LAB – 3

Sanjana R – CSE C

AIM: Evaluating User Interfaces – CLI, GUI, and VUI in File Renaming

Introduction

User interfaces determine how easily and efficiently a task can be performed on a computer. This experiment compares three types of interfaces—**Command Line Interface (CLI)**, **Graphical User Interface (GUI)**, and **Voice User Interface (VUI)**—by implementing a common task: renaming a file. Python was used in each case, with the help of:

- OS operations for CLI,
- Tkinter for GUI,
- SpeechRecognition library for VUI.

1. Command Line Interface (CLI)

Before starting, the folder contained a file named oldfile.txt. A terminal command was used to rename it. After execution, the file name was successfully changed to newfile.txt, as reflected in the folder.

```
import os

import sys


def  rename_file(old_name, new_name):

   try:

      os.rename(old_name, new_name)

      print(f"File renamed from {old_name} to {new_name}")

   except FileNotFoundError:

      print(f"Error: {old_name} not found.")

   except Exception as e:

      print(f"An error occurred: {e}")


if __name__ == "__main__":

   if len(sys.argv) != 3:

      print("Usage: python rename_file_cli.py <old_filename> <new_filename>")

   else:

      rename_file(sys.argv[1], sys.argv[2])
```

---

2. Graphical User Interface (GUI)

The user was prompted with a simple graphical window (built using Tkinter in Python). In this interface, the user typed the old and new filenames. After clicking a button, the file oldfile.txt was renamed to newfile.txt.

```
import tkinter as tk

from tkinter import messagebox

import os


def rename_file():

  old_name = old_filename_entry.get()

  new_name = new_filename_entry.get()


  try:

    os.rename(old_name, new_name)

    messagebox.showinfo("Success", f"File renamed from {old_name} to {new_name}")

  except FileNotFoundError:

    messagebox.showerror("Error", f"File {old_name} not found.")

  except Exception as e:

    messagebox.showerror("Error", f"An error occurred: {e}")


# Set up the main window

root = tk.Tk()

root.title("File Renamer")

# Create and place labels, entries, and buttons

tk.Label(root, text="Old Filename:").grid(row=0, column=0)

tk.Label(root, text="New Filename:").grid(row=1, column=0)


old_filename_entry = tk.Entry(root)

old_filename_entry.grid(row=0, column=1)


new_filename_entry = tk.Entry(root)
```

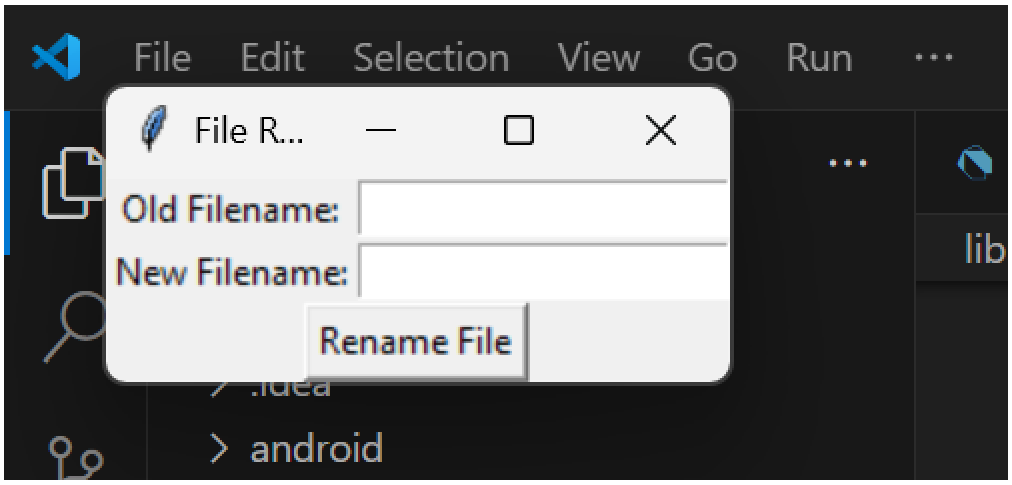
new_filename_entry.grid(row=1, column=1)

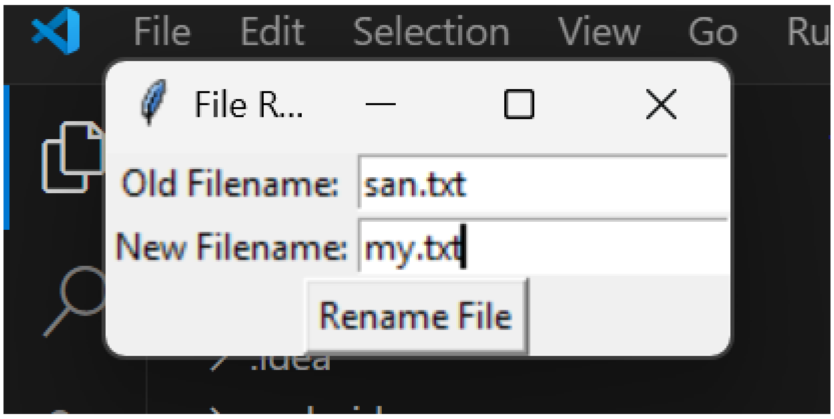
rename_button = tk.Button(root, text="Rename File", command=rename_file)

rename_button.grid(row=2, columnspan=2)

# Start the Tkinter event loop

root.mainloop()





Old Filename: san.txt

New Filename: my.txt

---

3. Voice User Interface (VUI)

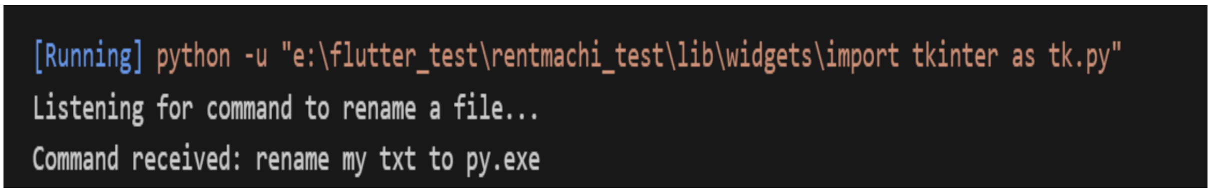
Initially, the folder contained a file named Sample. Using the SpeechRecognition library, the user gave a voice command to rename the file. After processing, the file was renamed to file, showcasing the convenience of hands-free control.

import speech_recognition as sr

import os

def rename_file_from_voice_command(command):

```
    # Extracting old and new filename from the command
    try:
        words = command.split(" ")
        old_name = words[1]
        new_name = words[3]

        os.rename(old_name, new_name)
        print(f"File renamed from {old_name} to {new_name}")
    except Exception as e:
        print(f"Error: {e}")

def listen_for_command():
    recognizer = sr.Recognizer()
    mic = sr.Microphone()

    print("Listening for command to rename a file...")
    with mic as source:
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)

    try:
        command = recognizer.recognize_google(audio)
        print(f"Command received: {command}")
        rename_file_from_voice_command(command)
    except sr.UnknownValueError:
        print("Sorry, I couldn't understand the command.")
    except sr.RequestError as e:
        print(f"Could not request results from Google Speech Recognition service; {e}")

if __name__ == "__main__":
    listen_for_command()
```



```
[Running] python -u "e:\flutter_test\rentmachi_test\lib\widgets\import tkinter as tk.py"
Listening for command to rename a file...
Command received: rename my txt to py.exe
```