Phase 3 : IoT Development part 1

To send real-time occupancy and cleanliness data to a restroom information platform involves a combination of hardware and software components. I'll provide you with a high-level example of how you might structure such a project. Keep in mind that the specific implementation will depend on the hardware and communication protocols used in your IoT sensors

Using MQTT for communication between the IoT sensors and the platform. MQTT is a lightweight messaging protocol commonly used in IoT applications.

Step 1: Setting up the IoT Sensors

You'll need IoT sensors capable of detecting restroom occupancy and cleanliness. These sensors should be programmed to send data to a central server. Below is a simplified example for an occupancy sensor.

Program:

```
import paho.mqtt.client as mqtt
import time
import random
 def simulate_occupancy_sensor():
    client = mqtt.Client()
    client.connect("mqtt.broker.com", 1883)
    while True:
        occupancy = random.choice([0, 1])  # 0 for empty, 1 for occupied
        client.publish("restroom/occupancy", payload=occupancy, qos=1)
        time.sleep(5)  # Send data every 5 seconds
 if __name__ == "__main__":
    simulate_occupancy_sensor()
```

You would have a similar script for cleanliness data.

Step 2: Setting up the Restroom Information Platform

Your platform should have an MQTT broker and a backend to process incoming data. You can use a library like Eclipse Paho to create an MQTT broker.

Step 3: Receive and Process Data on the Restroom Platform

On your platform, you need to create a Python script to receive data from the sensors, process it, and update the restroom information accordingly.

Program:

```
import paho.mqtt.client as mqtt
 def on_message(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode("utf-8")
     if topic == "restroom/occupancy":
        process_occupancy_data(payload)
    elif topic == "restroom/cleanliness":
        process_cleanliness_data(payload)
 def process_occupancy_data(data):
       print(f"Occupancy data received: {data}")
 def process_cleanliness_data(data):
       print(f"Cleanliness data received: {data}")
client = mqtt.Client()
client.connect("mqtt.broker.com", 1883)
client.on_message = on_message
client.subscribe("restroom/occupancy", qos=1)
client.subscribe("restroom/cleanliness", qos=1)
 client.loop_forever()
```

This script subscribes to MQTT topics for occupancy and cleanliness data, processes the data, and updates the restroom information accordingly.

Step 4: Run the Scripts

Now you can run both the sensor and platform scripts. Make sure to replace `"mqtt.broker.com"` with the actual MQTT broker address we are using.

This is a basic example, and in a real-world scenario, we would need to handle more complex tasks such as data persistence, security, and potentially use a proper IoT framework depending on the scale of your project.