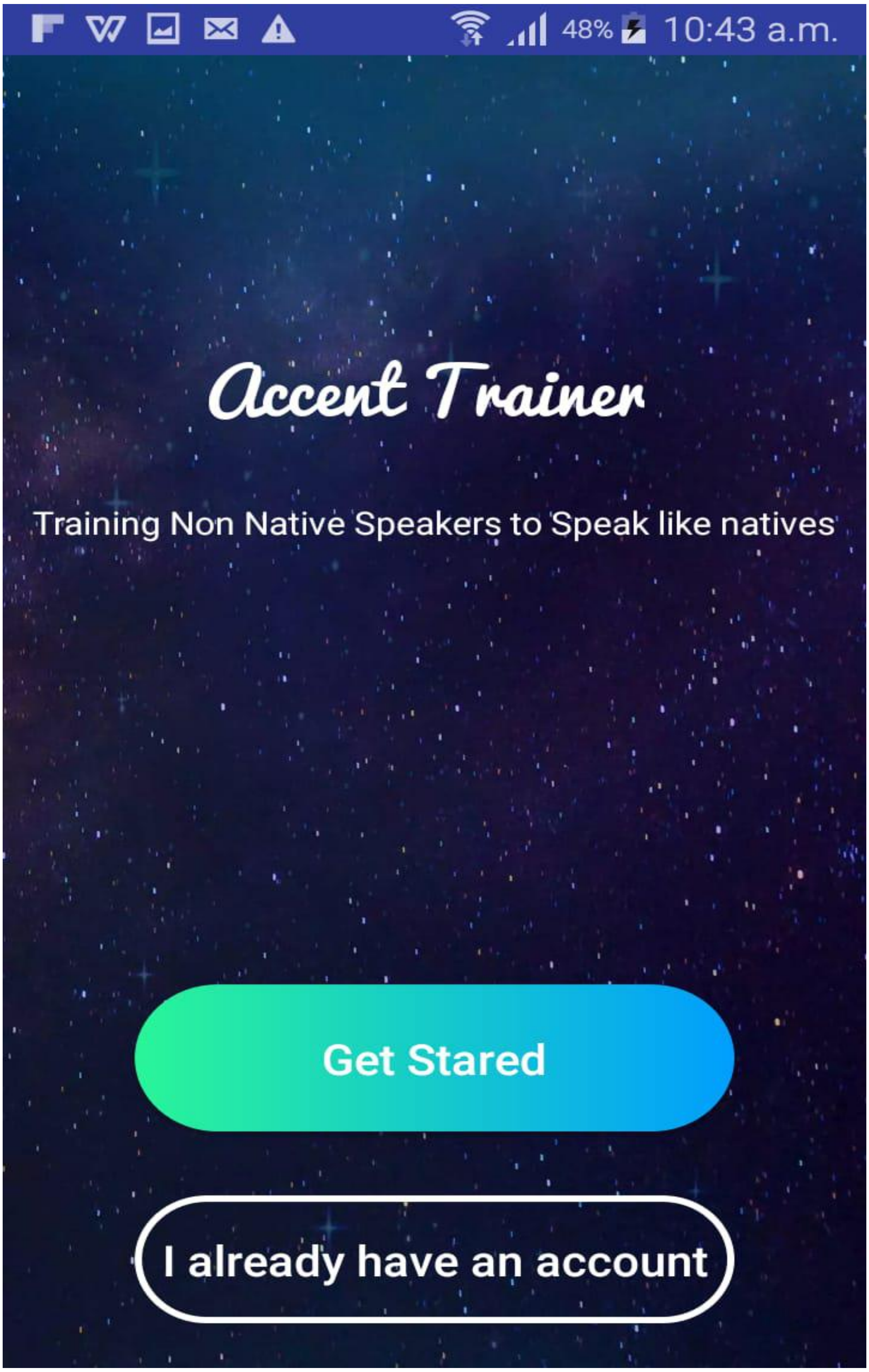


Figure 5.5 Mozilla Voice Data

5.2. External APIs:

We haven't used any external API. We have used "Flask" which is a web freamework and is for python. It not just manage HTTP requests but also render template but we have concern only with request, we used it to make an API for communication of our backend and front-end. As our back-end is in python so we don't need any external API. We achieve this functionality by Flask.

5.3. Interface



Accent Trainer

Training Non Native Speakers to Speak like natives

Figure 5.6 Start Page

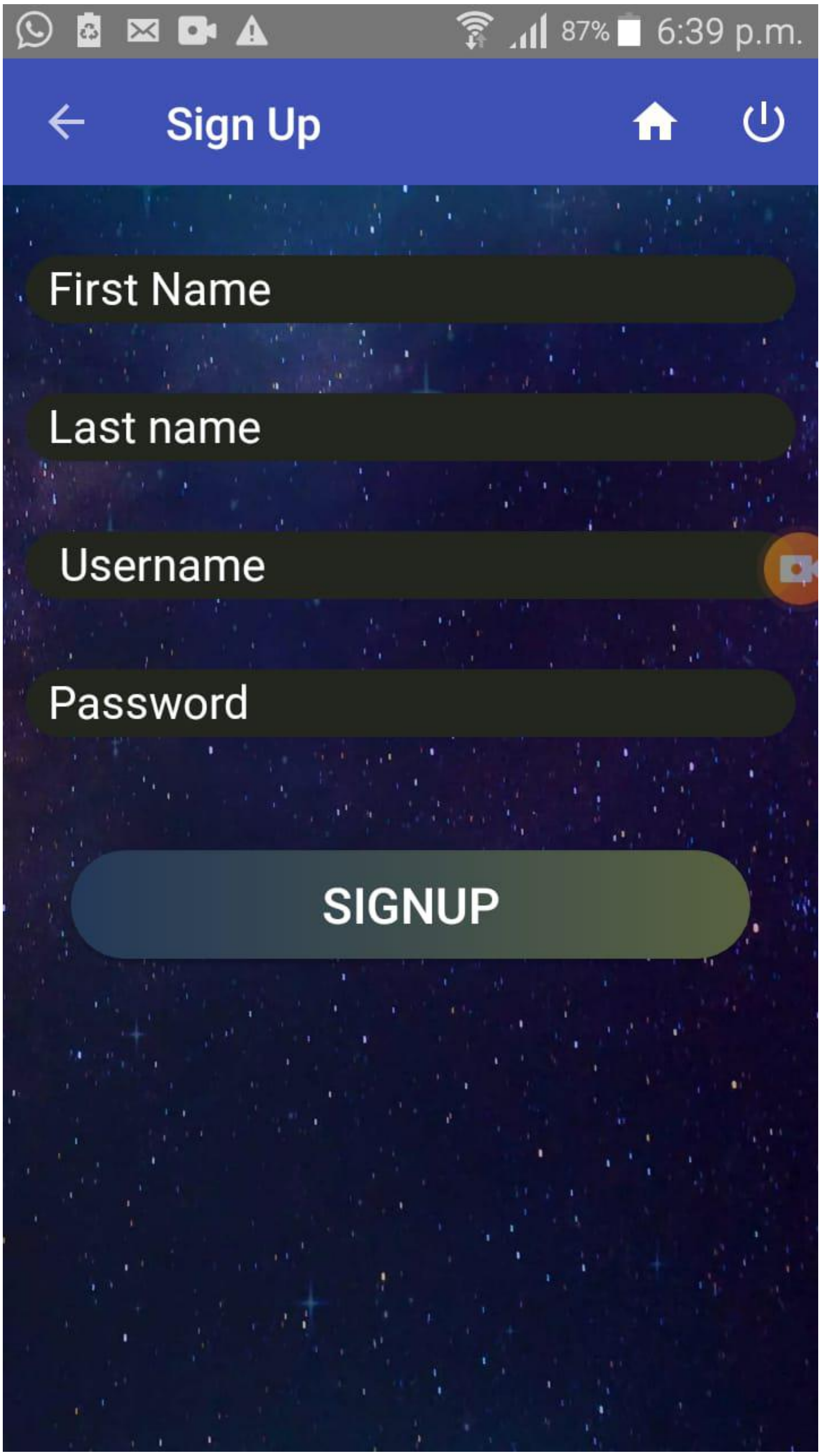


Figure 5.7 Sign up Page

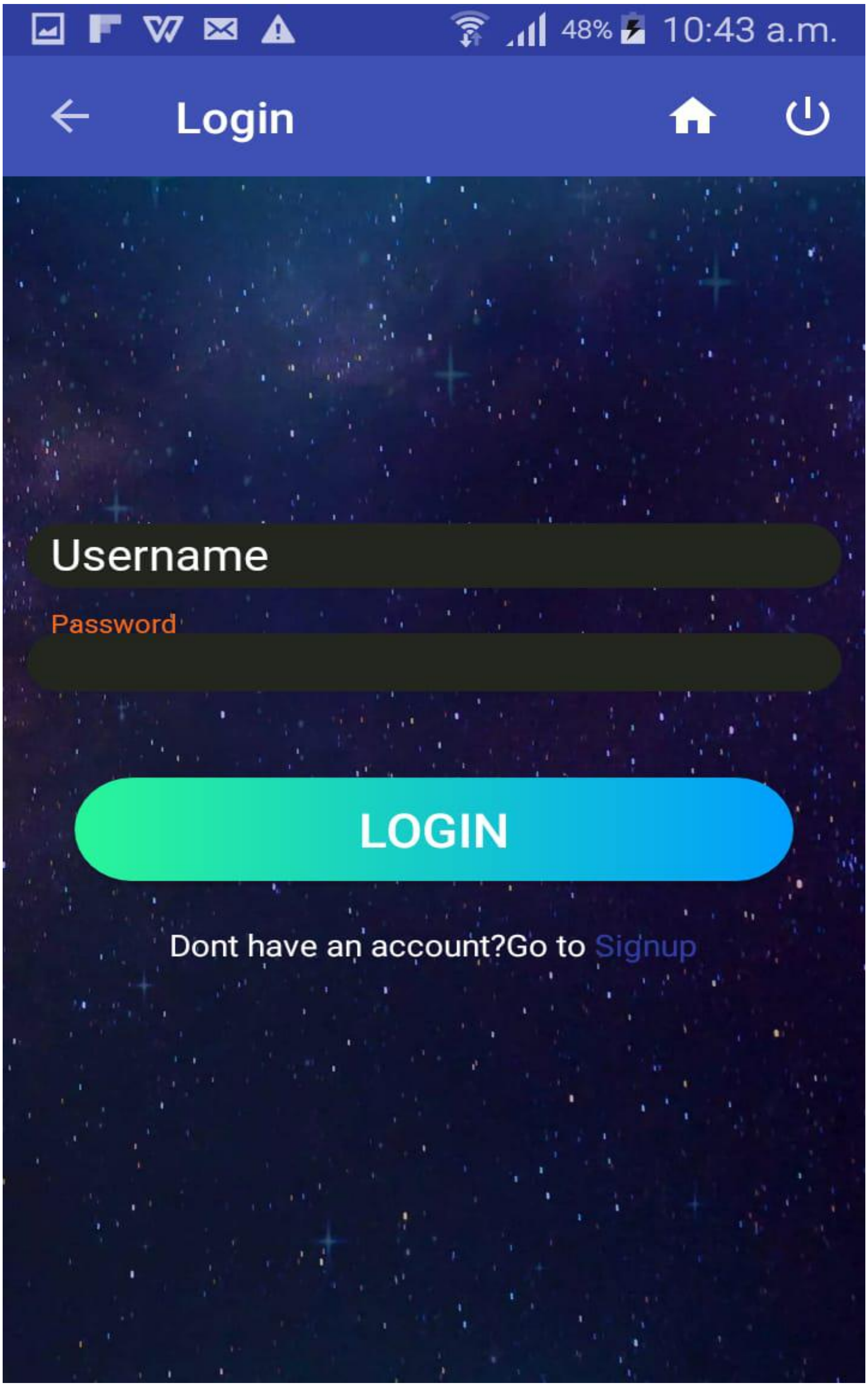
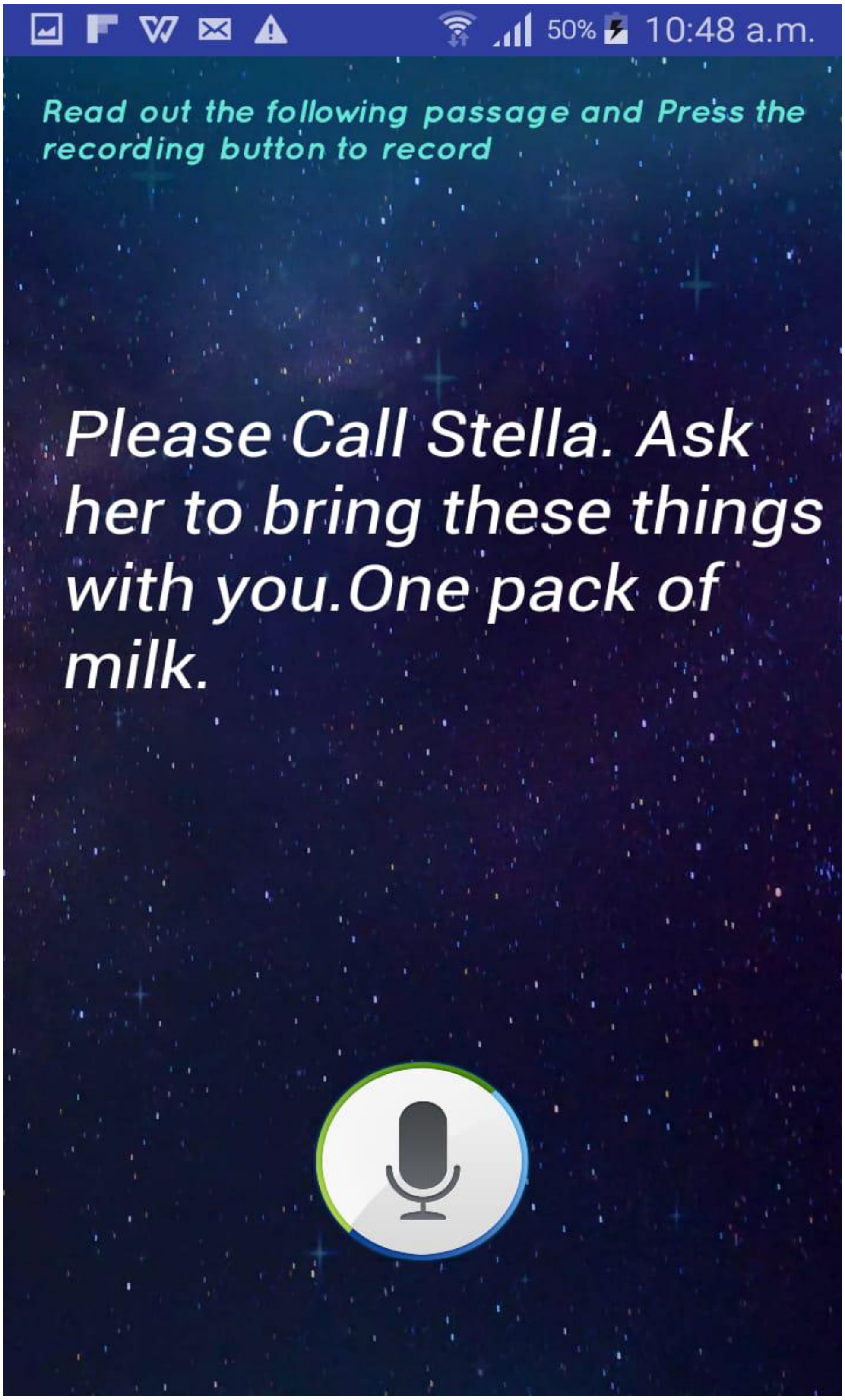


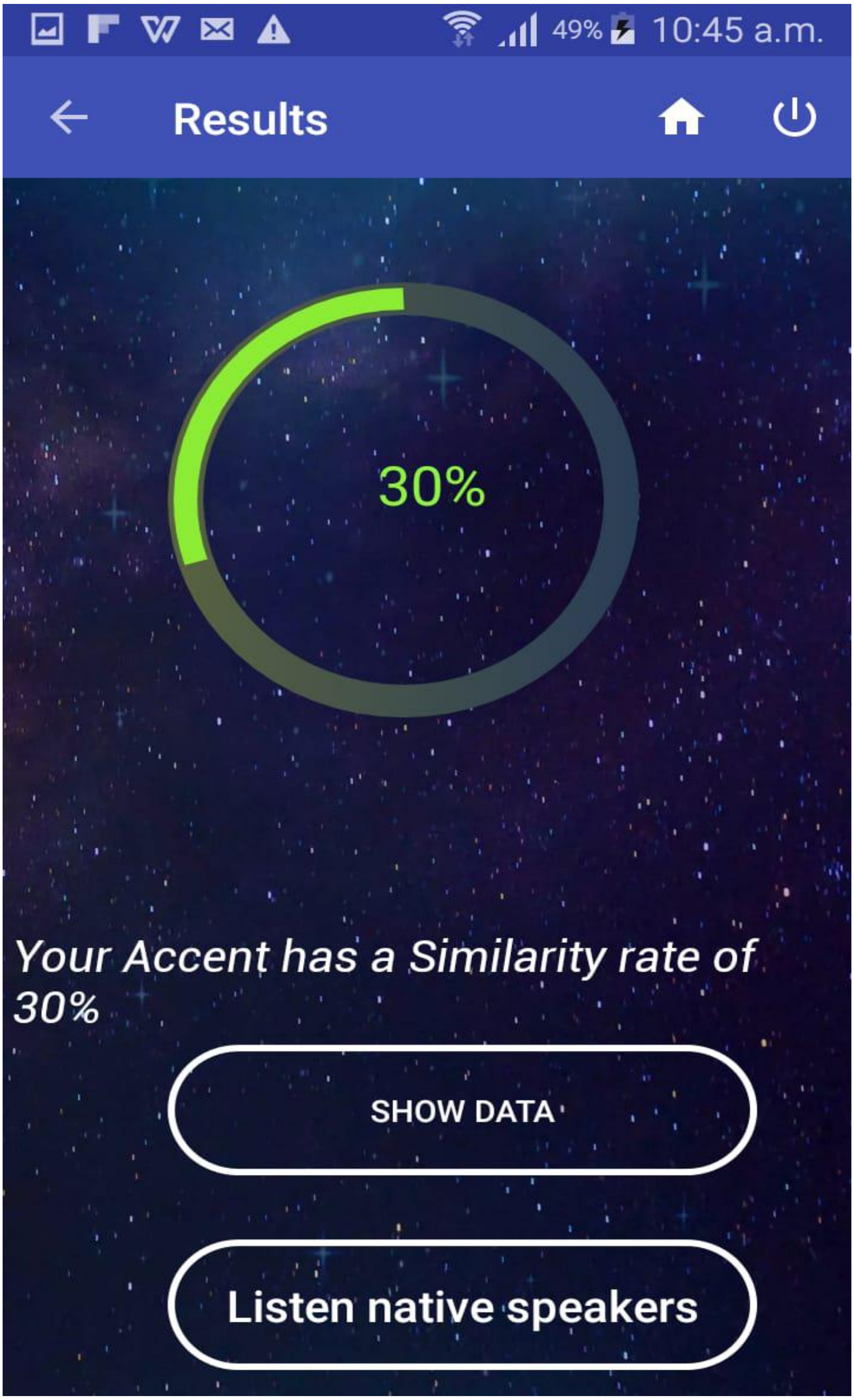
Figure 5.8 Login



Read out the following passage and Press the recording button to record

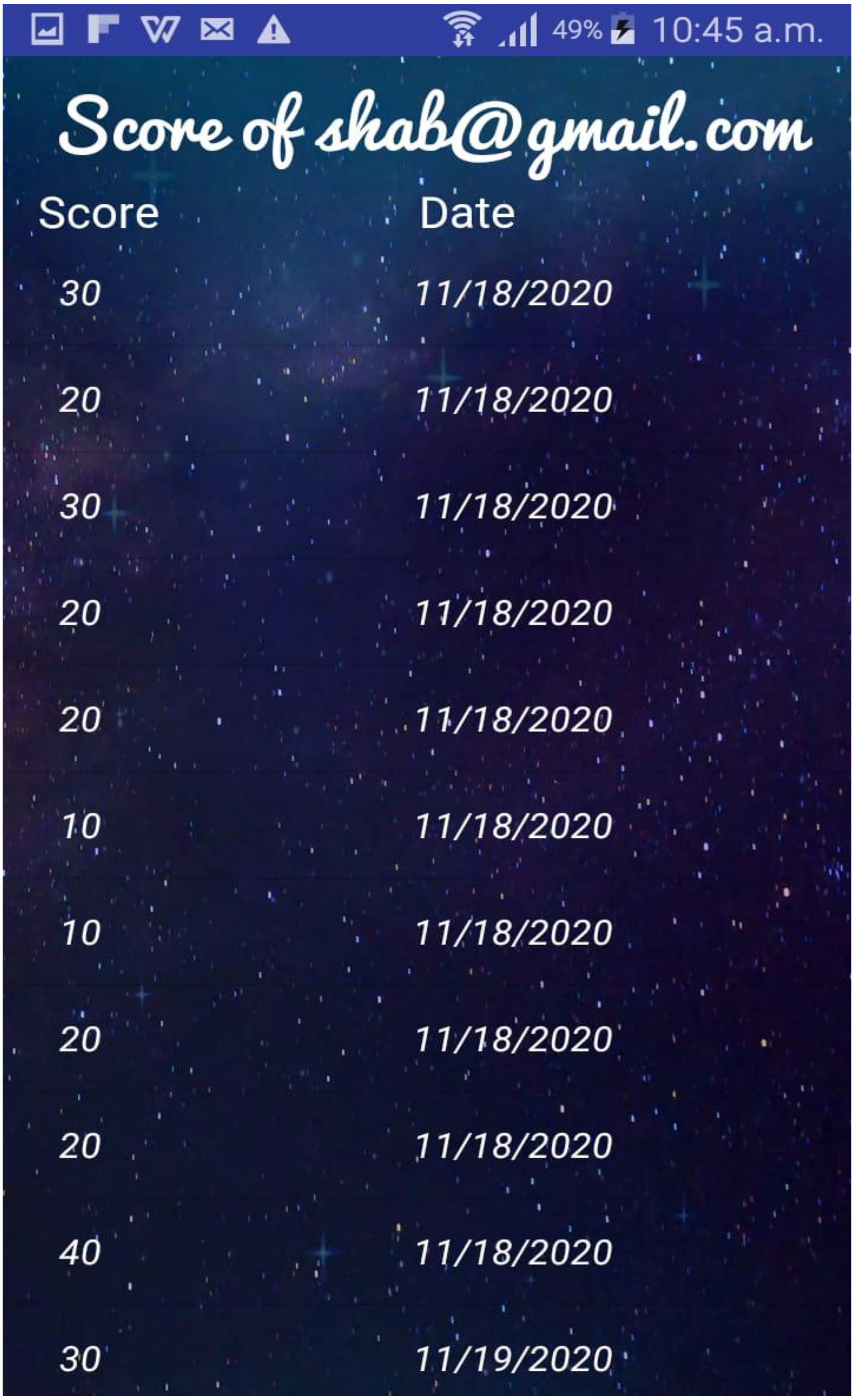
Please Call Stella. Ask her to bring these things with you.One pack of milk.

Figure 5.9 Voice Recorder



Your Accent has a Similarity rate of 30%

Figure 5.10 Results



Score of shab@gmail.com

Score	Date
30	11/18/2020
20	11/18/2020
30	11/18/2020
20	11/18/2020
20	11/18/2020
10	11/18/2020
10	11/18/2020
20	11/18/2020
20	11/18/2020
40	11/18/2020
30	11/19/2020

Figure 5.11 Profile

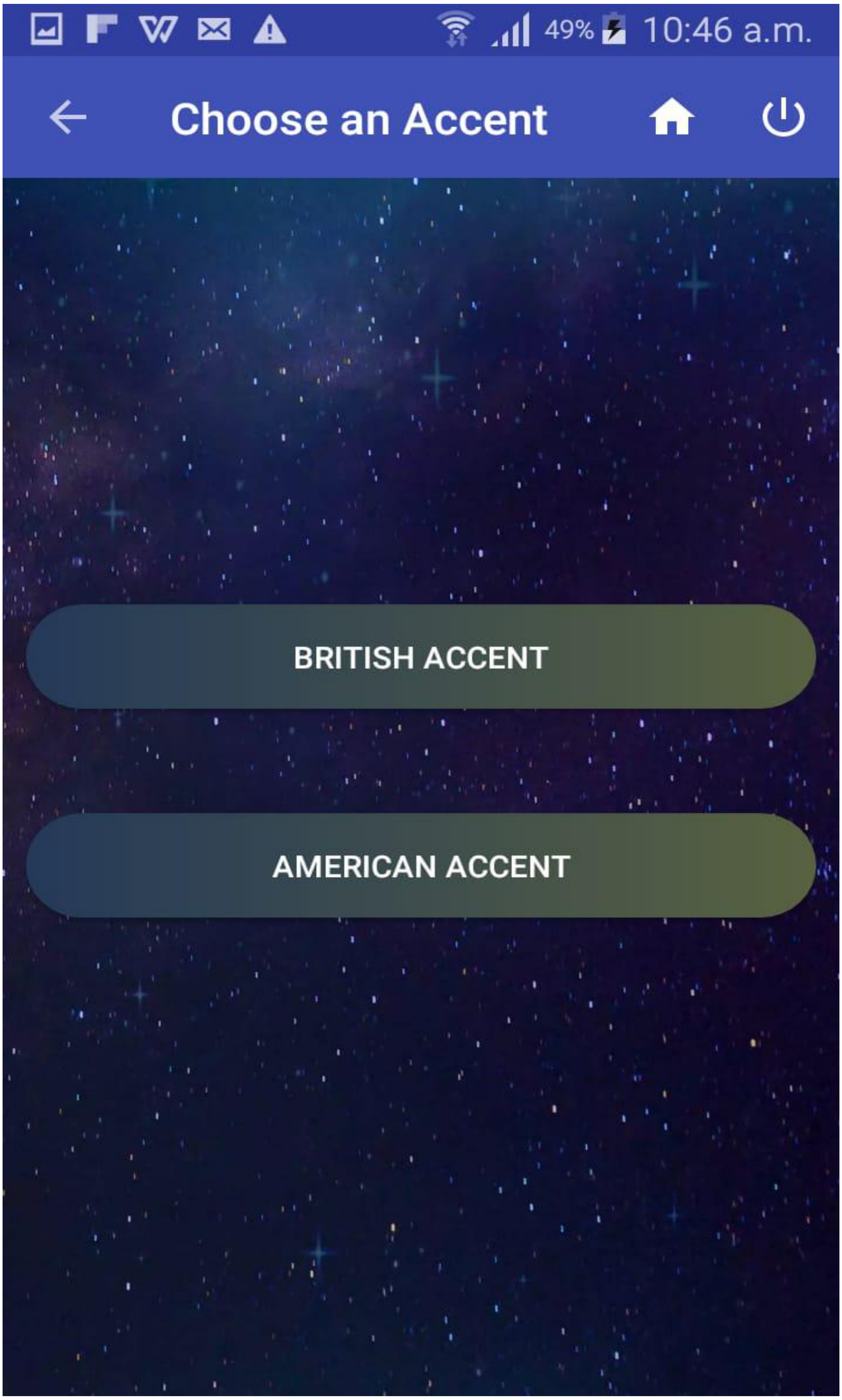
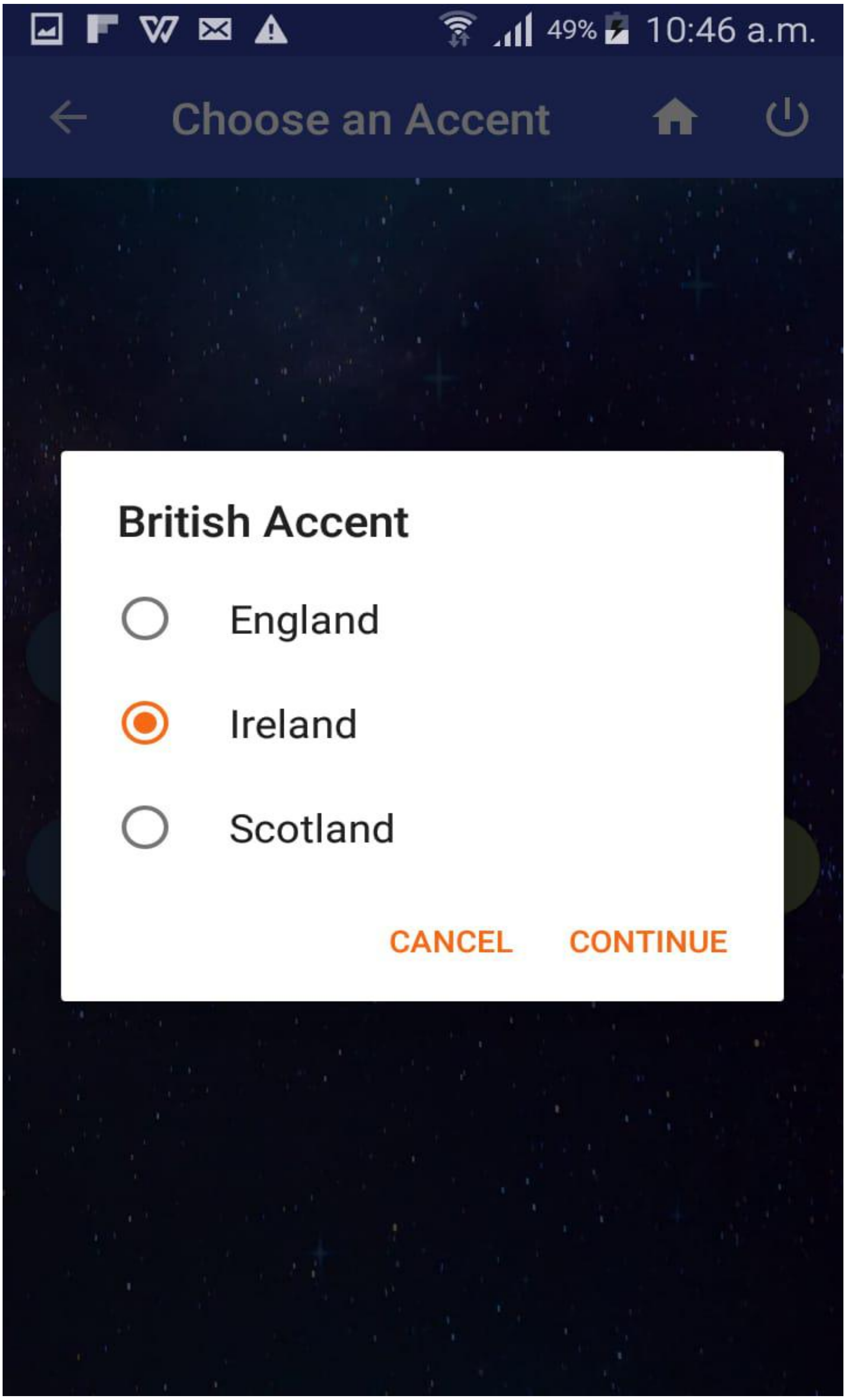


Figure 5.12 Accent Selection



British Accent

○ England

● Ireland

○ Scotland

Figure 5.13 Accent Selection

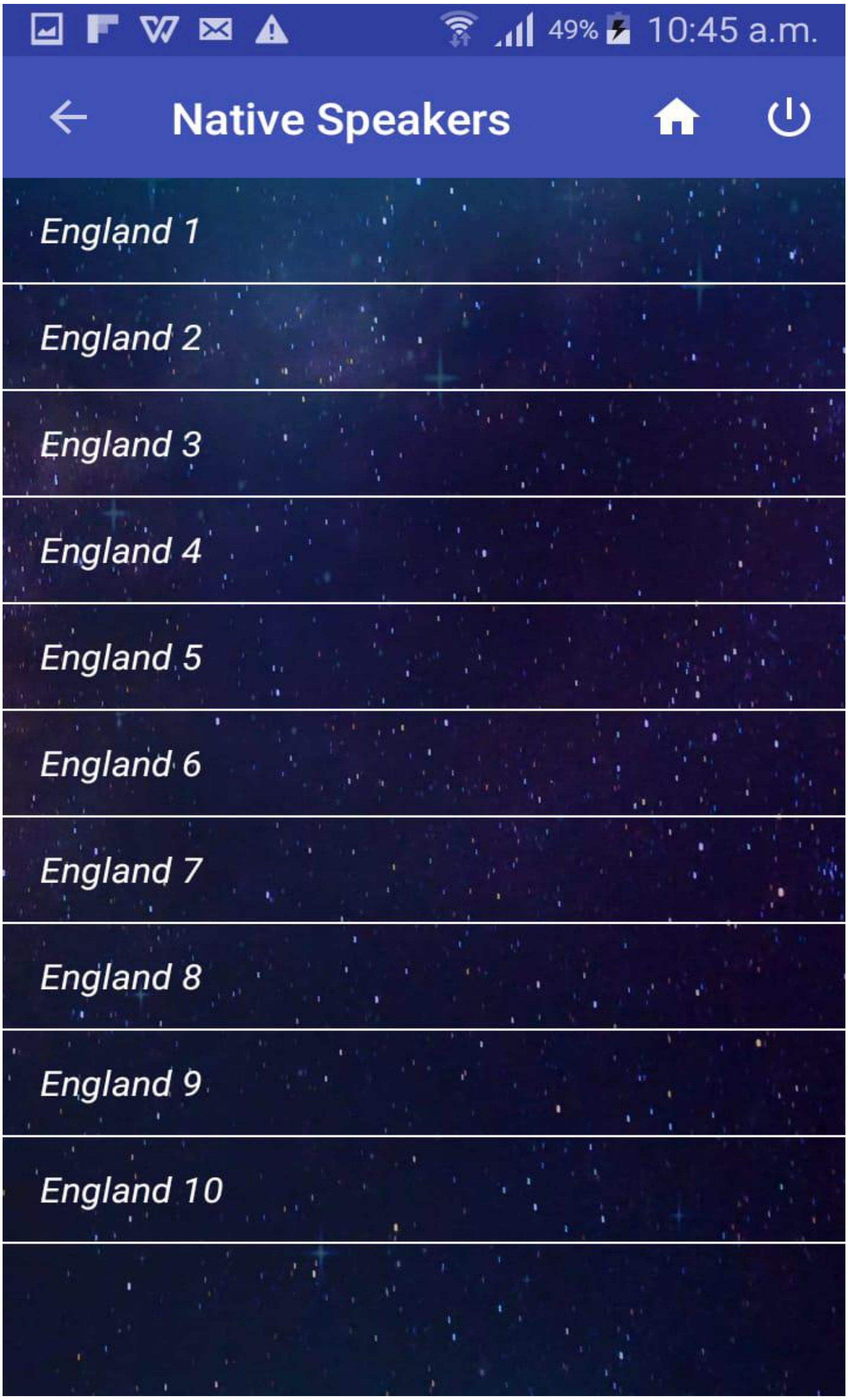


Figure 5.14 Practice Exercise

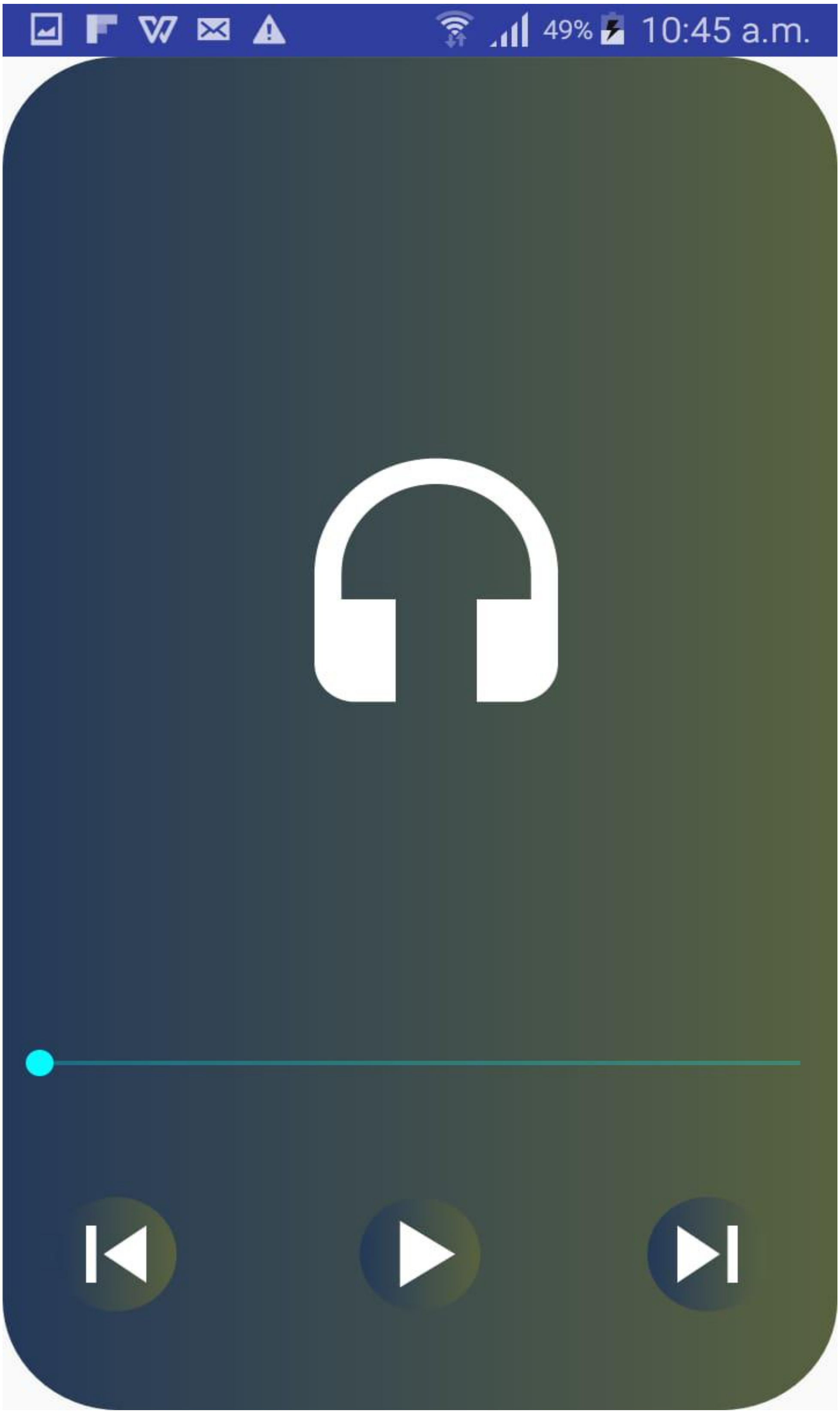


Figure 5.15 Audio Player

CHAPTER 6

TESTING AND EVALUATION

6. Testing And Evaluation

6.1. Manual Testing

Testing your project is a crucial and important part of any project and the performance and reliability of a project depends on this phase because here we evaluate the working of our system. We evaluate whether each and every part of this system is working and should be working as expected. It is a part of software development of high importance and we should not consider it separate from development process and leave it. Because without this step, you can not rely on software because you have not tested it.

Due to all these factors we have tested our project very carefully and try to find out each and every possible solution that a user can face in future and after identification of the errors, we fixed them. We perform following testing steps in order to evaluate our system :

1. System Testing
2. Unit Testing
3. Functional Testing
4. Integration Testing

After all of these steps we will perform Automated testing, which we will explain next in respective section of automated testing.

6.1.1. System Testing

It is defined as the testing of fully integrated and complete software product in all aspects. This testing comes under the term “black-box testing category” of testing in which the knowledge of the inner designing of the code is not required to start testing and testing is done by the testing team.

This testing is always performed while keeping the System Requirement Specification (SRS) in mind. This testing is considered as the final test in order to verify that the software/product to be delivered meets the requirements and specifications mentioned in requirement document. Purpose is to investigate both functional and non-functional requirements.

So in our system testing of this project we have to check that whether after integrating all modules of our project did it works according to our requirement. So as our project is completed and after that we use it to calculate our accent similarity. We perform following steps to investigate our system :

- We start by doing registration as a new user.
- Read the passage given by application in order to perform similarity test.
- After reading passage, similarity index was displayed in terms of graph in percentage.
- Now we choose our interested accents in which we want to improve our accent.
- After that practice exercises were given by application.
- Now we check our "Profile Management", it was maintaining our record and performance according to our progress so from here we can track our overall progress.

6.1.2. Unit Testing

It is the testing technique where one can test independent units of the software i.e. a group of related computer program modules that in combination makes one module, usage procedures (the way our user will use a certain module) and operating procedures (the way our module will work) are tested individually in this step in order to determine whether they work accordingly or not. With the help of this testing method each and every independent module is tested to verify if any issue exists in the module by the development team. By this testing we verify correctness of each modules independently.

As in unit testing each and every component of a software is tested individually so this testing is carried out during the process of development of an application where after completion of modules, they are being tested before integration.

We should keep on most important thing in mind that we are performing this testing as the point of view of the user. We perform our project's user testing and have following test cases:

- **Registration Module**
- **Login Module**
- **Similarity Test Module**
- **Similarity Index Module**
- **Practice Exercises Module**
- **Profile Management Module**

Now we will precisely document the test cases related to each module:

- Registration: For registration, we have to go to home page first so here we will have two test cases for this module:

  Test Case for Home Page for Registration Button:

  - Verified that "Get Started" button is present.(Get Started button will lead to registration page).
  - Verified that "Get Started" button is clickable or not i.e, it is working or not.
  - Verified that on clicking the button it takes us to registration page.

  Test case for Registration Page:

  - Verified that "First Name" field is present.
  - Verified that "Last Name" field is present.
  - Verified that "Email" field is present.
  - Verified that "Password" field is present.
  - Verified that all the above mentioned fields are editable.

- ◆ Verified that “SignUp” button is present.
- ◆ Verified the tab functionality- by pressing tabs for data input is working properly or not.
- ◆ Verified that signup button is working and after clicking it takes user to home page.

- **Login :** As like registration module testing, for testing of login module we need to go to home page also because “I Already Have an Account” button is present on home page. So we will divide our test case for login module into two parts.
  - ◆ Verified that “I already have an account” button is present on home page.
  - ◆ Verified that this button is clickable.
  - ◆ Verified that after clicking this button it takes us to Login page.

  Test Case for Login Page itself:

  - ◆ Verified that “Username” field is present.
  - ◆ Verified that “Password” field is present.
  - ◆ Verified that “Login” button is present.
  - ◆ Verified that “Don’t have an account?” link is present.
  - ◆ Verified that above link is clickable.
  - ◆ Verified that the above mentioned two fields are editable.
  - ◆ Verified that the login button is clickable and on clicking and logs us in.
  - ◆ Verified that signup link is clickable and it takes us to SignUp page.

Similarity Test Module:

Test case for similarity test module contains the testing of choosing an accent screen to the recording screen. First go to Choose Accent Screen and verify following test case:

- Verified that choose accent "Alert Dialogue" pop-ups.
- Verified that "Single Choice Item" is displayed.
- Verified that options are clickable.
- Checking the accent selection features by selecting accent and verified that it takes to reading paras screen.
- Verified that a passage is given.
- Verified that a "Recording Image Button" is available.
- Verified that this button is clickable.
- Verified that our audio is recorded by looking out output file.

Similarity Index Module:

Following is the test case for the similarity index module:

- Verified after reading passage and recording audio, Result page is displayed.
- Verified that a result graph with percentage is displayed on screen.
- Verified that result is also displayed as a message.
- Verified that "Show Data" button is present.
- Verified that "Listen Native Speakers" button is present.
- Verified that both of above buttons are clickable.
- Verified that both buttons takes to correct screen.

Practice Exercise Module:

Following is the test case for practice exercise module:

- Verified that “Native Speakers” screen is displayed.
- Verified that list view item of required audio is present.
- Verified that list view items are clickable.
- Verified and tested audio player functionality by pressing an exercise.
- Verified that a new screen with audio player appears.
- Verified that player is working and user can listen native speaker audio.

6.1.3. Functional Testing

This testing is that type in which product is evaluated to ensure that it matches the functional requirements. Whole project or system is tested for evaluation of the functional requirements so that we check they are completed. Functional testing verifies that the requirements or specifications which were committed/required at the start of project are fully completed by this developed system. Functional testing is basically related about the results of processing.

The objective that we achieve by using this type of testing is to verify that each functionality of the software application works in accordance to the requirement and specification. We can say this step is not concerned about the source code of the system. The process to test the functionality of the system is to provide a test input with a known output and we evaluate it by processing this test input and after that comparing the actual output with our expected output. The aim of this type of testing is to check the user interface, APIs if any involved, databases etc and working and functionality of the application when tested.

In functional requirements we need to test whether our system is actually performing the similarity test accurately and in practice exercises portion is it providing the audio of the asked accents.

So wee have to design a functional test for each test case.

For Accent Similarity Test:

We will first:

- Prepare values to give as input.
- Then test cases are executed
- Comparison of results that we get by these inputs and results which were expected.

The aim of similarity test is to calculate similarity index between the user accent and the native speaker's accent so we want that if record a native speaker's voice in our system, it should tell that accent of user is similar to the native accent or not.

So we take 2 voice sample, one from our data-set so we have a native speaker's sample and the other recording was my own.

Expected Results:

On recording native speaker's voice result of similarity should be a higher similarity index.

On recording my own voice it should be low similarity index.

By similarity index, it means the percentage of relevance of accent which is being tested.

Actual Results:

We got correct results. On my own recording it shows low similarity index and on recording of native speaker's audio, it shows higher similarity index.

Functional Testing for Practice Exercises:

In the practice exercise portion we want to provide the exercises containing voice recording of native speakers so that user can practice it. If user perform similarity test for any of accents from British Accent class than it should get British accents in practice exercise. As we have the dataset and we know which one is British Accent so we perform a similarity test by selecting one of the accent from British Accents and go to practice exercise look for accents and we found it was showing the same voice samples.

6.1.4. Integration Testing

Integration testing is the type of testing where we actual test the linking and integration of independent software modules which are integrated logically and now they are tested as a group. When we look a typical software project, it consists of multiple stand-alone and independent software modules, developed by different programmers. The main purpose and idea behind this level of testing of the project is to expose or identify defects in the interaction and communication between different software modules which perform correctly when they were tested in unit testing and now are to be tested when they are integrated.

Test Case #	Objective	Description	Expected Result	Actual Result
1	Check that registration button takes to registration form.	Press the Registration Button on the landing page	To be directed to the "Registration Page"	**User is directed to the "Registration Page".**
2	Check that login button takes to home screen of the app.	Enter valid credentials and press the login button.	On entering valid credentials should be directed to home page.	**User is directed to Home page on entering valid credentials.**
3	Check that after recording you are directed to the results page.	Record your voice and check for results.	After recording user is redirected to the results page	**User is redirected to results page.**

4	Check that on pressing "Listen Native Speakers", it takes to practice exercises page.	Press the "Listen Native Speakers" button and check if it redirects to practice exercises.	After pressing the button user must be redirected to practice exercises page.	**User is redirected to practice exercises page after pressing button.**
5	Check that on pressing any practice exercise, it redirects user to audio player.	Press all the practice exercises one by one and check whether it takes to player.	After pressing any exercise, user must be redirected to player.	**User is redirected to player after pressing all the practice exercises.**

Table 2 Integration Testing

CHAPTER 7

CONCLUSION AND FUTURE WORK

7. Conclusion And Future Work

7.1. **Conclusion**:

The research and development in field of Accent Classification has increased and although the improvement and work done in the field of "Automatic Speech Recognition" has open the paths for the smart gadgets like Alexa and Siri but still the term accent is a big problem in the way of these speech recognition systems to becom more robust. These speech recognition systems like Google Now and Siri are developed and usually trained on American and ritish accents. So that why they perform best for mentioned accents.

The results that we have gathered from the project that we completed shows us how much strength deep neural network architectures have in classifying both native and non-native english speakers. The MFCC features that we have extracted from audio recordings were used to feed our CNN model and our model was succesful in performing the task of classification and was most accurate than other algorithms we tested. From this we also conclude that pre processing of audio is major factor involved in behaviour and performance of CNN.

A Training Accuracy of 90% and Testing Accuracy of 88% has been achieved.

7.2. Future Work

This project can be extended to conduct more investigation on the next step of the recognition of audio, in which we recognize the content of the audio file given the accent. We can do this by using different models for every accent used, or we can attach the classified accent as an addition feature dimension to input vector. An increase in model accuracy is expected, and a possible decrease in model complexity.

This model and its training for accent identification for any other popular language. Given that many other languages has increased in popularity, it would be very interesting to train a model to classify various accents of that particular language e.g, a CNN model could be trained in order to classify Cantonese mandarin and Taiwanese mandarin.

As we know there is very progress in automatic speech recognition (ASR) systems that have helped us in develop the applications like Alexa and Siri, but still the problem in developing the independent and robust ASR projects and system that can give overcome the problems we face now a days. Studies and analysis based on data has revealed that gender and accent are very crucial factors involved in a speaker voice which effects the effectiveness of ASR systems so with the high accuracy of our model we can develop systems which will work only working with ASR implemented in them.

CHAPTER 8

REFERENCES

8. References

1. Asad Ahmed, Pratham Tangri, Anirban Panda, Dhruv Ramani, Samarjit Karmakar, (2019), VFNet: A Convolutional Architecture for Accent Classification, National Institute of Technology, Warangal, India

2. Muhammad Rizwan, David V. Anderson,(2017), A weighted accent classification using multiple words.

3. Jordan J. Bird, A Ekart, Elizabeth F. Wanner, Diego R. Faria,(2019), Accent Classification in Human Speech Biometrics for Non-Native and Native Speakers

4. Rishabh Upadhyay, Simon Lui, (2018), Foreign English Accent Classification Using Deep Brief Networks.

5. Emily Ehn, (2016), A computational Approach to Foreign Accent Classification.

6. J. H. L. Hansen P. Angkititrakul, "Advances in phone-based modeling for automatic accent classification," IEEE Trans. on Audio, Speech and Language Proc., vol. 14, pp. 634–646, March 2006.

7. S. Gray and J. H. L. Hansen, "An integrated approach to the detection and classification of accents/dialects for a spoken document retrieval system," in Proc. Automatic Speech Recognition and Understanding Workshop, San Juan, Perto Rico, 2005, pp. 35–40.

8. Pedro A. Torres-Carasquillo, Douglas Sturium, Douglas A. Reynolds, Alan McCree,(2008), Eigen channel compensation and Discriminatively Trained Gaussian.

9. Rishabh Upadhyay, (2017), Accent Clasification Using Deep Belief Network.

10. Maryam Najafian, Saeid Safavi, Phil Weber, Martin Russel, (2016), Identification of British English regional accents using fusion of I-vector and multi-accent phonotactic systems

11. Ghinwa F.Choueiter, G.Zweig, P Nguyen, (2008), An Empirical study of automatic accent classification.

12. Rongqing Huang; John H. L. Hansen; Pongtep Angkititrakul, (2007), Dialect/Accent Classification Using Unrestricted Audio, IEEE Transactions on Audio, Speech, and Language Processing (Volume: 15, Issue: 2), Page(s): 453 - 464.

13. Saiprasad Duduka, Henil Jain, Virik Jain, Harsh Prabhu, Prof. Pramila M. Chawan, (2020), Accent Classification using Machine Learning, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 11

14. Arlo Faria, (2005), Accent Classification for Speech Recognition, International Workshop on Machine Learning for Multimodal Interaction: Machine Learning for Multimodal Interaction pp 285-293

15. Abulsaoud Hanani,(2012), Human And Computer Recognition Of Regional Accents And Ethnic Groups From British English Speech.

16. Singh, Utkarsh et al. 'Foreign Accent Classification Using Deep Neural Nets'. 1 Jan. 2020: 6347 – 6352.

17. Keven Chionh Maoyuan Song, Yue Yin, Application of Convolutional Neural Networks in Accent Identiication,http://www.andrew.cmu.edu/user/yyin1/resume/cnn_project_report.pdf

18. Neha Rao, Binary Accent Classification in Speech, https://github.com/nkrao220/accent-classification/blob/master/README.md#pca

19. Singh Utkarsh et al. Foreign Accent Classification Usin2020, https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs179715

20. Ahn,Emily (2016), Computational Approach to Foreign Accent https://repository.wellesley.edu/object/ir631

21. Karsten Kumpf 1 and Robin W. King (2006), AUTOMATIC ACCENT CLASSIFICATION OF FOREIGN ACCENTED AUSTRALIAN ENGLISH SPEECH, http://www.asel.udel.edu/icslp/cdrom/vol3/004/a004.pdf

22. Phumchanit Watanaprakornkut, Chantat Esksombatchai, Peter Chien, Accent Classification, http://cs229.stanford.edu/proj2010/WatanaprakornkulEksombatchaiChien-AccentClassification.pdf

23. Pratik Parikh, Ketaki Velhal, Sanika Potdar, Aayushi Sikligar, Ruhina Karani, (2020), English Language Accent **Classification** and Conversion using Machine Learning, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=360074

.