

PROJECT REPORT ON

SPEECH BASED MUSIC PLAYER

PROJECT MADE BY
POOJA SINGH (184101028)
SALONI RATHI (184101047)
SANJANA AGRAWAL (184101029)
SUSRITA SAHA (184101038)

Under The Guidance of
PRADIP KR. DAS
Professor
Indian Institute of Technology, Guwahati

ACKNOWLEDGEMENT

Working on this project of “SPEECH BASED MUSIC PLAYER” has been a very pleasurable activity. A project is never an output of one’s individual effort it has always been a collaborative effort. Hence we would like to express our gratitude to some people without whom this project would have not been completed.

Firstly, we extend a very deep sense of gratitude to respected Dr. Pradip Kr Das for his patience support, guidance and continuous encouragement throughout the making of the project.

Secondly, we would like to express our gratitude to the TAs of Speech Processing who have been a great help throughout the project and for being available whenever we required any help.

Next we would like to thank our Department for providing us with the facilities without which we would not be able to start the project itself.

Last but not the least we would like to thank our fellow batch mates who have also helped us in this project by providing their views and important points for the successful completion of this project.

Despite the contribution of all the people mentioned above, needless to say, if there are any lapses in this project, we ourselves take the responsibility for them.

INDEX

S.No	Topic	Page No.
1	Abstract	4
2	Information to New Users about Usage	4
3	Project Execution	5
4	Guide Description	6
5	Project Functionality	15
6	Preprocessing	16
7	Training New User	19
8	Issues Faced	19
9	Target User Base	19
10	Future Scope	20

ABSTRACT

This project “Speech based Music Player” is a Windows Forms based VC++ application that uses the SoundPlayer feature of Windows Forms to play a song of the language chosen by the user. The project records the choice of language (word like “Hindi” or “English” spoken by the user, recognizes the word spoken and plays a song of that language, on the basis of the recognized word. The recognition is done using HMM, and the project also provides speech based navigation around the GUI.

INFORMATION TO NEW USERS ABOUT USAGE

To run this project on a new machine, the following need to be done:

Open the “SpeakToPlay.sln” Visual Studio Solution File in Microsoft Visual Studio (2010 or higher).

Open from Solution Explorer, the file “SelectLanguage.h”.

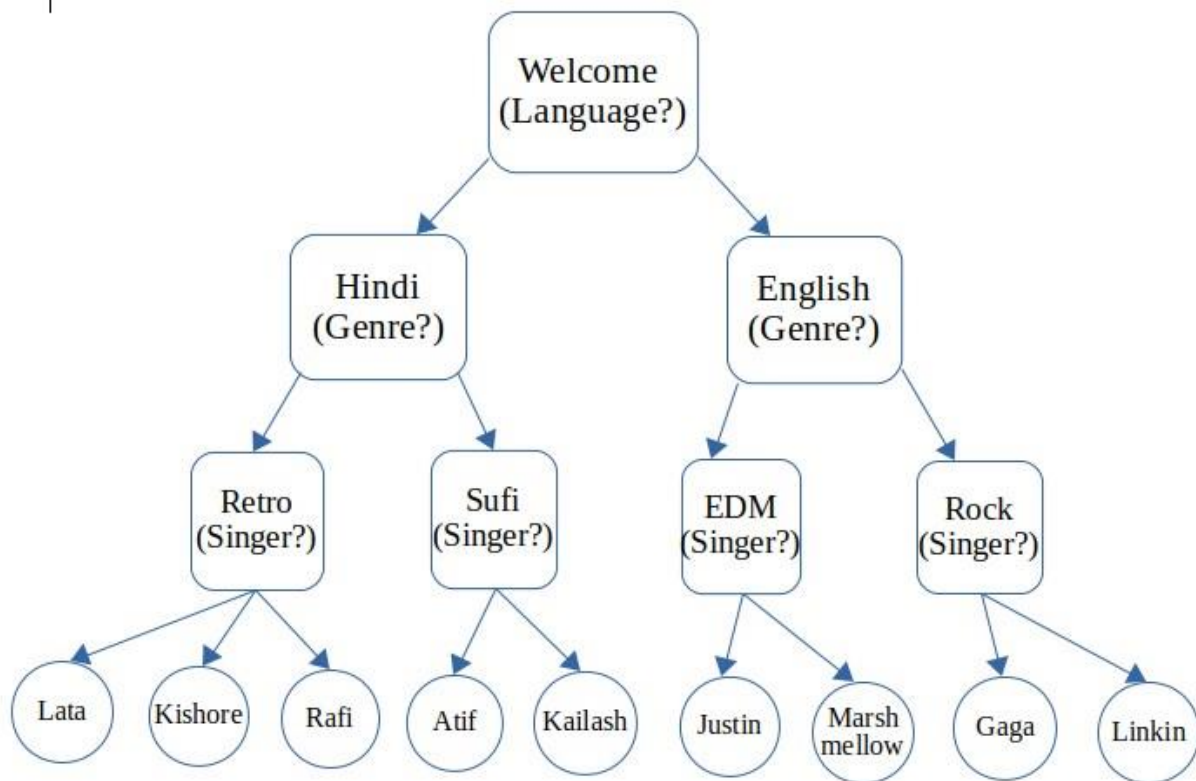
- Change variable “player->SoundLocation” to the path where the song is located. 2 such changes need to be made, one for hindi and the other for English song.
- Note that only .wav files can be played by this player. Two reference .wav songs have been made available with the project, which can be used for testing purpose.
- Change variable “path[]” to the system call used to invoke the recording_module.exe, according to the usage of the same.
- Change variable “ROOT_SAVE[]” to the path where the recording in txt format will be saved (also present in the path[] variable).
- Note that the “ANSWER.txt” file will be programmatically saved in “ROOT_SAVE” location only.

Open from Solution Explorer, the file “SpeakToPlay.cpp”.

- Change variable “ROOT_SAVE[]” to the same value as that of “ROOT_SAVE[]” in SelectLanguage.h”.
- Note that the “SilenceUnknown.txt” file also needs to be in this location.
- Also, note that all intermediate files will be stored in this location only.
- Change variable “ROOT_DATA[]” to hold path where the code book, HMMs, and Hinglish.txt files are. As such, all these files are present in the “Model” folder inside the project folder.

PROJECT EXECUTION

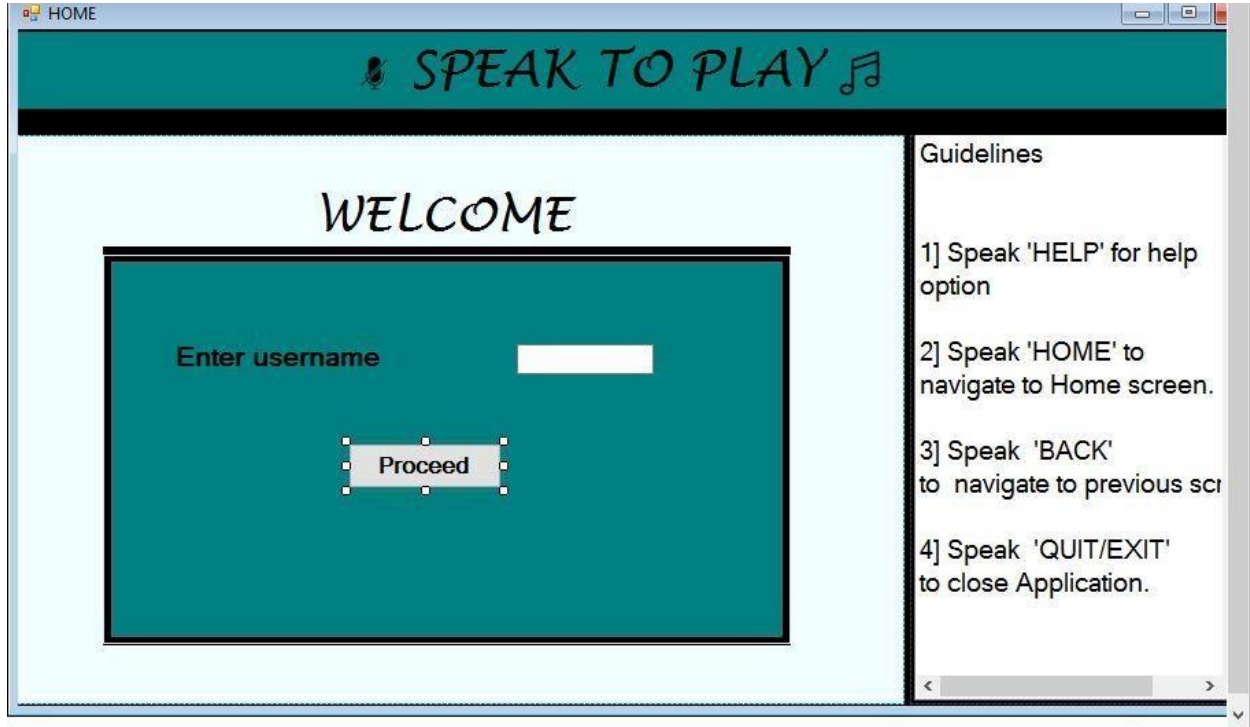
FLOWCHART



GUI DESCRIPTION

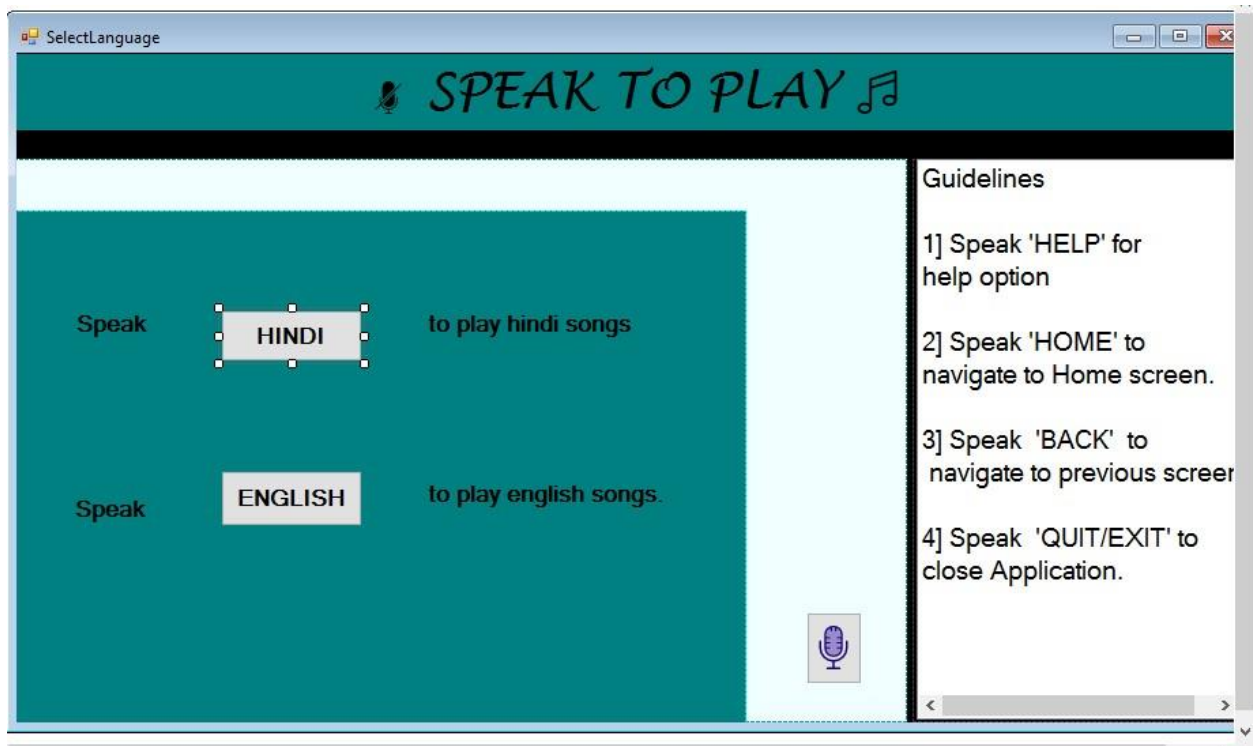
On every page, before the user starts speaking his command, he has to click on the microphone button to start the recording module.

WELCOME



The “WELCOME” page is the starting point of the project application. It asks for a username to determine which voice models to use for recognition. Then the user clicks on the button “Proceed” to navigate to the next page.

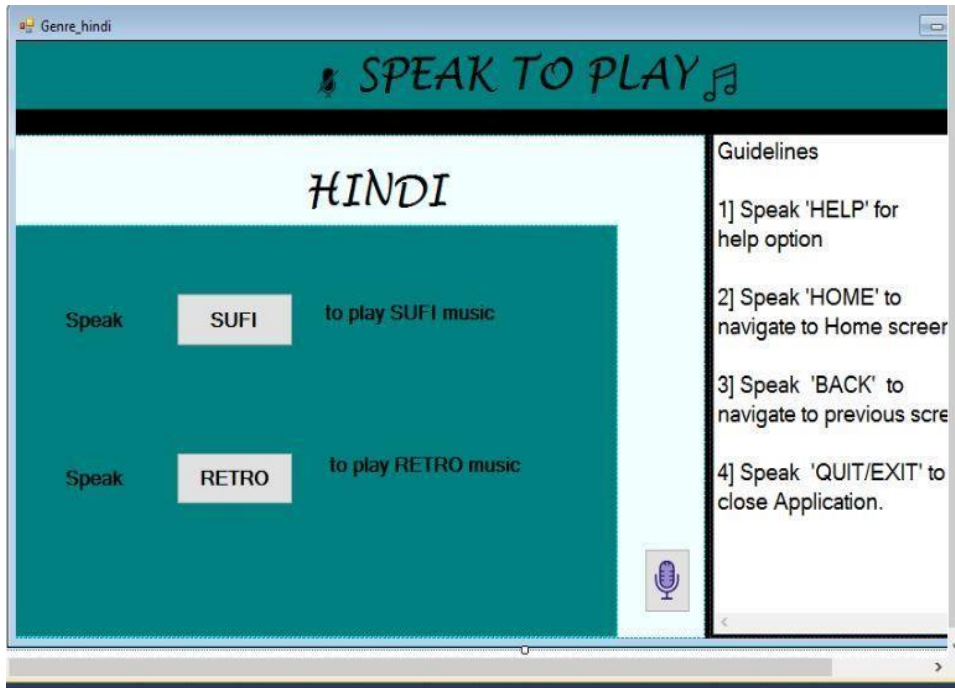
LANGUAGE



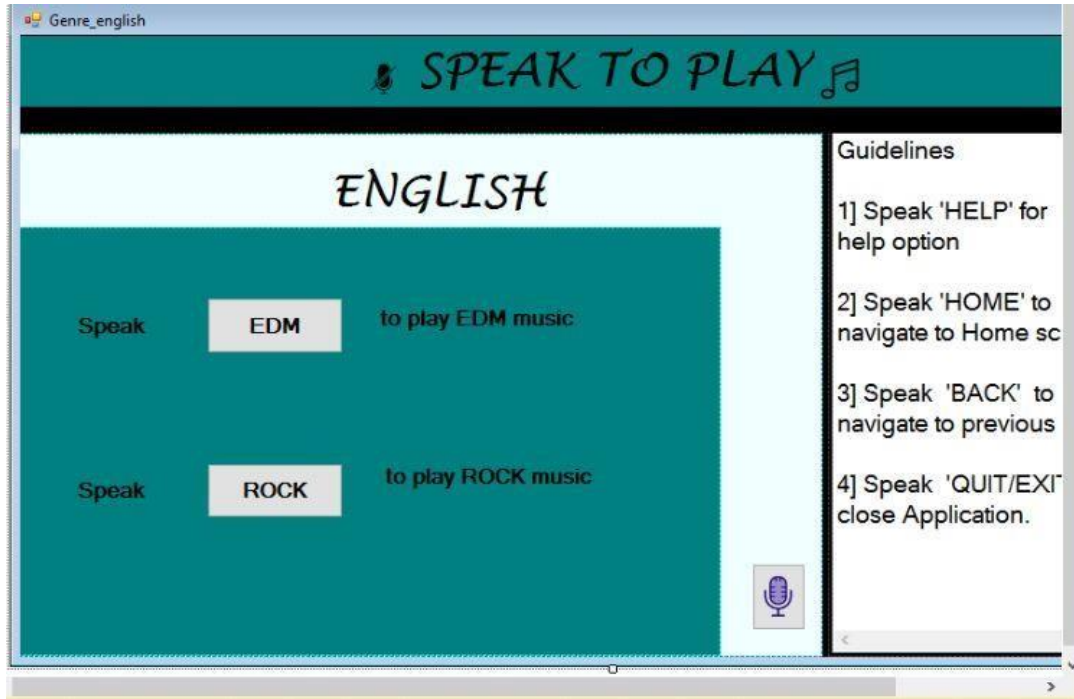
The next page that appears requires the user to speak his language choice for the song to be played, that is, "HINDI" or "ENGLISH".

A confirmation dialog box appears after a word has been recognized by the application, that requires the user to speak "YES" or "NO". If the user speaks "YES", next page appears. If he/she speaks "NO", the user remains on the same page he/she was in.

GENRE



If “HINDI” was spoken and confirmed, the next page that appears asks the user to speak his choice of “SUFİ” or “RETRO”.



If “ENGLISH” was spoken and confirmed, the next page that appears asks the user to speak his choice of “EDM” or “ROCK”.

SINGER

Sufi:



If "SUFI" was spoken and confirmed, the next page that appears asks the user to speak his choice of singer among "ATIF" or "KAILASH".

Retro:



If “RETRO” was spoken and confirmed, the next page that appears asks the user to speak his choice of singer among “LATA” or “RAFI” or “KAILASH”.

EDM:



If “EDM” was spoken and confirmed, the next page that appears asks the user to speak his choice of singer among “JUSTIN” or “MARSHMELLOW”.

Rock:



If “ROCK” was spoken and confirmed, the next page that appears asks the user to speak his choice of singer among “LINKIN” or “GAGA”.

SONGS

On speaking the choice of singer, the next page that appears displays three to four songs of that singer that matches the chosen genre and language as well. While on this page, the user needs to speak (and confirm) the number against the song of his/her choice to play it. The player closes after the song finishes playing.

ATIF ASLAM:



LADY GAGA :



KAILASH KHER:



LATA MANGESHKAR



SPEAK TO PLAY

HINDI/RETRO/LATA

LATA MANGESHKAR SONGS

Speak 1 to play Kabhi Kabhi .

Speak 2 to play Meri awaaj hi pehchaan hai.

Speak 3 to play Ae watan ke logo.

Guidelines

- 1] Speak 'HELP' f help option
- 2] Speak 'HOME' navigate to Home
- 3] Speak 'BACK' navigate to previo
- 4] Speak 'QUIT/E close Application

PROJECT FUNCTIONALITY (EXTRA)

DESCRIPTION

On any page, the user has the option to navigate to other pages using the following voice commands. These details are mentioned on every page on the right hand side panel.

“HOME”

From any page, if this command is spoken, the user is taken to the “WELCOME” page.

“BACK”

From any page, if this command is spoken, the user is taken to the previous page he/she had been on.

“HELP”

From any page, if this command is spoken, the user is taken to the “HELP” page. This page contains detailed information on how to use the project, and answers to common queries that a user might have.

“QUIT” OR “EXIT”

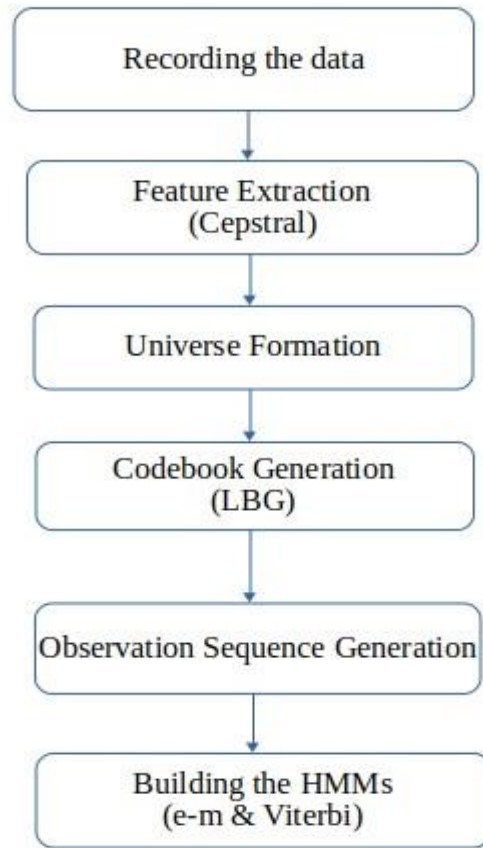
From any page, if this command is spoken, the project is terminated.

On any page, once the user has spoken his choice, a confirmation dialog box pops up, which asks the user if the word that is recognized is the same word he has spoken. The user answers “YES” or “NO” accordingly.

After the user has spoken the number (“ONE”, “TWO”, “THREE”, or “FOUR”, etc) against the song of his/her choice, the song starts playing.
(SCREENSHOT)

PREPROCESSING

FLOWCHART



DESCRIPTION

Recording the data:

Each command that will be spoken by the user is recorded a certain number of times (20 or 15), using the recording module. The recording module will store the recorded data in .txt and .wav file formats. The sampling frequency is 16kHz and the bit rate is 16bps.

List of commands that are used in the project:

Home
Help
Yes
No
Back
Hindi
English
Sufi
Retro
Rock
EDM
Atif
Kailash
Lata
Kishore
Rafi
Gaga
Linkin
Justin
Marshmellow
One
Two
Three
Four
Quit
Exit

Feature Extraction:

Each .txt file now has a number of samples. The data is then made to undergo the following operations:

DC-Offsetting
Normalization
Frame blocking
Application of Hamming Window
Calculation of R_i 's
Calculation of LPC coefficients (A_i 's)
Calculation of Cepstral coefficients (C_i 's)
Application of Raised Sine Window on the C_i 's
The windowed Cepstral coefficients are now stored in text files.

Universe Formation

All the cepstral coefficient files thus formed are used to create the universe of data.

Code Book Generation (LBG)

On the universe file thus formed, LBG algorithm is run to form the code book of size 32 (32 code vectors).

Observation Sequence Generation

Using the code book, each cepstral coefficient file is converted to an observation sequence file by comparing each cepstral vector to each code vector and noting the code book index of the code vector that gives the smallest Tokhura's distance.

Building the HMMs (E-M and Viterbi)

Using all the observation sequence files and an initial Bakis model as input, we run the E-M algorithm on the model till it converges. To check after each iteration the quality of the HMM formed, we run the Viterbi algorithm on it. Now we have 20 HMMs for each word (as there are that many iterations per word). We average these models out and use this as our initial model, to run E-M algorithm on it again. This process is repeated 3-4 times to get the best possible model.

TRAINING NEW USER

To make a new user capable of using the player, on the welcome screen, if the entered username is different from already present user (i.e. for starting we are the only valid users) then a console based application runs and asks the user to record the words to be used. The console application then after recording trains the HMM using new user's voice data (using the same preprocessing steps described previously), and stores the models in a folder of the same name as the username entered.

Then, opening the project application, the new user enters the same username that he/she entered in the console application. The project then accesses the model files from the folder with same username. This ensures that the project recognizes the commands spoken by the new user.

ISSUES FACED

- We were unable to run certain snippets of code in the header files, so finding a way to run those code snippets was difficult. We decided to run cpp files having those codes instead.
- We found it slightly challenging to find a solution to the synchronization problem between the header files and the cpp files running behind them.
- We could not find a way to eliminate clicks from the project, as we could not find a way to keep recording continuously and trim the recording to the actual commands spoken.

TARGET USER BASE

This application can be used by anyone who does not like cluttered interfaces of applications he/she must use. On using this application, he/she may simply speak his choice depending on his/her mood to start playing a song.

FUTURE SCOPE

This project can be expanded to include the new user training module as part of the GUI application itself.

The options at each stage can be increased.

There is scope to be able to stop the song that is playing using voice command itself, if there is some way to distinguish between the sound of the song and that of the command.

Instead of having to click the microphone button to start recording the word being spoken, some provision can be made to make the recording a continuous process, and the user need only speak whenever he/she is ready.