UNIVERSITÀ DEGLI STUDI DELL'AQUILA

DISIM
Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

IoT for art: management and monitoring of an art exhibition

Submitted to Professor **Davide Di Ruscio**
Software Engineering for Internet of Things

Version	1.0

Date	14/02/23
Team Name	MAGI Team

Team Members		
Name and Surname	**M. Number**	**Email**
Matteo Leoncini	288518	matteo.leoncini@student.univaq.it
Giorgio Morico	280856	giorgio.morico@student.univaq.it

Index

Introduction

The field of art exhibitions is constantly evolving and advancing, and with it, the need for innovative and efficient systems to manage and monitor these events. This software system was designed to address the specific challenges faced by curators, exhibitors, and collectors when it comes to organizing, showcasing, and preserving artwork. The goal of this system is to provide a comprehensive and reliable solution that can streamline the management and monitoring of an art show from start to finish.

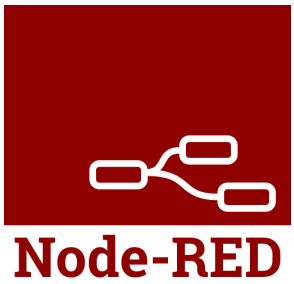
At the heart of this system is a sophisticated network of sensors, placed near each artwork and around the exhibition space. These sensors collect real-time data about the environment surrounding each piece, including visitor presence, average time spent viewing the artwork, temperature, and humidity. This data is transmitted wirelessly to the central software, where it can be analyzed and used to ensure the safe and secure preservation of the artwork, as well as to gather insights about visitor engagement with the exhibition.

The purpose of this software system is to provide an integrated and comprehensive solution for art show management and monitoring, allowing organizers to focus on the creative aspects of the exhibition, while relying on the software to handle the technical details. Whether you are organizing a small local show or a large international exhibition, this system is designed to meet the unique needs of your event and help you achieve your goals.

Used technologies

In this section are shown the technologies used to realize this system.
In this system we have the following sensors:

- **Proximity sensor**: placed near the artworks to check the presence of visitors.
- **Light sensor**: placed near the artworks to control the amount of light to which the artwork is exposed.
- **Humidity sensor**: placed near the artworks to control the amount of humidity to which the artwork is exposed.
- **Temperature sensor**: placed in the rooms to measure the temperature
- **Air quality sensor:** placed in the rooms to measure the air quality
- **Movement sensor**: placed in the enter and the exit of the building to keep the count of the people inside



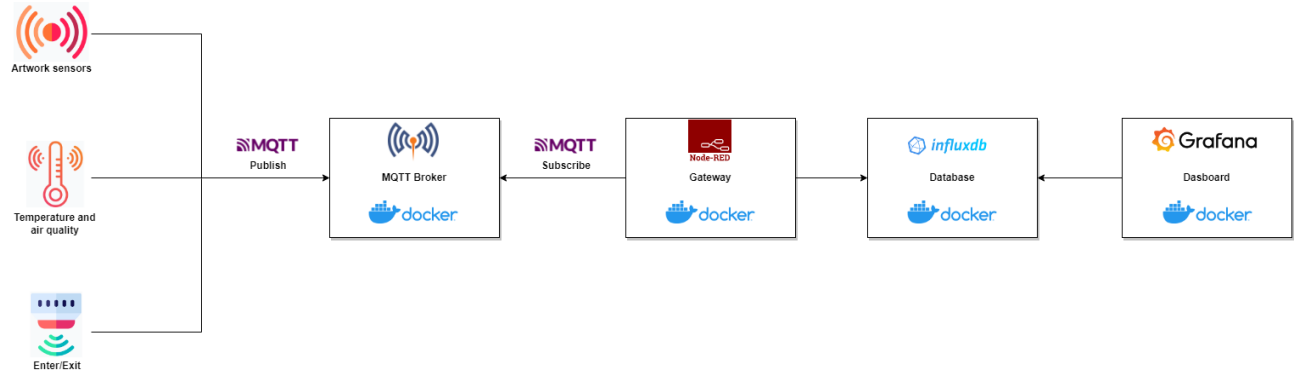
Node-RED provides a drag-and-drop interface for creating and deploying custom integrations, which allows users to connect sensors, data sources, and other components of the system quickly and easily. In this system, Node-RED is the glue that holds the art show management and monitoring system together, providing a powerful and flexible platform for integrating data from all the different sensors and data sources involved.

InfluxDB is a time series database that is used to store the real-time data collected by the sensors in the art show management and monitoring system. In this system, InfluxDB is a time series database that is used to store the real-time data collected by the sensors in the art show management and monitoring system.

MQTT is a lightweight publish/subscribe messaging protocol that is used in this system to transmit real-time data from the sensors to the central software. In this system, MQTT is a crucial component of the art show management and monitoring system, providing fast and reliable communication between the sensors and the central software, and enabling real-time monitoring and management of the exhibition.

Grafana is a powerful data visualization tool that is used in this system to provide real-time insights into the data collected by the sensors. Grafana allows users to create custom dashboards that display data from a variety of sources, including InfluxDB, the time-series database used in this system to store real-time data from the sensors. In this system, Grafana is a critical component of the art show management and monitoring system, providing real-time data visualization and advanced analytics capabilities to help curators, exhibitors, and collectors make informed decisions about the exhibition.

Architecture of the system



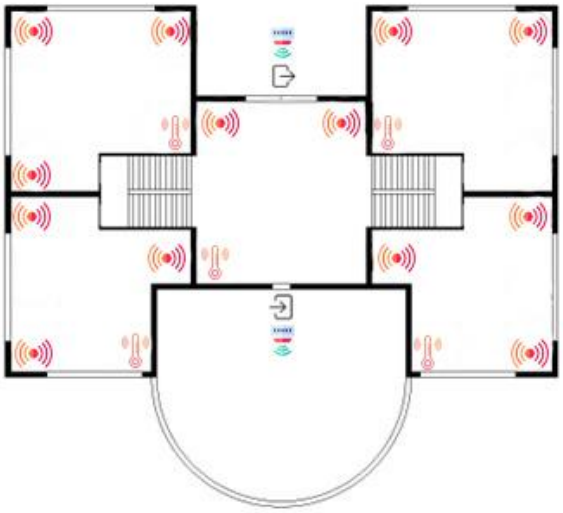
In this system all the components are stored and managed by Docker. Docker provides a consistent environment for each component allowing it to run and test each component individually, without the risk of affecting other parts of the system. This greatly improves the reliability and stability of the system as a whole. Another reason why I have chosen Docker is its portability: Docker containers can be easily moved from one environment to another, making it easier to deploy the system to different environments such as development, testing, and production. This also helps in reducing the time and effort required for deployment, as the same container can be used across multiple environments.

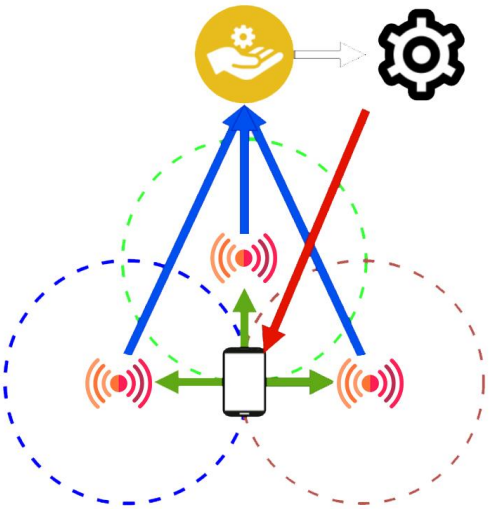
The MQTT Broker container allows us to have a broker through which to exchange messages between sensors and gateways. Our broker includes the following topics:

- artworks/id/visitors
- artworks/id/leave
- artworks/id/humidity
- artworks/id/light
- rooms/id/temperature
- rooms/id/air
- building/visitors

The Gateway component represents the Node-RED container, in particular its ability to make different technologies communicate easily. This container subscribes to the MQTT broker to read the published data from the sensors and send it to the InfluxDB database container.

The Database component represents the InfluxDB container, which contains the significant data collected by the sensors. Thanks to the connection with the Dashboard component, which represents the Grafana container, it is possible to display this data in the form of graphs on special dashboards to carry out accurate analyses.



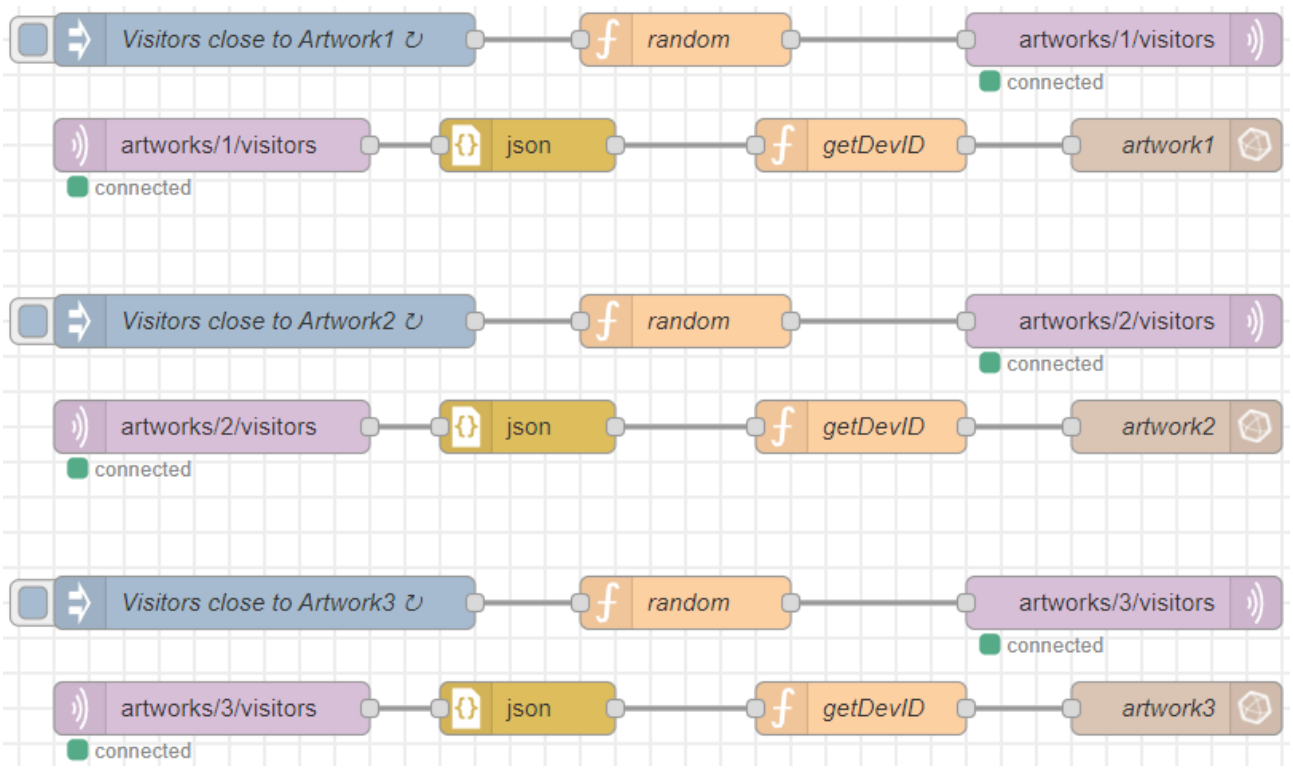


Each artwork has different sensors:
- one for detecting proximity
- one for gathering light intensity
- one for gathering humidity informations

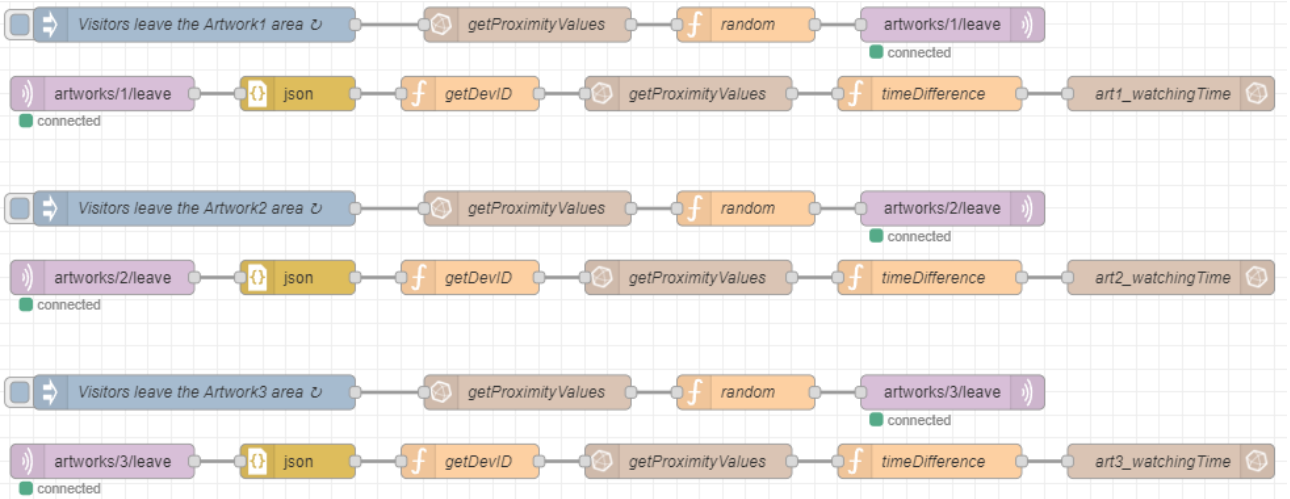
Each area has a thermometer to check the room's temperature and an air quality sensor.
What we show here is how IoT components and devices interact with the user devices.
The intercommunication is done through telemetry sensors for the acquisition of the needed data and a publish subscribe broker makes it possible to handle and use such data.
As we can see in the above diagram, a user device communicates with the iot sensors (bluetooth transreceivers) by sending a broadcasted message which alerts the sensors of the presence of the device.
At the receiving of said message the telemetry side of the system itself publishes on the broker a message containing an object that specifies the DevID.
The data then gets elaborated by the nodeRed flow below (*watching time figure*) and the aggregated result is given back.
For our purposes the protocol used is MQTT.
Due to the low power consumption IoT devices can be powered with batteries or low power sources like *beacon bluetooth low energy*. These are hardware devices that transmit their id to allow other devices (e.g. smartphones, tablets) to make some actions when they get close to them. In this way no electrical manipulation to the electrical network of the heritage site has to happen.

Node-Red implementation

Watching time for each artwork

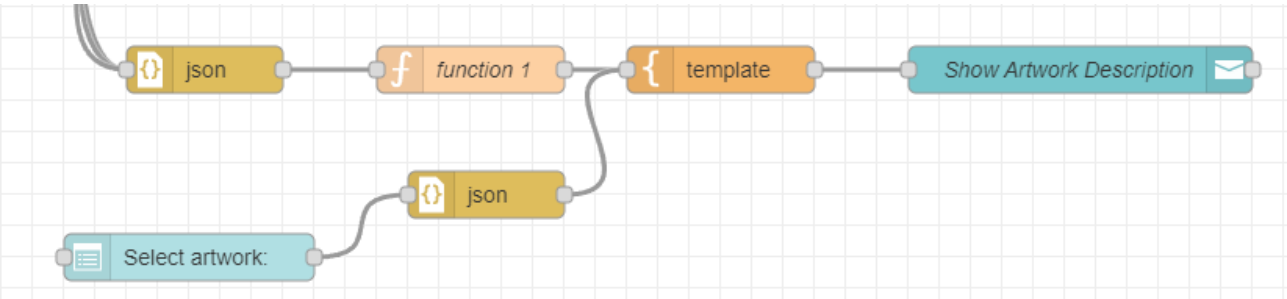


In this flow, the data is simulated using a random function and published on the topic "artworks/*/visitors" which identifies the visitors who have entered the reception range of the sensor. The information is then put in a json file that will be parsed to get the DevID that identifies the visitor. The data is then saved on InfluxDB, divided by artwork.

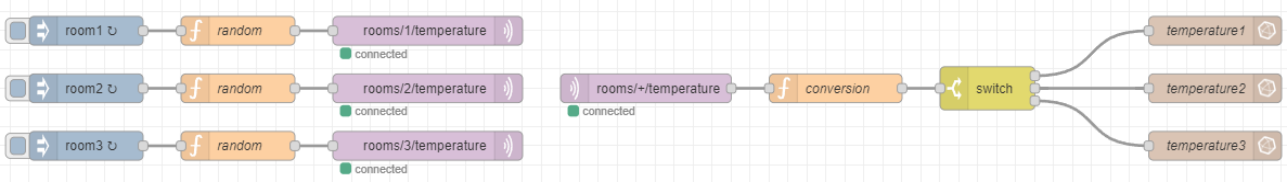


Subsequently, the situation in which a visitor leaves the sensor's reception area is represented. We get from the DB the time in which the user entered in the proximity sensor area, publish it on the topic "artworks/*/leave" and, using the "timeDifference" function, the watching time of that visitor for that specific work is calculated by getting the current time and subtracting to it the time of first contact. In this way, it gets the watchtime which is written in the DB as an analytics related to the user.

As soon as a user gets close to an artwork, a popup shows up with a brief description of the artwork and a link for more details (*see frontend*).

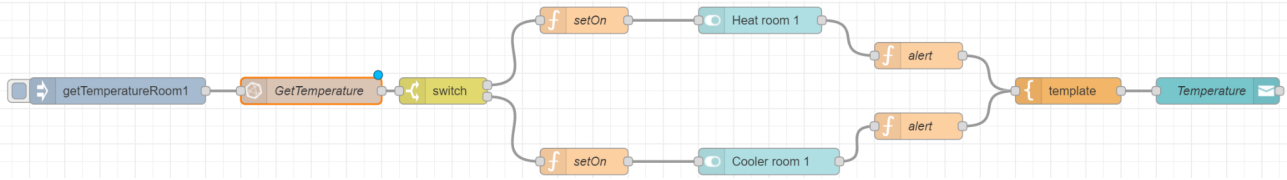


Rooms' temperatures



This flow shows the measurement of the temperature in the different rooms. The sensed data are simulated through a random function that publishes on the topic "rooms/*/temperature". In the end, the data is saved on InfluxDB in the relative location.

Monitor temperature

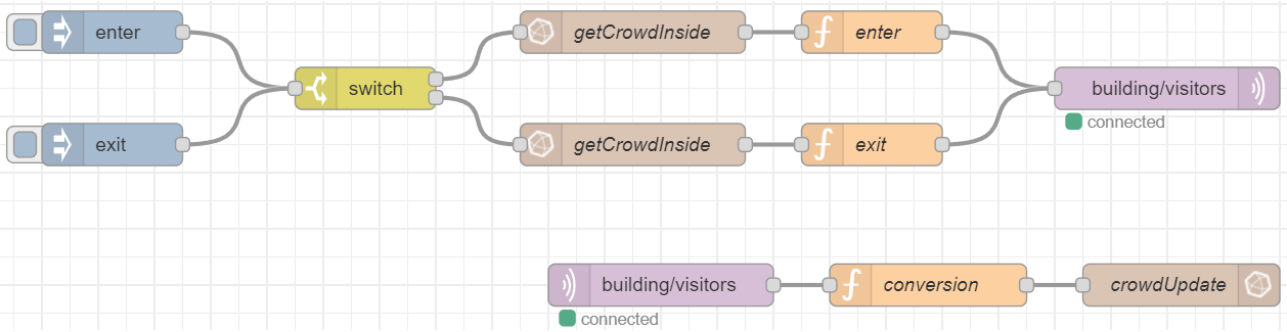


The thermometer spread in the location gets the temperature every minute, these data are then stored on InfluxDB. The flow shown in the figure gets the current temperature from the DB and:

- if it is <= 18°C the heat will be turned on
- if it is >= 25°C the cooling will be turned on

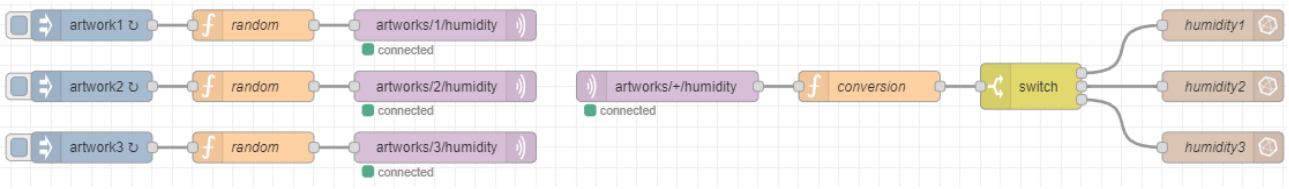
In this way we can automatically maintain the temperature inside between 18 - 25 °C.

People inside the building

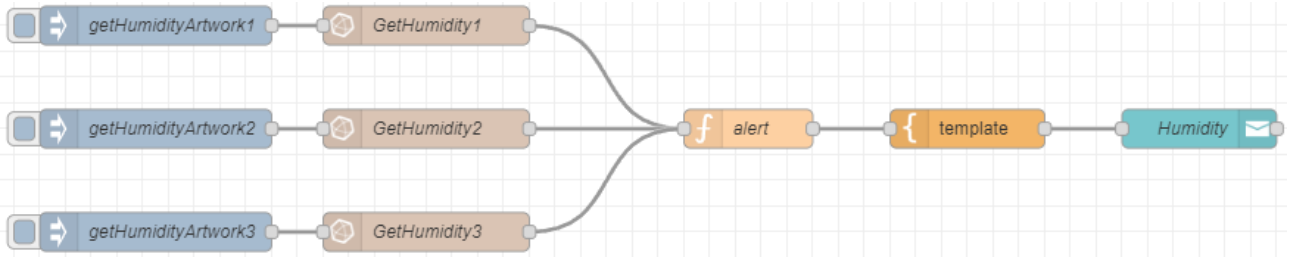


As soon as someone enters or exits, the current number of people inside will be obtained from the DB and updated with the enter or exit function that will just increase or decrease the number by one.

Humidity

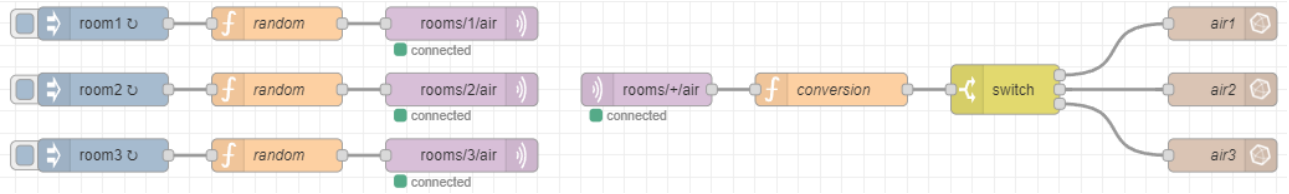


This flow shows the measurements of humidity on the different artworks. The sensed data are simulated through a random function that publishes on the topic "artworks/*/humidity". In the end, the data is saved on InfluxDB in the relative location.

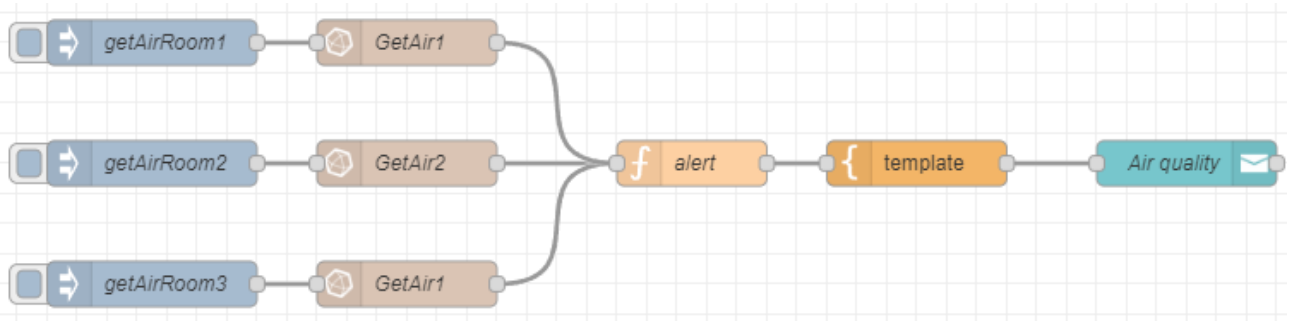


This flow gets the humidity data from the database and, if the current humidity in the area of a specific artwork is out of the expected range, an alert is shown on the admin view as a warning. (*see frontend*)

Air quality

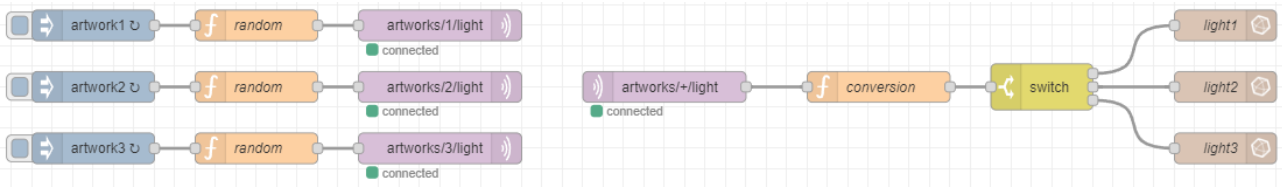


This flow shows the measurements of air quality in the different rooms. The sensed data are simulated through a random function that publishes on the topic "rooms/*/air". In the end, the data is saved on InfluxDB in the relative location.

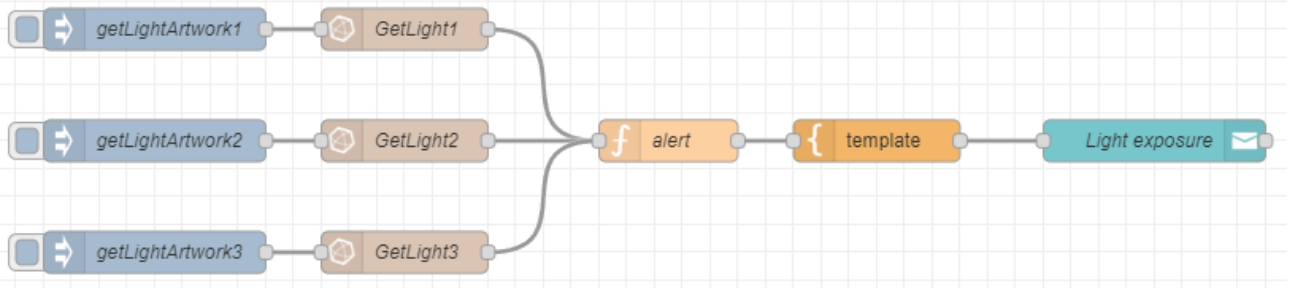


This flow gets the air quality data from the database and, if the current AQI in some room is not good as expected, an alert is shown on the admin view as a warning. (*see frontend*)

Light



This flow shows the measurements of light intensity on the different artworks. The sensed data are simulated through a random function that publishes on the topic "artworks/*/light". In the end, the data is saved on InfluxDB in the relative location.

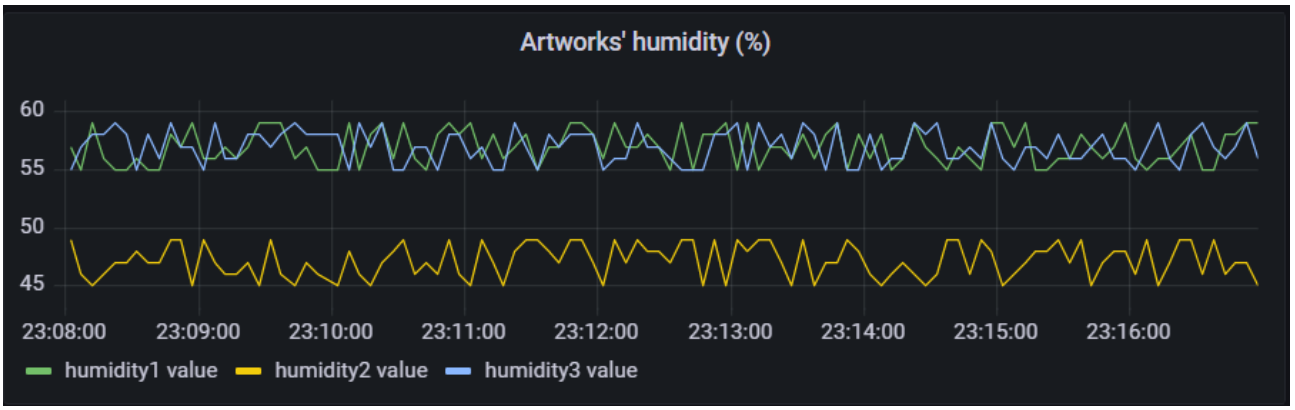


This flow gets the light intensity data from the database and, if the current data in some specific artwork is too high, an alert is shown on the admin view as a warning. (*see frontend*)

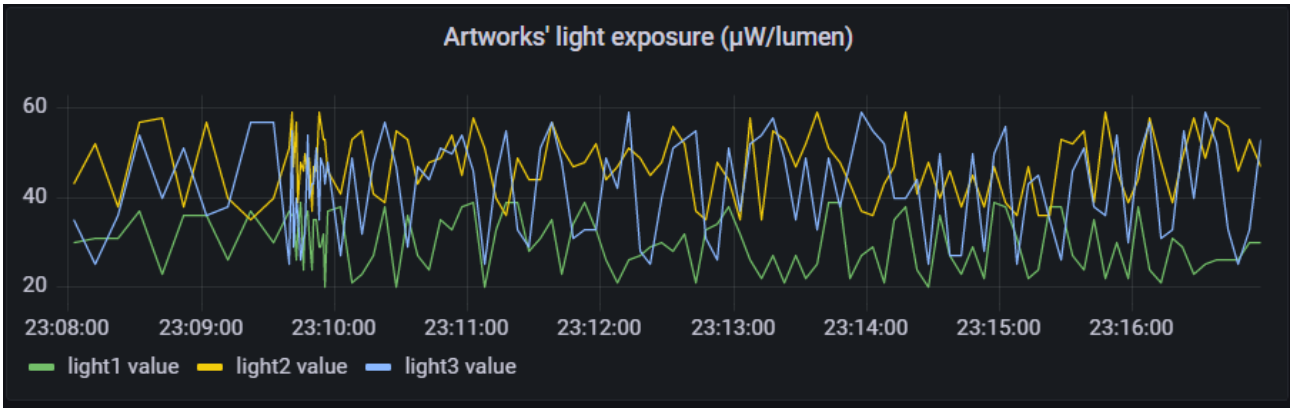
Dashboards for data visualization

Through Grafana, we have created the following Art Exhibition Control Panel composed by different dashboards, to show the data collected by the sensors:

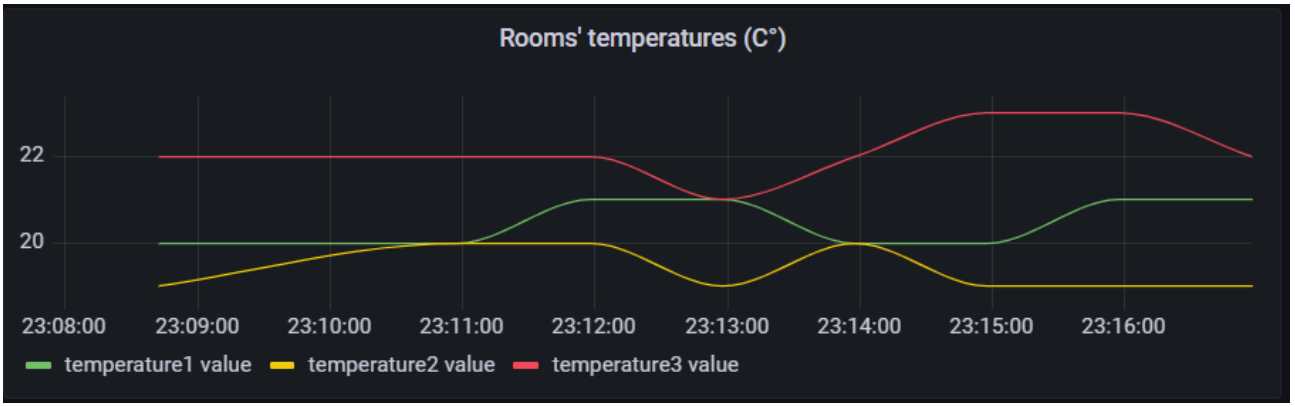
In detail:



With this dashboard it is possible to consult the trend of humidity data on the individual artworks: this information is important to ensure the integrity of the artwork.

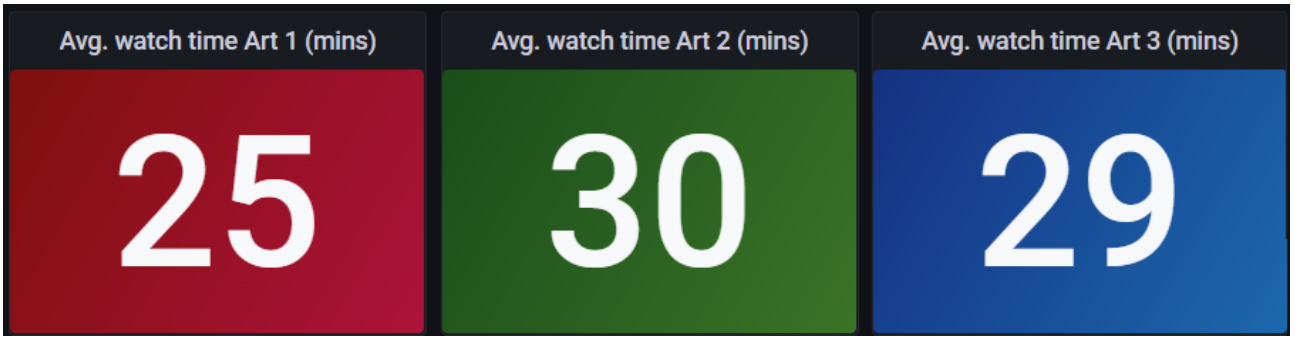


With this dashboard it is possible to consult the trend of light exposure data on the individual artworks: as the previous one, also this information is important to ensure the integrity of the artwork.

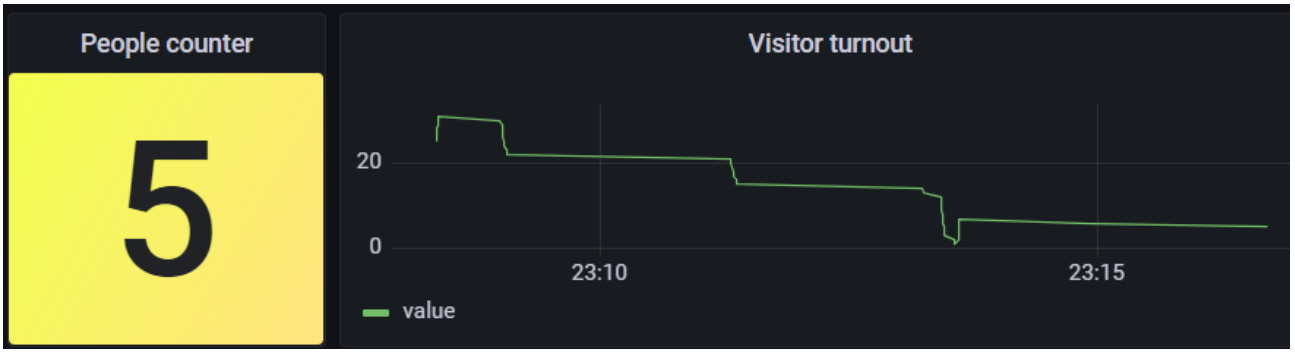


With this dashboard it is possible to consult the trend of the temperature in the different rooms.

With this dashboard it is possible to consult the trend of the Air Quality Index (AQI) in the different rooms.



With these dashboards it is possible to consult the average time spent by the visitors in front of the different artworks.



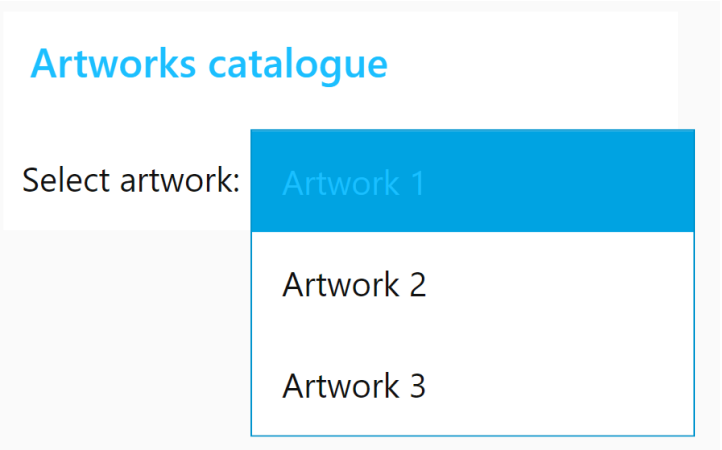
With this dashboard it is possible to consult how many people are in the building through the people counter and also see the trend over time.

Frontend

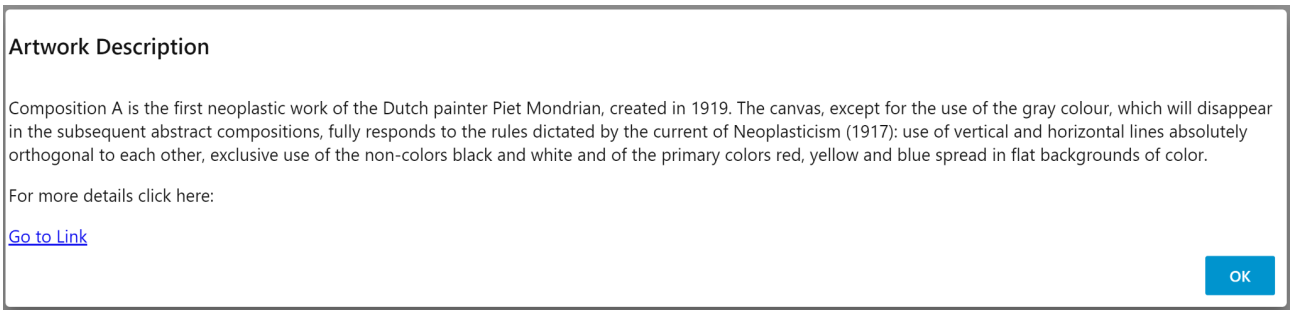
We have implemented an example of an application to handle the data shown above. It provides two different views, one for the visitor (to improve his experience visiting this art exhibition), one for the admin to help him manage the data gathered from the IoT system.

User View

On the user view we show a list of the artworks present in the building:



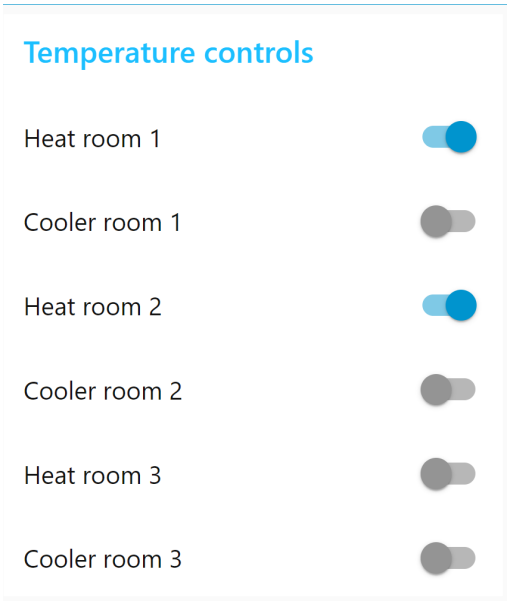
As soon as the user selects an artwork, this popup shows up:



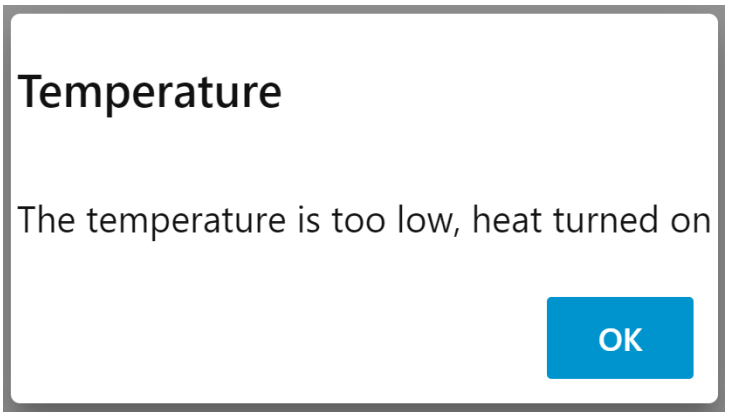
where the user can see a brief description of the artwork and a link for more details. This popup also automatically shows up when the user gets close enough to a specific artwork in the building.

Admin View

The admin can control the temperature system (turning on and off heating and cooling devices) using the following view:

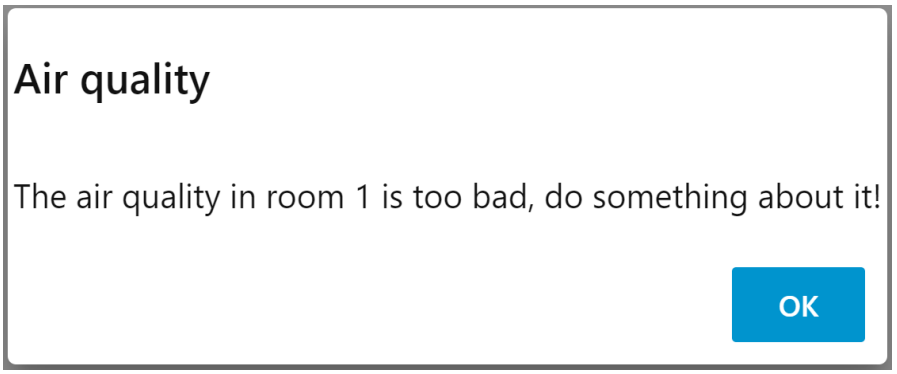


When a switch is turned a popup shows up:



This popup also automatically shows up when the cooling or heating system is turned on/off automatically by the IoT system (when the temperature is out of the optimal range).

We have also implemented other notification popups to alert the admin about some danger or possible hazards in one or more rooms: humidity out of range, light intensity too high and bad air quality. This is an example:



Conclusion

The ability to collect data from a variety of sensors and transmit it in real-time to a central location has revolutionized the way many industries operate. By providing real-time insights into a wide range of factors, IoT technologies have enabled organizations to make more informed decisions, improve efficiency and provide better experiences for their customers.

The utilization of IoT technologies has become increasingly widespread in recent years, with applications in a wide range of fields: this art exhibition management and monitoring system is just one example of the many innovative uses of IoT technologies to collect, process, and analyze real-time data.

With its use of cutting-edge technologies like Node-RED, InfluxDB, MQTT and Grafana, this system provides a complete and integrated solution for monitoring and managing all aspects of an art exhibition.

How to run the application

- Run the “docker-compose up” command inside the main folder
- InfluxDB credentials:
    - user: matteo, password: matteopass
- Grafana credentials:
    - user: admin, password: adminpass