AIR QUALITY MONITORING SYSTEM

1. SENSOR CONNECTIONS:

Connect the sensor's pins to the corresponding pins on your Raspberry Pi or Arduino.

Sensor VCC to Raspberry Pi 3.3V (or Arduino 3.3V)

Sensor GND to Raspberry Pi GND (or Arduino GND)

Sensor SDA to Raspberry Pi SDA (or Arduino A4 on an Arduino Uno)

Sensor SCL to Raspberry Pi SCL (or Arduino A5 on an Arduino Uno)

2.PYTHON SCRIPT ON RASPERRY PI:

import time

import smbus2

import bme680

# Create a BME680 sensor object

bus = smbus2.SMBus(1) # I2C bus 1

bme = bme680.BME680(i2c\_addr=0x76) # Sensor address may vary

# Set up the sensor for air quality monitoring

bme.set\_humidity\_oversample(bme680.OS\_2X)

bme.set\_pressure\_oversample(bme680.OS\_4X)

bme.set\_temperature\_oversample(bme680.OS\_8X)

bme.set\_filter(bme680.FILTER\_SIZE\_3)

# Define a function to read and print sensor data

def read\_sensor\_data():

if bme.get\_sensor\_data():

temperature = bme.data.temperature

humidity = bme.data.humidity

pressure = bme.data.pressure

air\_quality = bme.data.gas\_resistance

return temperature, humidity, pressure, air\_quality

else:

return None

try:

while True:

sensor\_data = read\_sensor\_data()

if sensor\_data:

temperature, humidity, pressure, air\_quality = sensor\_data

print(f"Temperature: {temperature:.2f}°C")

print(f"Humidity: {humidity:.2f}%")

print(f"Pressure: {pressure:.2f} hPa")

print(f"Air Quality: {air\_quality} Ohms\n")

else:

print("Failed to read data from the sensor.")

time.sleep(10) # Read data every 10 seconds

except KeyboardInterrupt:

pass

3.CLOUD OR MOBILE APP INTEGRATION:

On your cloud server or mobile app, set up an endpoint to receive data from the Raspberry Pi. Process the incoming data and update the parking space occupancy status accordingly.

Ensure that your server has proper security measures, like HTTPS and authentication, to protect the data transmission.

4.TESTING:

Run the Python script on your Raspberry Pi by executing python example.py in the terminal.

Monitor the output to ensure the correct distance measurements are being obtained.

Verify that data is being sent to your cloud server or mobile app and that the occupancy status is being updated correctly.

Depending on our project requirements, we might need to add more sensors, implement error handling, or optimize the code for better performance and reliability. Additionally, security considerations are crucial, especially if we are dealing with sensitive data