Experiment - 2

NAME: ASWIN SIDDHARTH A ROLL NO: 240701064

Develop and compare CLI, GUI, and Voice User Interfaces (VUI) for the same task and assess user satisfaction using Python (Tkinter for GUI, Speech Recognition for VUI), Terminal

AIM:

The aim is to develop and compare Command Line Interface (CLI), Graphical User Interface (GUI), and Voice User Interface (VUI) for the same task, and assess user satisfaction using Python (with Tkinter for GUI and Speech Recognition for VUI) and Terminal.

PROCEDURE:

i) CLI (Command Line Interface)

CLI implementation where users can add, view, and remove files using the terminal.

```
import os
import glob

def add_file():
  file = input("Enter the name of the file: ")
  with open(f"{file}.txt", "w") as f:
    pass
  print(f"file '{file}.txt' was added.")

def remove_file():
  file = input("Enter the name of the file to remove: ")
```

```
    file = file + ".txt"
    if os.path.exists(file):
        os.remove(file)
        print(f"file '{file}' was removed.")
    else:
        print(f"file '{file}' was not found.")

def view_files():
    files = glob.glob("*.txt")
    if files:
        print("files:")
        for i in range(len(files)):
            print(f"{i + 1} - {files[i]}")
    else:
        print("No files.")

while True:
    choice = input("Choose an action: \n(1) Add file \n(2) Remove file \n(3) View files \n(4) Exit\n>>> ")
    if choice == '1':
        add_file()
    elif choice == '2':
        remove_file()
    elif choice == '3':
        view_files()
    elif choice == '4':
        print("Exiting...")
        break
```

```
    else:
        print("Invalid choice. Please try again.")
```

Sample Output:

```
Choose an action:
(1) Add file
(2) Remove file
(3) View files
(4) Exit
>>> 1
Enter the name of the file: example
file 'example.txt' was added.
Choose an action:
(1) Add file
(2) Remove file
(3) View files
(4) Exit
>>> 3
files:
1 - aswin.txt
2 - example.txt
3 - new.txt
4 - something.txt
Choose an action:
(1) Add file
(2) Remove file
(3) View files
(4) Exit
>>> 2
Enter the name of the file to remove: aswin
file 'aswin.txt' was removed.
Choose an action:
(1) Add file
(2) Remove file
(3) View files
(4) Exit
>>> 4
Exiting...
```

ii) GUI (Graphical User Interface)

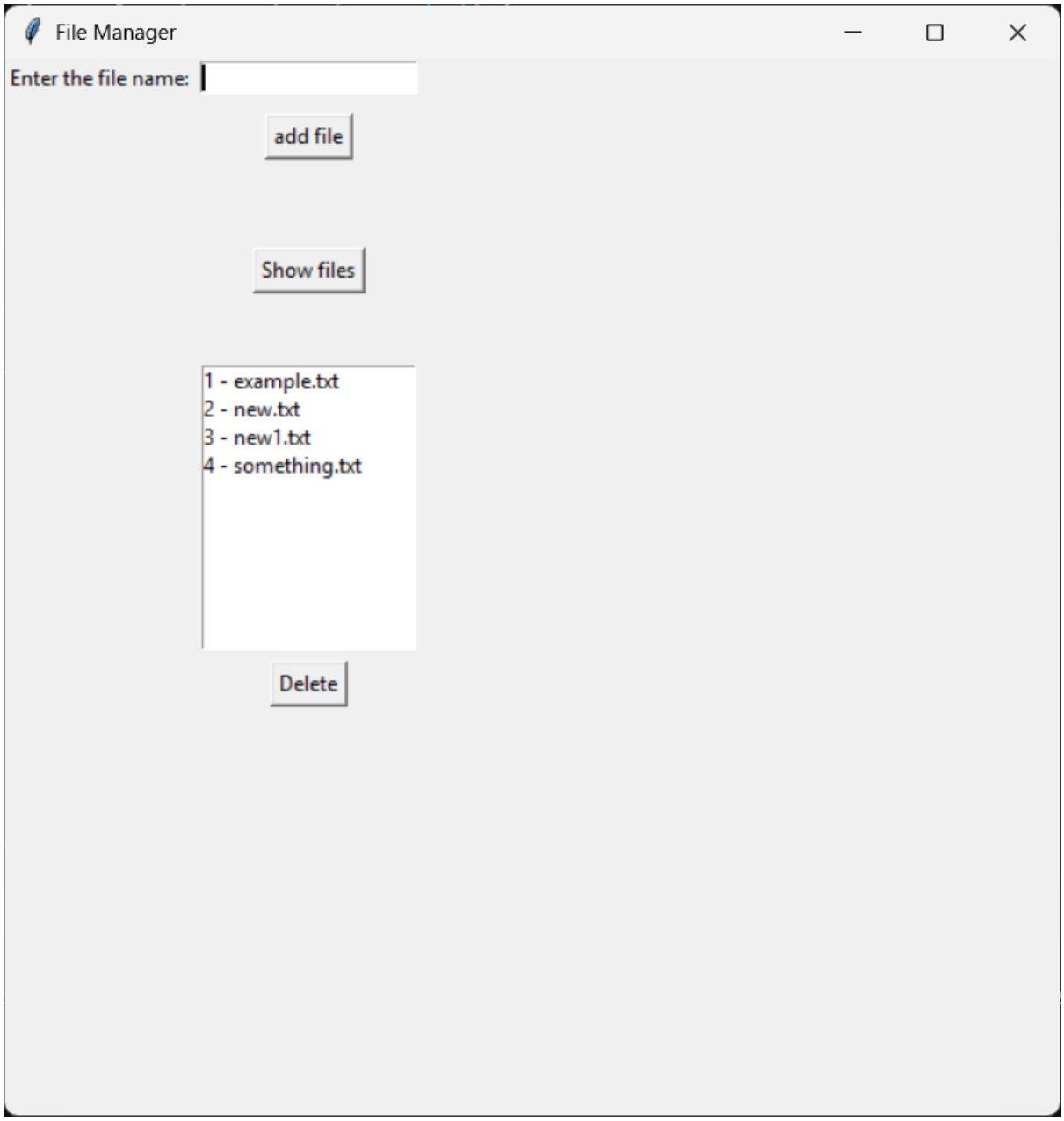
Tkinter to create a simple GUI for our file manager application.

```
import tkinter as tk
from tkinter import messagebox
import glob
import os


def add_file():
    file = enter.get()
    if file != "":
        enter.delete(0, "end")
        with open(f"{file}.txt", "w") as f:
            pass
    else:
        messagebox.showwarning("Warning", "Field should not be empty.")


def show_files():
    lb.delete(0, "end")
    files = glob.glob("*.txt")
    if files:
        for i in range(len(files)):
            lb.insert(i + 1, f"{i + 1} - {files[i]}")
    else:
        messagebox.showinfo("Info", "No files found.")


def remove_file():
    selected = lb.curselection()
    if selected:
```

```
        idx = selected[0]
        file = lb.get(idx)
        lb.delete(tk.ANCHOR)
        fileName = file.split(" - ")
        os.remove(fileName[1])
        lb.delete(0, "end")
        files = glob.glob("*.txt")
        if files:
            for i in range(len(files)):
                lb.insert(i + 1, f"{i + 1} - {files[i]}")
    else:
        messagebox.showwarning("Warning", "Select any file.")

app = tk.Tk()
app.title("File Manager")
app.geometry("600x600")

text = tk.Label(app, text="Enter the file name: ")
text.grid(row=0, column=3)

enter = tk.Entry(app)
enter.grid(row=0, column=4)

submit = tk.Button(app, text="add file", command=add_file)
submit.grid(row=1, column=4, pady=10)

lb = tk.Listbox(app)
lb.grid(row=3, column=4)
```

```
showBu = tk.Button(app, text="Show files", command=show_files)
showBu.grid(row=2, column=4, pady=40)

deleteBu = tk.Button(app, text="Delete", command=remove_file)
deleteBu.grid(row=4, column=4, pady=5)
app.mainloop()
```

Sample Output:



iii) VUI (Voice User Interface)

speech_recognition library for voice input and the pyttsx3 library for text-to-speech output. Make sure you have these libraries installed (pip install SpeechRecognition pyttsx3).

```
import speech_recognition as sr
import pyttsx3
import glob
import os

def speak(audio):
    engine = pyttsx3.init()
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[0].id)
    engine.say(audio)
    engine.runAndWait()

def command():
    reco = sr.Recognizer()
    with sr.Microphone() as ans:
        print("Listening...")
        reco.pause_threshold = 1
        audio = reco.listen(ans)
        try:
            print("Recognizing...")
            command_ans = reco.recognize_google(audio, language="en-in")
            print(command_ans)
            return command_ans
        except Exception as e:
```

```
        print(e)
        return None


while True:
    speak("Choose an action: Add file, Remove file, View files, Exit.")
    ans = command()
    if ans is None:
        continue
    ans = ans.lower()
    if "add file" in ans:
        speak("Whats the file name?")
        filename = command()
        with open(f"{filename}.txt", "w") as f:
            pass
    elif "remove file" in ans:
        speak("What is the name of the file?")
        filename = command()
        filename = filename + ".txt"
        if os.path.exists(filename):
            os.remove(filename)
            speak(f"file '{filename}' was removed.")
        else:
            speak(f"file '{filename}' was not found.")
    elif ("view files" in ans) or ("view file" in ans):
        files = glob.glob("*.txt")
        speak("The files are:")
        for filename in files:
            name = filename.split(".")
```

```
        speak(name[0])
    elif "exit" in ans:
        speak("exiting.")
        print("exiting...")
        break
    else:
        speak("I didn't understand your command.")
```