**VOICE VAULT**

Main Project Report

submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

BODDU NITISH KUMAR RAO 18L31A05B7

KAVALAPATI CHANDRA PRIYA 18L31A05F6

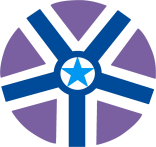
SINGULURI SAI MITHILESH 18L31A05F9

RAJAMUDILI AJAY 18L31A05G0

Under the guidance of

**Mrs.Ch. Swapna Priya**

**Asst Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VIGNAN’S INSTITUTE OF INFORMATION TECHNOLOGY**

(Autonomous) Affiliated to JNTU Kakinada & Approved by AICTE, New Delhi

Re-Accredited by NBA & NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2008, ISO 14001:2004, OHSAS 18001:2007 Certified Institution

VISAKHAPATNAM – 530 039

**MAY 2022**

**VIGNAN’S INSTITUTE OF INFORMATION TECHNOLOGY**

**Department of Computer Science and Engineering**



**CERTIFICATE**

**This is to certify that the project report entitled ― “ Voice Vault ” is the bonafide record of project work carried out under my supervision by B. Nitish Kumar Rao (18L31A05B7), S. Sai Mithilesh (18L31A05F9), K. Chandra Priya (18L31A05F6), and R. Ajay (18L31A05G0), during the academic year 2021-2022, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University, Kakinada. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.**

**Head of the Department**  **Signature of Project Guide**

Mr.B. Dinesh Reddy Mrs.Ch. Swapna Priya

Assoc. Professor Assistant Prof.

**DECLARATION**

We hereby declare that the project report entitled ― **Voice Vault** has been written by us and has not been submitted either in part or whole for the award of any degree, diploma or any other similar title to this or any other university.

1. **Mr. B. Nitish Kumar Rao (18L31A05B7)**
2. **Mr. S. Sai Mithilesh (18L31A05F9)**
3. **Miss**. **K. Chandra Priya (18L31A05F6)**
4. **Mr. R. Ajay (18L31A05G0)**

Date:

Place: VISAKHAPATNAM

**Signature Of External Examiner:**

**ACKNOWLEDGEMENT**

I consider it as a privilege to thank all those people who helped me a lot for successful completion of the project **"Voice Vault"**.

First of all I would like to thanks to my Project Supervisor **CH. Sekhar, Asst Professor, Department of Computer Science & Engineering** for helping me a lot in completing any project work and for enlightening me with constructive suggestions for solving my problems patiently and helping me to improve the quality of work.

I would like to thank our ever-inspiring **Head of the Department of Computer Science and Engineering, B. Dinesh Reddy**, for his spontaneous response to every request though he is busy with his hectic schedule of administration and teaching.

I would like to acknowledge with much appreciation the crucial role of my **U.G Co-Ordinator, Ch. Swapna Priya**, Associate Professor for helping me a lot in completing my project through the academic year. I just wanted to say thank you for being such wonderful educator as well as a person.

I would be to thank **Principal** of Vignan's Institute of Information Technology **Dr. B. Arundhati**, for her encouragement to me doing the course of this project.

I also express my sincere thanks to our beloved **Chairman** of Vignan's Institutions **Dr. L. Rathaiah**, who has given me a lot of support and freedom during my project work.

I thank to the entire **faculty** who have been a constant source of support during my study tenure.

**BODDU NITISH KUMAR RAO (18L31A05B7)**

**KAVALAPATI CHANDRA PRIYA (18L31A05F6)**

**SINGULURI SAI MITHILESH (18L31A05F9)**

**RAJAMUDILI AJAY (18L31A05G0)**

**ABSTRACT**

Voice Vault is a desktop-primarily based totally utility designed and evolved the usage of the python programming language. The device ambitions to make cloud garage with right encryption. The encryption set of rules is self-evolved, accurate, and secured. Along with Encrypted cloud garage, we also are offering right consumer authentication. User authentication is executed the usage of Voice Recognition System. Voice traits are frequently measured the usage of liveness detection with the consumer induced to utter a word for the contemporary action, or they may be measured passively and offer a sturdy extra layer of non-stop security. A consumer`s voice is tested regionally in opposition to itself, a token is despatched to the carrier provider, and get right of entry to is granted.

We evolved a voice popularity machine the usage of python & system learning. It won`t provide get right of entry to to the consumer till the voice doesn`t match. We used `python\_speech\_features` and `pyaudio` modules of python for taking voice samples and for preprocessing it, we used the GMM model (Gaussian Mixture Model).

**CONTENTS**

Declaration iii

Acknowledgement iv

Abstract v

List of Figures ix

List of Abbreviation x

**Chapter 1: INTRODUCTION 1**

1.1 Introduction to computer science 2

1.2 About the project 2

1.3 Overview 3

1.4 Software Development Life Cycle 6

**Chapter 2: PROJECT OVERVIEW 9**

2.1 Technologies Description 10

2.2 Front End Project 10

2.3.1 Registration 11

2.3.2 Login 11

2.3.3 Uploading a file 12

2.3.4 Downloading a file 12

**Chapter 3: SYSTEM ANALYSIS**  **13**

3.1 Proposed model 14

3.2 Existing model 14

**Chapter 4: REQUIREMENTS AND SPECIFICATION 15**

4.1 Functional Requirements 16

4.2 Non- Functional Requirements 16

4.3 Software Requirements 17

**Chapter 5: SYSTEM IMPLEMENTATION 18**

5.1 Frontend 19

5.1.1 About Python 19

5.1.1.1 Tkinter 19

5.1.1.2 PIL 19

5.1.1.3 Wave 20

5.1.1.4 Pickle 20

5.1.1.5 Pyauio 20

5.1.1.6 Numpy 20

5.1.1.7 Sklearn 20

5.1.1.8. Scipy 21

5.1.1.9 PythonSpeechFeatures 21

5.1.1.10 Threading 21

5.1.1.11 MySqlConnector 21

5.1.1.12 EssentialGenerator 21

5.1.1.13 GaussianMixture 22

5.1.1.14 Speech\_Recognition 22

**Chapter 6: DESIGN 23**

6.1 Introduction 24

6.2. UML Diagrams 24

6.2.1 Use Case Diagram 25

6.2.2 Sequence Diagram 27

**Chapter 7: CODING 28**

7.1 Sample Code 29

**Chapter 8: TESTING 63**

8.1 Types of Testing 64

8.1.1 Unit Testing 64

8.1.2 Integration Testing 64

8.1.3 Functional Testing 65

8.1.4 System Testing 65

8.1.5 White box Testing 65

8.2 Test Procedures 66

8.3 Test Cases 66

8.4 Experimental Results 68

**Chapter 9: CONCLUSION AND FUTURE SCOPE 71**

9.1 Conclusion And Future Scope 72

9.2 References 72

**LIST OF FIGURES**

**Figure Page No**

* 1. ASR 3
  2. Software Development Life Cycle 4
  3. Block Diagram 4
  4. Markov Process 5

2.1 Registration 11

2.2 Login 11

2.3 Uploading a file 12

2.4 Downloading a file 12

6.1 Use Case Diagram 26

6.2 Sequence Diagram 27

8.1 Table 67

8.2 Login 68

8.3 Registration 68

8.4 Upload 69

8.5 Download 69

8.6 About 70

**LIST OF ABBREVIATIONS**

HMM – Hidden Markov Model

ASR – Automated Speech Recognition

MFCC – Mel-frequency cepstral coefficients

ML – Machine Learning

GMM – Gaussian Mixture Model

GUI - Graphical user interface

**Chapter 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Computer science is the study of manipulating, managing, transforming and encoding information. It deals with the theory and methods of processing information in digital computers, the design of computer hardware and software and applications of computers. This field of study utilizes theories on how computers work to design, test and analyze concepts. Computer science also has strong connections to other disciplines, many problems in science, engineering, healthcare, business, banking and other areas can be solved effectively with computers, but finding a solution requires both computer science expertise and knowledge of the particular application domain. Finally, Computer science has a wide range of specialties. These include computer architecture software systems, graphics, AI, computational science and software engineering. Engineering provides the techniques for building hardware and software.

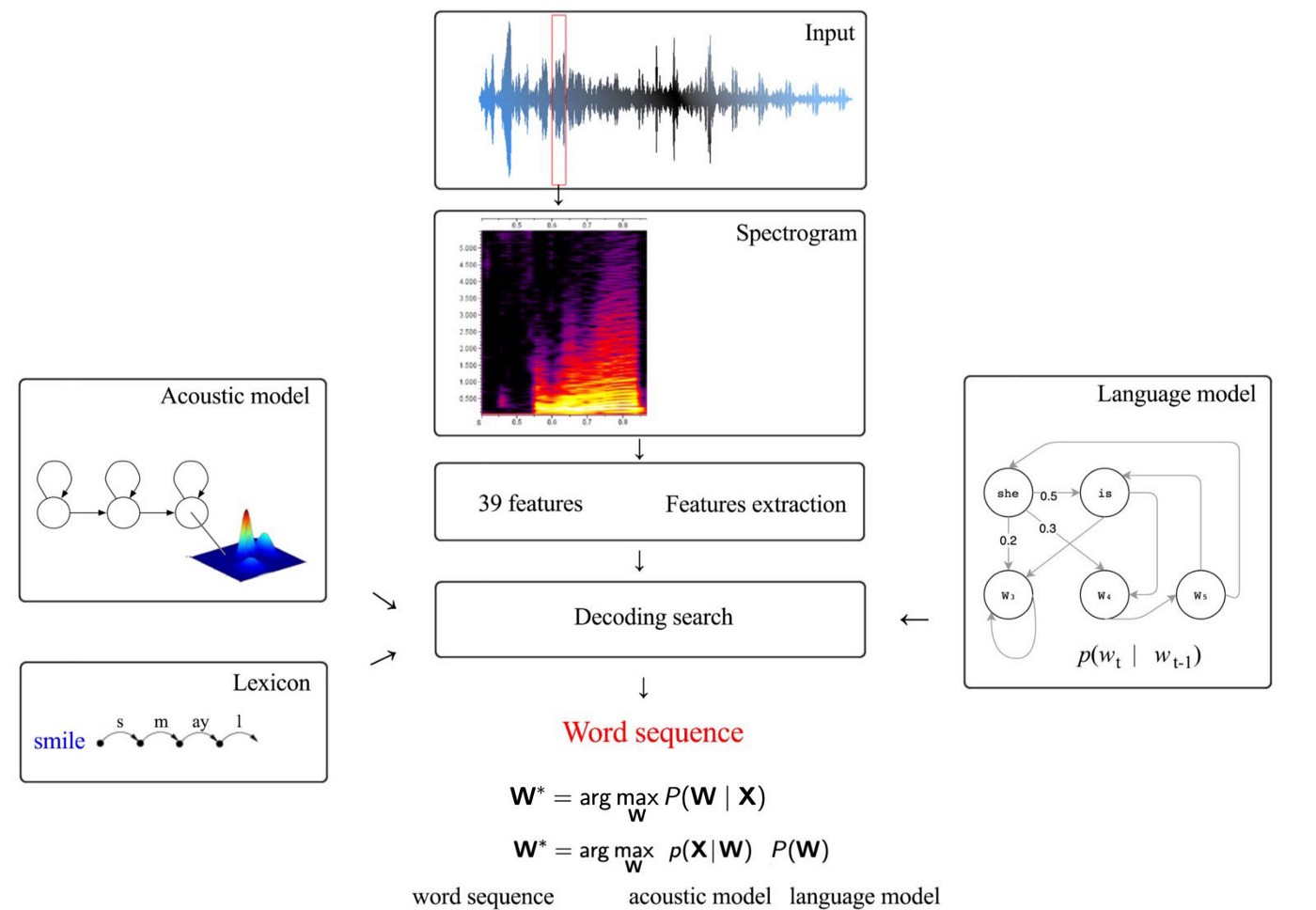
**1.2 About the Project**

It is intended to develop a real-time voice identification system. The voice is first captured using a microphone, and then voice features are retrieved. The hamming window is used to reduce dis-continuities at the frame's edge, resulting in smooth frequency transmission in speech signals. Mel Frequency Cepstral Coefficients produces 15 MFCC coefficients using 40 Mel filters. These coefficients are subsequently sent to GMM to be used in the training phase. Users are identified by comparing the logarithmic probability with the system specified threshold. It decreases the amount of work required to remember all of the passwords. The application aids in the security of all text files, photos, videos, and other media. It prevents unauthorized users from gaining access. The application is made even more secure by using a voice security technique for authentication and then encrypting the files. Cryptography technology (AES Encryption) increases the security system by virtually eliminating the possibility of a compromise. As a result, a new security solution has been developed that allows you to lock and unlock your files while also storing them in the cloud.

**1.3 Overview**

**Automatic Speech Recognition (ASR)**

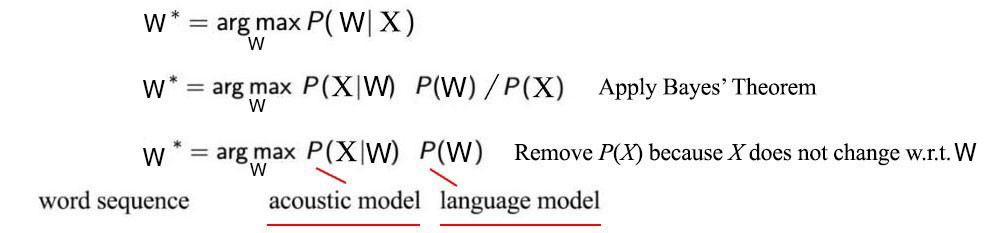
Let’s get a high-level overview first. The diagram below is a high-level architecture for speech recognition that links HMM (Hidden Markov Model) with speech recognition.



**Figure 1.1 ASR**

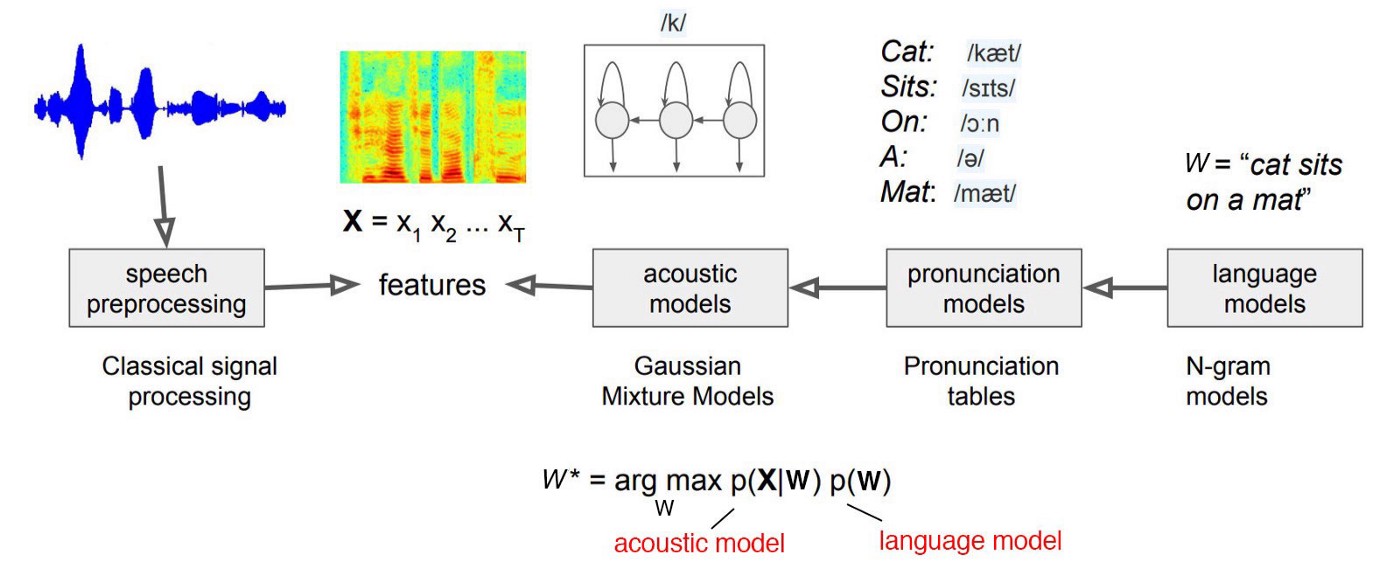
Starting from an audio clip, we slide windows of 25 ms width and 10 ms apart to extract MFCC features. For each window frame, 39 MFCC parameters will be extracted. The primary objective of speech recognition is to build a statistical model to infer the text sequences W (say “cat sits on a mat”) from a sequence of feature vectors X.

One approach looks for all possible sequences of words (with limited maximum length) and finds one that matches the input acoustic features the best.



**Figure 1.2**

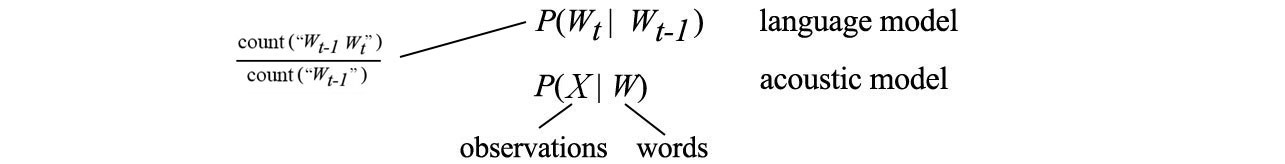
This model depends on building a language model P(W), a pronunciation lexicon model, and an acoustic model P(X|W) (a generative model) as below.



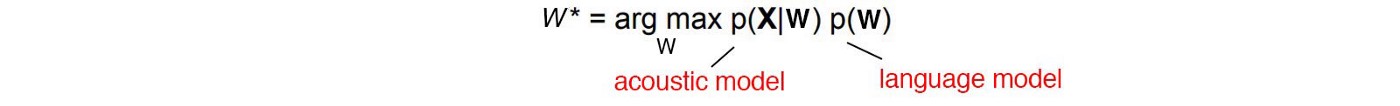
**Figure 1.3 Block diagram**

A pronunciation model can use tables to convert words to phones, or a corpus is already transcribed with phonemes already. The acoustic model is about modeling a sequence of feature vectors given a sequence of phones instead of words. But we will continue the use of the notation p(X|W) for the acoustic model. Just be aware.

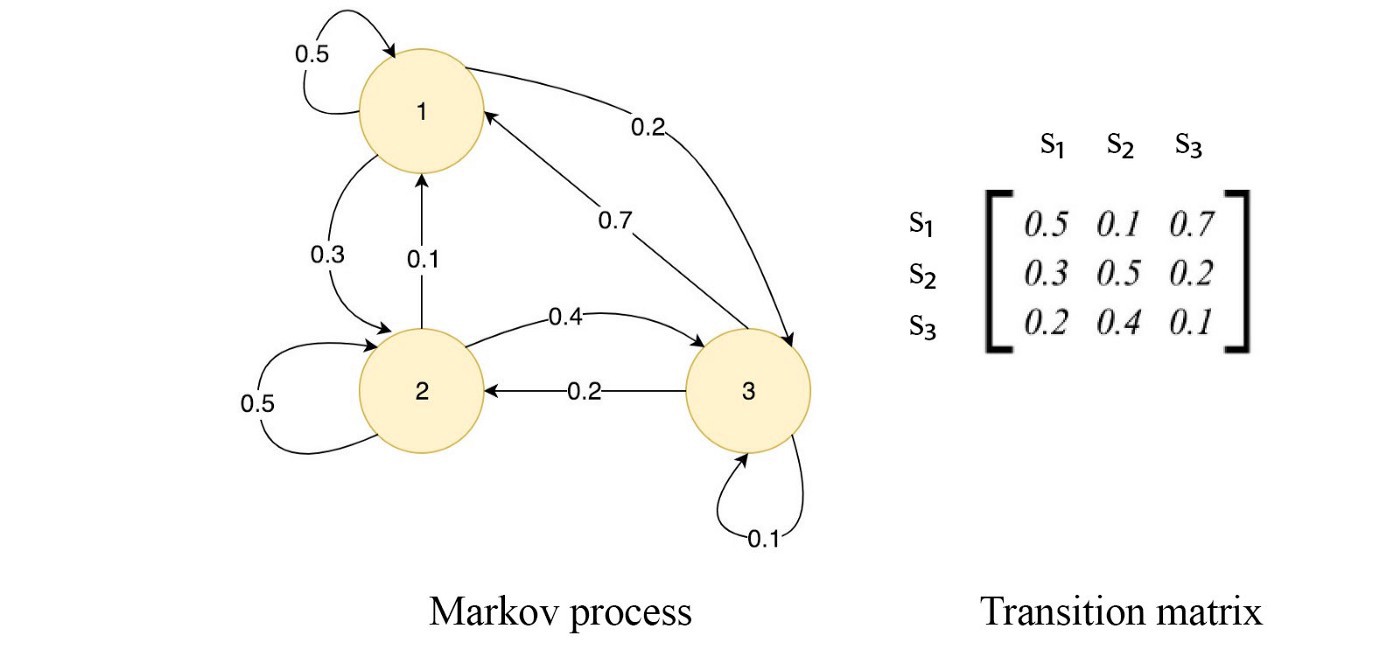
The language model is about the likelihood of the word sequence. For example, “I watch a movie” will be more likely than “I you movie watch” or “I watch an apple”. It predicts the next word given the previous words. If we approximate it with a first-order Markov chain, the next word will depend on the current word only. We can estimate it by counting the occurrence of word pairs in a corpus.



By combining the acoustic model and the language model, we search for the text sequence with the maximum likelihood.

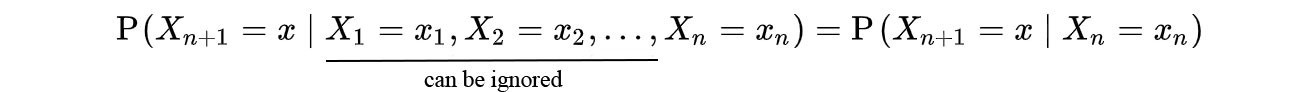
Hidden Markov Model (HMM)

A Markov chain contains all the possible states of a system and the probability of transiting from one state to another.

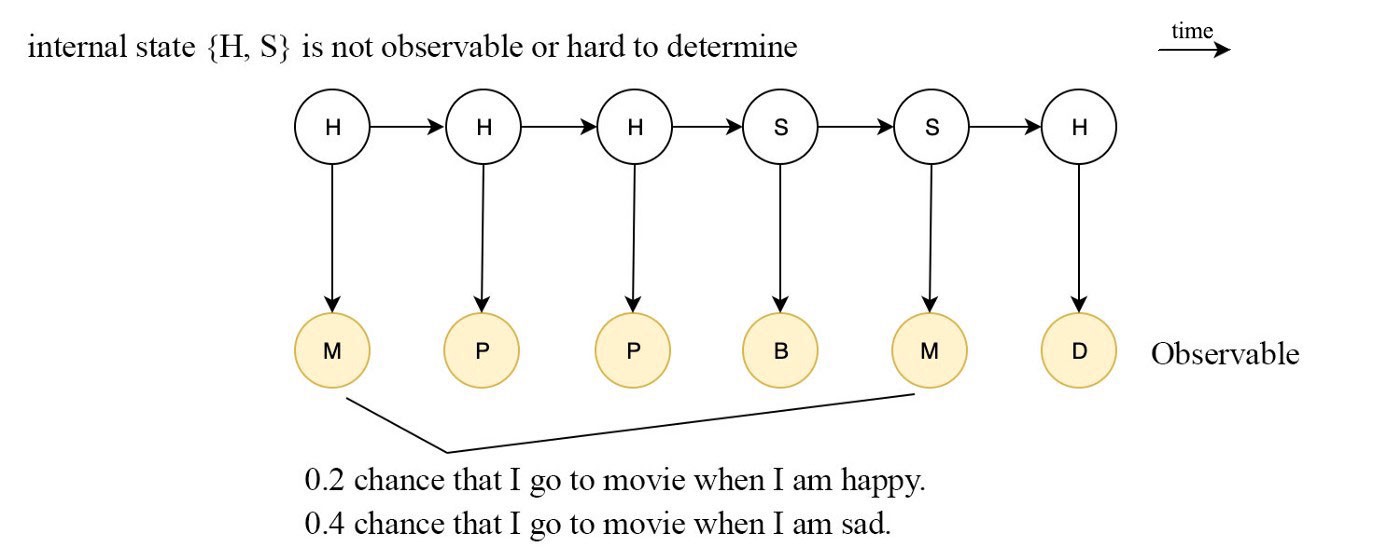


**Figure 1.4 Markov Process**

A first-order Markov chain assumes that the next state depends on the current state only. For simplicity, we often call it a Markov chain.



This model will be much easier to handle. However, in many ML systems, not all states are observable and we call these states hidden states or internal states. Some may treat them as latent factors for the inputs. For example, it may not be easy to know whether I am happy or sad. My internal state will be {H or S}. But we can get some hints from what we observe. For example, when I am happy I have a 0.2 chance that I watch a movie, but when I am sad, that chance goes up to 0.4. The probability of observing an observable given an internal state is called the **emission probability**. The probability of transiting from one internal state to another is called the **transition probability**.



For speech recognition, the observable is the content in each audio frame. We can use the MFCC parameters to represent it. Let’s see what we can do with an HMM.

**1.4 Software Development Life Cycle**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software’s. The SDLC aims to produce a high - quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

• SDLC is the acronym of Software Development Life Cycle.

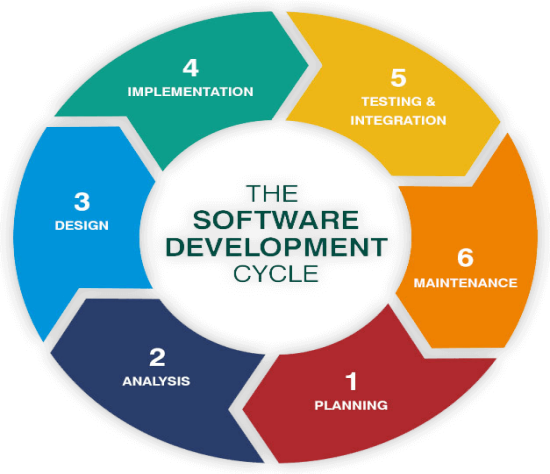
• It is also called as Software Development Process.

• SDLC is a framework defining tasks performed at each step in the software development process.

• ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The below figure shows the Software Development Life Cycle .



**Figure 1.2 Software Development Life Cycle**

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS

- Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules. The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high-level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

**Chapter 2**

**PROJECT OVERVIEW**

**2. PROJECT OVERVIEW**

**2.1 Technologies Description**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like OpenCV, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

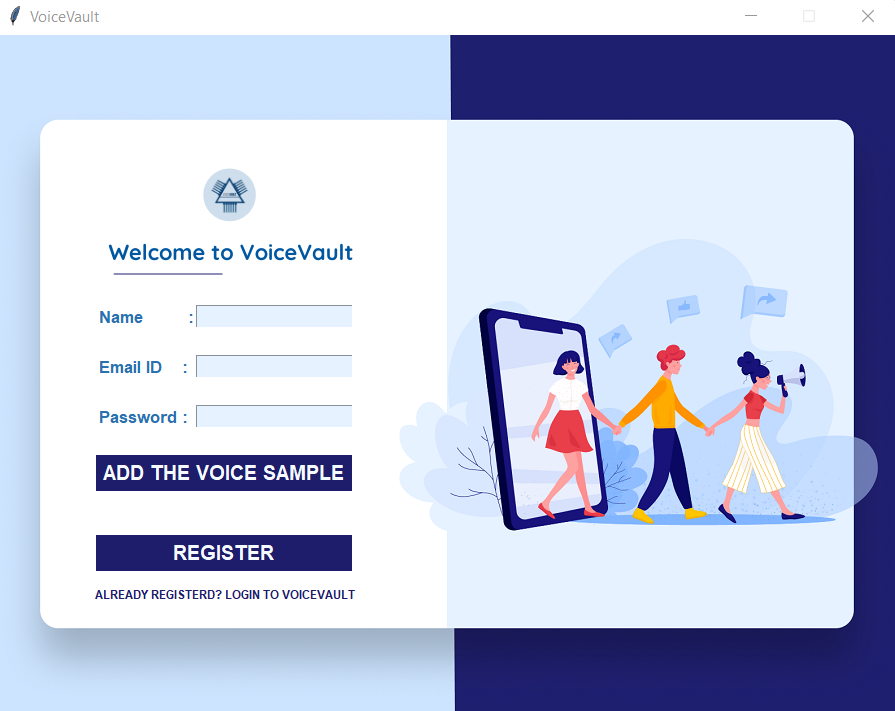
**2.3 Front End of Project**

Our application consists of few main aspects. They are

* Registration
* Login
* Uploading a file
* Downloading a file

**2.3.1 Registration**

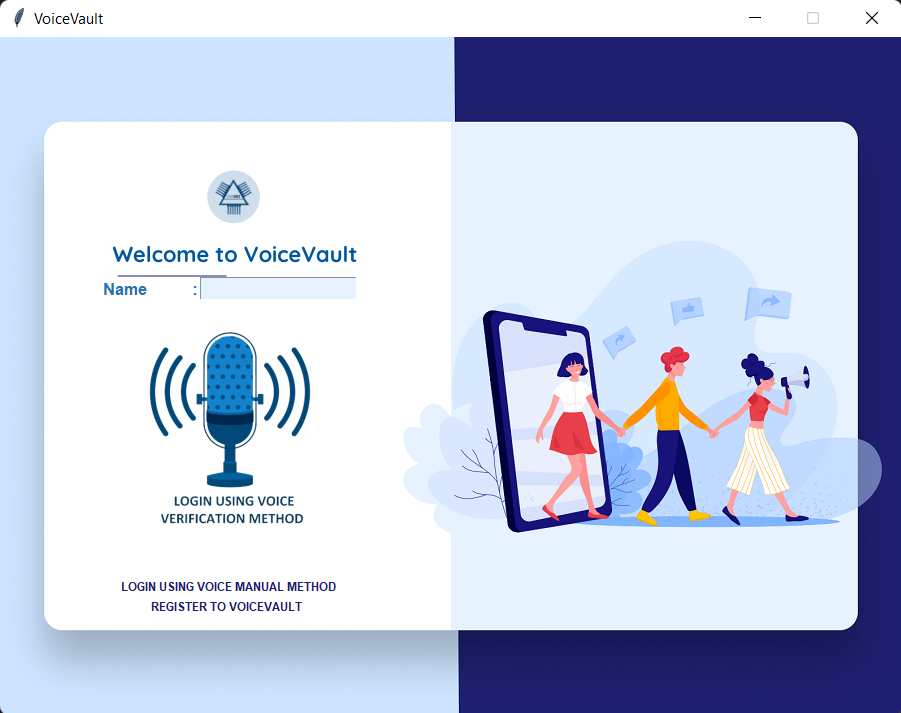
The below figure shows the registration page.



**Figure 2.1 Registration**

**2.3.2 Login**

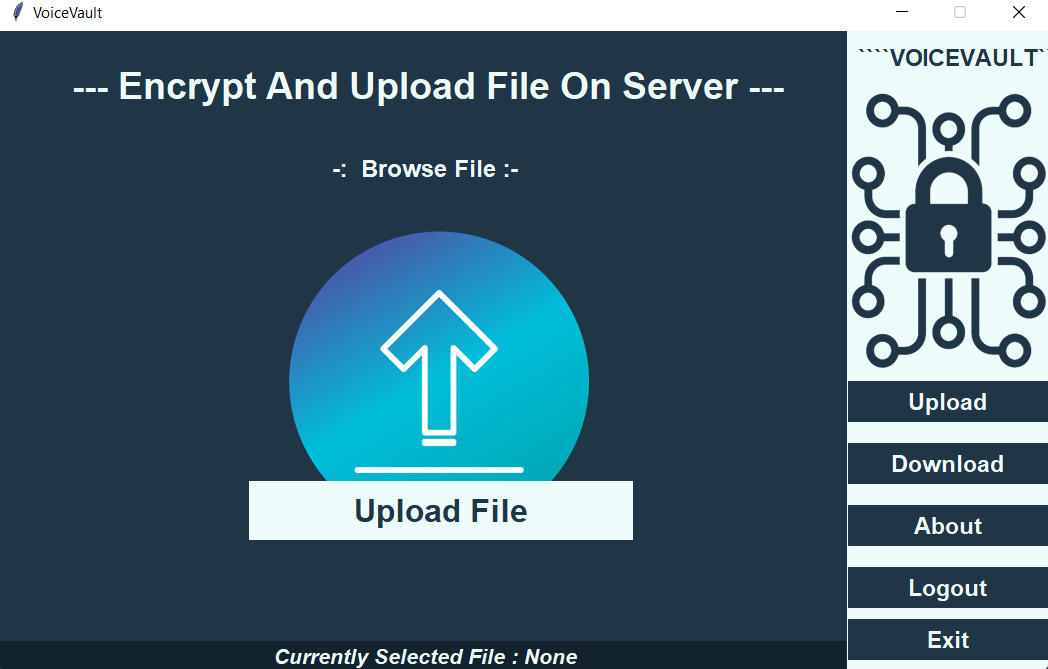
Here the user login with his voice.



**Figure 2.2 Login**

**2.3.3 Uploading a file**

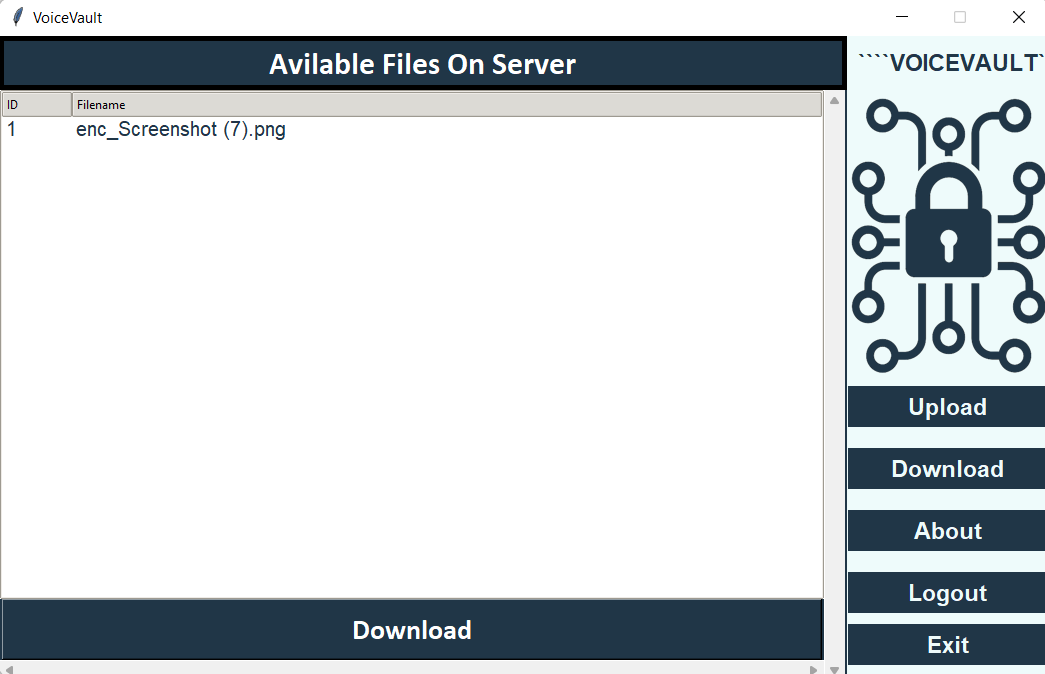
The user upload his file which will be encrypted automatically and get stored in the cloud.



**Figure 2.3 Uploading a file**

**2.3.4 Downloading a file**

The user downloads the encrypted file from the cloud database.



**Figure 2.3 Downloading a file**

**Chapter 3**

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

Often many software/tools enable you to prevent other people from viewing, editing, and deleting files and directories. These programs use encryption techniques for security. These tools are designed for protecting files/folders on your local computer. Many such programs can be used to lock directories on a flash drive, external USB drive, internal hard drive, and more. These tools have ads and charge you money for the protection of files.

**3.2 Proposed System**

The proposed tool can be accessed in the real world from any device. There are hacking software’s that can pick up everything entered through the keyboard and store it into a buffer. Hence a password entered through the keyboard can be easily hacked. However, if a password is spoken and the frequencies are considered, there is no software to date that can hack frequencies. The authors present an innovative technique for desktop file hiding. The software-based on this work is capable of locking files such that only the user who has locked them with his spoken password can access them.

**Chapter 4**

**REQUIREMENTS**

**AND SPECIFICATIONS**

**4.REQUIREMENTS AND SPECIFICATIONS**

Requirements play a vital role in the software development life cycle (SDLC).as it describes the complete requirements of the system, it is meant for use by the developers and will be the basic during testing phase. Any changes made to the future will have to go through formal change approval process.

**4.1 Functional Requirements**

It describes what the system should do i.e. the services provided for the users and for the system.

The functional requirements should include

1. Everything that a user of the system would need to know regarding what the system does.

2. Everything that would concern any other that has to interface to this system.

The functional requirements can be further Categorized as follows:

• What input the system should accept and under what conditions. This includes data and command both from the user and the other system.

• What computations the system should produce.

**4.2 Non-Functional Requirements**

Non-functional requirements describe user-visible aspects of the system that are not directly related to functionally of the system.

These requirements define what qualities is exhibited by the system. The following are the non- function requirements of the system:

1. Performance (arrives at solution faster than hard computing approach)

2. Usability

3. Scalability

4. Maintainability

5. Interoperability

**4.3 Software Requirements**

**4.3.1** Windows / Linux / Mac OS

The proposed application is system independent.

**4.3.2**. Microsoft Visual Studio 2010 Or Higher

This is an IDE that is used to develop this windows based desktop application.

**4.3.3**. SQL

To query the data My SQL is used which is an open source relational database management system.

**4.3.4**. Tkinter

Used to create Graphical User Interface for the application.

* + 1. **Frontend**
* **Python**

1. Tkinter
2. PIL
3. WAVE
4. Pickle
5. Pyaudio
6. Numpy
7. Sklearn
8. Scipy
9. PythonSpeechFeatures
10. Threading
11. MySqlConnector
12. EssentialGenerator
13. GaussianMixture
14. SpeechRecognition

**Chapter 5**

**SYSTEM IMPLEMENTATION**

**5.SYSTEM IMPLEMENTATION**

**Software Description**

**5.1 Frontend**

**5.1.1 About PYTHON**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**5.1.1.1 Tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

* The GPL version of PyQt5 can be installed from PyPI:

pip install tk

**5.1.1.2 PIL**

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

* Execute the following command to install the library:

pip install pillow

**5.1.1.3 Wave**

The wave module in Python's standard library is an easy interface to the audio WAV format. The functions in this module can write audio data in raw format to a file like object and read the attributes of a WAV file.

* Execute the following command to install the library:

pip install waves

**5.1.1.4 Pickle**

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling.

* Execute the following command to install the library:

pip install pickle5

**5.1.1.5 Pyaudio**

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms.

* Execute the following command to install the library:

pip install PyAudio

**5.1.1.6 Numpy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

* Execute the following command to install the library:

pip install numpy

**5.1.1.7 Sklearn**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

* Execute the following command to install the library:

pip install scikit-learn

**5.1.1.8 Scipy**

**SciPy in Python** is an open-source library used for solving mathematical, scientific, engineering, and technical problems. It allows users to manipulate the data and visualize the data using a wide range of high-level Python commands. SciPy is built on the Python NumPy extention.

* Execute the following command to install the library:

pip install SciPy

**5.1.1.9 PythonSpeechFeatures**

This library provides common speech features for ASR including MFCCs and filterbank energies.

* Execute the following command to install the library:

pip install python\_speech\_features

**5.1.1.10 Threading**

In Python, the threading module is a built-in module which is known as threading and can be directly imported. Holding data, Stored in data structures like dictionaries, lists, sets, etc. Can be passed as a parameter to a function.

* Execute the following command to install the library:

pip install threaded

**5.1.1.11 MySqlConnector**

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It also contains an implementation of the X DevAPI, an Application Programming Interface for working with the MySQL Document Store.

* Execute the following command to install the library:

pip install mysql-connector-python

**5.1.1.12 EssentialGenerator**

Whether it's testing database performance or a new web interface, we've all needed a dead simple solution that'a flexible enough to generate a complex data set. If this is one of those times, you've come to the right place. Essential Generators uses Markov chains to generate 'realistic' data - and you can train them on your own data to make it even more real.

* Execute the following command to install the library:

pip install essential\_generators

**5.1.1.13 Gaussian Mixture**

sklearn-mixture is a package which enables one to learn Gaussian Mixture Models (diagonal, spherical, tied and full covariance matrices supported), sample them, and estimate them from data. Facilities to help determine the appropriate number of components are also provided. A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

* Execute the following command to install the library:

pip install scikit-learn

**5.1.1.14 Speech Recognition**

Speech recognition, as the name suggests, refers to automatic recognition of human speech. Speech recognition is one of the most important tasks in the domain of human computer interaction.Speech recognition has various applications ranging from automatic transcription of speech data (like voicemails) to interacting with robots via speech.

Several speech recognition libraries have been developed in Python. However we will be using the SpeechRecognition library, which is the simplest of all the libraries.

* Execute the following command to install the library:

pip install SpeechRecognition

**CHAPTER- 6**

**DESIGN**

**6. DESIGN**

**6.1 Introduction**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer‘s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word ―Quality‖. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer‘s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**6.2. UML Diagrams**

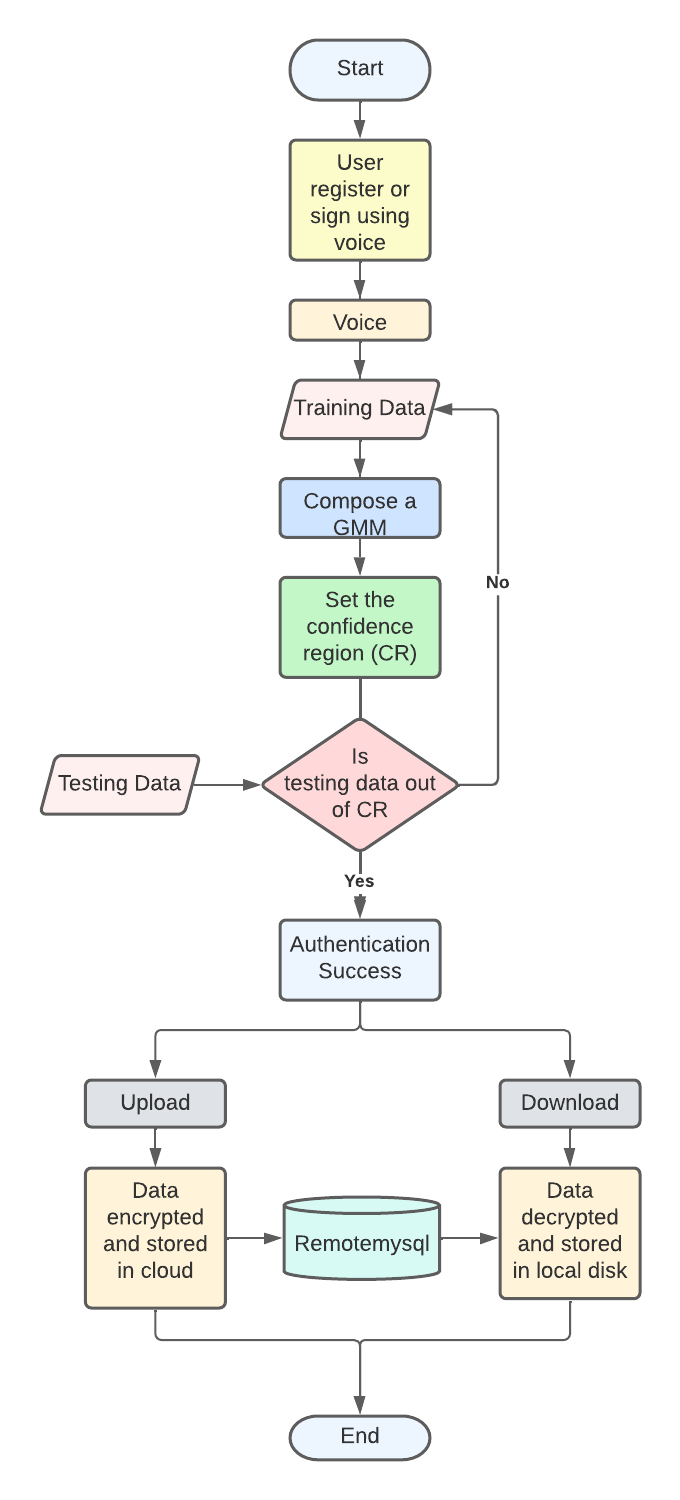
**6.2.1 Use Case Diagram**

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

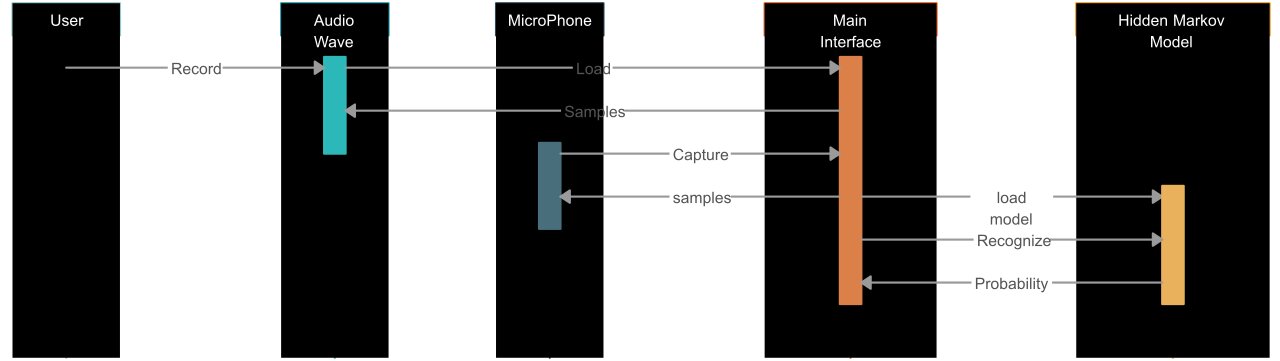
Hence to model the entire system, a number of use case diagrams are used.



**Fig 6.1: Use Case Diagram**

**6.2.2 Sequence Diagram**

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations.



**Fig 6.2: Sequence Diagram**

**Chapter 7**

**CODING**

**7. CODING**

**7.1 Sample Code**

from tkinter import \*

import ctypes,os,sys

from tkinter.tix import \*

from PIL import ImageTk, Image

import tkinter.filedialog as filedialog

import tkinter.messagebox as tkMessageBox

import wave

import time

import pickle

import pyaudio,sqlite3

import warnings

import numpy as np

from sklearn import preprocessing

from scipy.io.wavfile import read

import python\_speech\_features as mfcc

from sklearn.mixture import GaussianMixture

import threading as T

import shutil

from tkinter import ttk

import mysql.connector as msc

from shutil import copyfile

from tkinter import ttk

import threading # To perform multi threading

import pyjokes

warnings.filterwarnings("ignore")

os.makedirs('testing\_set', exist\_ok=True)

os.makedirs('training\_set', exist\_ok=True)

path = 'C:/ProgramData/VoiceVault/'

ImagePath='./Images/'

D\_path = "./Downloads/"

if not os.path.exists(D\_path):

os.mkdir(D\_path)

if not os.path.exists(path):

os.mkdir(path)

mydb = msc.connect(

host="remotemysql.com",

user="oYLTRlQjWj",

password="LETG04XLSX",

database="oYLTRlQjWj",

)

def Database():

#Function For Database Connectivity

global cursor #Making conn,cursor a globle variable

cursor = mydb.cursor()

table\_name = "users"

cursor.execute(f'''create table IF NOT EXISTS {table\_name}(

ID int NOT NULL AUTO\_INCREMENT,

Name varchar(255) NOT NULL,

Username varchar(255) NOT NULL,

Password varchar(255) NOT NULL,

PRIMARY KEY (ID)

)''')

def create\_user\_table(table\_name="Data"):

try:

mycursor = mydb.cursor()

mycursor.execute(f'''create table IF NOT EXISTS {table\_name}(

ID int NOT NULL AUTO\_INCREMENT,

FileName varchar(255) NOT NULL,

File LONGBLOB NOT NULL,

PRIMARY KEY (ID)

)''')

print(mycursor)

print("Successfully Table Created.")

except Exception as e:

print(e)

def calculate\_delta(array):

rows,cols = array.shape

print(rows)

print(cols)

deltas = np.zeros((rows,20))

N = 2

for i in range(rows):

index = []

j = 1

while j <= N:

if i-j < 0:

first =0

else:

first = i-j

if i+j > rows-1:

second = rows-1

else:

second = i+j

index.append((second,first))

j+=1

deltas[i] = ( array[index[0][0]]-array[index[0][1]] + (2 \* (array[index[1][0]]-array[index[1][1]])) ) / 10

return deltas

def extract\_features(audio,rate):

mfcc\_feature = mfcc.mfcc(audio,rate, 0.025, 0.01,20,nfft = 1200, appendEnergy = True)

mfcc\_feature = preprocessing.scale(mfcc\_feature)

print(mfcc\_feature)

delta = calculate\_delta(mfcc\_feature)

combined = np.hstack((mfcc\_feature,delta))

return combined

def train\_model():

source = './training\_set/'

dest = "./"

train\_file = "./training\_set\_addition.txt"

file\_paths = open(train\_file,'r')

count = 1

features = np.asarray(())

for path in file\_paths:

path = path.strip()

sr,audio = read(source + path)

vector = extract\_features(audio,sr)

if features.size == 0:

features = vector

else:

features = np.vstack((features, vector))

if count == 1:

gmm = GaussianMixture(n\_components = 6, max\_iter = 200, covariance\_type='diag',n\_init = 3)

gmm.fit(features)

# dumping the trained gaussian model

picklefile = path.split("-")[0]+".gmm"

pickle.dump(gmm,open(dest + picklefile,'wb'))

features = np.asarray(())

count = 0

count = count + 1

def voiceverificationlogin():

#-----------------------------------------------------------------------

loginwindow = Toplevel()

loginwindow.title('VoiceVault')

loginwindow.resizable(0,0)

NAME = StringVar()

#-----------------------------------------------------------------------

user32 = ctypes.windll.user32

user32.SetProcessDPIAware()

[w, h] = [user32.GetSystemMetrics(0), user32.GetSystemMetrics(1)]

lt = [w, h]

a = str(lt[0]//2-451)

b= str(lt[1]//2-339)

loginwindow.geometry("902x678+"+a+"+"+b)

#-----------------------------------------------------------------------

img = Image.open(r"images/registerPage.png")

img = ImageTk.PhotoImage(img)

panel = Label(loginwindow, image=img)

panel.pack(side="top", fill="both", expand="yes")

#-----------------------------------------------------------------------

def verify():

n = (NAME.get()).lower()

if n=='':

tkMessageBox.showinfo('VoiceVault','All Fields Are Mendatory To Fill')

else:

FORMAT = pyaudio.paInt16

CHANNELS = 1

RATE = 44100

CHUNK = 512

RECORD\_SECONDS = 10

device\_index = 2

audio = pyaudio.PyAudio()

index = 1

stream = audio.open(format=FORMAT, channels=CHANNELS,

rate=RATE, input=True,input\_device\_index = index,

frames\_per\_buffer=CHUNK)

Recordframes = []

for i in range(0, int(RATE / CHUNK \* RECORD\_SECONDS)):

data = stream.read(CHUNK)

Recordframes.append(data)

stream.stop\_stream()

stream.close()

audio.terminate()

OUTPUT\_FILENAME="sample.wav"

WAVE\_OUTPUT\_FILENAME=os.path.join("testing\_set",OUTPUT\_FILENAME)

trainedfilelist = open("testing\_set\_addition.txt", 'w')

trainedfilelist.write(OUTPUT\_FILENAME+"\n")

trainedfilelist.close()

waveFile = wave.open(WAVE\_OUTPUT\_FILENAME, 'wb')

waveFile.setnchannels(CHANNELS)

waveFile.setsampwidth(audio.get\_sample\_size(FORMAT))

waveFile.setframerate(RATE)

waveFile.writeframes(b''.join(Recordframes))

waveFile.close()

source = "./testing\_set/"

modelpath = "./"

test\_file = "./testing\_set\_addition.txt"

file\_paths = open(test\_file,'r')

gmm\_files = [os.path.join(modelpath,fname) for fname in

os.listdir(modelpath) if fname.endswith('.gmm')]

#Load the Gaussian gender Models

models = [pickle.load(open(fname,'rb')) for fname in gmm\_files]

speakers = [fname.split("\\")[-1].split(".gmm")[0] for fname

in gmm\_files]

# Read the test directory and get the list of test audio files

for path in file\_paths:

path = path.strip()

sr,audio = read(source + path)

vector = extract\_features(audio,sr)

log\_likelihood = np.zeros(len(models))

for i in range(len(models)):

gmm = models[i] #checking with each model one by one

scores = np.array(gmm.score(vector))

log\_likelihood[i] = scores.sum()

winner = np.argmax(log\_likelihood)

if n in speakers[winner]:

loginwindow.destroy()

home(n)

else:

tkMessageBox.showinfo('VoiceVault','Unable To Verify The User')

def verifyv():

x = T.Thread(target=verifyvoice)

x.start()

def switchwin4():

loginwindow.destroy()

signinmanual()

def switchwin5():

loginwindow.destroy()

register()

photo = (Image.open("images/voiceverificationlogin.png")).resize((300,300))

img2 = ImageTk.PhotoImage(photo)

b1 = Button(loginwindow, highlightthickness = 0, bd = 0,bg='white', image = img2,command=verify)

b1.place(x=80,y=240)

l1 = Label(loginwindow, text="Name : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l1.place(x=100, y=240)

e1 = Entry(loginwindow, textvariable=NAME, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af")

e1.place(x=200, y=240)

b2 = Button(loginwindow, text="LOGIN USING VOICE MANUAL METHOD", width="38", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white",activebackground ="white", command=switchwin4)

b2.place(x=94, y=540)

b3 = Button(loginwindow, text="REGISTER TO VOICEVAULT", width="40", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white",activebackground ="white", command=switchwin5)

b3.place(x=85, y=560)

loginwindow.mainloop()

def signinmanual():

#-----------------------------------------------------------------------

mloginwindow = Toplevel()

mloginwindow.title('VoiceVault')

mloginwindow.resizable(0,0)

#-----------------------------------------------------------------------

user32 = ctypes.windll.user32

user32.SetProcessDPIAware()

[w, h] = [user32.GetSystemMetrics(0), user32.GetSystemMetrics(1)]

lt = [w, h]

a = str(lt[0]//2-451)

b= str(lt[1]//2-339)

mloginwindow.geometry("902x678+"+a+"+"+b)

#-----------------------------------------------------------------------

img = Image.open(r"images/registerPage.png")

img = ImageTk.PhotoImage(img)

panel = Label(mloginwindow, image=img)

panel.pack(side="top", fill="both", expand="yes")

#-----------------------------------------------------------------------

EMAIL = StringVar()

PASSWORD = StringVar()

def login():

e,p=EMAIL.get(),PASSWORD.get()

print(e,p)

if e=='' or p=='':

tkMessageBox.showinfo('VoiceVault','All Fields Are Mendatory To Fill')

elif len(p)<6:

tkMessageBox.showinfo('VoiceVault','Password Length Must Be Greater Then Six')

PASSWORD.set('')

else:

Database()

cursor.execute('''

SELECT \* FROM users

WHERE(Name = %s and Password = %s)''', (e,p))

verifyLogin =cursor.fetchone()

try:

if(e in verifyLogin and p in verifyLogin):

tkMessageBox.showinfo(title="Login info", message="Logged In")

mloginwindow.destroy()

home(e)

except:

tkMessageBox.showinfo(title="Login info", message="Unable To Login !!!")

#-----------------------------------------------------------------------

def switchwin1():

mloginwindow.destroy()

voiceverificationlogin()

def switchwin2():

mloginwindow.destroy()

register()

def sysVerify():

steg.stegano()

l1 = Label(mloginwindow, text="Name : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l1.place(x=100, y=270)

l2 = Label(mloginwindow, text="Password : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l2.place(x=100, y=320)

e1 = Entry(mloginwindow, textvariable=EMAIL, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af")

e1.place(x=200, y=270)

e2 = Entry(mloginwindow, textvariable=PASSWORD, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af",show='\*')

e2.place(x=200, y=320)

b1 = Button(mloginwindow, text="LOGIN", width="21", bg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 1, borderwidth=0,font = ('Arial',15,'bold'), fg="#F5FAFE", command=login)

b1.place(x=100, y=370)

b2 = Button(mloginwindow, text="VERIFY SYSTEM", width="38", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white", command=sysVerify)

b2.place(x=94, y=420)

b3 = Button(mloginwindow, text="LOGIN USING VOICE VERIFICATION METHOD", width="40", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white", command=switchwin1)

b3.place(x=94, y=440)

b3 = Button(mloginwindow, text="NEW TO VOICEVAULT? REGISTER TO VOICEVAULT", width="40", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white", command=switchwin2)

b3.place(x=94, y=460)

mloginwindow.mainloop()

def register():

#-----------------------------------------------------------------------

regwindow = Toplevel()

regwindow.title('VoiceVault')

regwindow.resizable(0,0)

#-----------------------------------------------------------------------

user32 = ctypes.windll.user32

user32.SetProcessDPIAware()

[w, h] = [user32.GetSystemMetrics(0), user32.GetSystemMetrics(1)]

lt = [w, h]

a = str(lt[0]//2-451)

b= str(lt[1]//2-339)

regwindow.geometry("902x678+"+a+"+"+b)

#-----------------------------------------------------------------------

img = Image.open(r"images/registerPage.png")

img = ImageTk.PhotoImage(img)

panel = Label(regwindow, image=img)

panel.pack(side="top", fill="both", expand="yes")

#-----------------------------------------------------------------------

NAME = StringVar()

EMAIL = StringVar()

PASSWORD = StringVar()

#-----------------------------------------------------------------------

def addv():

n,e,p=(NAME.get()).lower(),EMAIL.get(),PASSWORD.get()

fname = n+'.gmm'

flist = os.listdir('./')

if n=='' or e=='' or p=='':

tkMessageBox.showinfo('VoiceVault','All Fields Are Mendatory To Fill')

else:

Name =n

FORMAT = pyaudio.paInt16

CHANNELS = 1

RATE = 44100

CHUNK = 512

RECORD\_SECONDS = 10

device\_index = 2

audio = pyaudio.PyAudio()

index = 1

stream = audio.open(format=FORMAT, channels=CHANNELS,

rate=RATE, input=True,input\_device\_index = index,

frames\_per\_buffer=CHUNK)

b1["text"]="Speak Now"

b1.config(command=None)

Recordframes = []

for i in range(0, int(RATE / CHUNK \* RECORD\_SECONDS)):

data = stream.read(CHUNK)

Recordframes.append(data)

b1["text"]="Audio Sample Taken"

OUTPUT\_FILENAME=Name+"-sample"+".wav"

WAVE\_OUTPUT\_FILENAME=os.path.join("training\_set",OUTPUT\_FILENAME)

trainedfilelist = open("training\_set\_addition.txt", 'w')

trainedfilelist.write(OUTPUT\_FILENAME+"\n")

trainedfilelist.close()

stream.stop\_stream()

stream.close()

audio.terminate()

waveFile = wave.open(WAVE\_OUTPUT\_FILENAME, 'wb')

waveFile.setnchannels(CHANNELS)

waveFile.setsampwidth(audio.get\_sample\_size(FORMAT))

waveFile.setframerate(RATE)

waveFile.writeframes(b''.join(Recordframes))

waveFile.close()

train\_model()

#-----------------------------------------------------------------------

def addvoice():

x = T.Thread(target=addv)

x.start()

def reg():

n,e,p=(NAME.get()).lower(),EMAIL.get(),PASSWORD.get()

fname = n+'.gmm'

flist = os.listdir('./')

if n=='' or e=='' or p=='':

tkMessageBox.showinfo('VoiceVault','All Fields Are Mendatory To Fill')

elif '@' not in e or '.' not in e:

tkMessageBox.showinfo('VoiceVault','Please Enter A Valid Email Address')

EMAIL.set('')

elif len(p)<6:

tkMessageBox.showinfo('VoiceVault','Password Length Must Be Greater Then Six')

PASSWORD.set('')

elif fname not in flist:

tkMessageBox.showinfo('VoiceVault','Please Add Voice Sample')

else:

Database()

print("---",n,e,p)

cursor.execute('''INSERT INTO users(Name, Username, Password) VALUES(%s,%s,%s)''' , (n,e,p))

create\_user\_table(n)

mydb.commit()

tkMessageBox.showinfo('VoiceVault','User Registered Successfully')

regwindow.destroy()

voiceverificationlogin()

#-----------------------------------------------------------------------

def switchwin3():

regwindow.destroy()

voiceverificationlogin()

l1 = Label(regwindow, text="Name : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l1.place(x=100, y=250)

l2 = Label(regwindow, text="Email ID : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l2.place(x=100, y=280)

l3 = Label(regwindow, text="Password : ", bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

l3.place(x=100, y=310)

l4=Label(regwindow,text="Read the phrase :",bg="white",font=('Arial',13,'bold'), fg="#2a72af")

l4.place(x=100,y=340)

e1 = Entry(regwindow, textvariable=NAME, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af")

e1.place(x=200, y=250)

e2 = Entry(regwindow, textvariable=EMAIL, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af")

e2.place(x=200, y=280)

e3 = Entry(regwindow, textvariable=PASSWORD, bg="#e6f2ff",width = 17,font = ('Arial',13,'bold'), fg="#2a72af")

e3.place(x=200, y=310)

m=pyjokes.get\_joke(language="en", category="neutral")

e4 = Label(regwindow, text=m, bg="white",font = ('Arial',13,'bold'), fg="#2a72af")

e4.place(x=70, y=380)

b1 = Button(regwindow, text="ADD THE VOICE SAMPLE", width="21", bg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 1, borderwidth=0,font = ('Arial',15,'bold'), fg="#F5FAFE", command=addvoice)

b1.place(x=100, y=490)

b2 = Button(regwindow, text="REGISTER", width="21", bg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 1, borderwidth=0,font = ('Arial',15,'bold'), fg="#F5FAFE", command=reg)

b2.place(x=100, y=530)

b3 = Button(regwindow, text="ALREADY REGISTERD? LOGIN TO VOICEVAULT", width="38", fg="#1E1D6C", highlightthickness = 0,relief = FLAT, bd = 0, borderwidth=0,font = ('Arial',9,'bold'), bg="white",activebackground ="white", command=switchwin3)

b3.place(x=94, y=570)

regwindow.mainloop()

def home(username):

global table\_name

table\_name = username

def load\_key():

with open(key\_path, 'rb') as key\_file:

key = key\_file.read()

print(key)

return key

def encrypt(file):

filename = file.split("/")[-1]

file\_ext = filename.split(".")[-1]

print(filename)

if (file\_ext == "jpg" or file\_ext == "jpeg" or file\_ext == "png"):

path = file

key = 21

# open file for reading purpose

fin = open(path, 'rb')

# storing image data in variable "image"

image = fin.read()

fin.close()

# converting image into byte array to

# perform encryption easily on numeric data

image = bytearray(image)

# performing XOR operation on each value of bytearray

for index, values in enumerate(image):

image[index] = values ^ key

# opening file for writting purpose

fin = open("enc\_" + filename, 'wb')

# writing encrypted data in image

fin.write(image)

fin.close()

print('Encryption Done...')

elif file\_ext == "txt":

f1 = open("enc\_" + filename, "w")

with open(file) as f:

data = f.read()

for i in data:

sc = 0

enc\_char = ord(i)

if (enc\_char % 2 != 0):

enc\_char += 1

sc = 224

enc\_char = enc\_char // 2

enc\_char = chr(enc\_char)

if (sc == 0):

f1.write(enc\_char)

else:

f1.write(enc\_char + chr(sc))

print(i, "Encrypt to ", enc\_char, ord(enc\_char))

f1.close()

# encrypt("x.txt")

def decrypt(file):

filename = file.split("/")[-1]

file\_ext = filename.split(".")[-1]

if (file\_ext == "jpg" or file\_ext == "jpeg" or file\_ext == "png"):

path = file

key = 21

fin = open(path, 'rb')

# storing image data in variable "image"

image = fin.read()

fin.close()

# converting image into byte array to perform decryption easily on numeric data

image = bytearray(image)

# performing XOR operation on each value of bytearray

for index, values in enumerate(image):

image[index] = values ^ key

# opening file for writting purpose

print(filename)

fin = open(D\_path + "dec\_" + filename, 'wb')

# writing decryption data in image

fin.write(image)

fin.close()

print('Decryption Done...')

elif file\_ext == "txt":

f1 = open(D\_path + "dec\_" + filename, "w")

with open(file) as f:

data = f.read()

for i in range(len(data)):

enc\_char = ord(data[i]) \* 2

if (ord(data[i]) == 224):

continue

try:

if (ord(data[i + 1]) == 224):

enc\_char = enc\_char - 1

except:

pass

enc\_char = chr(enc\_char)

f1.write(enc\_char)

# print(i,"Decrypt to ",enc\_char)

def convertToBinaryData(filename):

# Convert digital data to binary format

with open(filename, 'rb') as file:

binaryData = file.read()

return binaryData

def getUrlnName(fileurl):

all\_f = fileurl.split('/')

url = ""

filename = all\_f[-1]

for i in range(len(all\_f) - 1):

url += all\_f[i] + '/'

return url, filename

def upload\_file(fileurl):

global statusbar

path, filename = getUrlnName(fileurl)

print(path, filename)

encrypt(fileurl)

file\_ext = filename.split('.')[-1]

filename = "enc\_" + filename

print(filename)

if (file\_ext == "jpg" or file\_ext == "png" or file\_ext == "jpeg"):

copyfile(filename, "C:/Users/nitis/Desktop" + filename)

fileurl = filename

try:

mycursor = mydb.cursor()

file\_data = convertToBinaryData(fileurl)

query = """INSERT INTO """ + table\_name + """ (FileName,File) VALUES (%s,%s) """

values\_tuple = (filename, file\_data)

mycursor.execute(query, values\_tuple)

mydb.commit()

# print(mycursor)

print("Successfully File Uploaded.")

tkMessageBox.showinfo("Success", "Successfully Uploaded")

statusbar['text'] = ''' UPLOADED '''

except Exception as e:

print(e)

tkMessageBox.showwarning("Warning!", e)

def getImageFile(filename):

D\_path = "Downloads\\"

print(D\_path + filename)

copyfile("C:/Users/nitis/Desktop" + filename, D\_path + filename)

def write\_file(data, filename):

# Convert binary data to proper format and write it on Hard Disk

with open(filename, 'wb') as file:

file.write(data)

def download\_file(id=None):

D\_path = "./Downloads/"

try:

mycursor = mydb.cursor()

query = """select \* from """ + table\_name + """ where ID=""" + str(id)

mycursor.execute(query)

for i in mycursor:

print("id : ", i[0])

print("filename:", i[1])

# filename = i[i]

# print('File data:',i[2])

filename = i[1]

file\_ext = filename.split('.')[-1]

if (file\_ext == "png" or file\_ext == "jpg" or file\_ext == "jpeg"):

getImageFile(filename)

else:

write\_file(i[2], D\_path + i[1])

decrypt(D\_path + i[1])

print("\n")

# print(mycursor)

print("Successfully Files Saved.")

tkMessageBox.showinfo("Success", "Successfully Downloaded")

D\_path = "./Downloads/"

os.remove(D\_path + filename)

except Exception as e:

print(e)

tkMessageBox.showwarning("Warning!", e)

# ----------------------------------------------------------------------

window = Toplevel()

window.title('VoiceVault')

window.resizable(0, 0)

# -----------------------------------------------------------------------

user32 = ctypes.windll.user32

user32.SetProcessDPIAware()

[w, h] = [user32.GetSystemMetrics(0), user32.GetSystemMetrics(1)]

lt = [w, h]

a = str(lt[0] // 2 - 525)

b = str(lt[1] // 2 - 345)

window.geometry("1050x640+" + a + "+" + b)

window.config(bg='#203647')

# ---------------------------------------------------------------------------

def upload():

global filename, statusbar

canvas = Canvas(window, height=800, width=847, highlightthickness=0, bd=0, bg='#203647')

canvas.place(x=0, y=0)

statusbar = Label(canvas, font=("Arial", 16, "italic bold"), background='#12232E', fg='#EEFBFB', width=65, height=1,

highlightthickness=0, bd=3)

statusbar.place(x=0, y=610)

statusbar["text"] = '''Currently Selected File : None'''

def browse():

global filename

filename = filedialog.askopenfilename(initialdir=os.getcwd(), title="Select file",

filetypes=(("All Files", "\*.\*"), ("Text Files", ".txt")))

f = filename.split('/')[-1]

lt = list(f)

c = 0

x = ''

for i in lt:

c += 1

if c < 40:

x += i

if c == 40:

x = x + '...'

break

statusbar["text"] = '''Currently Selected File : ''' + x

extension = "." + f.split(".")[-1]

vid\_fm = [".flv", ".avi", ".mp4", ".3gp", ".mov", ".webm", ".ogg", ".qt", ".avchd"]

aud\_fm = [".flac", ".mp3", ".wav", ".wma", ".aac"]

if extension in vid\_fm:

tkMessageBox.showerror(title="Video File!",

message="Free version of this server doesn't allow storing of video file!")

browse()

elif extension in aud\_fm:

tkMessageBox.showerror(title="Audio File!",

message="Free version of this server doesn't allow storing of audio file!")

browse()

heading = Label(window, text="--- Encrypt And Upload File On Server ---", fg="#EEFBFB",font=('Calibiri', 28, 'bold'), bg="#203647")

heading.place(x=70, y=30)

subheading = Label(window, text="-: Browse File :-", fg="#EEFBFB", font=('Calibiri', 18, 'bold'), bg="#203647")

subheading.place(x=330, y=120)

imgfile = Image.open(ImagePath+'browse.png')

imgfile = imgfile.resize((300, 300))

browse\_img = ImageTk.PhotoImage(imgfile)

panel2 = Button(canvas, image=browse\_img, bg="#203647", highlightthickness=0, bd=0, activebackground="#203647",

command=browse)

panel2.image = browse\_img

panel2.place(x=290, y=200)

b = Button(window, text="Upload File", width=20, bg="#EEFBFB", highlightthickness=0, font=('Calibiri', 24, 'bold'),

bd=0, fg="#203647", activebackground="#4DA8DA", command=lambda: upload\_file(filename))

b.place(x=250, y=450)

def download():

global flag, id\_selected

global statusbar

flag = False

canvas = Canvas(window, height=800, width=847, highlightthickness=0, bd=0, bg='#203647')

canvas.place(x=0, y=0)

# Creating A View Item Window

Topwindow = Frame(canvas, bd=5, relief=SOLID)

Topwindow.pack(side=TOP, fill=X)

Midwindow = Frame(canvas, width=500, bg='#203647')

Midwindow.pack()

lbl\_text = Label(Topwindow, text="Avilable Files On Server", font=('calibri', 23, 'bold'), width=52, bg="#203647",

fg='white')

lbl\_text.pack()

scrollbarx = Scrollbar(Midwindow, orient=HORIZONTAL)

scrollbary = Scrollbar(Midwindow, orient=VERTICAL)

tree = ttk.Treeview(Midwindow, columns=('ID', 'FileName'), selectmode="extended", height=24, yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)

scrollbary.config(command=tree.yview)

scrollbary.pack(side=RIGHT, fill=Y)

scrollbarx.config(command=tree.xview)

scrollbarx.pack(side=BOTTOM, fill=X)

style = ttk.Style(Midwindow)

# set ttk theme to "clam" which support the fieldbackground option

style.theme\_use("clam")

style.configure("Treeview",background="white", fieldbackground="white", foreground="#203647", font=(3))

tree.heading('ID', text="ID", anchor=W)

tree.heading('FileName', text="Filename", anchor=W)

tree.column('#0', stretch=NO, minwidth=0, width=0)

tree.column('#1', stretch=NO, minwidth=0, width=70)

tree.column('#2', stretch=NO, minwidth=0, width=750)

def selected(a):

global flag, id\_selected

flag = True

curItem = tree.focus()

try:

id\_selected = tree.item(curItem)['values'][0]

except:

pass

def download\_item():

global flag, id\_selected

if flag == True:

download\_file(id\_selected)

flag = False

else:

tkMessageBox.showerror(title="File Not Selected", message="Select File to Download !")

def view\_files():

try:

mycursor = mydb.cursor()

query = """select \* from """ + table\_name

mycursor.execute(query)

for i in mycursor:

lt = [i[0], i[1]]

tree.insert('', 'end', values=(lt))

except Exception as e:

print(e)

tree.bind('<ButtonRelease-1>', selected)

tree.pack()

Button(Midwindow, text='Download', font=('calibri', 21, 'bold'), width=54, bg="#203647", fg='white',

command=download\_item).pack()

thread = threading.Thread(target=view\_files)

thread.start()

def about():

canvas = Canvas(window, height=800, width=847, highlightthickness=0, bd=0, bg='#203647')

imgq = Image.open(ImagePath+"abc.png")

image1 = ImageTk.PhotoImage(imgq)

panel1 = Label(canvas, image=image1, highlightthickness=0, bd=0)

panel1.image = image1 # keep a reference

panel1.pack(side='top', fill='both', expand='yes')

canvas.place(x=0, y=0)

def Exit():

# Function To Get Pop Up A Exit Window

result = tkMessageBox.askquestion('VoiceVault', 'Are you sure you want to exit?', icon="warning")

if result == 'yes':

window.destroy()

exit()

else:

tkMessageBox.showinfo('Return', 'You will now return to the application screen')

def logout():

window.destroy()

voiceverificationlogin()

canvas = Canvas(window, height=800, width=700, highlightthickness=0, bd=0, bg='#EEFBFB')

canvas.place(x=848, y=0)

head = Label(window, text=" ````VOICEVAULT````", bg="#EEFBFB", font=('Calibiri', 17, 'bold'), fg="#203647")

head.place(x=850, y=10)

file\_img = Image.open(ImagePath+'icon.png')

file\_img = file\_img.resize((195, 275))

icon\_img = ImageTk.PhotoImage(file\_img)

panel = Label(window, image=icon\_img, bg="#EEFBFB")

panel.place(x=850, y=60)

b1 = Button(window, text="Upload", width=14, fg="#EEFBFB", highlightthickness=0, font=('Calibiri', 17, 'bold'), bd=0, bg="#203647", activebackground="#EEFBFB", command=upload)

b1.place(x=849, y=350)

b2 = Button(window, text="Download", width=14, fg="#EEFBFB", highlightthickness=0, font=('Calibiri', 17, 'bold'), bd=0,

bg="#203647", activebackground="#EEFBFB", command=download)

b2.place(x=849, y=412)

b3 = Button(window, text="About", width=14, fg="#EEFBFB", highlightthickness=0, font=('Calibiri', 17, 'bold'), bd=0, bg="#203647", activebackground="#EEFBFB", command=about)

b3.place(x=849, y=474)

b4 = Button(window, text="Logout", width=14, fg="#EEFBFB", highlightthickness=0, font=('Calibiri', 17, 'bold'), bd=0, bg="#203647", activebackground="#EEFBFB", command=logout)

b4.place(x=849, y=536)

b4 = Button(window, text="Exit", width=14, fg="#EEFBFB", highlightthickness=0, font=('Calibiri', 17, 'bold'), bd=0, bg="#203647", activebackground="#EEFBFB", command=Exit)

b4.place(x=849, y=588)

about()

window.mainloop()

# sysVerify()

voiceverificationlogin()

**Chapter 8**

**TESTING**

**8.TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**8.1 Types of Testing**

**8.1.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches an internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic test at component level and test a specific business process, application, and system configuration. Unit test ensures that each unit path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.1.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens of fields. Integration test demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**8.1.3 Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user

manuals. Functional testing is centered on the following items:

• Valid input: Identified classes of valid input must be accepted.

• Invalid input: Identified classes of invalid input must be rejected

• Functions: Identified functions must be exercised.

• Output: Identified classes of application outputs must be exercised.

• Systems/Procedures: Interfacing systems or procedures must be invoked.

Organizations and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify business process flows; Data fields, pre-defined processes, and successive processes must be considered for testing. Before functional test is complete, additional tests are identified and the effective value of current test is determined.

**8.1.4 System testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system

testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven links and integration points.

**8.1.5 White box testing**

White box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at ease its purpose. It is used to test areas that cannot be reached from a black box level.

**8.1.6 Black box Testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system, acceptance.

**8.2 Test Procedures**

Specific knowledge of the application's code, internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

**8.3 Test Cases**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test Objectives**

1. All field entries must work properly.

2. Pages must be activated from the identical link.

3. The entry screen, messages and responses must not be delayed.

**Features to be tested**

1. Verify that the entries are of the correct format.

2. Verify that the entries are within the range.

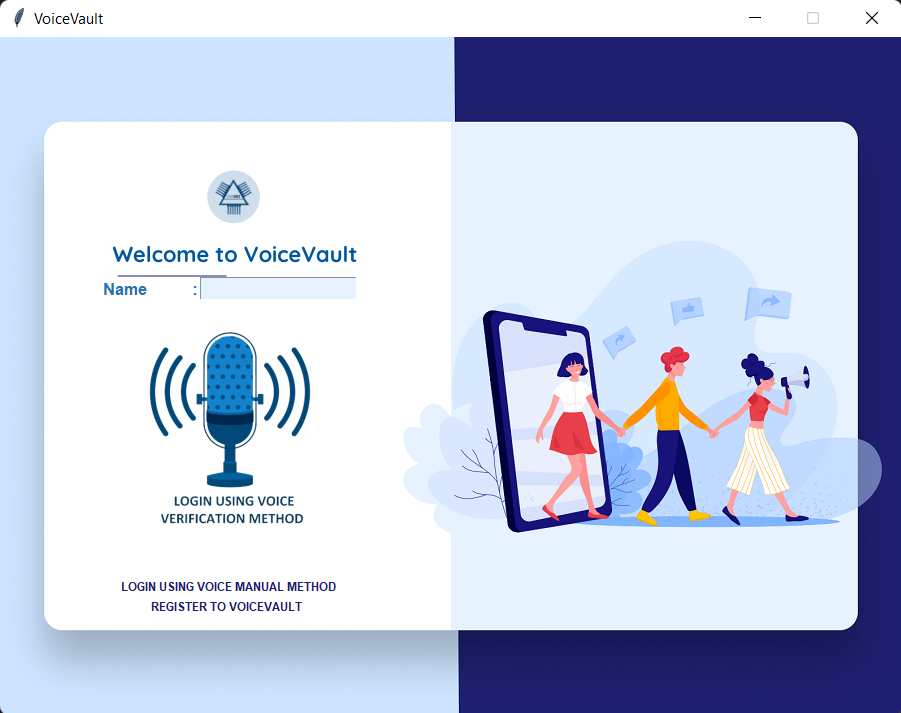
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case | Test Case | Test Case | Test Steps | | | Test | Test |
| id |  | Description | Step | Expected | Actual | Case | Priority |
|  |  |  | Status |  |
|  |  |  |  |  |
| 01 | Start the | Host the | If it | We | The | High | High |
| Applicatio | application | doesn't | Cannot | application |
| N | and test if it | Start | run the | hosts |
|  | Starts |  | applicati | success. |
|  | making sure |  | on. |  |
|  | the required |  |  |  |
|  | software is |  |  |  |
|  | available |  |  |  |
| 02 | Home Page | Check the | If it | We | The | High | High |
| deployment | doesn‘t | Cannot | application |
| environmen | load. | Access | is running |
| t for |  | The | successfully |
| properly |  | applicati | . |
| loading the |  | on. |  |
| application. |  |  |  |
| 03 | User | Verify the | If it | We | The | High | High |
| Mode | working of | doesn‘t | Cannot | application |
|  | The | Respond | use the | displays the |
|  | application |  | Freestyle | Freestyle |
|  | in freestyle |  | mode. | Page |
|  | Mode |  |  |  |
| 04 | Data Input | Verify if the | If it fails | We | The | High | High |
| application | to take the | Cannot | application |
| takes input | input or | Proceed | updates the |
| and updates | store in | Further | input to application |
|  | The |  |
|  | Database |  |  |

**8.3.1** **TEST CASES**

**Table 8.1**

**8.4. Experimental Results**

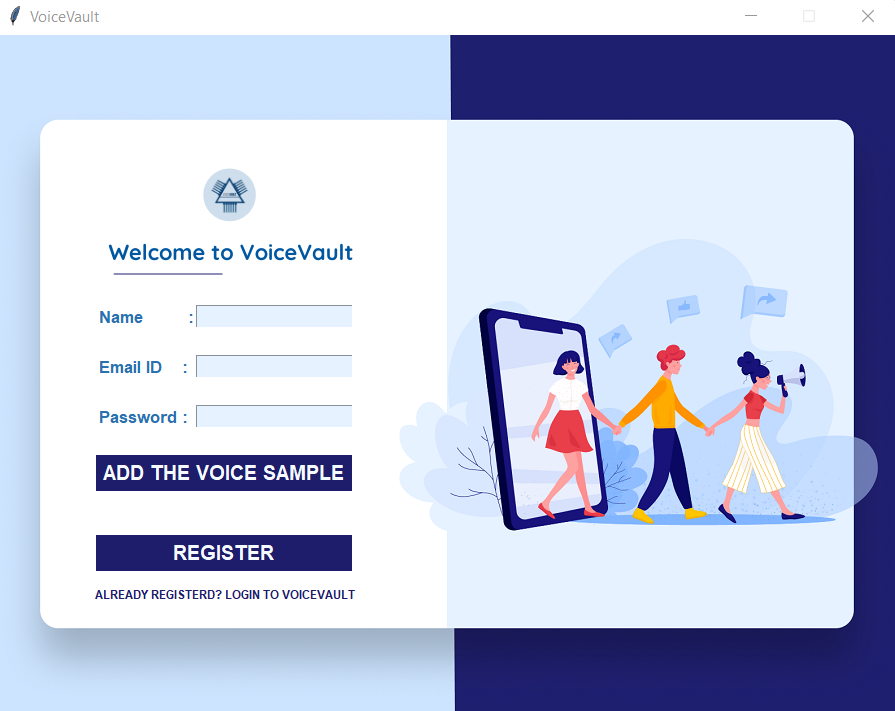
**8.4.1 Login**

 The below figure shows the Login Window where the user authenticates himself with their voice.

**Figure** **8.2 Login**

**8.4.2 Registration**

The below figure shows the Registration Window where the user register himself with the name, email-id, password and with their voice.



**Figure 8.3 Registration**

**8.4.3 Upload**

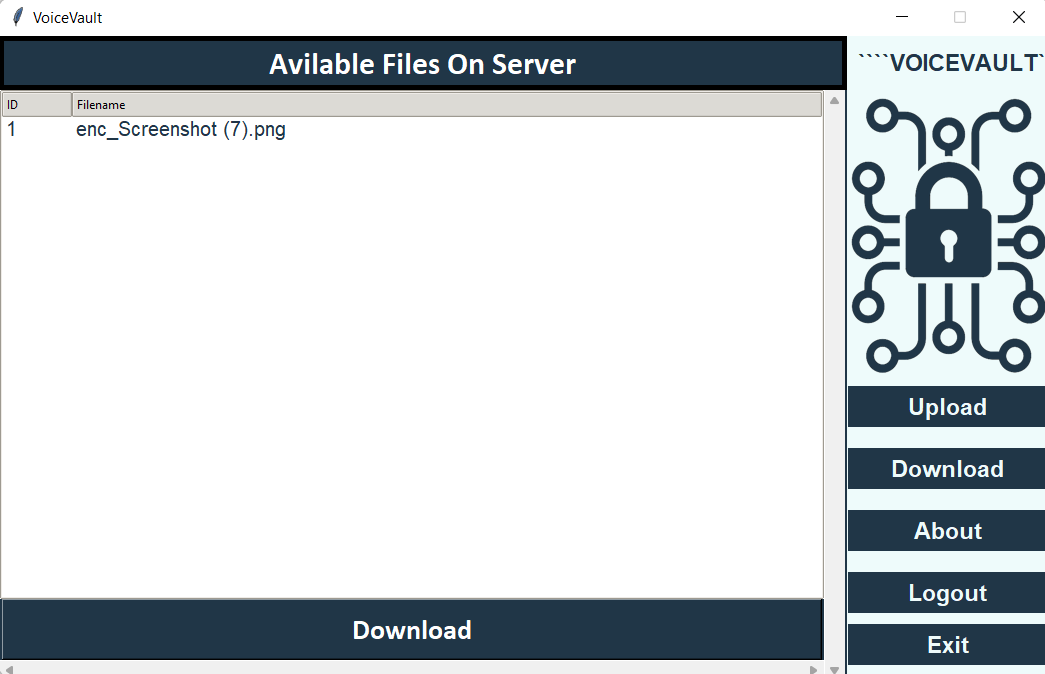
The below figure shows the Upload section where the user upload his important file which is then stored in cloud database in the encrypted format.



**Figure 8.4 Upload**

**8.4.4 Downloading**

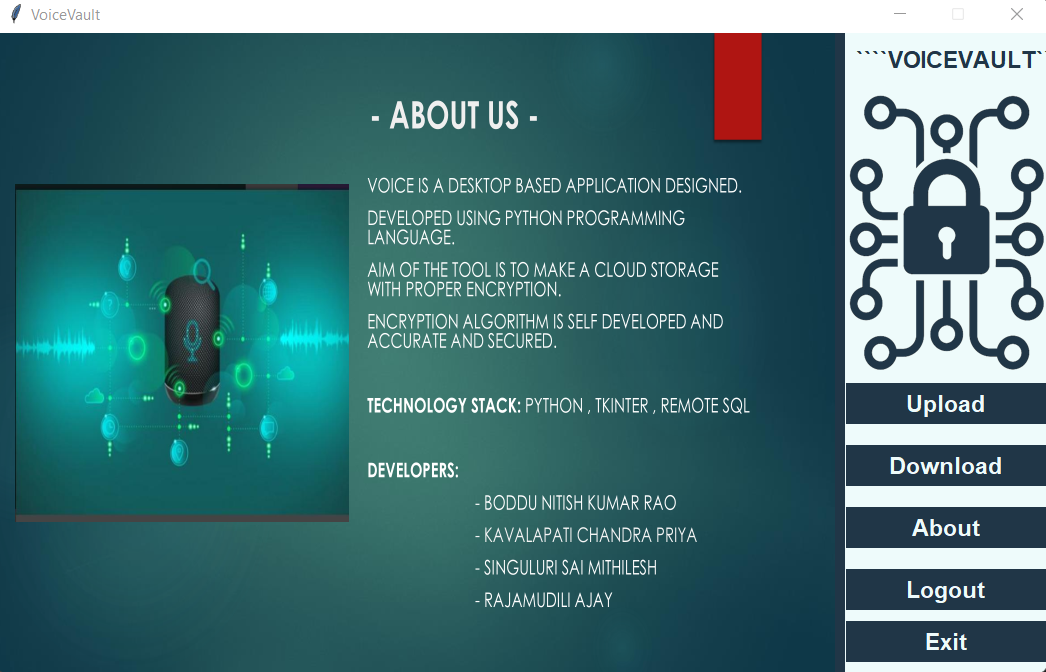
The below figure shows the Download section where the user download his important file from the cloud database.



**Figure 8.5 Download**

**8.4.5 About**

The below figure shows the About section which is about our project.



**Figure 8.6 About**

**CHAPTER 9**

**CONCLUSION AND FUTURE SCOPE**

**9.1 CONCLUSION AND FUTURE SCOPE**

This part closes the proposed framework. Fostering a constant voice acknowledgment system is planned. For highlight extraction, it utilizes MFCC, and for preparing, it utilizes GMM. The voice is first caught utilizing a receiver, and afterward voice highlights are recovered. The Hamming window is utilized to diminish discontinuities at the casing's edge, bringing about smooth recurrence transmission in discourse signals. Mel Recurrence Cepstral Coefficients produces 15 MFCC coefficients utilizing 40 Mel channels. These coefficients are hence shipped off GMM to be utilized in the preparation stage. The speaker is recognized by contrasting the log likelihood with the framework's predefined edge. It diminishes how much work expected to recall the passwords in general. The application helps with the security of all text records, photographs, recordings, and different media. It keeps unapproved clients from getting entrance. The application is made considerably safer by involving a voice security strategy for verification and afterward encoding the records. Cryptography innovation (AES Encryption) expands the security framework by basically dispensing with the chance of a split the difference. Accordingly, another security arrangement has been fostered that permits you to lock and open your records while likewise putting away them in the cloud. This program is likewise incredibly easy to understand and practical.

**9.2 REFERENCES**

* A Programmed Speaker Acknowledgment Framework Jun 2005
* Do, Minh N. "A Programmed Speaker Acknowledgment Framework", Computerized Signal Handling Smaller than normal Task. 14 June 2005.
* Md. Rashidul Hasan, Mustafa Jamil, Md. Golam Rabbani and Md. Saifur Rahman, "Speaker ID utilizing Mel recurrence Cepstral coefficients", third Worldwide Gathering on Electrical and PC Designing ICECE 2004, 28-30 December 2004.
* Productive Speaker Recognizable proof and Recovery Examination of various boundaries utilized in GMM based programmed voice acknowledgment Aug 2011
* Hagai Aronowitz, David Burshtein, " Productive Speaker Recognizable proof and Recovery http://en.wikipedia.org/wiki/MATLAB. http://www.mathworks.com/help/pdf\_doc/matlab/getstart.pdf.