150160509 – Yunus Emre BALAK

# COMPUTER PROJECT 1 – PROJECT 1/REPORT 2

## VOICE BASED MAIL SYSTEM FOR VISUALLY IMPAIRED PEOPLE

An executable programme which is a mail system that can be controlled by sound for visually impaired people. The logic of the programme will be similar to a voice assistance logic. Visually impaired person will easily be able to send and receive mails via gmail.com. Programme will read received messages and communicate with the user. There will be 3 main commands which can create new mail, open inbox, open outbox.

- **Create New Mail Task:** It will get to, cc, subject informations from user by sound. Then it will get text message to be sent. According to these informations. It will connect to gmail.com and send the message.

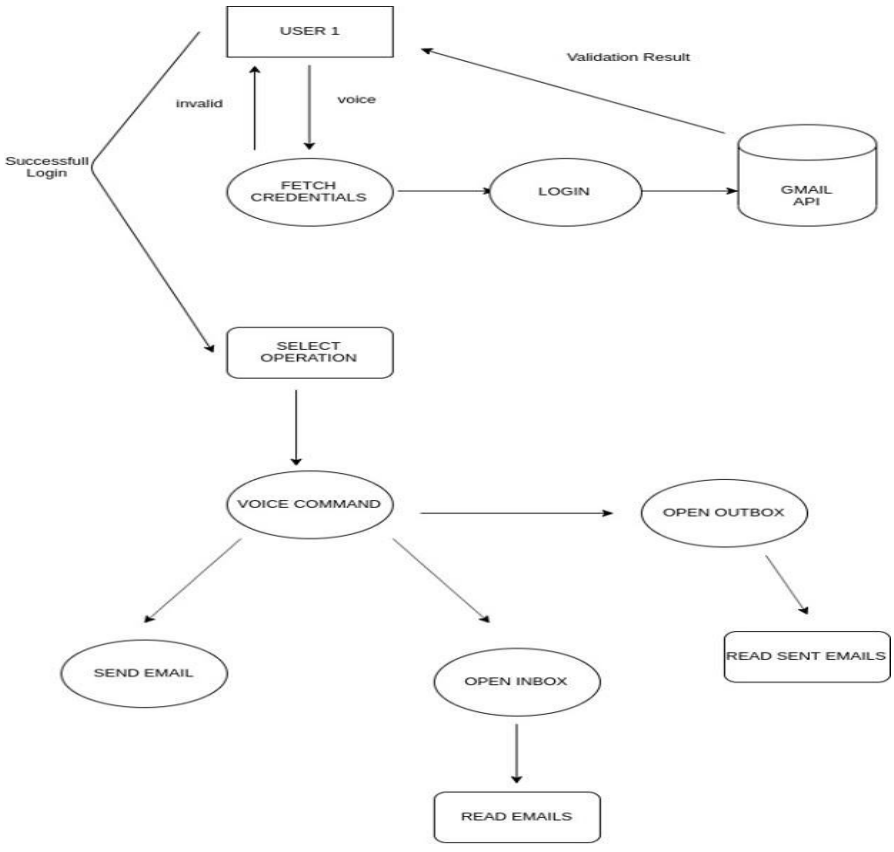
- **Open Inbox:** It will connect to gmail.com and read messages in inbox. User will determine how many messages will be read by the programme. User will also be able to stop programme to read the message.

- **Open Outbox:** It will connect to gmail.com and read messages in outbox. User will determine how many messages will be read by the programme. User will also be able to stop programme to read the message.

**Technologies Planned to be Used:**

- Python's pygame, speech_recognition, and gTTS libraries.
- Gmail API

### Flowchart of the Programme



```python
import speech_recognition as sr
from gtts import gTTS
import pygame

def speak(text):
    tts = gTTS(text=text, lang='en')
    filename = 'voice.wav'
    tts.save(filename)
    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(filename)
    pygame.mixer.music.play()

    #pygame.event.wait()
    return


def get_audio():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source, duration=5)
        audio = r.listen(source)
        said = ""

        try:
            said = r.recognize_google(audio)
            print(said)
        except Exception as e:
            print("Exception: " + str(e))

    return said

speak("you can talk")

text = get_audio()
speak()
```

The code above uses Python’s pygame, speech_recognition, and gTTS libraries. It communicates with the user and says ‘you can talk’. After user gives command. Programme gets the command and transform speech to text.

150160509 – Yunus Emre BALAK

```python
from __future__ import print_function
import pickle
import os.path
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request

# If modifying these scopes, delete the file token.pickle.
SCOPES = ['https://www.googleapis.com/auth/gmail.readonly']

def main():
    """Shows basic usage of the Gmail API.
    Lists the user's Gmail labels.
    """
    creds = None
    # The file token.pickle stores the user's access and refresh tokens, and is
    # created automatically when the authorization flow completes for the first
    # time.
    if os.path.exists('token.pickle'):
        with open('token.pickle', 'rb') as token:
            creds = pickle.load(token)
    # If there are no (valid) credentials available, let the user log in.
    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
                'credentials.json', SCOPES)
            creds = flow.run_local_server(port=0)
        # Save the credentials for the next run
        with open('token.pickle', 'wb') as token:
            pickle.dump(creds, token)

    service = build('gmail', 'v1', credentials=creds)

    # Call the Gmail API
    results = service.users().labels().list(userId='me').execute()
    labels = results.get('labels', [])

    if not labels:
        print('No labels found.')
    else:
        print('Labels:')
        for label in labels:
            print(label['name'])

if __name__ == '__main__':
    main()

```

This code will provide us to access Gmail mailboxes. We will be able to read and send email messages. When the code is executed, it will attempt to open a new window which wants you to enter your email and password. After the authorization, information about user is stored and subsequent executions will not want you to authorize.

**The code above is for command-line application.**