1.INTRODUCTION

1.INTRODUCTION

The purpose of this project is to create an AI enabled voice assistant for PESU which will allow students to get their information via voice commands, this is going to make PESU access easier and user friendly, as they will be able to get MARKS, ATTENDANCE, COURSES INFO and RESUTLS easily using voice commands only.

The requirements for voice assistant are students can get their information without going into website and just by talking to the assistant, which will fetch their data directly from the database without going in depths of the website.

The product is a PESU of software which is going to:

- Speech recognition: The assistant will have ability to understand and interpret spoken language.
- Natural language processing: The assistant will have ability to understand and interpret human language, including the nuances of grammar and context.
- Text-to-speech synthesis: The assistant will have ability to convert written text into spoken language.
- Voice commands: The assistant will have ability to respond to spoken commands and perform tasks based on those commands.
- Personalization: The ability to recognize individual users and provide personalized responses based on their preferences and past behaviour.

2.PROBLEM STATEMENT

Voice assistants can help individuals manage their personal lives by setting reminders, scheduling appointments, showing results, reminding about attendance and read notifications to students. The scope of a PESU voice assistant is vast and continues to expand as new use cases and functionalities are developed. The potential for voice assistants is limited only by the imagination of developers and the needs of users.

PRODUCT PERSPECTIVE:

Integration: PESU voice assistant will be seamlessly integrated into the product ecosystem, whether it is a standalone device or part of a larger system.

Compatibility: PESU voice assistant will be designed to work with a wide range of devices and platforms, including smartphones, tablets, smart speakers, and other connected devices.

User interface: The user interface of our PESU voice assistant will be intuitive and easy to use, allowing users to interact with the device using natural language commands.

Functionality: PESU voice assistant will be designed to perform a wide range of tasks, from simple tasks like setting reminders and playing music to more complex tasks.

Performance: PESU voice assistant will be designed to provide fast, accurate, and reliable performance, even in complex and dynamic environments

3. LITERATURE REVIEW

LITERATURE REVIEW

- **DATA EXTRACTION:** The data extraction component will help in extraction of data from pdf format, image documents to extract data from image has to be in good quality. That's why we are using Logistic Regression to classify image as good or bad.
- For Extraction of data from. We are going to use OpenCV and PY tesseract for text extraction from each cell.
- Then Cells are detected and text are extracted and CSV/JSON is generated
- Logistic regression is used for checking quality of image
- Performance Is optimized by creating a background by calculating the height and width from all the cell by ROI crops

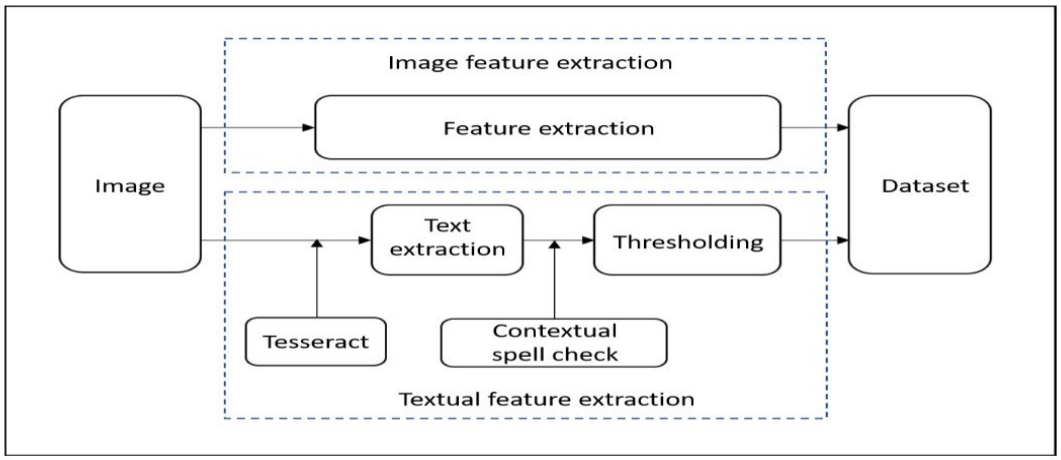


Fig. 3: Schematic flow of dataset compilation.

NARRATION:

The voice assistant will narrate every information the user will ask from it regarding himself.

Working:

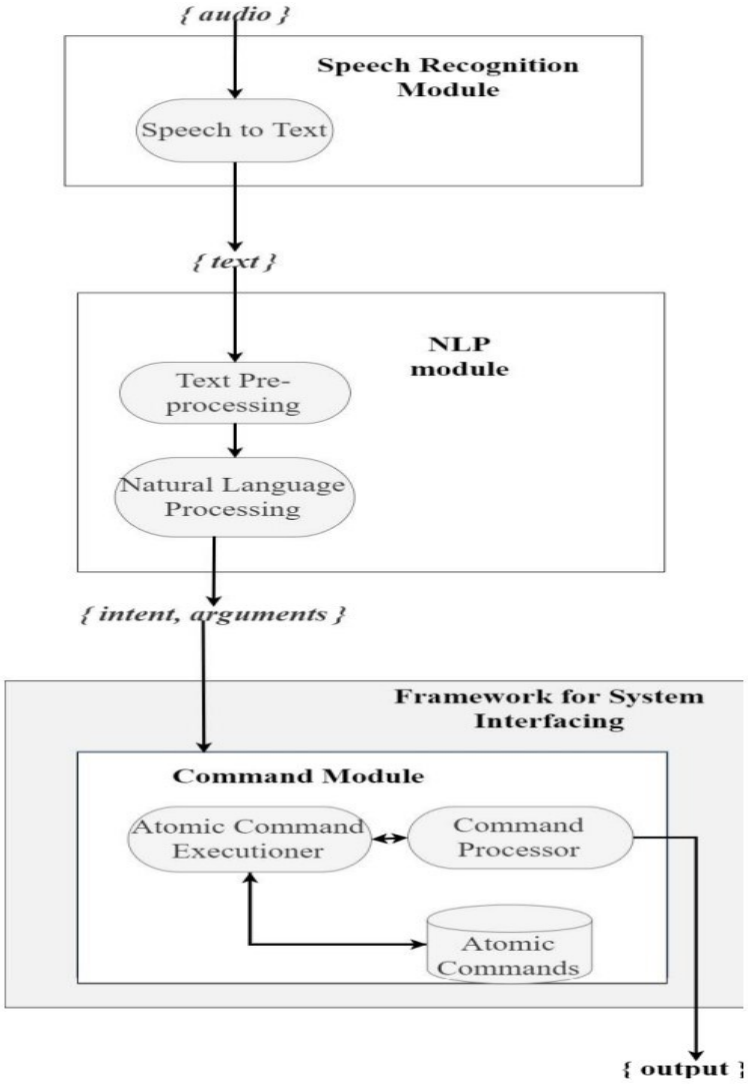
- Audio is broken down into 0.125 seconds clip, fed to feature extraction algorithm
- Then, weights are added and passed to KNN, KNN classifies as human voice and non-human voice

Audio Features: -

- Mel-frequency Cepstral Coefficients
- Speectral Roll – Off
- Speectral Centroid
- Zeroo Crossing Rate

The Speech Recognizing library will be used in 2 variants :

- online and offline modes, only difference is in online mode audio is passed to Certain API’s
- on the other hand, offline mode makes use of the popular “CMU Sphinx” library for conversion of speech to text.


{ audio }
Speech Recognition Module
Speech to Text
{ text }
NLP module
Text Pre-processing
Natural Language Processing
{ intent, arguments }
Framework for System Interfacing
Command Module
Atomic Command Executioner
Command Processor
Atomic Commands
{ output }

DATABASE HANDLING:

The data will be fetched directly from the PESU dummy database (which we are going to create then we'll put into production) , via queries generated by the assistant

Working:

- The assistant will generate queries from the voice model and every word will be tokenized as verbs and nouns
- The words will then be checked from selected key-words to fetch the information from the database

Security component:

- Since the voice assistant will handle lots of student's data and handling and managing of confidential data is a risk factor
- So different encrypting algorithms will be used as means of keeping data secure from the intruders
- The voice assistant model will keep the data secure within database and itself

4.DATA

4.1 Overview

We need to Capture the voice of student and sent it to server and Server will do the following steps

- Pre-processing Voice to remove noise from it.
- Convert it to text.
- Convert it into query, execute and store the result into database. Cache if necessary

4.2 Types of data required for the system

4.2.1 Input Data

- We mainly Require Data of the students in some kind of format.
- And Voice Command of Students. The size depends mainly on the Command and No. of Queries (ex: - Get me my latest GPA and What were my ISA Marks and their average)

4.2.2 Model Architecture

- **Tokenization: -** Spacy, NLTK.
- **PESU: -** HMM Model
- **Named Entity: -** CRF (Conditional Random Field)

4.2.3 Training Details

- **We're Going to use Pre-trained Models.**

4.2.4 Performance Metrics

- Accuracy of models are roughly 84% of all models

5. SYSTEM REQUIREMENTS SPECIFICATION

5. SYSTEM REQUIREMENTS SPECIFICATION

5.1 Product Perspective

Integration: PESU voice assistant will be seamlessly integrated into the product ecosystem, whether it is a standalone device or part of a larger system.

Compatibility: PESU voice assistant will be designed to work with a wide range of devices and platforms, including smartphones, tablets, smart speakers, and other connected devices.

User interface: The user interface of our PESU voice assistant will be intuitive and easy to use, allowing users to interact with the device using natural language commands.

Functionality: PESU voice assistant will be designed to perform a wide range of tasks, from simple tasks like setting reminders and playing music to more complex tasks.

Performance: PESU voice assistant will be designed to provide fast, accurate, and reliable performance, even in complex and dynamic environments.

5.1.1 Product Features

- Speech recognition: The assistant will have ability to understand and interpret spoken language.
- Natural language processing: The assistant will have ability to understand and interpret human language, including the nuances of grammar and context.
- Text-to-speech synthesis: The assistant will have ability to convert written text into spoken language.
- Voice commands: The assistant will have ability to respond to spoken commands and perform tasks based on those commands.
- Personalization: The ability to recognize individual users and provide personalized responses based on their preferences and past behavior.

.1.2 Operating Environment

1. Power source: Many voice assistants are designed to be always-on and require a stable power source to operate reliably.
2. Internet connectivity: A voice assistant needs a reliable and stable internet connection to function properly.
3. Compatibility with devices: Voice assistants are designed to work with various devices such as smartphones, smart speakers.
4. Access to data: To function effectively, a voice assistant needs access to data such as user preferences, history, and other relevant information.

2. Functional Requirements

- Voice Authorization
- Generating Intents and Arguments Then sending Arguments to database Caching most asked Queries
- NLP module for extracting intents and arguments
- Contextual awareness

3. External Interface Requirements

1. The project will be GUI based and will also support voice commands.
2. It will majorly be Voice based commands/Inputs

3.1 Hardware Requirements

- Any device with internet connection and has a screen with working microphone can use the voice assistant.

3.2 Software Requirements

- **None**

4. Non-Functional Requirements

- Security for student's Data
- It must have user friendly UI
- Must Be Accurate with the information it provides
- And Must authorize Correct user according to their SRN

4.1 Performance Requirement

- Response Time: The voice interface must respond to user requests within a specific time frame, usually measured in seconds or milliseconds.
- Throughput: The voice assistant must be able to handle a certain number of requests or transactions per second, depending on the expected usage.
- Availability: The assistant must be available for use during certain hours of the day, with a defined percentage of uptime (e.g., 99.9% uptime).
- Scalability: The project must be able to handle an increasing number of users or requests without sacrificing performance or reliability.
- Resource utilization: The project must use system resources efficiently, such as CPU, memory, and network bandwidth, to avoid performance degradation or system crashes.

4.1 Compatibility Requirements

- Security and Data Integrity: The assistant must be secure, with measures in place to prevent data breaches or unauthorized access, and must ensure the integrity of data.

- Maintenance and Support: The assistant must be easy to maintain, and support should be readily available in case of issues or bugs.

4.2 Usability Requirements

None, as It will be easy to use via voice commands only

6. SYSTEM DESIGN

6.SYSTEM DESIGN

6.1 Design Considerations

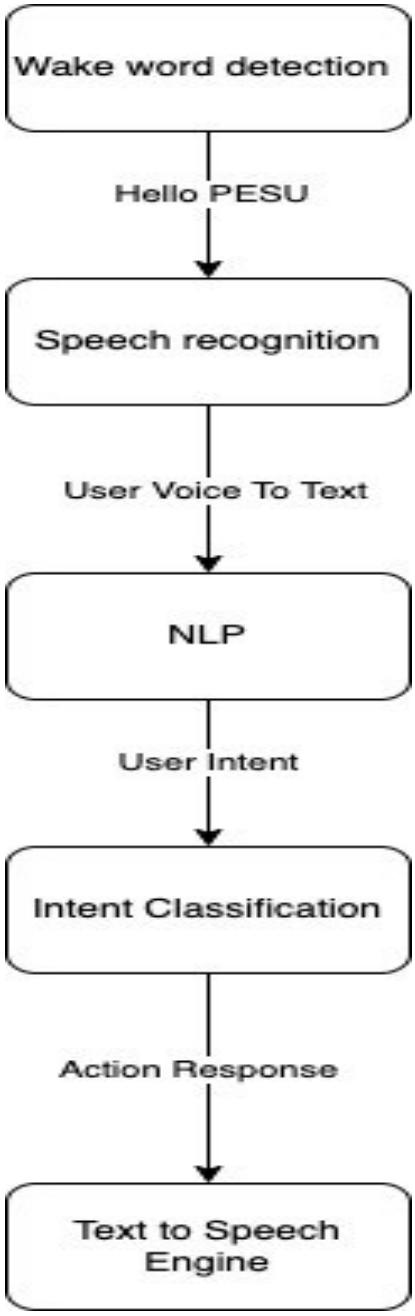
6.1.1 Design Goals

The main goal of designing a project is to describe the working process which will ease the work of developer. We are giving an analogy of how the components of product will be working.

Although the following model is predicated on one, it is not actually based on a client-server architecture.

6.1.2 Architecture Choices

We have Several choices when it comes to architecture, as voice assistant are booming these days. But we have decided to go with simplest approach below is diagram of our choice


Wake word detection
Hello PESU
Speech recognition
User Voice To Text
NLP
User Intent
Intent Classification
Action Response
Text to Speech Engine

6.1.3 Constraints, Assumptions and Dependencies

Assumptions made

- PESU will allow us to use Their Website/database.
- This is only limited to Students only

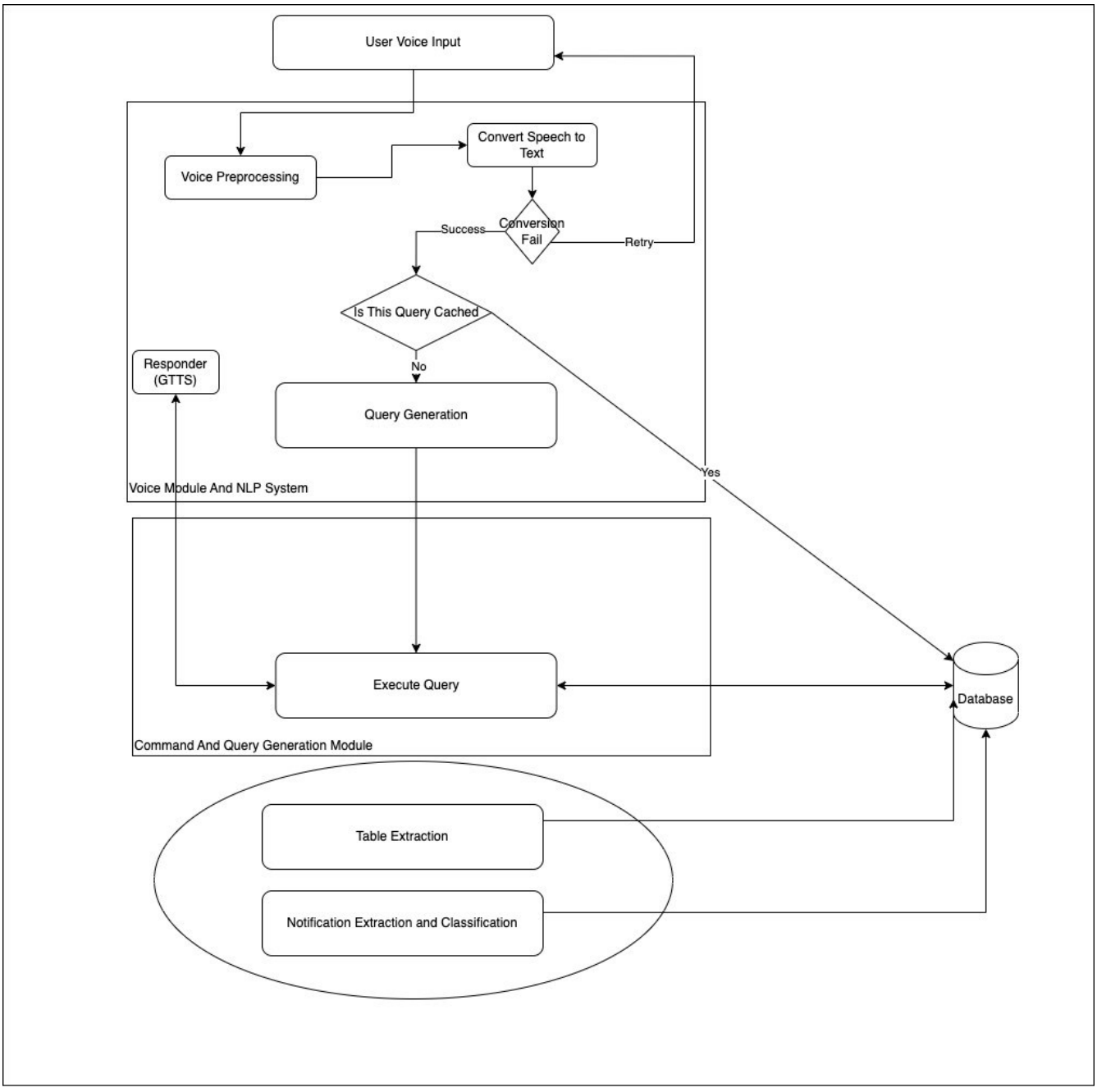
Risk involved

- Our only risk is of leakage of confidential data of students. We need Maximum Security for this

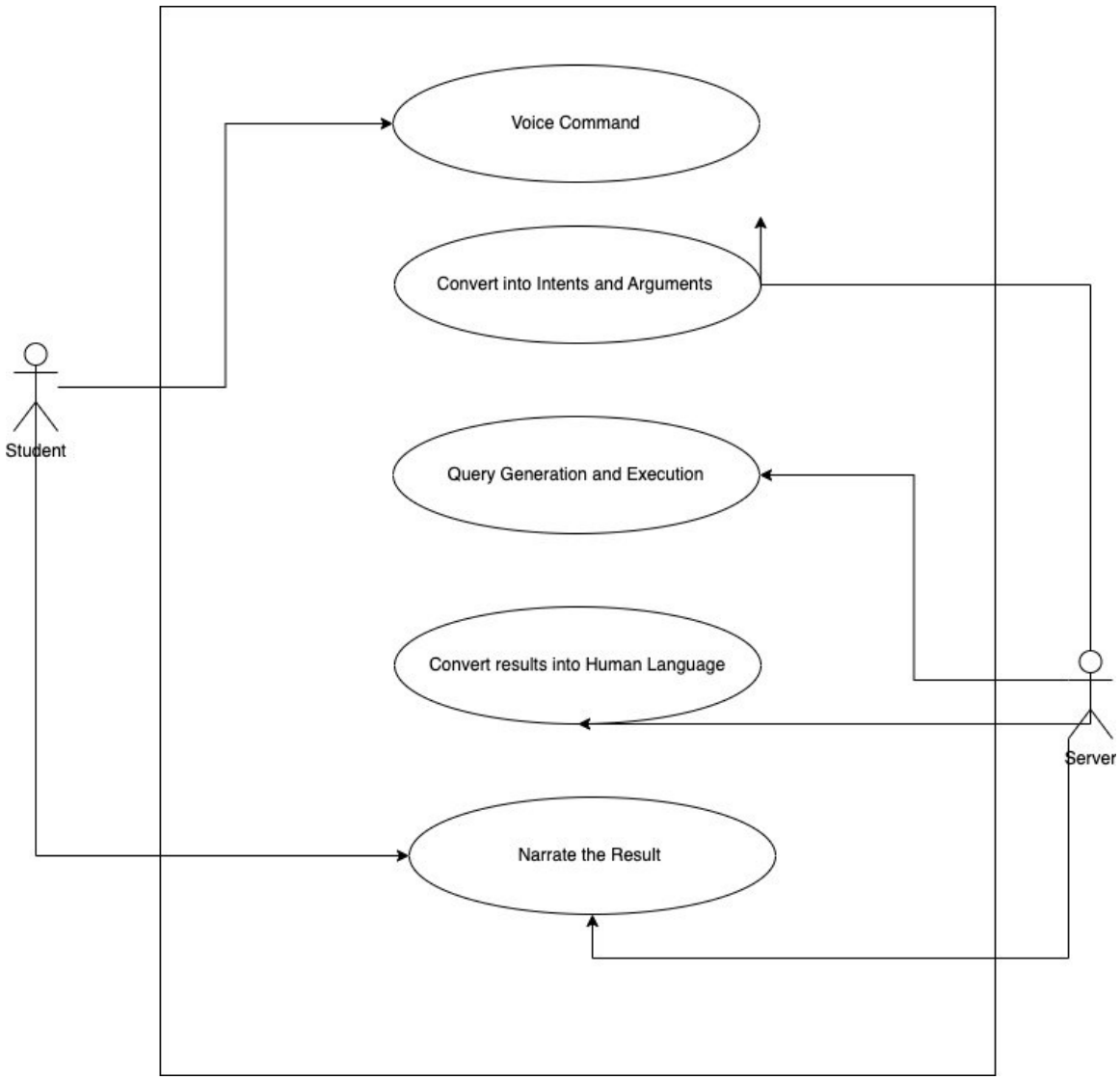
Usage Limitations

- Only applicable to Students for now. (May Extend to Professors also. Depends on work and feedback)

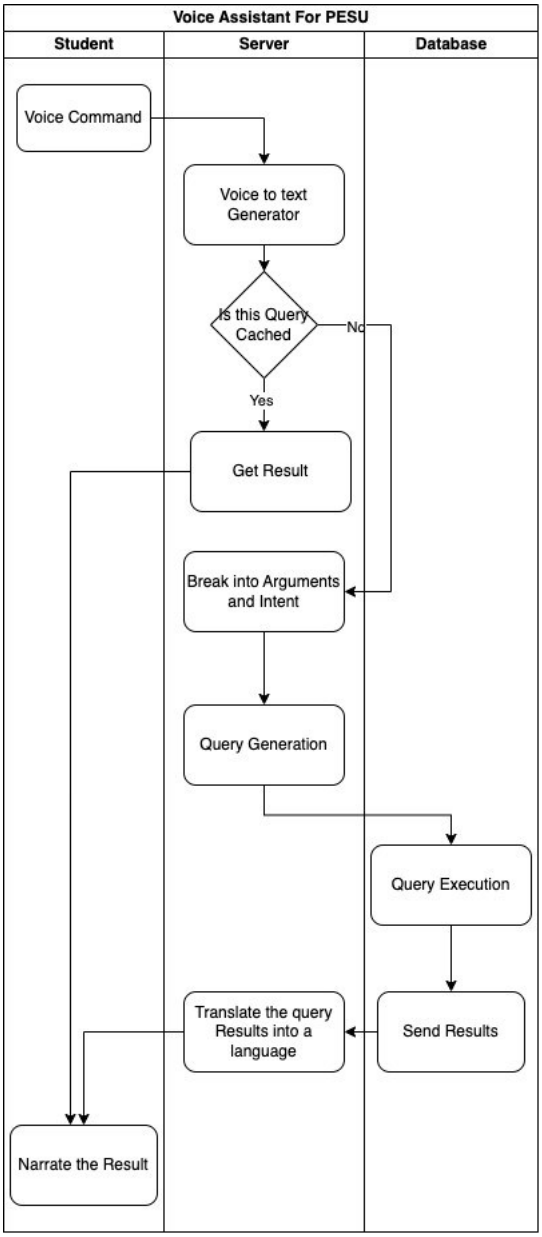
6.2 High Level System Design



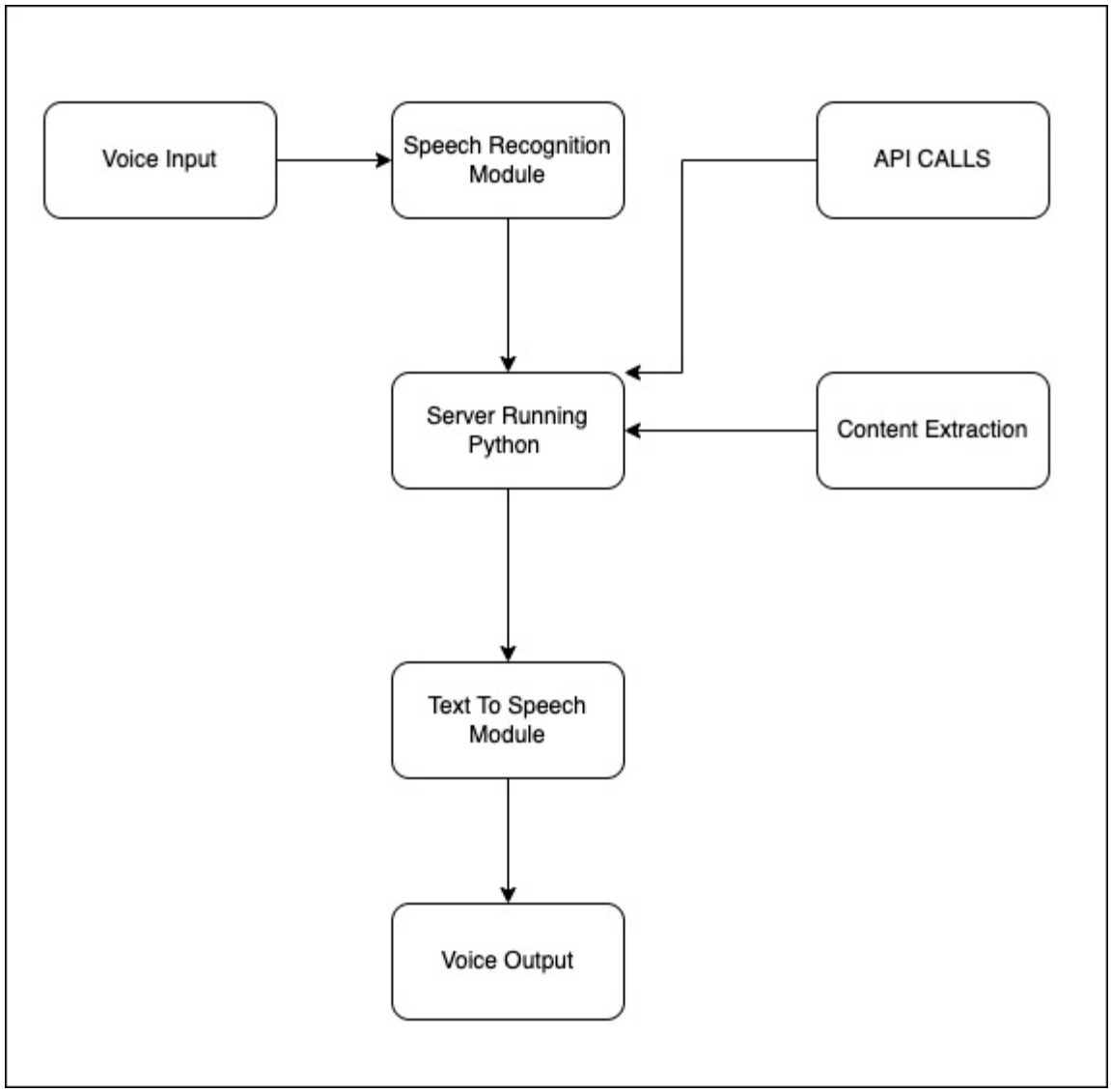
6.3.1 Use case Diagram



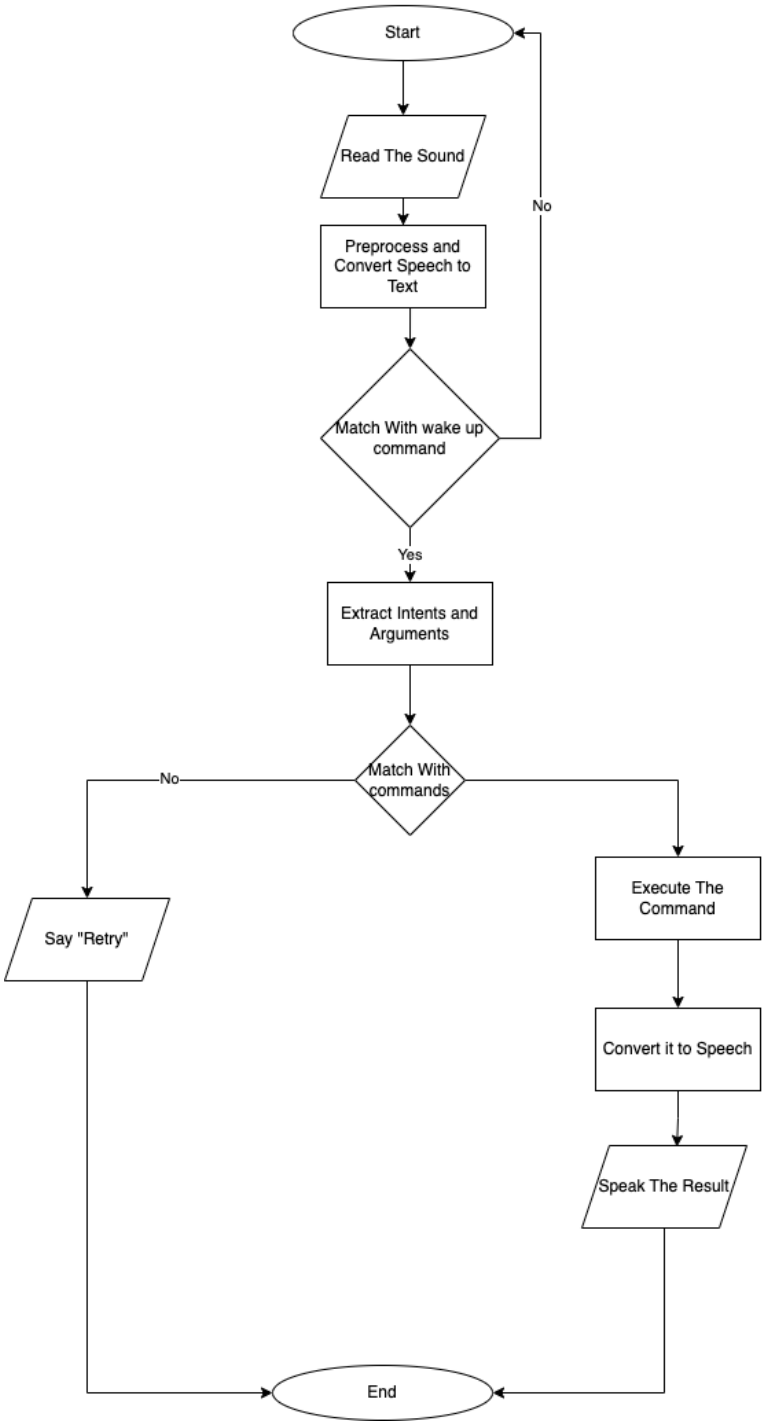
6.3.1 Swim Lane Diagram



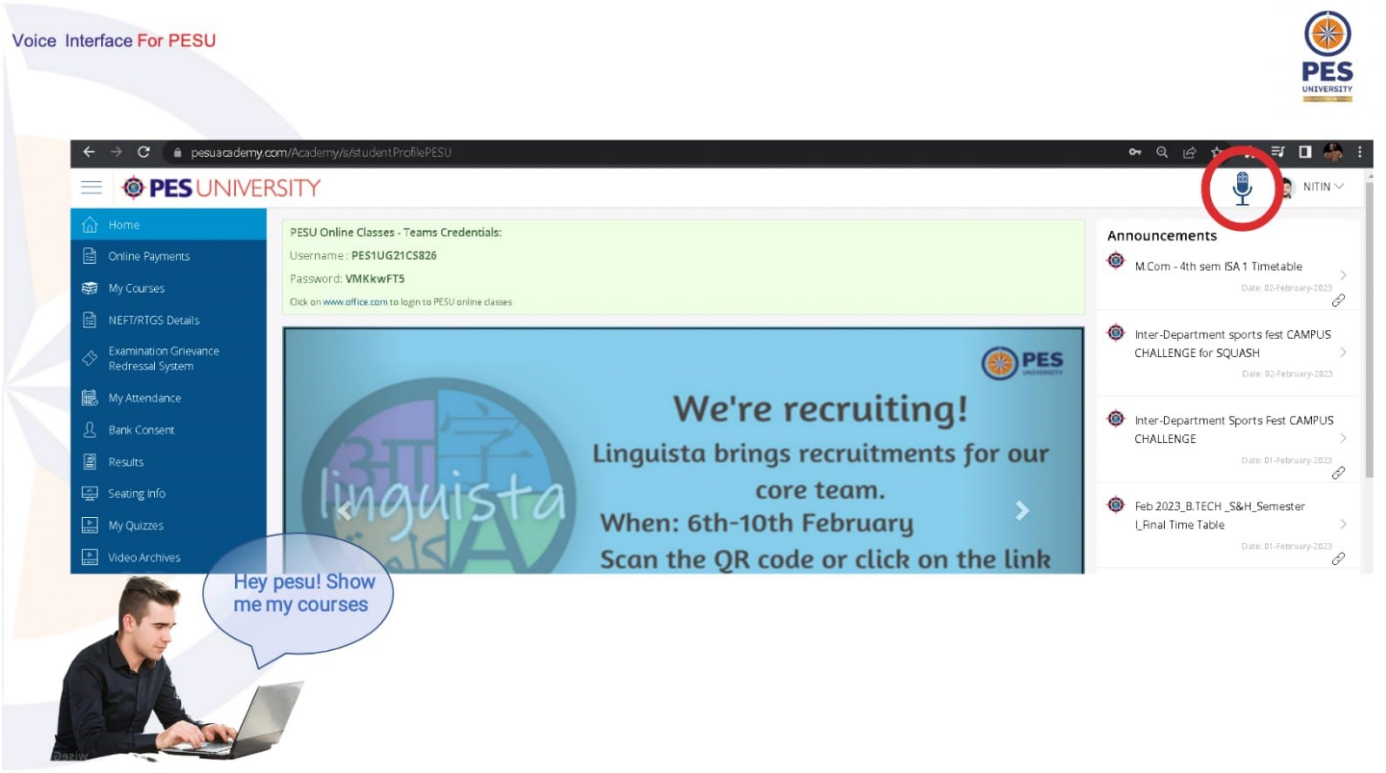
System design architecture for voice assistant:



Flow chart for voice assistant working:



6.3.2 User Interface Diagrams



7.IMPLEMENTATION AND PSEUDOCODE

7.IMPLEMENTATION AND PSEUDOCODE

7.1 Algorithm

```
import pandas as pd import cv2 import numpy as np img =
cv2.imread('test.png') gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
thresh =
cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY_INV,11,2) contours, hierarchy =
cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    area = cv2.contourArea(cnt)     perimeter =
cv2.arcLength(cnt,True)     corners =
cv2.approxPolyDP(cnt,0.01*perimeter,True)
if area > 5000 and len(corners) == 4:

cv2.drawContours(img,[cnt],0,(0,0,255)
,2) cv2.imshow('Detected Table', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

8.CONCLUSION OF CAPSTONE PROJECT PHASE-1

8.CONCLUSION

As Discussed by panel members and ourcapstone faculty, we have finalized the features what to use etc.

Main Key features of our project are :

- Getting information from database
- Getting voice from user and converting it to text,
- Pre-processing it and turning it into queries

9.PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2

- The main thing of part of our project is NLP, we are going to build a model for converting student query into executable query.
- We are going to design a program to pre-process all the data that we have extracted from table.
- Next part of our project is going to be database design and security, that is encryption
- We must look into handling of cached data.

REFERENCE/ BIBLIOGRAPHY

- Artificial_Intelligence-based_Voice_Assistant - Subhash S, Ullas A, Santhosh B, Siddesh S
- A Framework for System Interfacing of Voice User Interface for Personal Computers Preet Dabre, Rohit Gonsalves, Raj Chandvaniya, and Anant V. Nimkar Department of Computer Engineering Sardar Patel Institute of Technology Mumbai, India
- Table Detection and Extraction using OpenCV and Novel Optimization Methods – Nidhi, Karandeep Saluja, Asmita Mahajan, Akash Jadhav, Nakul Aggarwal
- A Method for Voice Activity Detection using K-Means Clustering - Atul Rohit Agarwal, Sourabh Tiwari, Sudhakar M S, Sankar Ganesh S

- Artificial_Intelligence-based_Voice_Assistant - Subhash S, Ullas A, Santhosh B, Siddesh S
- Table Detection and Extraction using OpenCV and Novel Optimization Methods – Nidhi, Karandeep Saluja, Asmita Mahajan, Akash Jadhav, Nakul Aggarwal

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS USED

AI: - Artificial Intelligence

Narrator: - converts text to speech

NLP: - Natural language processing helps computer to understand text