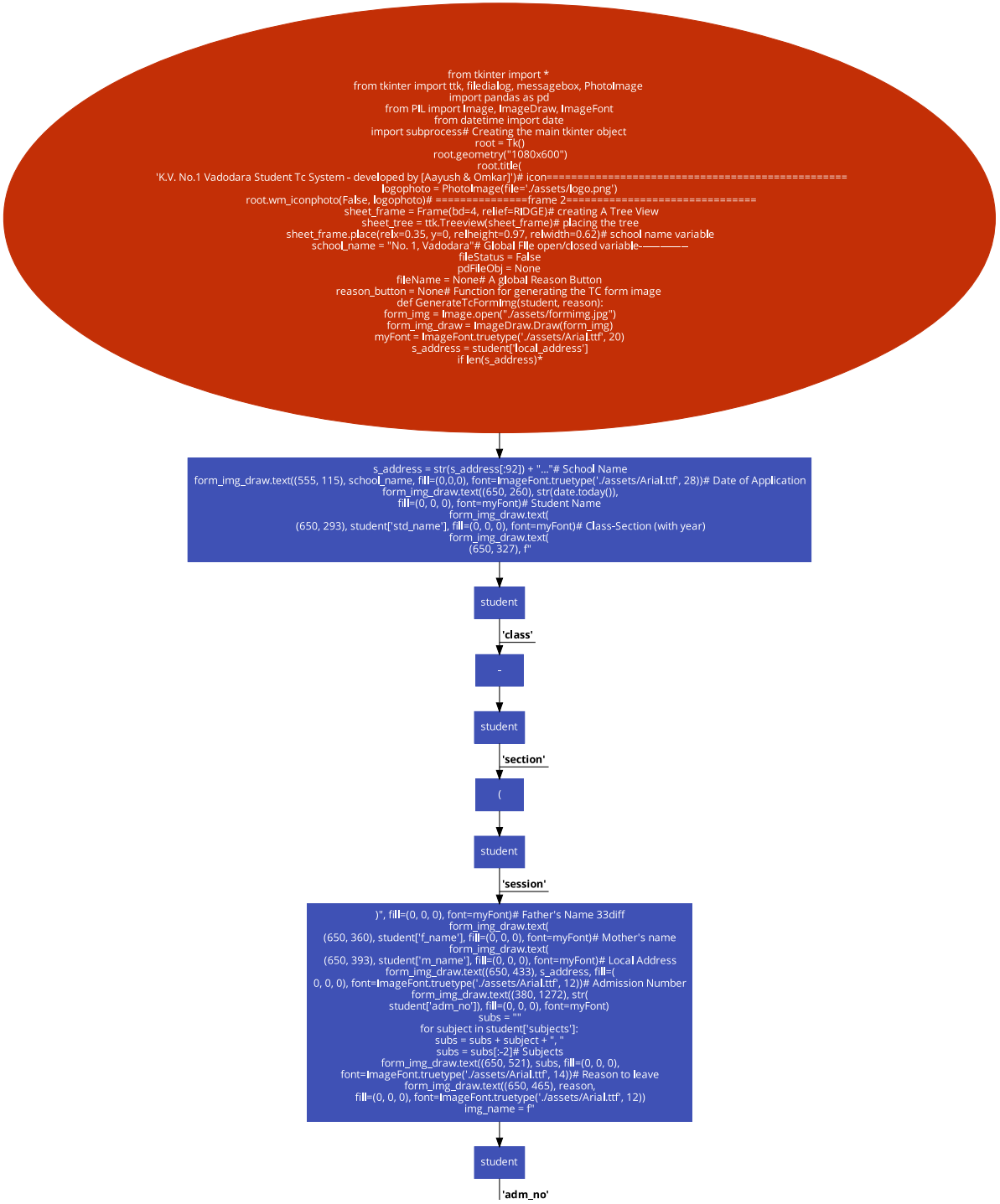
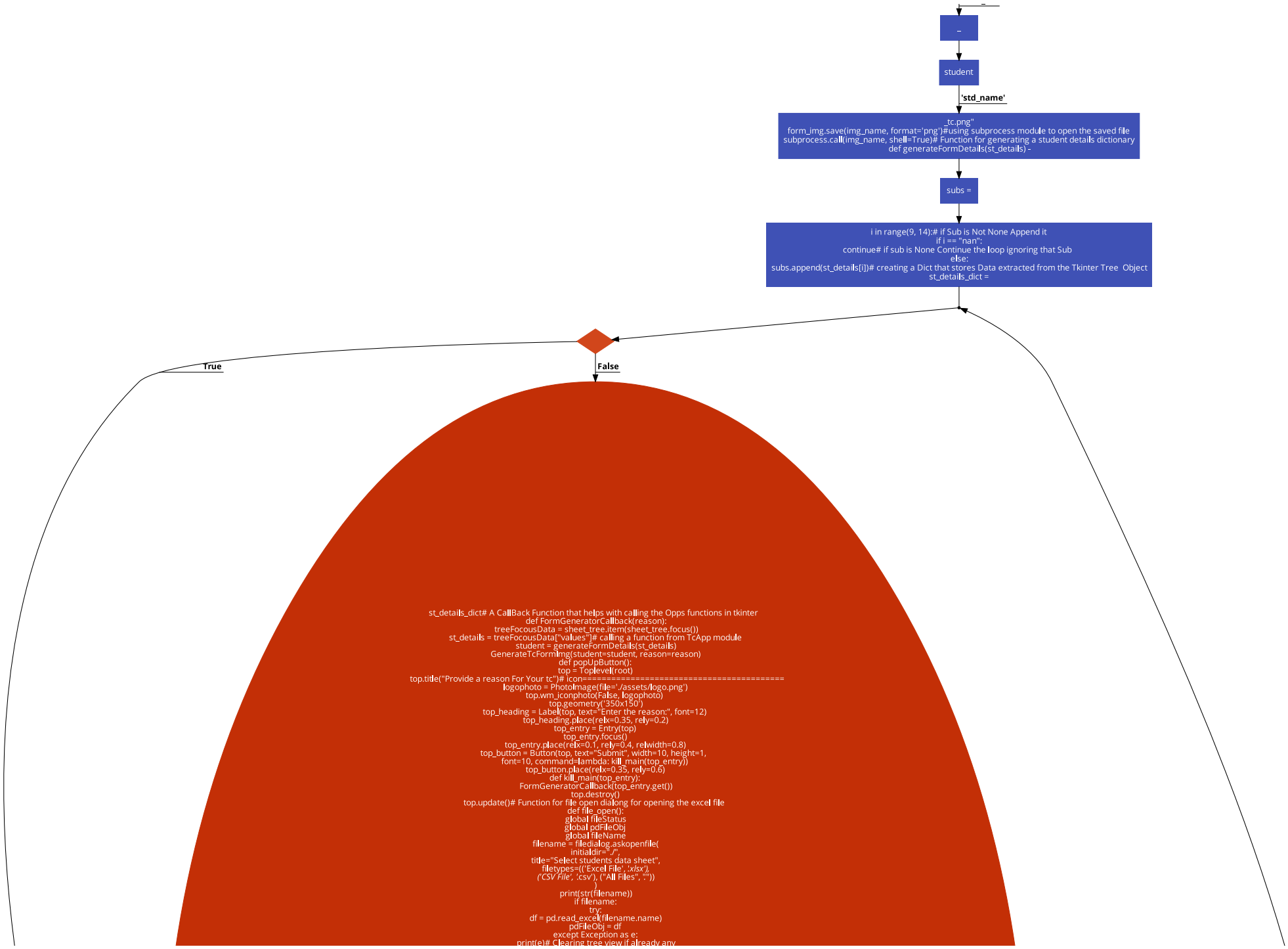

from tkinter import *
from tkinter import ttk, filedialog, messagebox, PhotoImage
import pandas as pd
from PIL import Image, ImageDraw, ImageFont
from datetime import date
import subprocess# Creating the main tkinter object
root = Tk()
root.geometry("1080x600")
root.title(
'K.V. No.1 Vadodara Student Tc System - developed by [Aayush & Omkar]')# icon==========
logophoto = PhotoImage(file='./assets/logo.png')
root.wm_iconphoto(False, logophoto)# ==========frame 2==========
sheet_frame = Frame(bd=4, relief=RIDGE)# creating A Tree View
sheet_tree = ttk.Treeview(sheet_frame)# placing the tree
sheet_frame.place(relx=0.35, y=0, relheight=0.97, relwidth=0.62)# school name variable
school_name = "No. 1, Vadodara"# Global File open/closed variable----------
fileStatus = False
pdFileObj = None
fileName = None# A global Reason Button
reason_button = None# Function for generating the TC form image
def GenerateTcFormImg(student, reason):
form_img = Image.open("./assets/formimg.jpg")
form_img_draw = ImageDraw.Draw(form_img)
myFont = ImageFont.truetype('./assets/Arial.ttf', 20)
s_address = student['local_address']
if len(s_address)*
s_address = str(s_address[:92]) + "..."# School Name
form_img_draw.text((555, 115), school_name, fill=(0,0,0), font=ImageFont.truetype('./assets/Arial.ttf', 28))# Date of Application
form_img_draw.text((650, 260), str(date.today()),
fill=(0, 0, 0), font=myFont)# Student Name
form_img_draw.text(
(650, 293), student['std_name'], fill=(0, 0, 0), font=myFont)# Class-Section (with year)
form_img_draw.text(
(650, 327), f"
student
'class'
-
student
'section'
(
student
'session'
)", fill=(0, 0, 0), font=myFont)# Father's Name 33diff
form_img_draw.text(
(650, 360), student['f_name'], fill=(0, 0, 0), font=myFont)# Mother's name
form_img_draw.text(
(650, 393), student['m_name'], fill=(0, 0, 0), font=myFont)# Local Address
form_img_draw.text((650, 433), s_address, fill=(
0, 0, 0), font=ImageFont.truetype('./assets/Arial.ttf', 12))# Admission Number
form_img_draw.text((380, 1272), str(
student['adm_no']), fill=(0, 0, 0), font=myFont)
subs = ""
for subject in student['subjects']:
subs = subs + subject + ", "
subs = subs[:-2]# Subjects
form_img_draw.text((650, 521), subs, fill=(0, 0, 0),
font=ImageFont.truetype('./assets/Arial.ttf', 14))# Reason to leave
form_img_draw.text((650, 465), reason,
fill=(0, 0, 0), font=ImageFont.truetype('./assets/Arial.ttf', 12))
img_name = f"
student
'adm_no'


_
student
'std_name'
_tc.png"
form_img.save(img_name, format='png')#using subprocess module to open the saved file
subprocess.call(img_name, shell=True)# Function for generating a student details dictionary
def generateFormDetails(st_details) -
subs =
i in range(9, 14):# if Sub is Not None Append it
if i == "nan":
continue# if sub is None Continue the loop ignoring that Sub
else:
subs.append(st_details[i])# creating a Dict that stores Data extracted from the Tkinter Tree Object
st_details_dict =
True
False
st_details_dict# A CallBack Function that helps with calling the Opps functions in tkinter
def FormGeneratorCallback(reason):
treeFocousData = sheet_tree.item(sheet_tree.focus())
st_details = treeFocousData["values"]# calling a function from TcApp module
student = generateFormDetails(st_details)
GenerateTcFormImg(student=student, reason=reason)
def popUpButton():
top = Toplevel(root)
top.title("Provide a reason For Your tc")# Icon==
logophoto = PhotoImage(file='./assets/logo.png')
top.wm_iconphoto(False, logophoto)
top.geometry('350x150')
top_heading = Label(top, text="Enter the reason:", font=12)
top_heading.place(relx=0.35, rely=0.2)
top_entry = Entry(top)
top_entry.focus()
top_entry.place(relx=0.1, rely=0.4, relwidth=0.8)
top_button = Button(top, text="Submit", width=10, height=1,
font=10, command=lambda: kill_main(top_entry))
top_button.place(relx=0.35, rely=0.5)
def kill_main(top_entry):
FormGeneratorCallback(top_entry.get())
top.destroy()
top.update()# Function for file open dialong for opening the excel file
def file_open():
global fileStatus
global pdFileObj
global fileName
filename = filedialog.askopenfile(
initialdir="./",
title="Select students data sheet",
filetypes=(('Excel File', '.xlsx'),
('CSV File', '.csv'), ("All Files", "*"))
)
print(str(filename))
if filename:
try:
df = pd.read_excel(filename.name)
pdFileObj = df
except Exception as e:
print(e)# Clearing tree view if already any


st_details[0],
"std_name": st_details[1],
"f_name": st_details[2],
"m_name": st_details[3],
"class": int(st_details[4][:-2]),
"section": st_details[5],
"session": st_details[6],
"local_address": st_details[7],
"subjects": subs
clear_tree()# Setting up new treeview
sheet_tree["column"] = list(df.columns)
sheet_tree["show"] = "headings"# Iterating through column list
for col in sheet_tree["column"]:
sheet_tree.heading(col, text=col)# Putting data in treeview
df_rows = df.to_numpy().tolist()
for row in df_rows:
sheet_tree.insert("", "end", values=row)# adding scroll bars———
if fileStatus == False:
sheet_scrolly = Scrollbar(sheet_frame)
sheet_scrollx = Scrollbar(sheet_frame, orient=HORIZONTAL)
sheet_scrollx.pack(side=BOTTOM, fill=X)
sheet_scrolly.pack(side=RIGHT, fill=Y)# packing the scroll bar ——————————————
sheet_tree.pack()# internal configs for scrollbars
sheet_scrolly.config(command=sheet_tree.yview)
sheet_scrollx.config(command=sheet_tree.xview)
sheet_tree.config(yscrollcommand=sheet_scrolly.set)
sheet_tree.config(xscrollcommand=sheet_scrollx.set)
sheet_tree.config(selectmode=BROWSE)
gen_button.config(state=ACTIVE)
else:
pass
sheet_tree.place(x=0, y=0, relheight=1, relwidth=1)
fileStatus = True# showing the file again
def show_file_again():
global fileStatus
global pdFileObj
df = pdFileObj# Clearing tree view if already any
clear_tree()# Setting up new treeview
sheet_tree["column"] = list(df.columns)
sheet_tree["show"] = "headings"# Iterating through column list
for col in sheet_tree["column"]:
sheet_tree.heading(col, text=col)# Putting data in treeview
df_rows = df.to_numpy().tolist()
for row in df_rows:
sheet_tree.insert("", "end", values=row)
else:
pass# Some helper functions
def clear_tree():
sheet_tree.delete(*sheet_tree.get_children())#helper function that focous on a tkinter tree row
def get_selected():
row = sheet_tree.item(sheet_tree.focus())
print(row)#Searches for student details in the excel sheet provided by the user
def searchStudent():
if fileStatus == False:
messagebox.showerror(
"Error", "No Student Record File is Open\nPlease open a Student record file by using the open button")
return# Clearing tree view if already any
clear_tree()# Setting up new treeview
sheet_tree["column"] = list(pdFileObj.columns)
sheet_tree["show"] = "headings"# Iterating through column list
for col in sheet_tree["column"]:
sheet_tree.heading(col, text=col)# Putting data in treeview
df_rows = pdFileObj.to_numpy().tolist()
rowFound = False
for row in df_rows:
if row[0] == int(search_entry.get()):
sheet_tree.insert("", "end", values=row)
rowFound = True
if rowFound == None:
messagebox.showerror(
"Error", "No Student Data for Provided Admission Number found.\nPlease Check the data file or Admission Number Provided")
return
print(search_entry.get())# ==================frame1===# SETTING THE BUTTONS IN THE MENU THING
menu_frame = Frame(bd=4, relief=RIDGE)
menu_frame.place(relx=0, rely=0, relheight=0.97, relwidth=0.35)
sheet_button = Button(menu_frame, text="Open File", width=40,
height=2, font=20, fg="white", bg="#0078d7", command=file_open)
sheet_button.place(x=0, y=0)
#
gen_button = Button(menu_frame, text="Generate Tc", width=40, height=2,
font=20, fg="white", bg="#0078d7", command=popUpButton, state=DISABLED)
gen_button.place(x=0, y=60)# search ==
search_heading = Label(
menu_frame, text="Search Student details:\n(Enter admission number)", font=12)
search_heading.place(x=0, y=280)
search_entry = Entry(menu_frame)
search_entry.focus()
search_entry.place(x=5, y=320, relwidth=0.70)
search_button = Button(menu_frame, text="Search", width=10, height=1,
font=10, fg="white", bg="#0078d7", padx=4, command=searchStudent)
search_button.place(relx=0.65, y=320)
show_button = Button(menu_frame, text="show all", width=10, height=1,
font=10, fg="white", bg="#0078d7", padx=4, command=show_file_again)
show_button.place(relx=0.65, y=360)
search_heading = Label(
menu_frame, text="Made By :-\nAayush Mishra & Omkar Mahindrakar", font=12)
search_heading.place(x=0, y=480)# Running the tkinter mainloop
root.mainloop()
*