FINAL PROJECT
INTRODUCTION TO PROGRAMMING
(WEATHER FORECAST)

BINUS
UNIVERSITY
INTERNATIONAL

Alvian Wijaya

2301891595

2019

I. Specification

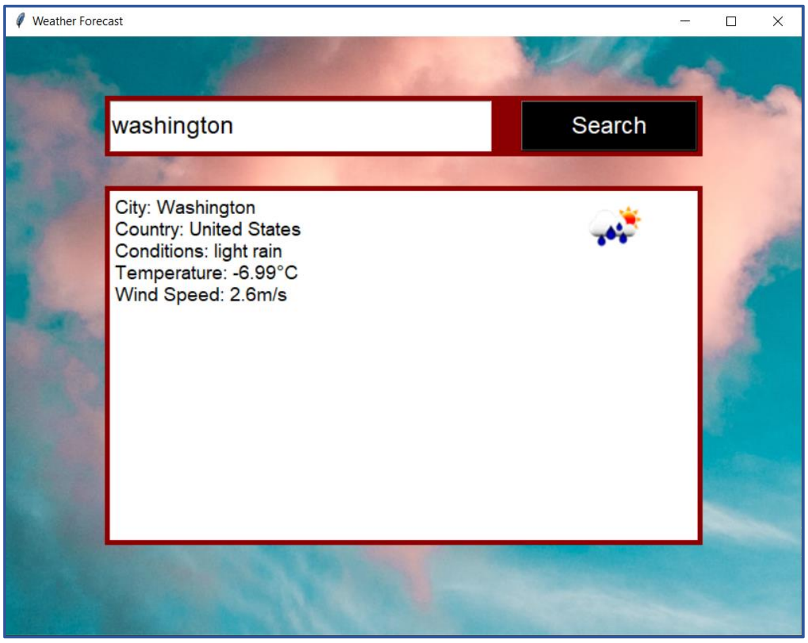
a. Description

The program is designed to tell the user what is the current weather of a certain location. The program shows the name of the city, the country of the city, the condition of the weather, the average temperature, and the wind speed.

Users can input either postal code, city, state, country, and area to the “text box” that is provided. After the users make an input, users can click “search” button to start searching the weather condition, Then the program check if the input is valid or not, if not it will tell the user that the input is not valid.

II. Solution Design

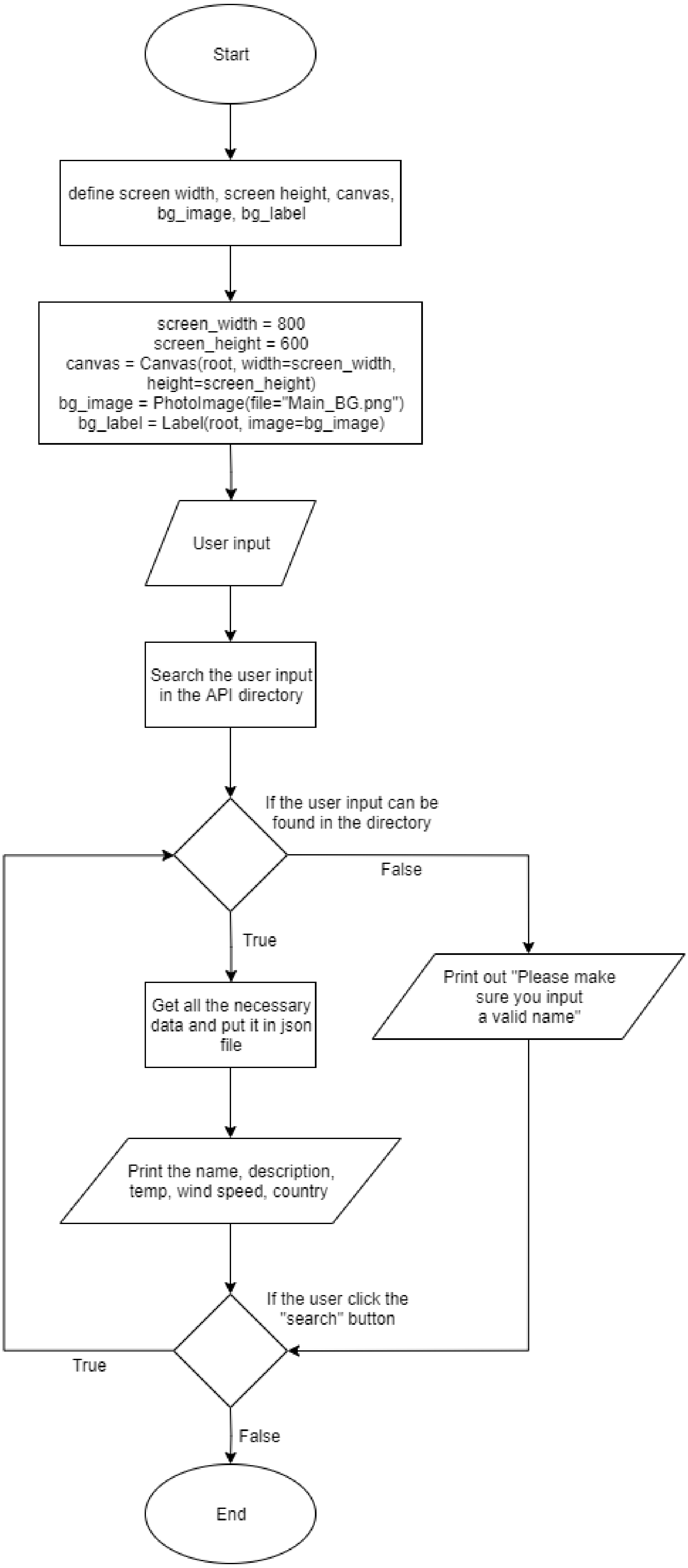
The design of this program is very simple, user just have to input any address (ex. City, postal code, state, province, country, area) and click the button to start searching about the weather of the area. The user gets a brief information about the weather and an icon that show the condition and also tell is the area is on daytime or night.



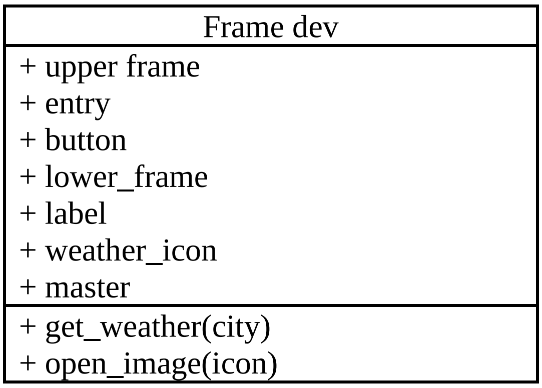
City: Washington
Country: United States
Conditions: light rain
Temperature: -6.99°C
Wind Speed: 2.6m/s

III. How It Works

a. Flow Chart


Start
define screen width, screen height, canvas, bg_image, bg_label
screen_width = 800
screen_height = 600
canvas = Canvas(root, width=screen_width, height=screen_height)
bg_image = PhotoImage(file="Main_BG.png")
bg_label = Label(root, image=bg_image)
User input
Search the user input in the API directory
If the user input can be found in the directory
False
True
Print out "Please make sure you input a valid name"
Get all the necessary data and put it in json file
Print the name, description, temp, wind speed, country
If the user click the "search" button
True
False
End

b. Class Diagram



Frame dev

+ upper frame
+ entry
+ button
+ lower_frame
+ label
+ weather_icon
+ master

+ get_weather(city)
+ open_image(icon)

c. Explanation of Each Function

Weather_Forecast.py

- Outside the class:
  - def format_weather(weather)
    - try if the city is valid or not
    - if valid, it will output name, desc, temp, wind_speed, country to the label which is located on lower_frame.
    - Open the csv file to change the country from alpha2 to a normal name.
    - if the city is not valid, it will output “please make sure you input a valid name”

- class Frame_dev
  - def __init__(self, master)
    - Place upper_button, entry, button, lower_frame, weather_icon, and label to the canvas/ tkinter window.

  - def get_weather(self, city)
    - give the API key, set up the city from the user input, and use the metric units for all the measurement(ex. Temperature and wind speed)
    - ask request from the API URL.
    - Change the given response to .json file to make it easier to work with python.
    - Select the correct weather_icon from the given json file.

  - def open_image(self, icon)
    - Change the size of the weather_icon to 0,20 of the width of lower_frame.
    - insert the weather_icon to the program and tell the program where the images are saved.
    - Delete the previous weather_icon so does not overlap the new one.

Get_Icon.py

- Check if the necessary image is already in the same folder of the program or not. If not it will automatically download all of the necessary image from https://openweathermap.org/img/w/ website and put it together in a folder called "img"
- After finish downloading all of the necessary images, it will automatically start the "Weather_Forecast.py" program.

d. The Evidence of Working Program

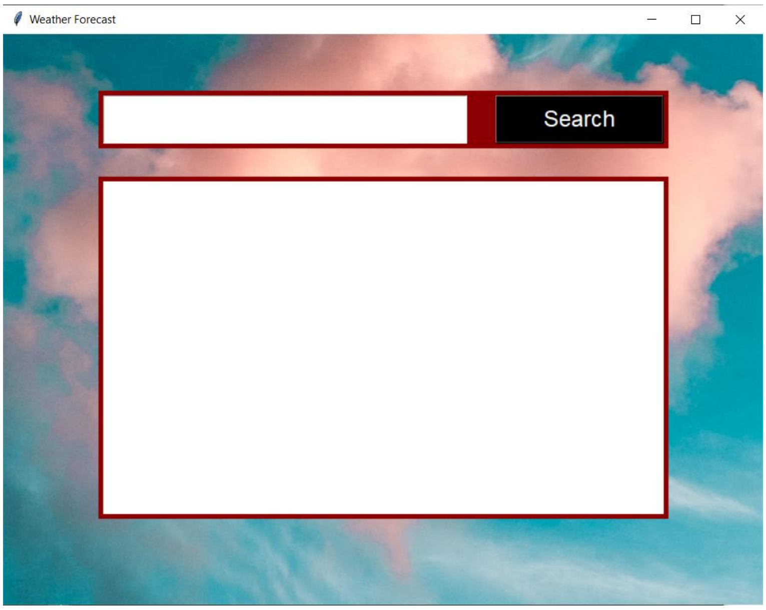
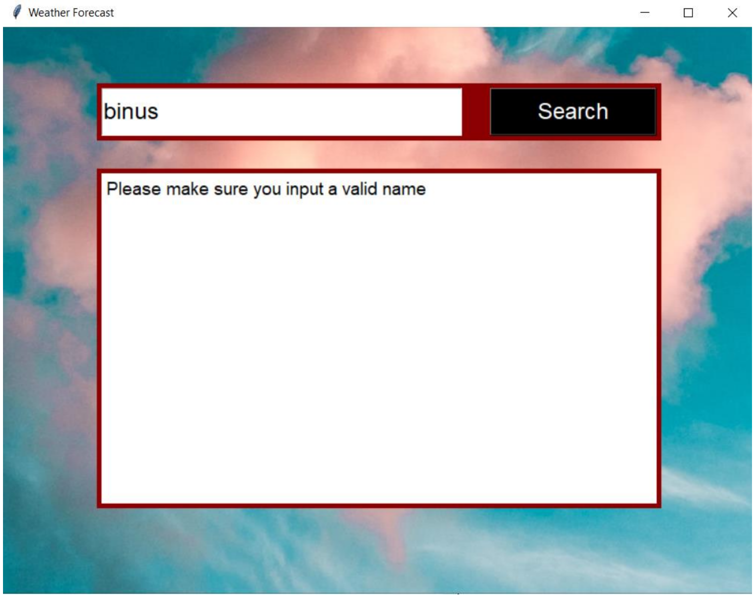
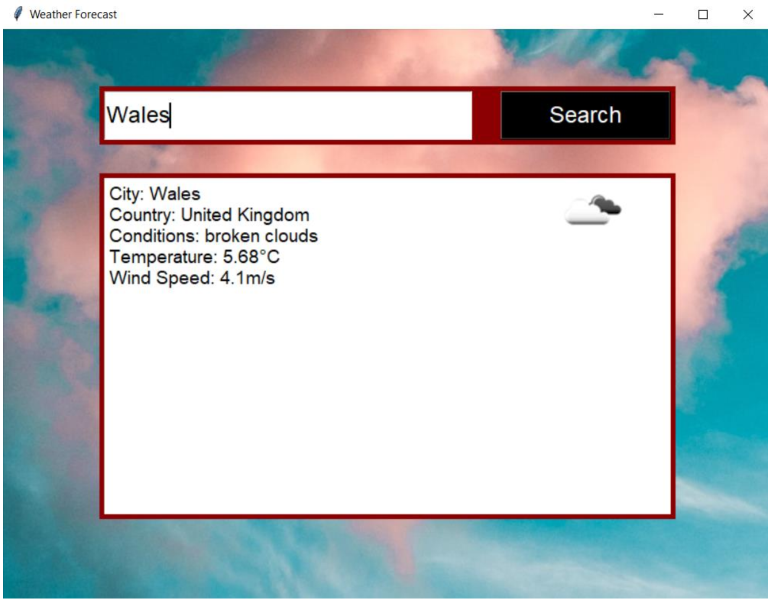


Figure 1
The main display when the program run before the user gives any actions



Please make sure you input a valid name

Figure 2
The output of the program if the user's input is not valid and cannot be find in the API directory



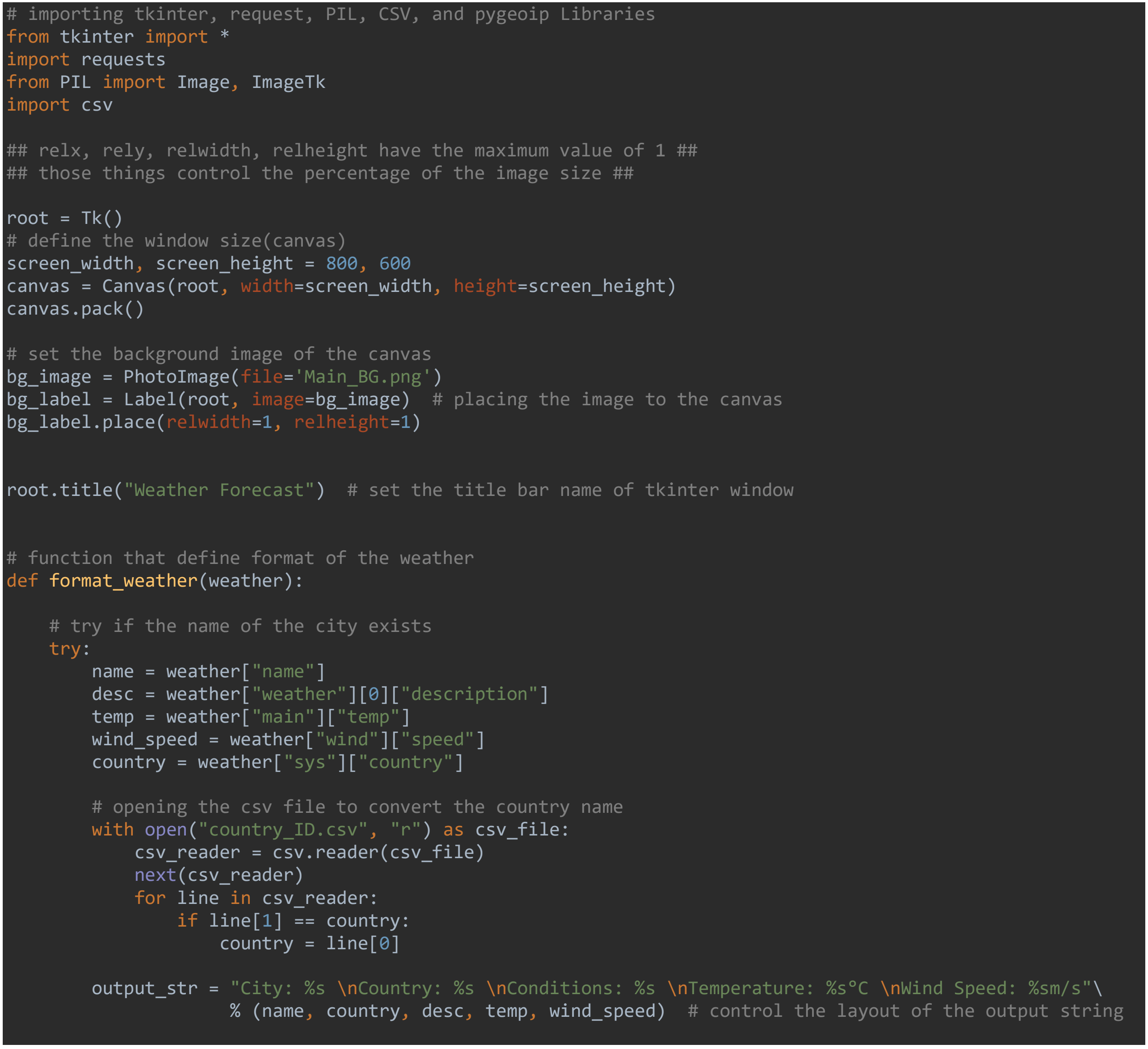
```
City: Wales
Country: United Kingdom
Conditions: broken clouds
Temperature: 5.68°C
Wind Speed: 4.1m/s
```

Figure 3

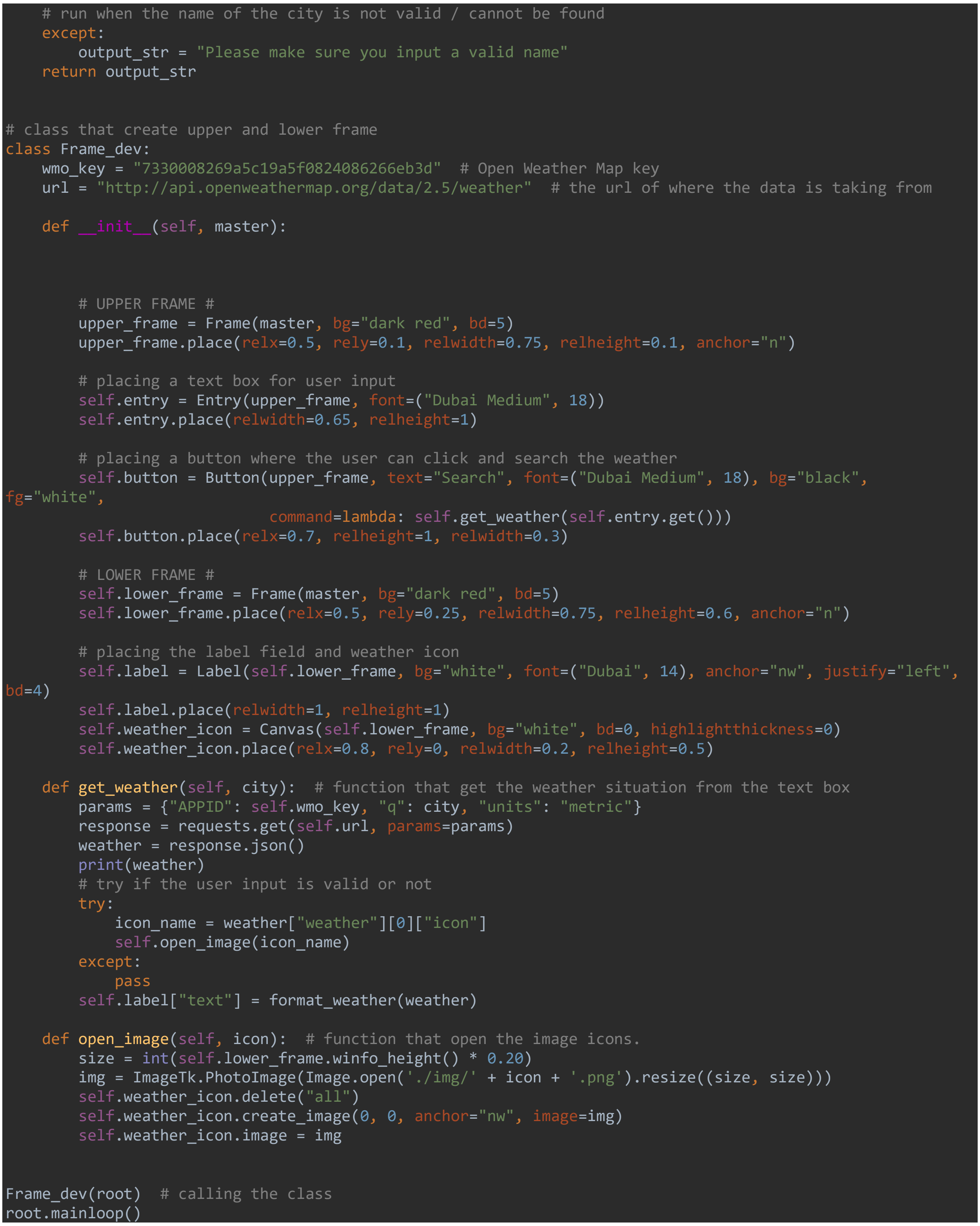
The output of the program if the user's input is valid

IV. Source Code

a. Weather_Forecast.py

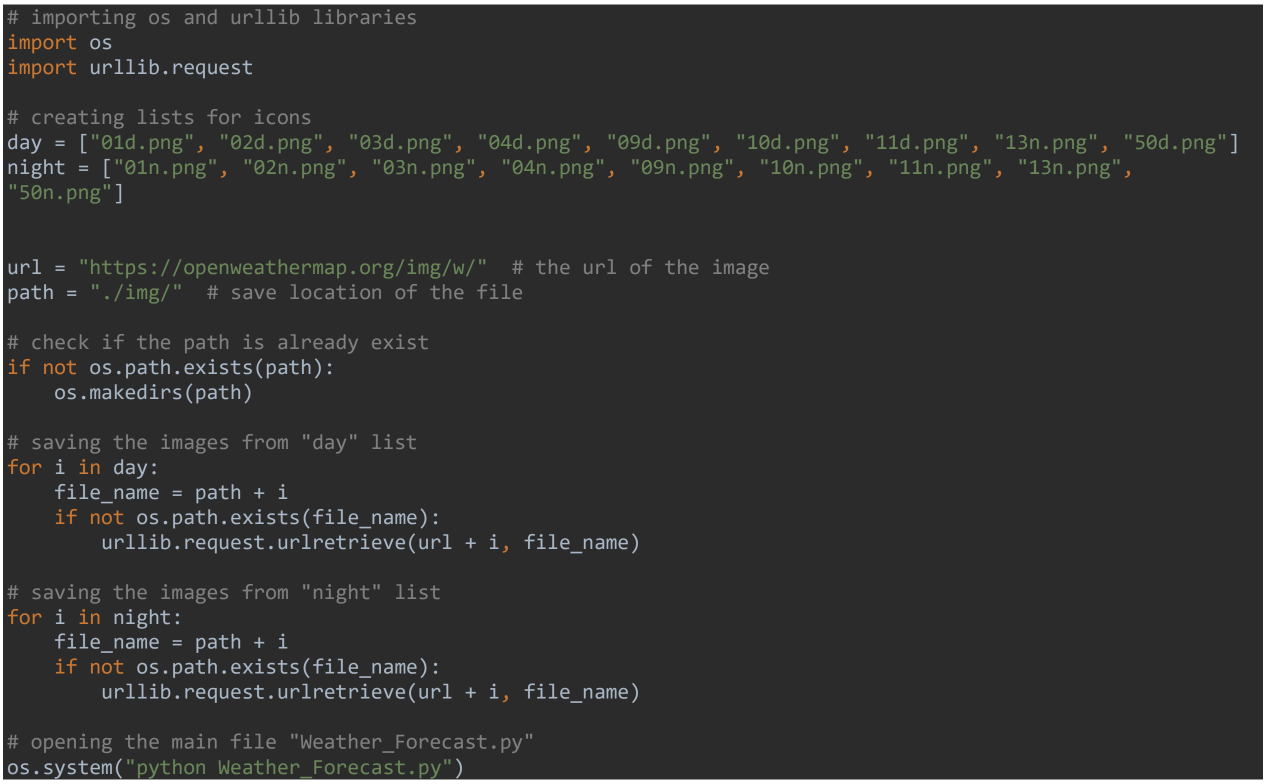


```python
# importing tkinter, request, PIL, CSV, and pygeoip Libraries
from tkinter import *
import requests
from PIL import Image, ImageTk
import csv

## relx, rely, relwidth, relheight have the maximum value of 1 ##
## those things control the percentage of the image size ##

root = Tk()
# define the window size(canvas)
screen_width, screen_height = 800, 600
canvas = Canvas(root, width=screen_width, height=screen_height)
canvas.pack()

# set the background image of the canvas
bg_image = PhotoImage(file='Main_BG.png')
bg_label = Label(root, image=bg_image)  # placing the image to the canvas
bg_label.place(relwidth=1, relheight=1)


root.title("Weather Forecast")  # set the title bar name of tkinter window


# function that define format of the weather
def format_weather(weather):

    # try if the name of the city exists
    try:
        name = weather["name"]
        desc = weather["weather"][0]["description"]
        temp = weather["main"]["temp"]
        wind_speed = weather["wind"]["speed"]
        country = weather["sys"]["country"]

        # opening the csv file to convert the country name
        with open("country_ID.csv", "r") as csv_file:
            csv_reader = csv.reader(csv_file)
            next(csv_reader)
            for line in csv_reader:
                if line[1] == country:
                    country = line[0]

        output_str = "City: %s \nCountry: %s \nConditions: %s \nTemperature: %s°C \nWind Speed: %sm/s"\
                     % (name, country, desc, temp, wind_speed)  # control the layout of the output string
```



```
    # run when the name of the city is not valid / cannot be found
    except:
        output_str = "Please make sure you input a valid name"
    return output_str


# class that create upper and lower frame
class Frame_dev:
    wmo_key = "7330008269a5c19a5f0824086266eb3d"  # Open Weather Map key
    url = "http://api.openweathermap.org/data/2.5/weather"  # the url of where the data is taking from

    def __init__(self, master):



        # UPPER FRAME #
        upper_frame = Frame(master, bg="dark red", bd=5)
        upper_frame.place(relx=0.5, rely=0.1, relwidth=0.75, relheight=0.1, anchor="n")

        # placing a text box for user input
        self.entry = Entry(upper_frame, font=("Dubai Medium", 18))
        self.entry.place(relwidth=0.65, relheight=1)

        # placing a button where the user can click and search the weather
        self.button = Button(upper_frame, text="Search", font=("Dubai Medium", 18), bg="black",
fg="white",
                             command=lambda: self.get_weather(self.entry.get()))
        self.button.place(relx=0.7, relheight=1, relwidth=0.3)

        # LOWER FRAME #
        self.lower_frame = Frame(master, bg="dark red", bd=5)
        self.lower_frame.place(relx=0.5, rely=0.25, relwidth=0.75, relheight=0.6, anchor="n")

        # placing the label field and weather icon
        self.label = Label(self.lower_frame, bg="white", font=("Dubai", 14), anchor="nw", justify="left",
bd=4)
        self.label.place(relwidth=1, relheight=1)
        self.weather_icon = Canvas(self.lower_frame, bg="white", bd=0, highlightthickness=0)
        self.weather_icon.place(relx=0.8, rely=0, relwidth=0.2, relheight=0.5)

    def get_weather(self, city):  # function that get the weather situation from the text box
        params = {"APPID": self.wmo_key, "q": city, "units": "metric"}
        response = requests.get(self.url, params=params)
        weather = response.json()
        print(weather)
        # try if the user input is valid or not
        try:
            icon_name = weather["weather"][0]["icon"]
            self.open_image(icon_name)
        except:
            pass
        self.label["text"] = format_weather(weather)

    def open_image(self, icon):  # function that open the image icons.
        size = int(self.lower_frame.winfo_height() * 0.20)
        img = ImageTk.PhotoImage(Image.open('./img/' + icon + '.png').resize((size, size)))
        self.weather_icon.delete("all")
        self.weather_icon.create_image(0, 0, anchor="nw", image=img)
        self.weather_icon.image = img


Frame_dev(root)  # calling the class
root.mainloop()
```

b. Get_Icon.py



```
# importing os and urllib libraries
import os
import urllib.request

# creating lists for icons
day = ["01d.png", "02d.png", "03d.png", "04d.png", "09d.png", "10d.png", "11d.png", "13n.png", "50d.png"]
night = ["01n.png", "02n.png", "03n.png", "04n.png", "09n.png", "10n.png", "11n.png", "13n.png",
"50n.png"]


url = "https://openweathermap.org/img/w/"  # the url of the image
path = "./img/"  # save location of the file

# check if the path is already exist
if not os.path.exists(path):
    os.makedirs(path)

# saving the images from "day" list
for i in day:
    file_name = path + i
    if not os.path.exists(file_name):
        urllib.request.urlretrieve(url + i, file_name)

# saving the images from "night" list
for i in night:
    file_name = path + i
    if not os.path.exists(file_name):
        urllib.request.urlretrieve(url + i, file_name)

# opening the main file "Weather_Forecast.py"
os.system("python Weather_Forecast.py")
```

V. Problem that Have Been Overcome

Creating this program is not as easy as I expected, because I use tkinter library which is pretty new for me. And also I am pretty unfamiliar with requests, PIL, and urllib. Although almost all of those library is build-in from python itself, but I never use one of them before. So I have to do a lot of searching from google and also watch a lot of tutorial explaining those library on youtube. The other problem that I overcome is the lack of class at the end of the program, so I have to modify the code to work with class, so the class implementation in this program is far from perfect and therefore it can be improve to make it simpler and easier to work with. Because of a pretty bad implementation of the class, I have to cancel my plan to use the IP of the device to determine the location and automatically show the weather when the program is run for the first time.

VI. Sources

- https://openweathermap.org
- https://www.youtube.com/watch?v=D8-snVfekto
- https://datahub.io/core/country-list#resource-data
- https://www.youtube.com/watch?v=IYHJRnVOFlw