

# Proposal Cloud and Edge computing

## Title: Smart Air Quality and Fire Monitoring System for University Classrooms

Groupmembers: Layton Berth, Kemp Dewulf Draft: 1

### Overview project

We found that our classrooms at Howest are poorly managed in terms of air quality, with no direct system in place to monitor or improve it, which led us to develop this solution.

This system monitors the air quality within university classrooms using multi-sensor integration. It continuously tracks pollutants (e.g., CO2, VOCs, particulate matter), temperature, and humidity to ensure a healthy environment. The system also incorporates infrared technology to detect early signs of fire by monitoring sudden temperature spikes.

When unsafe levels of pollutants or fire risks are detected, the system will trigger alarms and interact with smart devices to take appropriate actions—such as opening windows, activating air conditioning, or unlocking doors to improve ventilation and ensure safety. The system is controlled and monitored via a Flutter mobile app, which allows users to view real-time data, receive alerts, and manually control connected devices.

## Cloud & Edge Computing

### Used hardware

- IC: ESP32 Chip Sensors:
  - Humidity sensor
  - Temperature sensor
  - CO2 sensor

### Communication

The communication between the monitoring devices (ESP32) and the Android App (Android Phone) is via WiFi and MQTT protocol for real-time messaging.

#### **Sending from ESP32 to Base Station:**

Each ESP32 device is assigned to a specific classroom and includes this information in every message. The system collects air quality data, monitors fire risks, and sends the information to the Android app, tagging it with the classroom ID to differentiate between different locations.

The messages are in JSON, in the following format:

```
{
  "classroom": "BST.A.1.101",
  "action": "monitoring",
  "CO2": "500 ppm",
  "temperature": "22°C",
  "humidity": "50%",
  "fire_detected": "false"
}
```

### Data gathering

- Classroom ID: Identifies the specific classroom (saved as varchar)
- CO2 levels: in parts per million (ppm), saved as float
- Temperature: in degrees Celsius, saved as float
- Humidity: in percentage, saved as float
- Fire detection: Boolean (true/false)
- Start time: datetime
- End time: datetime
- Sensor health status: Indicates if sensors are working correctly (saved as boolean)
- Alarm status: Boolean (active/inactive)

### Edge computing

When the air quality is within safe levels, or when no fire risks are detected, the device reduces the frequency of data transmission to save energy. Data is processed locally on the ESP32 to trigger immediate actions (such as opening windows or activating air conditioning) without relying on cloud communication. Each device (in different classrooms) communicates directly with the smart home devices in the same room, ensuring quick responses to changes in air quality or fire detection. This device-to-device communication does not pass through the cloud unless long-term data logging or remote monitoring is required.

### Events

User event: A user can manually activate ventilation, windows, or alarms via the mobile app if they notice poor air quality or suspect a fire risk.

Sensor event: When the system detects unsafe CO2 levels or a sudden temperature spike (fire risk), it triggers the alarm and notifies nearby rooms and connected devices to take action (e.g., opening windows or turning on air purifiers).

Cloud event: The system logs the event and sends a notification to the university's central monitoring system to alert staff of potential hazards, prompting a review of air quality trends or emergency procedures.

## Computations

- Average CO2 levels over time for each classroom.
- Average temperature and humidity trends in different classrooms.
- Frequency of air quality threshold breaches (e.g., CO2, temperature) per classroom.
- Time taken to resolve air quality issues (e.g., how long it takes for CO2 levels to normalize after ventilation).
- Number of fire risks detected and duration until resolution.
- Device uptime and health status, calculating the percentage of time sensors are operational.
- Number of manual user interventions (e.g., activating ventilation or alarms via the app).

## CMS

---

The application continuously monitors air quality data and logs it into a centralized content management system (CMS). The CMS provides:

- A dashboard displaying real-time air quality metrics (CO2 levels, temperature, humidity) for each classroom.
- Historical data visualization for trends over time, allowing users to track air quality improvements or issues.
- Alerts and notifications logged for user reference, detailing when and why alarms were triggered.
- Educational content about air quality and fire safety, providing users with information on maintaining a healthy learning environment.

## Trending Topics

---

Application 1 in Rust (LANGUAGE TBD): Classroom Air Quality Dashboard with the following content:

- Real-time CO2 levels and alerts
- Historical air quality data trends for each classroom
- Rankings of classrooms based on air quality scores (e.g., classrooms with the best air quality)
- User feedback and reporting system for air quality issues
- Fire safety notifications and historical incidents in classrooms