UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

Smart Gym Monitoring System

Eric Rode
Göktuğ Gençkaya
Jorge Leon

CONTENTS

INTRODUCTION

For many gym-goers, nothing is more frustrating than dealing with overcrowded spaces, or uncomfortable workout conditions like poor air quality.

To address this issue, there is a growing need for smarter, more connected solutions that provide real-time insights into gym conditions and user activity. By bridging this gap, gyms can not only enhance customer satisfaction but also streamline their operations to create a safer, more comfortable, and more effective workout environment for everyone.

The primary objectives of this project are to monitor gym usage and environmental parameters, provide actionable data through visual dashboards, and enable real-time alerting mechanisms. These capabilities aim to streamline gym management, ensuring rapid responses to critical events such as overcrowding or overheating.

Technologically, the system integrates MQTT for data communication, Node-RED for workflow automation, InfluxDB for time-series data storage, and Grafana for real-time visualization. Mock sensors were used to simulate gym data, enabling development and testing under controlled conditions. Additionally, Telegram bots provide a real-time alerting mechanism to notify staff and users about potential issues.

This report covers the scope of the project, including sensor integration, data visualization, and alerting mechanisms. While the system focuses on monitoring and management within a gym setting, future enhancements could extend its capabilities, such as integrating wearable devices or equipment tracking for a more comprehensive fitness solution.

OBJECTIVES

1. Develop a Reliable IoT System for Real-Time Monitoring: The system is designed to ensure continuous, accurate monitoring of key gym metrics such as:

Occupancy levels: Track how many users are in the gym at any given time in each area.
Environmental factors: Measure air quality, temperature, humidity and noise to ensure a safe and comfortable workout environment.

Sensors collect data in real time, publishing it to an MQTT broker for efficient communication. Reliability is prioritized by implementing robust communication protocols (MQTT) and error-handling mechanisms to reduce data loss.

2. Implement Automated Detection and Real-Time Alerts: The system uses defined thresholds to detect abnormal conditions, including:

Overcrowding: When gym occupancy exceeds a preset limit.
Overheating: If temperature rises beyond a comfortable range.

Alerts are sent via **Telegram bots**, which provide immediate notifications to gym staff. The thresholds when a notification should be sent can be modified using a telegram bot.

3. Provide Intuitive and Understandable Dashboards: Dashboards are built using **Grafana**, which is connected to an InfluxDB time-series database to retrieve and display data. The design focuses on simplicity and clarity to ensure quick interpretation of key metrics. Visualizations include:

- Real-time graphs for occupancy, air quality, humidity and temperature
- Overview over each single sensor in a given gym
- Selection of time frame and gym of the shown data

FUNCTIONAL REQUIREMENTS

The Smart Gym Monitoring System is designed to simulate real-time data for critical metrics such as occupancy, temperature, humidity, noise and air quality, ensuring a comprehensive view of gym conditions (SGMS-FR001). This data is collected from sensors and transmitted to a centralized server for processing and storage (SGMS-FR002). The system detects abnormal conditions, such as overcrowding, and immediately notifies staff through an alerting mechanism (SGMS-FR003). A real-time monitoring interface is provided via dashboards, offering an intuitive display of environmental data (SGMS-FR004). In critical situations, a notification system sends alerts through Telegram to ensure rapid response.

Identifier	Name	Description	Priority
SGMS-FR001	Data Generation	Simulate real-time sensor data for occupancy, temperature, and air quality.	High
SGMS-FR002	Data Collection	Collect and transmit data from all sensors to a centralized message broker and processing system for analysis and storage.	High
SGMS-FR003	Alerts	Detect abnormal conditions and notify staff.	High
SGMS-FR004	Monitoring Interface	Display real-time data through a dashboard.	Medium
SGMS-FR005	Notification System	Send alerts via Telegram for critical conditions.	High

NON-FUNCTIONAL REQUIREMENTS

The system ensures responsiveness by triggering alerts within 10 seconds of detecting abnormal events, prioritizing real-time monitoring (SGMS-NFR001). Reliability is a cornerstone of the system, maintaining uptime with minimal disruptions for maintenance (SGMS-NFR002). The design supports scalability, allowing additional sensors and gyms to be integrated seamlessly as needed (SGMS-NFR003). Lastly, usability is achieved through intuitive dashboards and clear visualizations, enabling gym staff to monitor conditions effortlessly (SGMS-NFR004).

Identifier	Name	Description	Priority
SGMS-NFR001	Response	Alerts should trigger within 10 seconds of detecting an abnormal event.	High
SGMS-NFR002	Reliability	Ensure continuous operation of the system by minimizing disruptions during maintenance activities.	Low
SGMS-NFR003	Scalability	Support additional sensors and gyms as needed.	High
SGMS-NFR004	Usability	Provide intuitive dashboards and clear visualizations for easy monitoring.	Medium

SYSTEM OVERVIEW

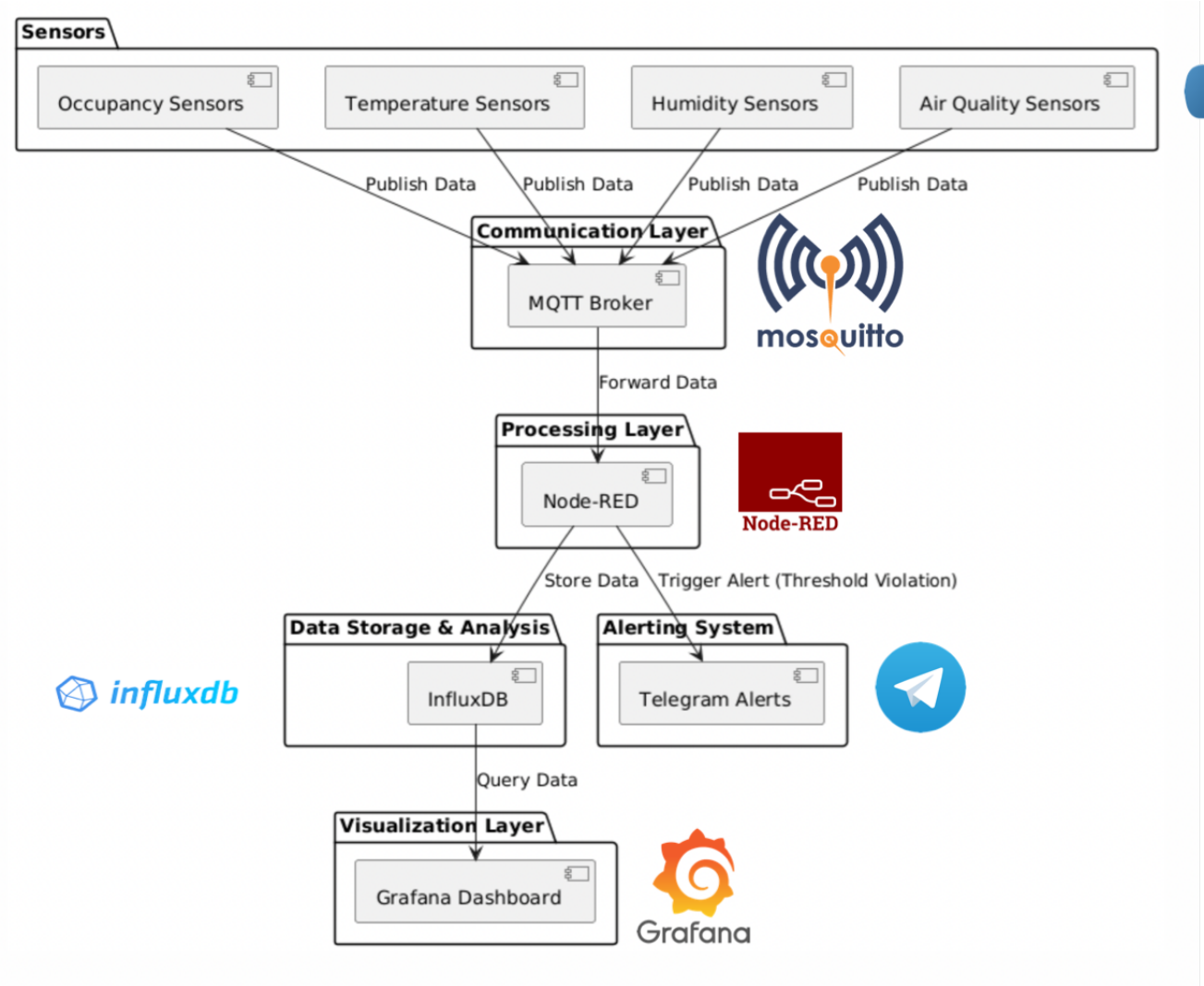
The Smart Gym Monitoring System integrates IoT technology to ensure real-time monitoring of gym conditions while providing actionable insights for both gym staff and users. The system leverages mock sensors to simulate critical metrics such as occupancy of each gym area, temperature, air quality (CO2, CO, NO2, PM2.5, PM10). These metrics are processed through the system's middleware to calculate additional insights, including Air Quality Index (AQI) and total occupancy, and are visualized in Grafana dashboards.

The system ensures efficient communication using MQTT to publish sensor data, which is collected and processed by Node-RED. Alerts for critical conditions, such as high occupancy or poor air quality, are sent via Telegram bots. Data is stored in InfluxDB, enabling real-time and historical analysis. Grafana dashboards provide detailed and interactive visualizations, allowing gym operators to monitor trends and respond to anomalies effectively.

SYSTEM ARCHITECTURE

The architecture of the Smart Gym Monitoring System is designed to process, analyze, and visualize gym data efficiently. It comprises the following components:

1. **Sensors**:
    - Mock sensors generate real-time data for key metrics, including:
        - **Occupancy**: Tracks gym usage.
        - **Environmental Sensors:** Temperature, Noise Level and Humidity, Monitors environmental conditions to maintain comfort.
        - **Air Quality**: Metrics such as CO2, CO, NO2, PM2.5, and PM10 are measured to calculate the AQI.
2. **Communication Layer (MQTT Broker)**:
    - The Mosquitto MQTT broker manages the real-time transfer of sensor data to downstream systems using structured topics like `/smartgym/{location}/{sensor_type}`.
3. **Data Collection and Middleware**:
    - **Node-RED**: Processes the collected data to:
        - Detect threshold violations and trigger alerts.
        - Perform calculations such as AQI based on air quality metrics.
        - Batch data for reducing traffic to database
        - Send alerts to Telegram bots.
4. **Data Storage (InfluxDB)**:
    - Stores time-series data for real-time and historical analysis. Metrics like AQI, noise levels, and temperature trends are retained for visualization and decision-making.
5. **Visualization (Grafana Dashboards)**:
    - Provides interactive dashboards to monitor:
        - Real-time occupancy, temperature and humidity levels.
        - AQI and air metrics.
        - Noise levels and occupancy of each gym area.
6. **Alerting System**:
    - Real-time notifications are sent via Telegram bots, ensuring prompt action for critical events like overcrowding or bad air quality.



SENSOR INTEGRATION

The system simulates sensor data to replicate real-world conditions, enabling testing and evaluation of the IoT architecture. The integration of mock sensors and the MQTT communication protocol ensures a reliable data flow from sensors to downstream systems for analysis and visualization.

The system simulates multiple sensor types, each providing critical metrics for monitoring gym environments:

- **Temperature Sensors**: Generate random values between 18°C and 30°C to reflect ambient temperature.
- **Humidity Sensors**: Provide values between 30% and 70% to monitor air moisture levels.
- **CO2 Level Sensors**: Calculate carbon dioxide levels based on occupancy, starting from a base value of 400 ppm with a dynamic increase per person.
- **CO Level Sensors**: Estimate carbon monoxide levels with a base value of 0.1 ppm, accounting for emissions from occupancy.

- **NO2 Level Sensors**: Simulate nitrogen dioxide levels, starting from a base of 20 ppb, with an increase proportional to occupancy.
- **PM2.5 and PM10 Sensors**: Measure fine and coarse particulate matter, calculated based on occupancy levels.
- **Noise Level Sensors**: Randomly generate noise levels between 40 dB and 90 dB to assess the acoustic environment.
- **Utilization Sensors**: Distribute occupants across gym areas (e.g., strength training, aerobic zones) using maximum capacity thresholds from the `.env` file.

```python
for sensor_type, sensors in location["sensors"].items():
    for sensor_id in sensors:
        value = None

        if sensor_type == "temperature":
            value = round(random.uniform(18, 30), 1)
        elif sensor_type == "humidity":
            value = round(random.uniform(30, 70), 1)
        elif sensor_type == "co2_level":
            value = get_co2_level(occupancy)
        elif sensor_type == "no2_level":
            value = get_no2_level(occupancy)
        elif sensor_type == "noise_level":
            value = round(random.uniform(40, 90), 1)
        elif sensor_type == "pm2_5":
            value = get_pm25_level(occupancy)
        elif sensor_type == "pm10":
            value = get_pm10_level(occupancy)
        elif sensor_type == "co_level":
            value = get_co_level(occupancy)
        elif sensor_type == "utilization":
            # Get the dictionary of distribution and publish each key/value separately.
            distribution_dict = get_people_distribution(occupancy)

            for dist_key, dist_val in distribution_dict.items():
                dist_topic = f"/smartgym/{name}/{dist_key}"
                dist_payload = {
                    "timestamp": simulated_time.isoformat(),
                    "sensor_id": sensor_id,
                    "value": dist_val,
                }
                result = client.publish(dist_topic, json.dumps(dist_payload))
                if result.rc == 0:
                    print(f"Published to {dist_topic}: {dist_payload}")
                else:
                    print(f"Failed to publish to {dist_topic}")
            # After publishing distribution, we can skip the single publish below
            continue  # Skip the default publish step
```

- Data is published every 10 seconds. Publishing frequency is controlled by the time.sleep() statement at the end of each loop.
- Each message is sent in JSON format, containing:

```
{
  "timestamp": "2025-01-19T12:34:56.789Z",
  "sensor_id": "sensor_Olympus_Gym_temperature_1",
  "value": 25.3
}
```

Communication Protocol

The system utilizes MQTT (Message Queuing Telemetry Transport) for lightweight, efficient, and real-time communication. The Mosquitto MQTT broker acts as the central hub, managing communication between sensors and downstream services. The broker is configured with credentials to ensure secure communication.

- Each sensor publishes data to structured MQTT topics, following the format: /smartgym/{gym_location}/{sensor_type}
  - /smartgym/Olympus_Gym/temperature
  - /smartgym/High_Tech_Gym/pm2_5
- Occupancy is distributed across gym areas and published to area-specific topics, such as: /smartgym/Olympus_Gym/strength_training

The `.env` file specifies thresholds for each sensor type, enabling anomaly detection. For example, CO2 levels exceeding 1500 ppm or noise levels above 85 dB trigger alerts. These thresholds are used downstream in Node-RED to generate real-time Telegram notifications.

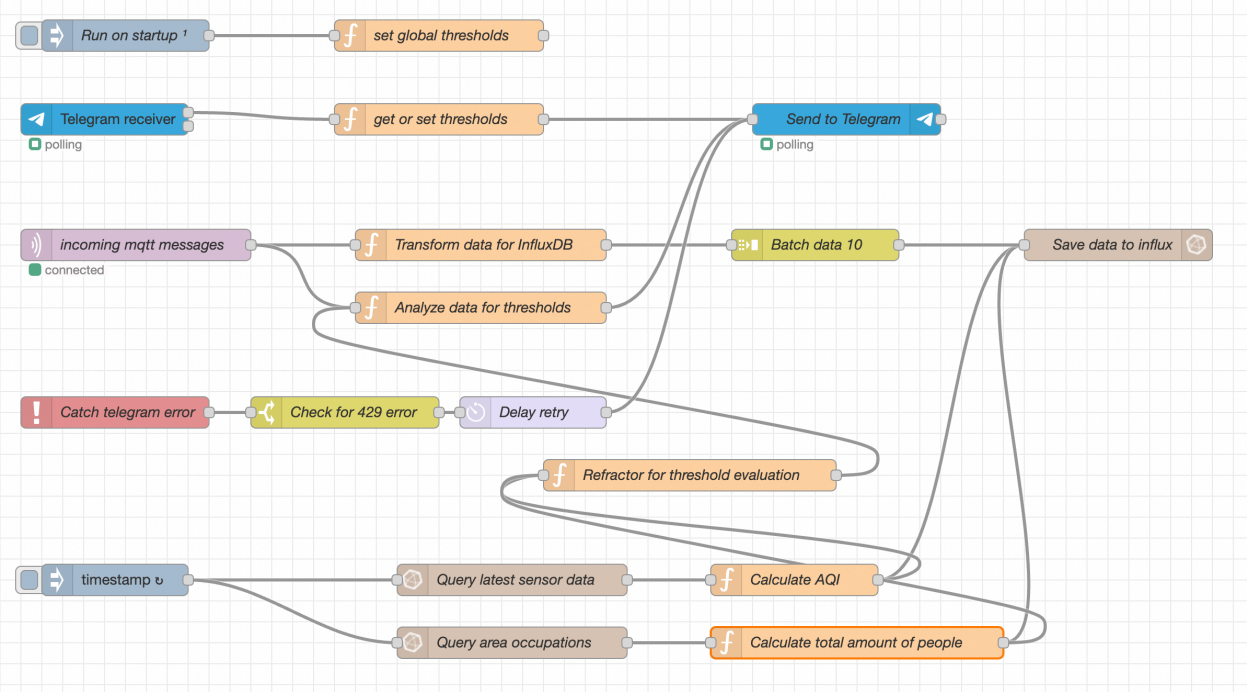
```
CO2_THRESHOLD=1500
CO_THRESHOLD=10
NO2_THRESHOLD=0.1
NOISE_THRESHOLD=85
AQI_THRESHOLD=100
TEMPERATURE_THRESHOLD=26
HUMIDITY_THRESHOLD=60
STRENGTH_TRAINING_THRESHOLD=40
AEROBIC_THRESHOLD=20
FUNCTIONAL_THRESHOLD=20
NO_EQUIPMENT_THRESHOLD=20
TOTAL_PEOPLE_THRESHOLD=100
PM10_THRESHOLD=100
PM2_5_THRESHOLD=35
```

MIDDLEWARE

The middleware in the Smart Gym Monitoring System is implemented using Node-RED, which acts as the central hub for data processing, transformation, and alerting. It facilitates seamless integration between data ingestion, preprocessing, AQI calculation, aggregation, and notification mechanisms, ensuring real-time monitoring.

Data Ingestion

- Data from simulated sensors is published to MQTT topics and ingested by Node-RED via the incoming mqtt messages node.
- Each message contains sensor_id, value and timestamp in JSON format. The location is transmitted via the MQTT topic. MQTT ensures reliable delivery, even in high-frequency scenarios, using structured topics (e.g., /smartgym/{gym_location}/{sensor_type}).



Data Preprocessing

- Sensor values are compared against predefined thresholds (e.g., CO2 > 1500 ppm, PM2.5 > 35 µg/m³) using the 'Analyze data for thresholds' function node.
- Thresholds for each sensor type are initialized globally (e.g., CO2 > 1500 ppm, AQI > 100) using the 'set global thresholds' function. Administrators can dynamically adjust these thresholds via Telegram commands (/set_threshold <sensor_type> <value>)
- Alerts are triggered for values exceeding thresholds, ensuring quick identification of critical issues.
- The 'transform data for InfluxDB' function reformats data for storage. Adds gym location, sensor_id and sensor_type as tags and ensures data integrity by rejecting malformed messages.

AQI Calculation

- Data for air quality metrics (PM10, PM2.5, CO, NO2) is grouped by gym and averaged. AQI is calculated using the U.S. EPA standards with linear interpolation, prioritizing the highest pollutant AQI. For detailed calculation methods, see U.S. Environmental Protection Agency. (2024).
- AQI is stored as a new sensor type (AQI) in InfluxDB for visualization and alerting.

Data Aggregation and Storage

- **Batching**: Data is aggregated into batches of 10 records using the Batch data node (join node) before storage. This reduces traffic to the database.
- **InfluxDB Storage**: Data is written to InfluxDB as time-series records.

Telegram Bot Integration

- The middleware analyzes data against thresholds using the 'Analyze data for thresholds' function. When violations occur, a formatted alert message is sent via Telegram to the SmartGyms_LAquila chat.
- Notifications for each sensor are rate-limited to avoid spamming (minimum interval: 1 minute).

Error Handling Mechanisms

- **Telegram Error Management**: The Catch telegram error node monitors for failures (e.g., rate-limiting errors imposed by telegram). A retry mechanism is implemented to resend notifications after a delay.
- **Data Integrity Checks**: Malformed data (e.g., missing values or invalid sensor types) is logged and excluded from processing.

DATA STORAGE AND VISUALIZATION

The **Smart Gym Monitoring System** utilizes InfluxDB, a high-performance time-series database, for efficient data storage and retrieval. It plays a crucial role in organizing and managing the large volumes of data generated by the gym's simulated sensors, enabling real-time and historical analysis.

Data Organization

Data in InfluxDB is structured into **measurements**, **tags**, **fields**, and **time**, providing a logical and efficient way to store and query time-series data.

- Each gym location is treated as a unique measurement. For example, Olympus_Gym, High_Tech_Gym, Fit4Fun… This organization helps to isolate data per gym for easier queries and visualization.

- Fields store the sensor values recorded over time. These include values: the measured data for metrics like temperature, humidity, occupancy, air quality, etc.
    - Example:

```
{
  "value": 56.36
}
```

- Tags add metadata to each data point for efficient filtering and querying. The tags used in this system are:
    - sensor_type: Identifies the type of sensor (e.g., AQI, co2_level, temperature).
    - sensor_id: Unique identifier for the sensor (e.g., AQI_Fit4Fun).
- Each record includes a timestamp to track when the data was generated or recorded. This allows for time-based analysis and trend visualization. {"_time": "2025-01-16T15:00:00.000Z"}

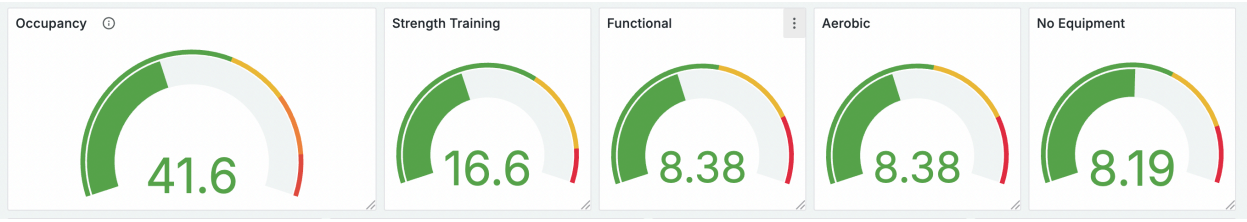
Data Ingestion

- Data from simulated sensors is formatted for InfluxDB by the middleware before storage.
- Example message for storage:

```
{
  "measurement": "Fit4Fun",
   "tags": {
      "sensor_type": "AQI",
      "sensor_id": "AQI_Fit4Fun"
   },
   "fields": {
      "value": 46.09
   },
   "timestamp": 1674567890123
}
```

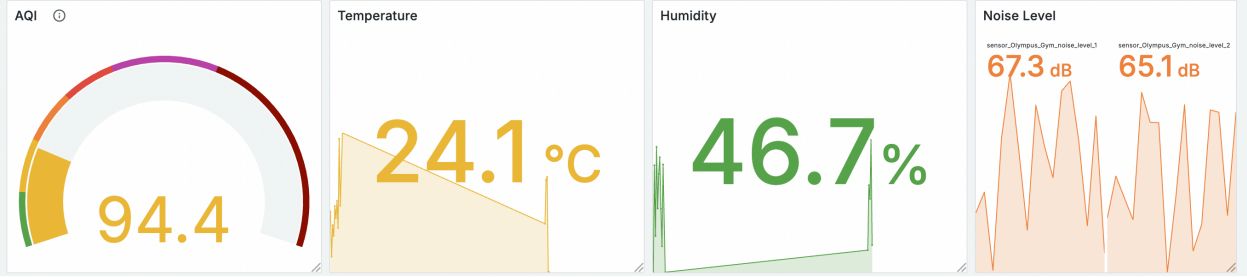
Occupancy

- For a given time period, mean of the sensor values are calculated for each sport area and for the total gym. Leftmost gauge tracks overall gym occupancy while the ones on the right display the distribution of these people in different parts of the gym. From Grafana, one can change the time period in order to keep track of real-time occupancy of each area of the gym.

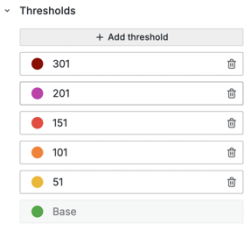


AQI and Environmental Metrics

- The AQI gauge reflects the overall air quality within the gym environment based on pollutants such as PM10, PM2.5, CO, and NO2.



- Thresholds for AQI are color-coded, making it easier to interpret:

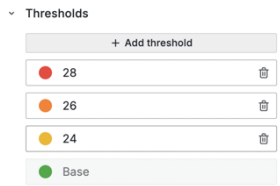


- **Green (0–50)**: Good air quality, safe for everyone.
- **Yellow (51–100)**: Moderate air quality, some pollutants present.
- **Orange (101–150)**: Unhealthy for sensitive groups; precautions needed.
- **Red (151–200)**: Poor air quality; action required to improve conditions.
- **Purple (201–300)**: Very unhealthy air quality, requiring immediate attention.
- **Maroon (>300)**: Hazardous, requires evacuation.

- Real-time AQI helps staff identify periods of poor air quality and adjust ventilation systems accordingly.

Environmental Metrics (Temperature, Humidity, and Noise Levels)

- **Temperature**: Tracks the average temperature of the gym areas in real-time. Thresholds for temperature are adjustable via Grafana:



  - **Green (below 24°C)**: Comfortable conditions.
  - **Yellow (24°C–26°C)**: Warmer conditions; cooling may be required.
  - **Red (above 26°C)**: Too hot; immediate action is needed.

- **Humidity**: Displays the percentage of air moisture in the gym, ensuring comfort and proper air quality. While no specific thresholds are indicated, maintaining a humidity level between 30–60% is ideal for user comfort.
- **Noise Levels**: Gauges show sound intensity in decibels (dB) for different gym areas. Noise levels are monitored to ensure a conducive workout environment, with 65 dB or lower being ideal. High noise levels may indicate overcrowding or misuse of equipment.

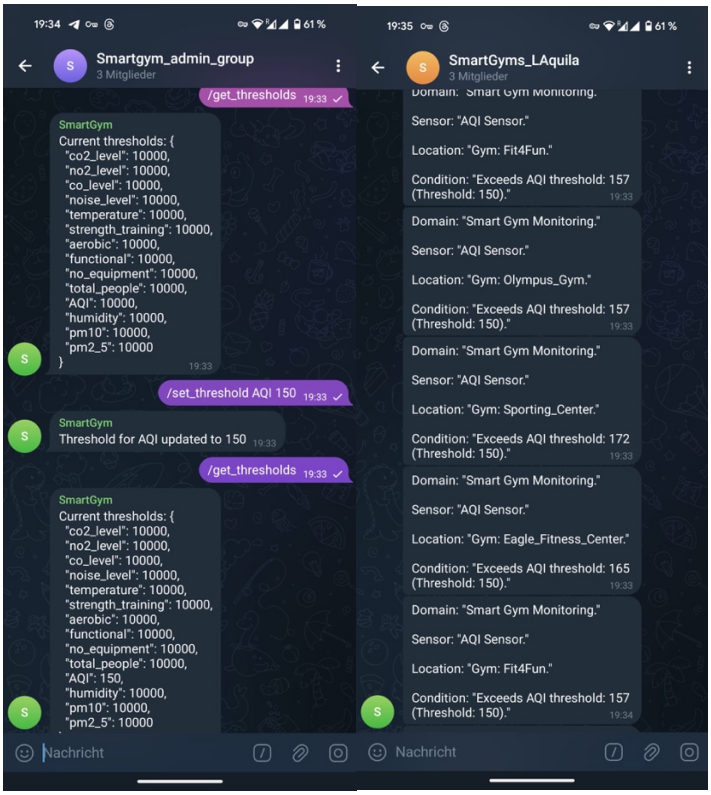
ALERTING MECHANISMS AND DEPLOYMENT

Alerting Mechanisms

- The alerting system for the **Smart Gym Monitoring System** ensures proactive responses to abnormal conditions.
    - Alerts are triggered when a sensor value exceeds the predefined thresholds.
        - AQI exceeding the threshold of 150.
        - Temperature or occupancy nearing critical levels.
- **Telegram Admin Group** allows gym administrators to retrieve and adjust thresholds using commands like /get_thresholds and /set_threshold <sensor_type> <value>.
    - Example: /set_threshold AQI 150 updates the AQI threshold to 150, as seen in the screenshots.
- **Telegram Alert Group** sends real-time alerts for threshold violations. Alerts include sensor type, gym location and threshold details.
    - Example Alert:

      Domain: "Smart Gym Monitoring."
      Sensor: "AQI Sensor."
      Location: "Gym: Fit4Fun."
      Condition: "Exceeds AQI threshold: 157 (Threshold: 150)."

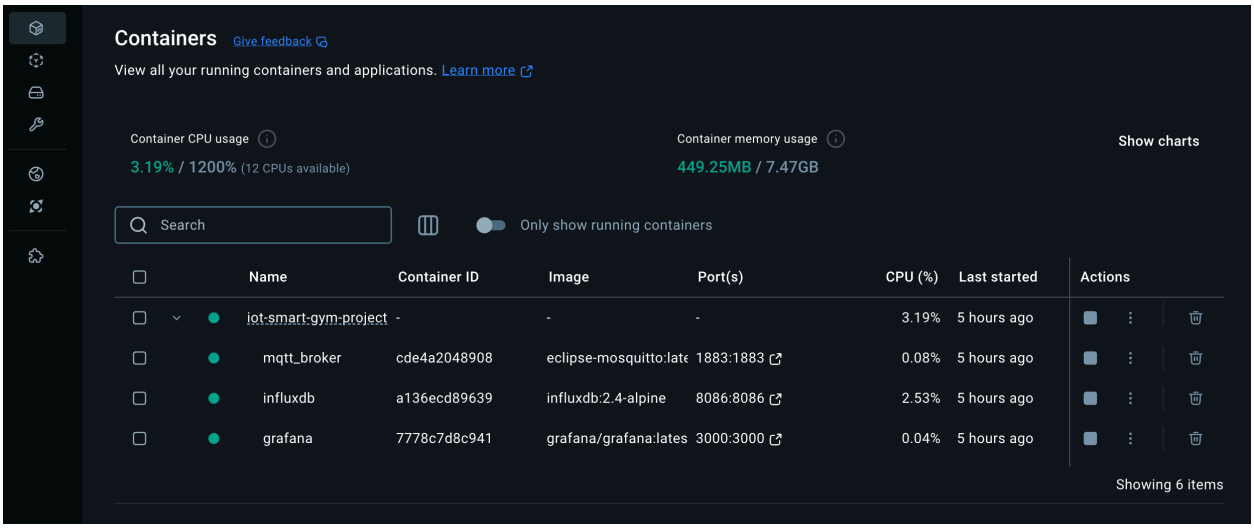
- The system handles errors, such as failed Telegram messages, by retrying with a delay.



Deployment

- The system is containerized using **Docker**, providing scalability, portability, and ease of management.

- Deployment ensures seamless integration between these services.
    - **MQTT Broker**: Handles communication between simulated sensors and the system.
        - Image: eclipse-mosquitto:latest.
        - Port: 1883 for communication over the MQTT protocol.
    - **InfluxDB**: Stores time-series data.
        - Image: influxdb:2.4-alpine.
        - Port: 8086 for database access.
    - **Grafana**: Provides the interface for visualizing and analyzing sensor data.
        - Image: grafana/latest.
        - Port: 3000 for dashboard access
    - Sensor_simulation: Simulates sensor data and publishes it to MQTT topics.
        - Image: custom-built for the project
    - Node-RED: handles data ingestion, preprocessing, threshold evaluations and Telegram alerts.
        - Image: Custom-built for Node-RED workflow
        - Port: 1880 for the access to the Node-RED interface.
- **Scalability**: Docker enables horizontal scaling of services, allowing the system to accommodate additional sensors or gym locations as needed.
- **Resource Utilization**: Efficient resource allocation ensures minimal CPU and memory usage, as shown in the screenshots, even under active monitoring and alerting.



CONCLUSION

The Smart Gym Monitoring System is a testament to how technology can bridge the gap between user experience and operational efficiency in gyms. By tackling everyday challenges like overcrowding and uncomfortable conditions, it ensures a safer, more enjoyable, and smarter environment for everyone involved.

Through seamless integration of IoT technologies, real-time dashboards, and instant alerting mechanisms, the system not only improves how gyms operate but also empowers staff to respond quickly to critical situations. Using tools like MQTT, Node-RED, and Grafana, alongside containerized deployment via Docker, the solution balances innovation with practicality, keeping scalability and reliability at its core.

This project reflects the potential of technology to solve real-world problems with simplicity and clarity. It's more than just a monitoring tool—it's a vision for creating spaces that enhance well-being. By taking this step, we've laid a foundation for future innovations, like integrating wearables or expanding data analytics, that could revolutionize fitness environments further.

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

REFERENCES

- U.S. Environmental Protection Agency. (2024). *Technical Assistance Document for the Reporting of Daily Air Quality – The Air Quality Index (AQI)* (EPA-454/B-24-002). Office of Air Quality Planning and Standards, Air Quality Assessment Division. Retrieved from https://www.epa.gov
- Bian, S., Rupp, A., & Magno, M. (2023). *Fully automatic gym exercises recording: An IoT solution*. arXiv preprint arXiv:2305.17594. https://arxiv.org/abs/2305.17594