FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

IoT Project
3^{rd} Assignment
CATS - Creative Art Tracking & Surveillance

Matilde Carvalho 2019233490 and Rafael Luís 2018289409

Internet of Things
Coimbra University
Faculty of Science and Technology

1 Introduction

This report presents the final IoT assignment of the CATS IoT Project that aims to protect and monitor works of art in museums. To achieve this objective, specific technology, software, sensors, and hardware were utilized. Throughout this report, we will provide details about the technology employed, the components used, and the communication protocols adopted. Furthermore, we will describe the system's functionality, scalability, and the usefulness of the collected information for the user, as well as the potential of the presented solution.

2 Technologies, Software, Protocols, Sensors, and Hardware

2.1 Integration Diagram

According to figure 1, there were used different technologies, software, protocols, sensors, and hardware used in the project.

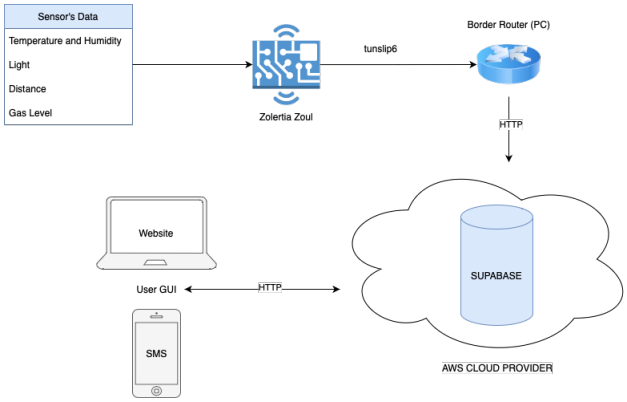


Fig. 1: Integration Diagram

2.2 Materials

The board along with the sensors in table 1 were used to monitor and protect art pieces in museums:

Material	**Specification**
Control Board	Zolertia zoul
Temperature and Humidity Sensor	DHT22
Ultrassonic Ranger Sensor	Grove Ultrassonic Ranger v2.0
Light Sensor	Grove - Light Sensor v1.2
Gas Sensor	MQ2

Table 1: List of materials

Contrary to the plan established in the 2^{nd} assignment, the vibration sensor is not used because it was not available in the lab while this project was developed. The specifications of some sensors also changed due to the material available.

Note: It was not possible to use the Ultrassonic Range Sensor due to library incompatibility with the *Zolertia* board. The data from this sensor was simulated.

The readings from the Temperature and Humidity sensor, Light Sensor and Gas Leak sensor are obtained using the ADC1 and ADC3 ports on the *Zolertia* Zoul board.

Due to the constraints of having a reduced number of ports on the *Zolertia* board and having only one board because of the low availability of boards in the lab, the code needed to be tested separately.

Using two *Zolertia* boards, it would be possible to demonstrate all the sensors simultaneously. Nevertheless, functional code was submitted along with the rest

of the project. So to observe the above described it would only be necessary to flash that same code on another board. Each set of boards would have an individual ID associated with each of the artworks to update the values to the database.

2.3 Technologies

To read the values from the sensors it was used a MQTT publisher made with C code that is running on the *Zolertia* board.

Since there is no Gateway to connect the *Zolertia* module to the Internet, it was used tunslip6 as a border router. Tunslip6 creates a virtual network interface (tun0) on the host side and uses SLIP (serial line internet protocol) to encapsulate and pass IP traffic to and from the other side of the serial line, i.e., the USB port connected to the *Zolertia* board.

To send the data to the cloud it was used a MQTT listener made with python code to retrieve the data and then to send it via HTTP to the Database. The MQTT listener calls other two codes in C: one to patch the data to the database and other to get the level of risk of the data and sends a SMS to the user when the artwork is in danger.

Instead of Azure as was planned in the previous assignment there was used an online SQL database: *Supabase* provided by Amazon's AWS Cloud Services.

The Database saves the sensors' data in a table and then processes it to determine if the artwork is in danger and makes a risk analysis. If a high level of risk is detected, the sensor's data is stored in another table so that a detailed history of the danger can be seen later.

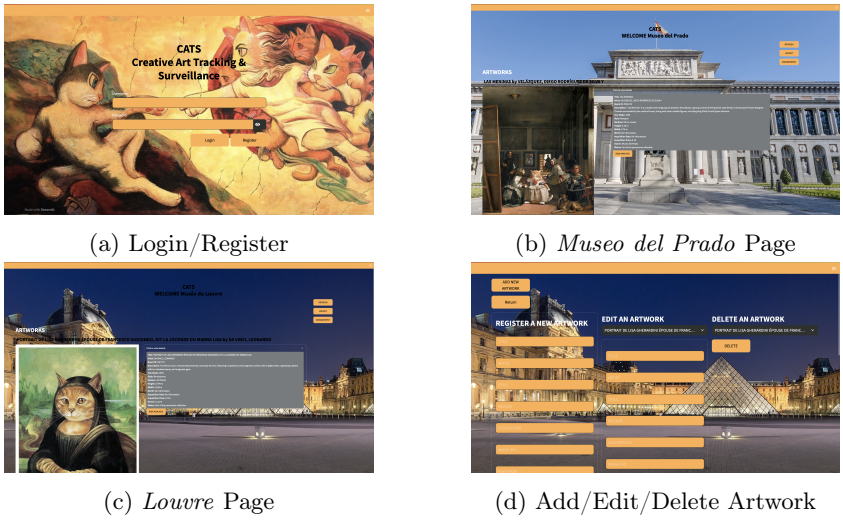
The Database also has data from museums such as name, artworks, owner and phone number.

For the app development it was used a framework named *streamlit* that allows an easy, fast and intuitive way to create a webapp. To exchange information between the Database to the webapp it was used a python program and *Github sites* as the host for the website.

To use the User GUI, there was created an website that allows the museum curator to login to the museum page, register the artworks and check the risk analysis for each artwork as well the history.

3 User Application

The Information Presented on the User Application can be seen on figure 2 and the web app can be accessed via the link: CATS-iot23.The Web app starts with a login page where the museum administrator can login or create a new account (Figure 2a). After logging in, the Web App shows the museum page and the artworks that have a *Zolertia* board, for example, figure 2b for *Museo del Prado* and figure 2c for *Louvre*. The user can also add, edit or remove an artwork (Figure 2d).



(a) Login/Register (b) *Museo del Prado* Page

(c) *Louvre* Page (d) Add/Edit/Delete Artwork

Fig. 2: Web App

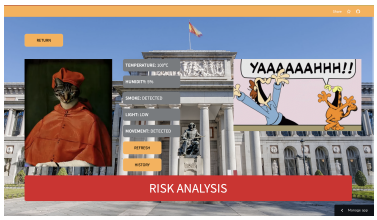
The user can see the data from the sensors connected to the *Zolertia* board as well a risk analysis to protect and monitor the pieces of art as can be seen in figure 3. There were defined 3 levels of risk:

- Green: Low risk. All values from the sensors are normal and there is no risk to the art piece. (Figure 3a).
- Yellow: Medium Risk. Some values from the sensors are out of normal but still in the appropriate range. In this case the system stays alert in case of 3 "Yellows", at the 4^{th} the risk analysis turns to Red and sends a SMS to the user. (Figure 3b)
- Red: High Risk. Some values from the sensors are completely off normal range. It means the artwork is in danger and the system immediately sends a SMS to the user. (Figure 3c and Figure)

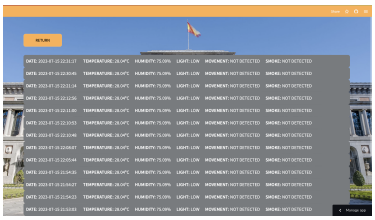
The user can see a history of all Red alerts like is shown in figure 3d.

(a) Green

(b) Yellow



(c) Red



(d) History of Red

Fig. 3: Risk Analysis

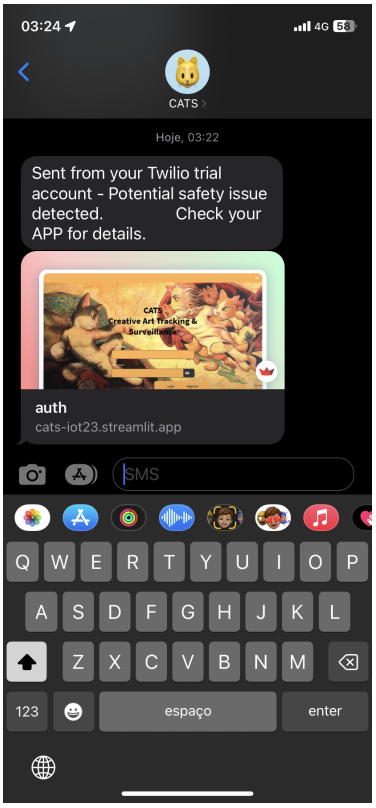


Fig. 4: SMS received

4 Potential of the Proposed Solution

This project has the potential to monitor and protect pieces of art in museums or private collections. It provides enhanced security, remote and continuous monitoring as it allows museum staff to remotely monitor the condition of art pieces and ensures that the conditions the artwork is exposed to are within optimal ranges to preserve the artwork and take immediate action in case of deviations. It also provides real-time alerts and notifications in case of risk.

Some limitations may arise such as reliability because the system relies on network connectivity and uninterrupted power supply. Any network or power disruptions can affect the functionality of the system, potentially leading to lapses in monitoring and protection.

5 Conclusion

Summary of the main findings

Recommendations for future work

This IoT project presented in this report aims to protect and monitor works of art in museums. By utilizing various technologies, software, protocols, sensors, and hardware, the project provides enhanced security and continuous monitoring for art pieces. The system utilizes MQTT and HTTP for data transmission, *Supabase* as the online SQL database, and a web application developed using the *Streamlit* framework.

Some limitations were found for the sensors since the *Zolertia* board is outdated as well as its supported libraries with very few online support. The Azure is not intuitive as it offers a lot of difficulties in order to do simple things such as creating a Database.

In conclusion this project was a challenging experience that contributed for our learning about Internet of Things.