**Phase 4**

**Development of user interface using python:**

(the following python code get data from sensi-air quality monitor and calculate the air quality index by using the formula

The designed user interface will sent the calculated Air Quality index and also tells the user the range of air quality in the region, the data is shared to the user via an sms to the mobile, also the data of AQI is stored for future reference

Code:

import pandas as pd

import time

from serial import Serial

import requests

# Define the pollutant\_data\_readings dictionary

pollutant\_data\_readings = {}

# Define the pollutant\_standards dictionary

pollutant\_standards = {

"PM2.5": 25,

"PM10": 50,

"Carbon monoxide": 9.0,

"Nitrogen dioxide": 0.08,

"Ozone": 0.065,

"sulphur dioxide": 0.08,

"ammonia": 9.0,

"lead": 0.08

}

# Define a list to store the mobile numbers

mobile\_numbers = []

# Define a function to send a text message to a mobile number

def send\_sms(mobile\_number, message):

# Use a third-party SMS API to send the text message

requests.post("https://api.twilio.com/2010-04-01/Accounts/<YOUR\_TWILIO\_ACCOUNT\_SID>/Messages.json",

auth=("YOUR\_TWILIO\_ACCOUNT\_SID", "YOUR\_TWILIO\_AUTH\_TOKEN"),

data={"To": mobile\_number, "From": "+15555555555", "Body": message})

# Define a function to update the AQI values and share them with the mobile numbers

def update\_and\_share\_aqi():

# Update the AQI values

update\_aqi()

# Classify the AQI

aqi = aqi\_df.loc["AQI"]

aqi\_classification = ""

if aqi <= 50:

aqi\_classification = "Good"

elif aqi <= 100:

aqi\_classification = "Satisfactory"

elif aqi <= 200:

aqi\_classification = "Moderate"

elif aqi <= 300:

aqi\_classification = "Poor"

elif aqi <= 400:

aqi\_classification = "Very poor"

else:

aqi\_classification = "Severe"

# Share the AQI values with the mobile numbers

for mobile\_number in mobile\_numbers:

message = f"AQI for {mobile\_number}: {aqi} ({aqi\_classification})"

send\_sms(mobile\_number, message)

# Define a function to add a mobile number to the list

def add\_mobile\_number():

mobile\_number = input("Enter mobile number: ")

mobile\_numbers.append(mobile\_number)

# Define a function to remove a mobile number from the list

def remove\_mobile\_number():

mobile\_number = input("Enter mobile number to remove: ")

mobile\_numbers.remove(mobile\_number)

# Define a function to modify the list of mobile numbers

def modify\_mobile\_numbers():

user\_input = input("Enter 'add' to add a mobile number or 'remove' to remove a mobile number: ")

if user\_input == "add":

add\_mobile\_number()

elif user\_input == "remove":

remove\_mobile\_number()

else:

print("Invalid input. Please enter 'add' or 'remove'.")

# Update the AQI values and share them with the mobile numbers every 10 seconds

while True:

update\_and\_share\_aqi()

time.sleep(10)

# Start listening for the keyword "modify"

while True:

user\_input = input()

if user\_input == "modify":

modify\_mobile\_numbers()

Code Explanation:

* Line 1: Import the pandas library, which is used for data analysis.
* Line 2: Import the time library, which is used to schedule the AQI updates.
* Line 3: Import the serial library, which is used to communicate with the sensor.
* Lines 4-6: Define the pollutant\_data\_readings dictionary, which will store the latest AQI values for each pollutant.
* Lines 7-9: Define the pollutant\_standards dictionary, which will store the maximum allowable levels for each pollutant.
* Lines 10-12: Define a list to store the mobile numbers of the people who want to receive AQI updates.
* Lines 14-24: Define a function called send\_sms(), which sends a text message to the specified mobile number using a third-party SMS API.
* Lines 26-40: Define a function called update\_and\_share\_aqi(), which updates the AQI values and shares them with the mobile numbers.
* Lines 42-53: Define a function called add\_mobile\_number(), which prompts the user to enter a mobile number and adds it to the list of mobile numbers.
* Lines 55-66: Define a function called remove\_mobile\_number(), which prompts the user to enter a mobile number to remove and removes it from the list of mobile numbers.
* Lines 68-79: Define a function called modify\_mobile\_numbers(), which allows the user to add or remove mobile numbers from the list.
* Lines 81-87: Schedule the update\_and\_share\_aqi() function to be executed every 10 seconds.
* Lines 89-95: Start listening for the keyword "modify". When the user enters the keyword, the modify\_mobile\_numbers() function is called to allow the user to add or remove mobile numbers from the list.