sustainability

MDPI

Article

CityAirQ—Pollution Tracking System

Mihnea Dinica [1], Denisa Popescu [1], Dan Tudose [1], Bianca Dumitru [1], Laura Ruse [1,*], Abhinuv Pitale [2] and Mihai Preda [1]

[1] Faculty of Automatic Control and Computers, National University of Science and Technology POLITEHNICA Bucharest, 060042 Bucharest, Romania; mihnea.dinica@stud.acs.upb.ro (M.D.); ioana.popescu1905@stud.acs.upb.ro (D.P.); dan.tudose@upb.ro (D.T.); bianca.dumitru0907@stud.acs.upb.ro (B.D.); mihai.preda1808@stud.energ.upb.ro (M.P.)

[2] Google, San Diego, CA 92121, USA; abhinuvpitale@gmail.com

* Correspondence: laura.ruse@upb.ro

Abstract: Air pollution represents a significant threat to human health and the environment, especially in densely populated metropolitan areas. Determining air pollution levels in urban areas is crucial for raising public awareness about air quality and potential health risks, empowering citizens to make informed decisions about their well-being, potentially leading to improved air quality and healthier communities in the long run. The project proposes CityAirQ, a reliable pollution tracking system, that uses air pollution parameters and environmental data to generate dynamic maps for metropolitan regions. CityAirQ includes the following components: energy-efficient and portable pollution tracking devices equipped with pollution and environmental sensors, a mobile application that displays real-time collected data, together with dynamic environmental maps and, lastly, a cloud-based data pipeline that ingests, processes and stores sensor data. Our system integrates an ultra-compact custom PCB that enables real-time tracking of a broader range of pollutants than any other mobile solution of comparable size, making it a uniquely efficient tool for urban air quality assessment. The system's performance was assessed in the final phase through testing and data collection in order to validate functionality and reliability. CityAirQ promotes environmental sustainability by providing the tools and information needed to understand, monitor, and mitigate air pollution in urban areas, ultimately contributing to a healthier and more sustainable future.

Keywords: Internet of Things; pollution; air quality; sensors; pollution monitoring; environmental sustainability

check for updates

Academic Editor: Giouli Mihalakakou

Received: 21 February 2025
Revised: 13 April 2025
Accepted: 24 April 2025
Published: 30 April 2025

Citation: Dinica, M.; Popescu, D.; Tudose, D.; Dumitru, B.; Ruse, L.; Pitale, A.; Preda, M. CityAirQ—Pollution Tracking System. *Sustainability* **2025**, *17*, 4062. https://doi.org/10.3390/su17094062

1. Introduction

1.1. Air Pollution

Air pollution poses a significant risk to human health, especially when living in polluted urban areas on the long term. Naturally, the fact that pollution is invisible, therefore hardly noticeable unless sensed [1], makes it even more dangerous, with historical root causes tracing back to major industrial events such as "The Great London Smog" in 1952 [2] or the The Industrial Revolution.

In recent years, emissions of smoke and pollutants have substantially increased, with various sources of pollution that are presented in Section 9.1. Exposure to pollution has a dangerous medical impact [3–5], including not only respiratory and cardiovascular diseases, central nervous system dysfunctions, and cancer [6], but also fertility and pregnancy issues [7], effects reported by biomedical research. Among demographic groups most likely to suffer, children are vulnerable by simply playing outside or living near busy, commuting zones [8]. Older adults are also heavily affected and different studies have shown a strong correlation between pollution and affections such as dementia or Alzheimer's disease [9].

On the other hand, pollution also has toxic effects on the environment. Emissions of pollutants have a solid impact on the ecosystem, which can be observed under scientific phenomena such as extreme weather, global warming, wildfire, acid rain, extinction and resources degeneracy [6].

1.2. Benefits of Air Quality Monitoring

Air quality monitoring is potentially the most important short-term solution for reducing health risks and preserving ecosystems in the future, especially in heavily populated cities. Geographically, the primary area of focus is the metropolitan area of Bucharest, Romania. Just like many major capitals, it suffers from severe traffic congestion, high vehicle emissions and a lack of recreational green areas [10]. In Bucharest, the network of air pollution monitoring stations is sparse and the public access to real-time data collection is not widespread. With not so many recent studies on this issue within the past five years, the available data is primarily sourced from online services, which will later be presented in Section 2.

By anticipating and visualizing this unseen phenomenon [11], metropolitan residents can stay in touch with the levels of air pollution in their area at all times and be alerted when air quality indicator values become remarkably high. This can inevitably lead to empowering decisions that can potentially improve the well-being of everyone around, resulting in raises in public awareness and healthier, stronger neighborhoods. A clear understanding of air quality patterns can also lead to more sustainable urban development and planning, on a higher level [12].

1.3. Proposed Solution

In this paper, we propose CityAirQ, an user-friendly, low-power Internet of Things (IoT) system for tracking urban pollution. The system includes pollution detection devices attached to bicycles and uses real-time and historical environmental data to generate dynamic pollution maps on a mobile application. Users can use CityAirQ to switch cycling routes to less polluted areas on the map, which is obviously beneficial for reducing exposure to contaminants.

The proposed IoT system includes three types of components:

- Hardware device: A portable and energy efficient hardware device from scratch. It is small in size, attachable to bicycles and equipped with multiple sensors for air quality measurements. The device plays an essential role, as the only real time raw data source, which builds the foundation for further functionalities of the system.
- Data Pipeline: A Cloud-based pipeline for data processing. The real-time data from the previously mentioned device is further collected by the mobile application, then forwarded through an Application Programming Interface (API) to the Cloud-based system. The pipeline is responsible for data aggregation, analysis, storage, and dynamic map generation.
- Mobile Application: An user-friendly, energy-efficient mobile application, focusing on data visualization. It renders real-time sensor data and historical charts, but also displays dynamic maps and alerts. It also provides secure and fast communication with both endpoint peers: the hardware device and the pipeline API.

The performance of the system was evaluated during the final phase of the project, with all components seamlessly integrated. Testing and data collection was conducted to validate system functionality, verify data reliability and gather user feedback.

1.4. Key Research Gaps

Traditional air quality monitoring systems are often limited by sparse spatial coverage, high costs, bulky designs, and poor user accessibility. CityAirQ addresses these gaps with a compact, cost-effective, energy-efficient device that integrates real-time data visualization and dynamic mapping via a mobile app. By leveraging bicycle-based data collection, it enhances spatial coverage, reduces costs, and provides detailed air quality insights, making it a transformative tool for urban environmental monitoring. Our system addresses several key research gaps:

- Spatial Coverage: Fixed air quality monitoring stations are sparse and cannot capture localized pollution hotspots. By attaching devices to bicycles, data can be collected across a wide range of locations, including narrow streets and residential areas. This fills the gap in spatial coverage and provides a more detailed understanding of air quality variations within a large city.
- Lack of Compact and Wearable Air Quality Monitoring Devices: Existing solutions like are either too large or lack the compactness required for easy wearability and portability. CityAirQ achieves an ultra-compact design through a custom PCB and optimized sensor placement, making it one of the smallest devices with comparable functionality.
- High Power Consumption in Existing Devices: Many commercial devices prioritize real-time sensing but suffer from high power consumption, leading to shorter battery life. CityAirQ implements selective sensor activation and duty cycling, reducing average power consumption by 30% and enabling extended battery life without compromising data reliability.
- High Cost of Existing Solutions: Similar commercial devices are expensive, limiting their accessibility to a broader audience. CityAirQ achieves a significantly lower cost ($\tilde{\$}$86) by carefully selecting and integrating cost-effective sensors without compromising functionality, making it more accessible.
- Limited Mobile Integration and Dynamic Mapping: Some existing solutions lack dedicated mobile applications, while others rely on web-based dashboards, limiting real-time accessibility and user engagement. CityAirQ integrates real-time data visualization through a dedicated mobile application, offering interactive dynamic pollution maps, which provide insights into personal exposure and dynamic pollution patterns.

1.5. Urban Sustainability

CityAirQ directly addresses urban sustainability challenges by deploying an IoT-based air pollution monitoring system in a large city. The system's core innovation lies in its ability to generate real-time, dynamic maps of air quality, empowering citizens with important information about their environment.

Air pollution monitoring is a fundamental aspect of environmental sustainability. Accurate envionmental data is essential for understanding the extent of pollution problems and developing effective mitigation strategies. By creating dynamic maps of air pollution, CityAirQ provides a visual representation of environmental data, making it easier to identify pollution hotspots and track changes over time.

By providing real-time data to citizens, CityAirQ empowers individuals to make informed decisions about their exposure to pollutants. Raising public awareness about air quality can lead to healthier habits, such as choosing to walk or bike instead of driving.

The project enables the collection and analysis of large air quality datasets, which can used to identify trends, evaluate the effectiveness of pollution control measures, and inform policy decisions. Transparent data, made available to the public, increases accountability and encourages sustainable practices. The collected data can be used by city planners and policymakers to develop sustainable urban development strategies.

The project's long-term goal is to enhance the quality of life in large cities by promoting a cleaner, more sustainable urban environment, directly contributing to the city's progress towards achieving its environmental goals.

1.6. Paper Structure

This paper is structured as follows: Section 2 presents current State of the Art solutions while Section 3 includes the general architecture. The following sections thoroughly describe the architecture and implementation of the pollution tracking devices (Section 4), the mobile application (Section 5) and the cloud-based data pipeline (Section 6). Section 7 presents the experimental evaluation and Section 8 includes the conclusions and future work.

2. State of the Art

This section outlines a brief presentation of similar pollution tracking solutions, as well as current industry standards regarding hardware components. A general understanding of air pollution monitoring practices, as presented in Section 9.2, is essential for comparing different approaches and comprehending environmental safety.

2.1. Industry Standards

In terms of hardware devices and components, last 10–15 years brought significant improvements when it comes to the size and accuracy of the majority of sensors used in such environmental sensing applications. Modern sensors offer more precise readings when exposed to harsher weather conditions, but usually at the cost of higher energy consumption. Moreover, modern hardware became even smaller in size, allowing us to shrink the dimensions of the project up to the point where it can be used as a mobile sensing device, with an area less than half of a standard mobile phone.

Particulate matter sensors provide valuable pollution data in any environmental sensing device, but at a cost: significantly large cases. This problem became less impactful with the release of PMSA003 sensor: low cost, small dimensions and accurate results as shown by both field and laboratory tests conducted [13]. Two different technologies are present on the market for PM sensors: infrared and laser; out of these two, the latter technology offers more accurate results [14]. The working principle is reflected in Figure 1.

Infrared technology for CO_2 sensors was an important improvement over the older metal oxide semiconductor sensors offering enhanced readings that are not altered by external factors such as temperature or humidity. In 2016 the family of MH-Z19 sensors were released, offering longer lifespan, stable readings at a relatively low price.

The RV3028 RTC module represents a major upgrade regarding current consumption: it only draws around 40 nA from a backup battery to keep time, compared to other more popular alternatives such DS3231 that need about 1000 nA to be drawn from the battery in order to assure time keeping (25 times less energy consumed). A relevant study in this area has already been conducted in 2020 [15], showing an actual use-case scenario with only 22 nA needed to run.

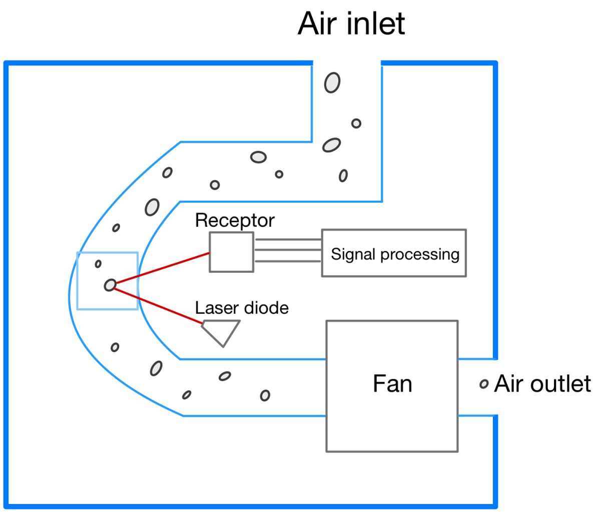

Figure 1. Particulate Matter (PM) sensor working principle.

2.2. Similar Projects

Extensive research has been conducted before determining how to build the device architecture and bind its elements together. It was of great help to examine other existing solutions on the market, particularly in our city, Bucharest, but also in other areas with comparable environments. By pointing out benefits and drawbacks on each of them, we were able to make informed decisions and maintain a simple, yet impactful design.

2.2.1. AirGradient

AirGradient devices feature sensors for measuring temperature, humidity, and key air pollutants like carbon dioxide, particulate matter (using the bulkier Plantower PMS5003T), TVOCs, and NOx, with CO_2 detection handled by the SenseAir S8 module. While they offer advantages such as real-time data reporting, historical data storage, and application-layer alerts, their size (13 $\times$ 13 cm, roughly the size of two iPhones side by side) makes them suitable only for fixed locations like walls or poles, as even the manufacturer targets static use cases. Though affordable at under 200 dollars, their design and functionality are unsuited for mobile air pollution tracking, a key requirement of the CityAirQ project.

2.2.2. uRADMonitor

uRADMonitor, a Romanian project launched in 2013, has developed a global IoT network of pollution monitoring devices, offering both fixed and portable designs to track air pollutants like carbon dioxide, carbon monoxide, particulate matter, VOCs, SO_2, NO_2, and environmental indices such as temperature, pressure, humidity, and noise. Among its offerings, the A3 model aligns closest with our objectives, featuring similar PM and CO_2 sensors with laser technology, though it lacks mobility and real-time pollutant display. However, uRADMonitor devices are costly, with the A3 priced at 589 dollars and other models reaching up to 3749 dollars.

2.2.3. Sodaq Air

Sodaq Air is a very similar project released in 2023, targeting the mobile usage, mounted on bicycles. It offers comparable achievements in the pollution tracking area: making use of PM, temperature and humidity sensors it is able to create a map of pollution over the areas one crossed by using a GPS module to keep accurate track of the routes.

Its usage and performance have already been tested and presented in the following article [16], outlining favorable results. However, a few identified shortcomings in the implementation are:

- Sodaq Air only keeps track of the PM pollutants, which is not sufficient for assessing a broader and more comprehensive pollution level tracking system
- It targets simplicity in being used as a semi-passive device, but it lacks on giving users detailed levels of the tracked air pollution and does not allow user interactions
- The service does not offer a mobile application and results can only be seen through accessing a web server [16], which diminishes mobility and accessibility

2.2.4. Airly

Airly is a prominent project with thousands of sensors across the UK and Europe, gathering over 40,000 data points from 50 countries to build a robust air quality database. Their stationary devices, roughly the size of a mobile phone but much thicker, primarily track PM pollutants, with some models also monitoring ozone, carbon monoxide, sulfur and nitrogen dioxide. While current pricing is unavailable, their standard sensor cost approximately 360€ in 2020. Additional pollution monitoring solutions are presented in Section 9.3.

3. General Architecture

Figure 2 shows the general architecture of the project, consisting of 3 main types of components, further described in detail throughout this paper:

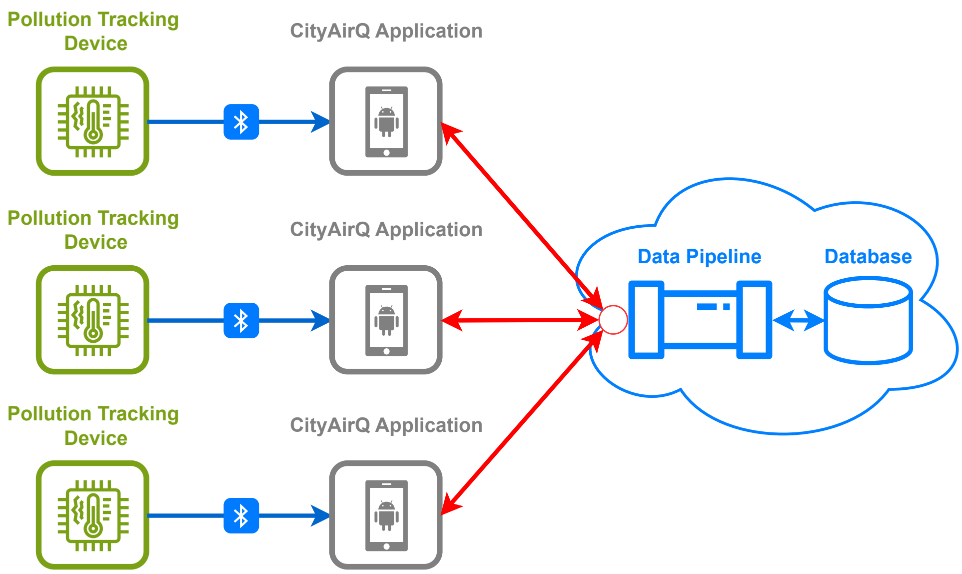

Figure 2. CityAirQ Architecture.

1. Pollution Tracking Devices are ESP32-based portable devices equipped with various air monitoring and environmental sensors, inside an enclosure with special brackets for mounting on bicycles. ESP32 offers Bluetooth Low Energy (BLE) [17] for persistent communication with the mobile application, allowing the transmission of real-time and historical sensor readings. Historical data is stored locally on the SD card prior to transmission.
2. The Mobile Application plays a critical role in the system, facilitating data transfer between components. It also provides data visualization, including real-time data, charts, warnings, and interactive maps. The application acts as a central hub which re-

ceives pollution indicators values from the tracking devices and renders the processed results from the cloud system.

3. A Cloud-based Data Pipeline features a custom API architecture, which enables seamless data transmission from the application to the cloud. After data ingestion, the system utilizes a queue structure to initiate storage and processing. Once the data is processed, it is fed back to the mobile application to render dynamic maps in real-time.

To enhance the value of CityAirQ's air pollution monitoring, multiple environmental indicators were selected from the potential options presented in Section Section 9.2, with their detailed ranges and measurement units provided in Table 1.

Table 1. Unit Measures and Ranges for CityAirQ Air Quality Indicators.

Parameter	Range	Unit Measure
Temperature	[−15.0, 57.0]	°C—degrees Celsius
Humidity	[0.0, 100.0]	%—percentage
Pressure	[950.0, 1047.25]	hPa—Hectopascals
Altitude	[−420.0, 8848.0]	m—meters
CO_2	[400.0, 2100.0]	ppm—Parts Per Million
PM1	[0.0, 250.0]	$\mu g/m^3$—micrograms of pollutant per air cubic metre
PM2.5	[0.0, 250.0]	$\mu g/m^3$
PM10	[0.0, 430.0]	$\mu g/m^3$
AQI	[0.0–500.0]	[18]

4. Pollution Tracking Device

This section introduces a cost-effective and energy-efficient IoT device designed for monitoring urban pollution. Its compact form factor allows for multiple usage scenarios, being suitable to be used both as a body-worn or bike-mounted device. These versatile deployment plans would allow us to create a network of interconnected pollution tracking devices, that enables continuous pollution monitoring throughout the day, with a focus on detecting and analyzing transient hotspot areas. Alternatively, utilizing the device as a wearable sensor allows users to actively monitor pollution levels in their immediate surroundings, providing real-time data on their personal exposure while traveling or engaging in outdoor activities.

4.1. Hardware Architecture

The hardware system depicted in Figure 3 integrates a wide range of advanced air monitoring sensors and off the shelf components to capture air pollution metrics from the surrounding environment.

The pollution tracking devices include the following hardware components:

- BME 680: this gas sensor is capable of measuring humidity, barometric pressure, temperature and volatile organic compounds (VOCs). It operates with exceptionally low power consumption, requiring only 3.7 μA for humidity, pressure, and temperature measurements, and up to 12 mA when also assessing the gas parameter. This sensor was selected for its multi-functional capabilities, replacing separate dedicated sensors for each parameter, thus reducing component count and overall cost.
- PMSA003: this particulate matter sensor operates with a power consumption of up to 100 mA in active mode. The majority of its power consumption is attributed to the internal fan, which must run continuously to ensure adequate airflow within the measurement chamber. We opted for this choice due to its compact size, laser-based

accuracy, and low power consumption, outperforming bulkier alternatives like the PMS5003T (used in AirGradient), which, while reliable, was significantly larger and consumed more power.

- MH-Z19B: this CO_2 sensor is built with the modern non-dispersive infrared (NDIR) technology. It has an average power consumption of 60 mA, with peak demands reaching up to 150 mA. For accurate measurements, the sensor requires a preheating phase of 3 min. It was chosen over older metal oxide semiconductor (MOS) sensors, which are more sensitive to environmental conditions like temperature and humidity.
- ESP32 S3: this microprocessor is responsible for controlling and handling sensor data acquisition, coordinating BLE communication, and managing user input/output logic. It features a dual-core CPU operating at a frequency of 240 MHz, includes 45 GPIO pins, and is equipped with 512 KB of internal SRAM. It was preferred over STM32 and Raspberry Pi alternatives due to its BLE connectivity, low power consumption, and cost-effectiveness, making it ideal for mobile IoT applications.
- Other components used: E-Ink display, RV-3028-C7 (real-time clock - RTC), MCP73831 (battery charge management controller), XC6220A331MR (3.3V low-dropout voltage power source), TPS63060 (buck-boost converter), SD card module.

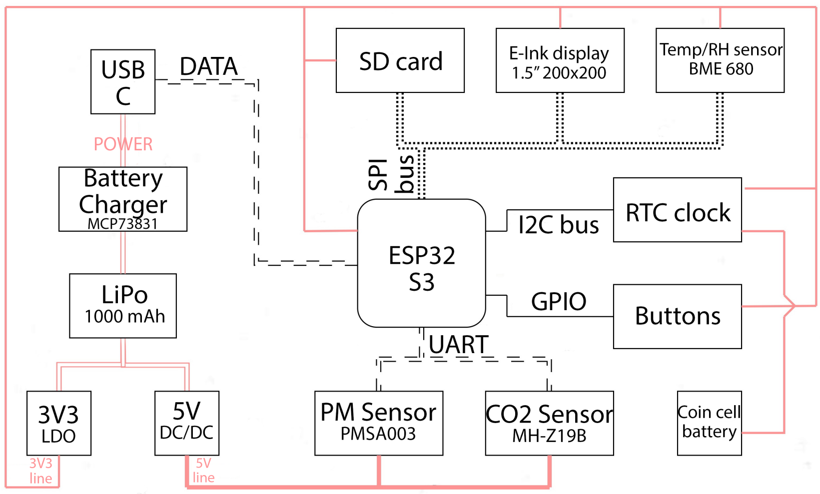

Figure 3. Hardware Architecture.

4.2. Hardware Implementation

The USB-C header serves a dual purpose: providing a data path to the ESP32 microprocessor for both chip programming and serial communication (primarily for debugging), and a power path for charging the main battery and supplying voltage to the system components. A simple power distribution switch is implemented to toggle the input voltage between battery and USB, when the latter source is detected, through the smart use of a PMOS transistor and a Schottky diode. More details on the integration of the RTC module are given in Section 9.4.1.

The final version of the PCB is shown in Figure 4. All sensors used for data acquisition are mounted on the back side, with careful consideration given to avoid interference with each other's intake openings and to ensure sufficient airflow for accurate measurements. The SMD coin battery holder is also located on the back. A key design decision was to position the CO_2 sensor above the card reader, utilizing its mounting points for a slightly elevated placement.

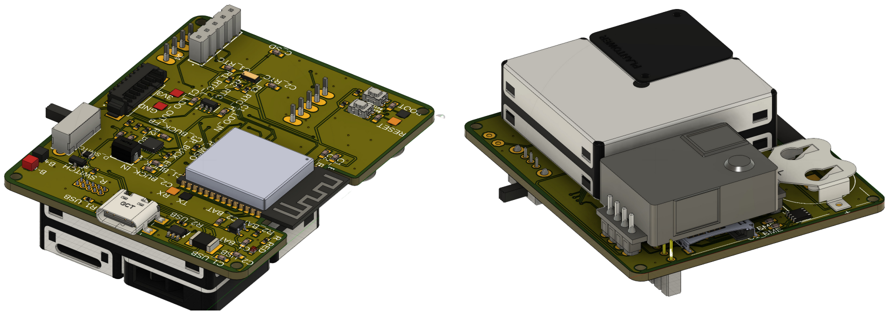

Figure 4. PCB rendering.

The battery and display were mounted externally to the PCB to further minimize space usage on the board. Wire headers were installed to ensure proper connections between the board and these external components. Additionally, the three GPIO buttons are placed on an external PCB for easier access, as illustrated in Figure 5.

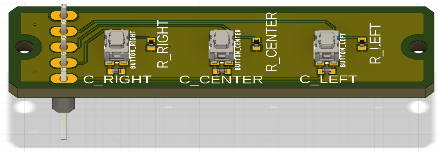

Figure 5. Buttons PCB rendering.

The overall PCB dimensions are just 63 × 53 mm. Further downsizing is nearly impossible due to the physical constraints of the PM and CO_2 sensors. The optimal arrangement is to mount them horizontally across the vertical axis of the PCB. The PM sensor is approximately a square with a 37 mm side, while the CO_2 sensor is 21 mm wide, resulting in a combined height of 58 mm—only 5 mm shorter than the PCB itself, allowing for a safety gap between components. This minimizes interference between sensors and ensures unobstructed air intake through the carefully designed air vents on all sides of the final enclosure.

To ensure that no overlaps occur between components, two sectional analyses are presented in Section 9.4.2.

4.3. Enclosure Design

The enclosure was precisely designed around the PCB's format, achieving the smallest possible form factor. Air vents were strategically placed on the bottom and 3 sides, aligned with the sensors' air intake openings, to allow optimal airflow through the case. Button covers were included to enhance aesthetics and usability. Rendering is shown in Figure 6.

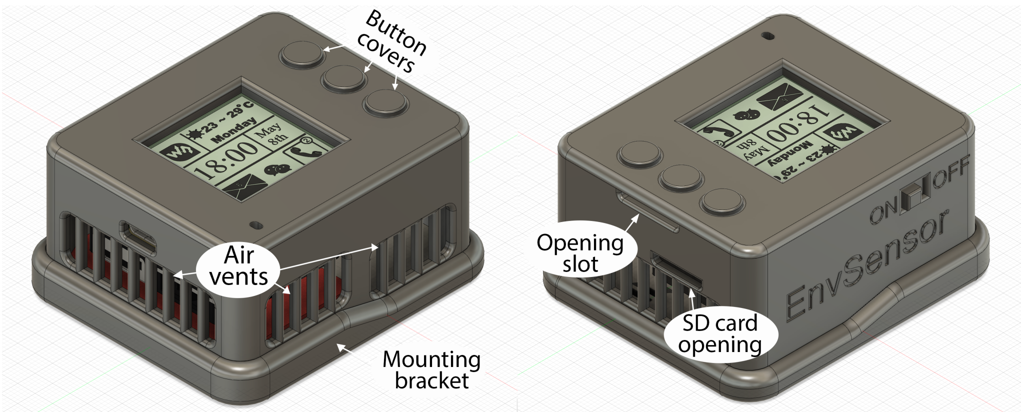

Figure 6. Enclosure rendering.

Its design features rounded corners, providing a more ergonomic feel for daily use. For user convenience, the power switch is labeled with engraved "On" and "Off" indicators. A screw-free design for both the lid and mounting bracket was achieved through a custom snap-fit system.

4.4. Software Architecture

Figure 7 (left side) provides a high-level overview on the software architecture. Upon device startup, the sensors and other software modules are initialized. Subsequently, the BLE server begins advertising its services, while the sensors enter a heating phase to ensure accurate readings. This is followed by a cyclic phase, where data is collected from the sensors and displayed on the screen. If a client is connected to the BLE server, it receives real-time updates; otherwise, the collected data is stored locally as historical data.

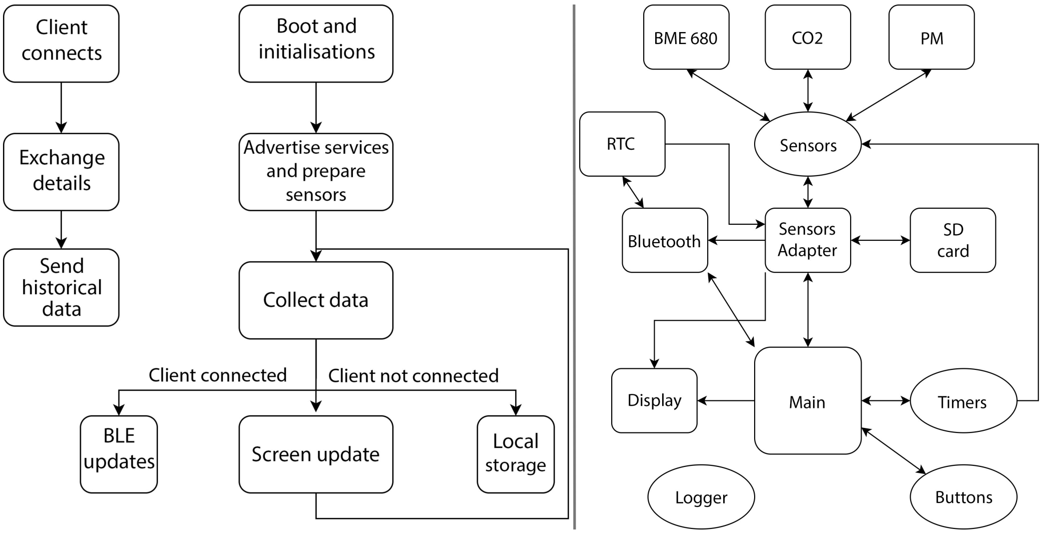

Figure 7. Software Architecture and Implementation.

In a separate flow, once a client connection is established, key details such as the current timestamp from the mobile device are synchronized and batches of previously stored historical data are transferred from the sensing device. The data collected are structured in a CSV format: [Timestamp, Voltage, Temperature, Humidity, Pressure, Gas, Altitude, CO_2, PM1, PM2.5, PM10].

Data acquisition from all sensors occurs once every 20 s in a sequential manner. If there is no client and connected and BLE connection does not need to be maintained, in between readings the device enters a sleep mode where all sensors are in standby in order to reduce battery consumption; it will recover from this state earlier than the exact time data collection is scheduled in order to prepare the sensors for accurate acquisition. Once collection is performed on the hardware device, it will be updated immediately on the screen, but due to e-paper screen limitations it will have up to 2 s of delay, if a full screen refresh cycle happens. At the same time, BLE updates containing live data values are sent to the mobile application, which processes them with no visible delay.

In all scenarios, considering the main use-case of being a bicycle mounted device, its average travel speed would be around 16 km/h, which means that each data collection will be in a radius of less than 100 m apart from the previous one, allowing for proper pollution mapping of the area even when the map is zoomed in. For areas in between reading points, pollution values can be interpolated in the data pipeline, outside the processing that takes place on the embedded device.

4.5. Software Implementation

The actual software implementation for the firmware is illustrated in Figure 7 (right side). In this diagram, rectangles denote modules designed as objects, while ellipses represent extensions of the main code. All sensors implement two common interfaces critical to the system's functionality: one for data handling and another for state and data acquisition. This ensures that sensors are accessed uniformly when passed as parameters to handler functions, providing a generalist and extensible format.

The Sensors Adapter module is responsible for managing the core logic of data acquisition and handling. It operates in coordination with both the Bluetooth and SD modules to fulfill these tasks. After a sensor completes a reading, the collected data is processed by this adapter and handled as follows:

- BLE is enabled & client connected: the corresponding BLE characteristics are updated.
- No client is connected: data is saved on the SD card as a batch of historical data.

The system is designed to be fault-tolerant under three scenarios, from which it will try to self-recover through a software designed mechanism by disconnecting the affected sensor and starting the pairing procedure again. During this operation, corresponding data is invalidated on the device itself, but can be virtually approximated through methods such as interpolation or prediction (based on AI methods that takes into account the other valid data) in the data pipeline.

- Sensor Not Found Error: This error suggests a hardware connection issue (e.g., power failure) if the sensor cannot be detected prior to the initialization phase.
- Sensor Initialization Error: This occurs when the sensor's physical connection is established, but initialization fails, potentially due to communication interference on the data lines.
- Sensor Read Error: This happens after successful initialization if issues arise during readings, such as false data, timeouts, or incorrect messages. False data is considered when the received values tend to be in the extremities of sensor's allowed measuring range.

In these cases, errors are logged, and a recovery procedure is triggered.

4.6. Innovation

The CityAirQ pollution tracking device introduces several key innovations that differentiate it from existing air quality monitoring solutions:

- Ultra-Compact Custom PCB Design for Wearability: Unlike existing solutions such as AirGradient or uRADMonitor, which offer larger sensor modules, CityAirQ achieves one of the smallest possible footprints (63 by 53 mm on PCB, while the actual enclosure only adds 1 mm of tolerance all around, with a height of just over 3 cm). This is achieved by strategically integrating off-the-shelf sensors into a custom developed PCB, optimizing their placement to minimize space while ensuring proper airflow for accurate readings (more details offered at the end of Section 4.2). Competing systems such as Sodaq Air or Airly, while portable, do not achieve a comparable reduction in device dimensions, while maintaining the same functionalities (CO_2 measurements, on-device display etc.).
- Optimized Power Consumption for Extended Use: Building a custom device allows for direct control over power distribution and system behavior. Unlike commercial sensor modules that operate with fixed power requirements, CityAirQ implements selective activation and duty cycling of sensors, especially the PM and CO_2 sensors, reducing average power consumption by 30% compared to continuous operation (detailed in Section 7.1.2). This results in longer battery life while maintaining reliable

data collection, an advantage over devices such as AirBeam3 or Atmotube Pro, which prioritize real-time sensing but sacrifice battery efficiency.

- Seamless Mobile Integration with Dynamic Pollution Mapping: CityAirQ integrates real-time sensor data visualization on both the device itself as well as through the dedicated mobikle application, with a cloud-based pipeline that generates dynamic pollution maps. Unlike Sodaq Air, which only tracks PM pollutants and lacks a dedicated mobile application, CityAirQ provides detailed air quality insights (CO_2, PM, temperature, humidity, pressure and more to be added) via a user-friendly Android application with interactive mapping and alerts. This enhances usability and accessibility beyond the web-based dashboards used by competitors like Airly and Oizom.
- Cost-Effectiveness Without Compromising Functionality: In comparison to existing solutions such as uRADMonitor (starting at 589 USD) or Airly (360 Euros per unit), CityAirQ offers a highly affordable alternative at approximately 86 USD, making it more accessible to a broader audience. This remarkably low cost achieved without sacrificing performance, primarily comes from having the ability to select, compare, and integrate the best price-to-performance sensors available on the market. Unlike pre-configured commercial solutions that often rely on fixed sensor modules with limited flexibility, our approach allowed for a meticulous selection process across multiple sensor categories, ensuring a great quality-to-price ratio for each component.

4.7. Sensors

4.7.1. Monitoring Ranges

CityAirQ is designed to provide accurate and reliable environmental data by covering a broad range of pollutant concentrations and environmental parameters. The system integrates carefully selected sensors that ensure high sensitivity across multiple air quality indicators.

- PMSA003:

  Particle range of measurement: 0.3–1.0 μm (PM1), 1.0–2.5 μm (PM2.5) and 2.5–10 μm (PM10)

  Particle counting efficiency: 50% for 0.3 μm particles and 98% for greater than 0.5 μm particles

  Effective measuring range: 0–500 $\mu g/m^3$

  Accuracy: $+-$ 10% for 100–500 $\mu g/m^3$ range and $+-$ 10 $\mu g/m^3$ for 0–100 $\mu g/m^3$
- MH-Z19B:

  Effective measuring range: 0–5000 ppm

  Accuracy: $+-$ 50 ppm + 3% of the reading value (if calibrated)
- BME680:

  Gas sensor: Accuracy is $+-$ 15 % and Measuring range 0–500 IAQ index

  Humidity sensor: Accuracy is $+-$ 10 % and Measuring range 0–100%

  Pressure sensor: Accuracy is $+-$ 1 % and Measuring range 300–1100 hPa

  Temperature sensor: Accuracy is $+-$ 1 °C and Measuring range is −40–85 °C

4.7.2. Calibration Procedures

Ensuring accurate sensor readings is essential for meaningful pollution tracking. A 2020 study [19] analyzed the performance and calibration of two low-cost particulate matter sensors, PMSA003 (the one used in CityAirQ) and Shinyei PPD42NS. The results demonstrated that:

- PMSA003 outperformed the Shinyei sensor, delivering more reliable readings.

- Factory calibration of PMSA003 is already highly accurate, with a Pearson correlation coefficient of 0.998 between two identical sensors placed in the same location, compared to 0.853 for the Shinyei sensor.
- While the study offers two potential calibration methods for PMSA003, they require long-term observational studies to adjust for specific environmental conditions (temperature, humidity, and PM levels). Since PMSA003 already provides consistent factory calibration, further tuning was not necessary at this stage, but will be addressed in future iterations.

For the MH-Z19B CO_2 sensor, multiple calibration techniques exist, each suited to different accuracy requirements and application needs:

- 0 point calibration: Enhances accuracy for lower-range readings (0–400 ppm) by calibrating the sensor in a sealed chamber with pure nitrogen or argon. While precise, this method is expensive and requires specialized calibration software. Given that typical outdoor CO_2 levels exceed 400 ppm, this calibration was not necessary for CityAirQ.
- Rich CO_2 mixture calibration: Achieved using a baking soda and vinegar controlled reaction. Although simpler, this method is less efficient and should be combined with other procedures.
- Fresh air calibration: Relies on exposing the sensor to ambient outdoor air, where CO_2 levels naturally stabilize slightly above 400 ppm. This method is effective when high precision is not the primary goal.
- Auto-calibration: The sensor firmware includes a built-in self-correction mechanism that automatically adjusts the reference baseline every 24 h. This method, however assumes that the sensor will be in contact with fresh outdoor air, when it should measure its lowest readings, and use those to adjust its baseline.

For CityAirQ, a hybrid calibration strategy was implemented:

- Exposing the sensor to a rich CO_2 mixture in a hermetic container for 30 min to adjust its response range.
- Allowing self-calibration in outdoor air for 24 h to establish an accurate baseline.
- Enabling the sensor's built-in auto-calibration function to maintain long-term stability.

4.8. Conclusions

The development of the CityAirQ pollution tracking device successfully balanced compactness, energy efficiency, affordability, and measurement accuracy, making it a viable alternative to existing air quality monitoring solutions.

- A highly compact form factor (63 × 53 mm PCB) was achieved through strategic sensor placement, ensuring that all components fit within the smallest possible footprint while maintaining adequate airflow for precise pollutant measurements. Unlike bulkier commercial alternatives, CityAirQ offers a wearable and bike-mounted form factor without compromising functionality.
- Power consumption was significantly optimized through the implementation of duty cycling, which selectively activates sensors only when necessary. This approach led to an extended battery life of approximately 5 h, striking a balance between continuous monitoring and energy efficiency, making it more practical for mobile applications.
- The affordable cost of CityAirQ was made possible by carefully selecting and comparing available sensors, ensuring that each component offered the best price-to-performance ratio. By avoiding pre-configured sensor modules with fixed specifications, the project achieved a low-cost yet highly functional pollution tracking system, making it accessible to a wider audience.

- Calibration procedures were applied to the CO_2 sensor to ensure accurate readings in real-world conditions. A hybrid calibration strategy was implemented, combining rich CO_2 mixture exposure, fresh air calibration, and auto-calibration. This method allowed the sensor to self-correct over time, reducing measurement drift and enhancing reliability.

The decision to focus on particulate matter (PM) and CO_2 monitoring in the first iteration of CityAirQ was driven by a careful balance between cost-effectiveness, sensor importance, and space constraints. PM and CO_2 are among the most critical air quality indicators, with well-documented health impacts. PM exposure is directly linked to respiratory and cardiovascular diseases, while CO_2 levels serve as an effective proxy for ventilation quality and urban pollution hotspots. Both parameters can be accurately measured using cost-efficient sensors, ensuring reliable data collection without significantly increasing the overall price of the device. However, CityAirQ was designed with modularity in mind, allowing for future sensor additions as technology advances and more compact, energy-efficient gas sensors become available.

4.9. Limitations & Challenges

Despite its advantages, the development process revealed certain technical limitations that may impact measurement accuracy and user experience:

- The range of tracked pollutants remains limited, as the current design does not include dedicated sensors for NO_2, SO_2, or additional VOCs. While this was primarily a result of space and cost constraints, future iterations could integrate additional sensors to improve coverage.
- The e-paper display, chosen for its low power consumption, has a slow refresh rate, which may result in a noticeable delay when updating real-time pollution readings. Although this helps conserve battery life, it slightly reduces the immediacy of displayed data.
- Temperature readings from the BME680 sensor were occasionally misleading due to its placement on the main PCB, where it was affected by heat dissipation from other electronic components. This resulted in higher-than-actual temperature readings, highlighting the need for a more thermally isolated placement in future versions.
- The PM sensor's air vents were designed larger than ideal, which, under high-speed movement (e.g., cycling or riding in windy conditions), allowed excessive airflow through the sensor chamber. Since the sensor is calibrated for a specific air volume intake, this led to inconsistent PM readings at higher speeds. A more controlled air intake mechanism could mitigate this issue.

The CityAirQ prototype successfully demonstrated that a wearable and bike-mounted air pollution monitoring system can be achieved with a highly compact, energy-efficient, and cost-effective design. While the current implementation has some hardware limitations, these insights provide valuable direction for future improvements, ensuring that the next iterations of the device offer even greater accuracy, pollutant coverage, and user experience.

4.10. Future Work

Building on the insights gained from the current CityAirQ prototype, several hardware and software enhancements are planned to further improve accuracy, efficiency, and usability.

- Addressing the Temperature Sensor Issue: To eliminate the heat dissipation effect that influenced temperature readings, the BME680 sensor has been relocated to the external buttons PCB, where it is less affected by heat from the main board. Additionally, cutouts were made into the PCB design to further reduce thermal interference, ensuring more reliable environmental measurements.

- Expanding the Range of Tracked Pollutants: Future iterations will integrate the MICS-6814 sensor, which is capable of detecting a broader range of harmful gases such as NO_2, NH_3, and CH_4. This addition will allow CityAirQ to provide a more comprehensive assessment of urban air quality, addressing a key limitation of the current design.
- Implementing On-Board Software Compensation Algorithms: To further increase measurement accuracy, software-based compensation techniques will be developed. These algorithms will focus particularly on PM sensor readings, adjusting values based on external conditions such as airflow variations, humidity, and temperature. This will help mitigate inconsistencies caused by high-speed movement and environmental fluctuations.
- Refining the Enclosure for Improved Aesthetics and Usability: While the current enclosure prioritizes functionality, future versions will undergo aesthetic refinements to enhance user experience and wearability while maintaining the device's compact and ergonomic design. Improved airflow control will also be integrated to further stabilize sensor readings.
- Further Optimizing Power Consumption: While the current duty cycling strategy has already extended battery life, additional power optimizations will be explored. A key improvement will be the potential integration of the MH-Z1311A CO_2 sensor, which offers an ultra-low power consumption of just 1 mA on average, compared to the current MH-Z19B sensor's 60 mA average consumption. This upgrade could significantly enhance energy efficiency, further extending operational time on a single charge.

5. Mobile Application

The mobile application plays the role of a middle ground, or interface, that can facilitate user interaction and easily turn any bicycle, due to its attached device, into a powerful environmental tool.

5.1. General Architecture and Design Pattern

The mobile application enables communication between different components of the system, following the data life cycle, from collection to visualization. It begins with initial data collection through pollution tracking devices (via BLE, left side of Figure 8), followed by sending collected values to the cloud-based pipeline (right side of Figure 8). After thorough analysis and processing, the application receives and displays results in the user interface (UI), in form of dynamic maps and alerts (Figure 9).

Managing data flow comes with a set of performance requirements, such as fast processing of live data, high bandwidth networking and no frame rendering bottlenecks.

Once endpoint connections are established, the next important step is to decide how to manage information and integrate data into the application architecture (Figure 9), taking into consideration visualization, accessibility [20] and distraction optimization.

By making comparisons between fundamental existing design patterns for Android, the chosen option for the application, which best fits the objectives and functionalities is Model-View-ViewModel-Coordinator (MVVMC) [21], an extension of classic Model-View-ViewModel (MVVM). The reasons for this architectural choice are quite straightforward, ranging from navigation features, encapsulation and single responsibility to separation of back end flow from the UI and memory usage [22].

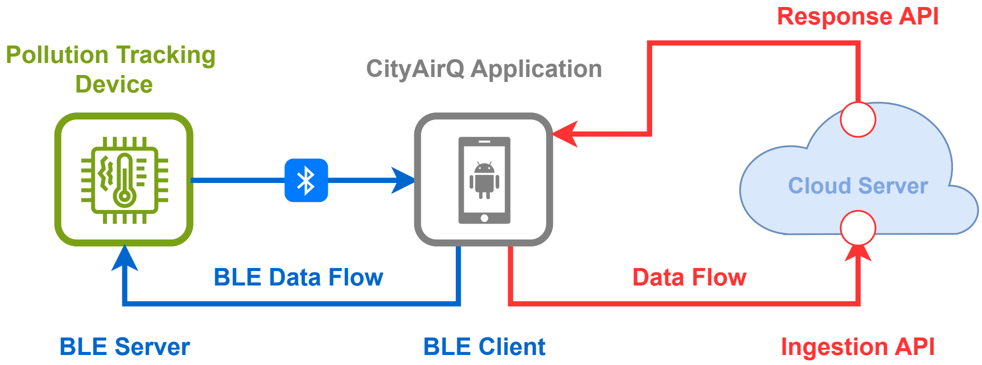

Figure 8. CityAirQ Communication Data Flow.

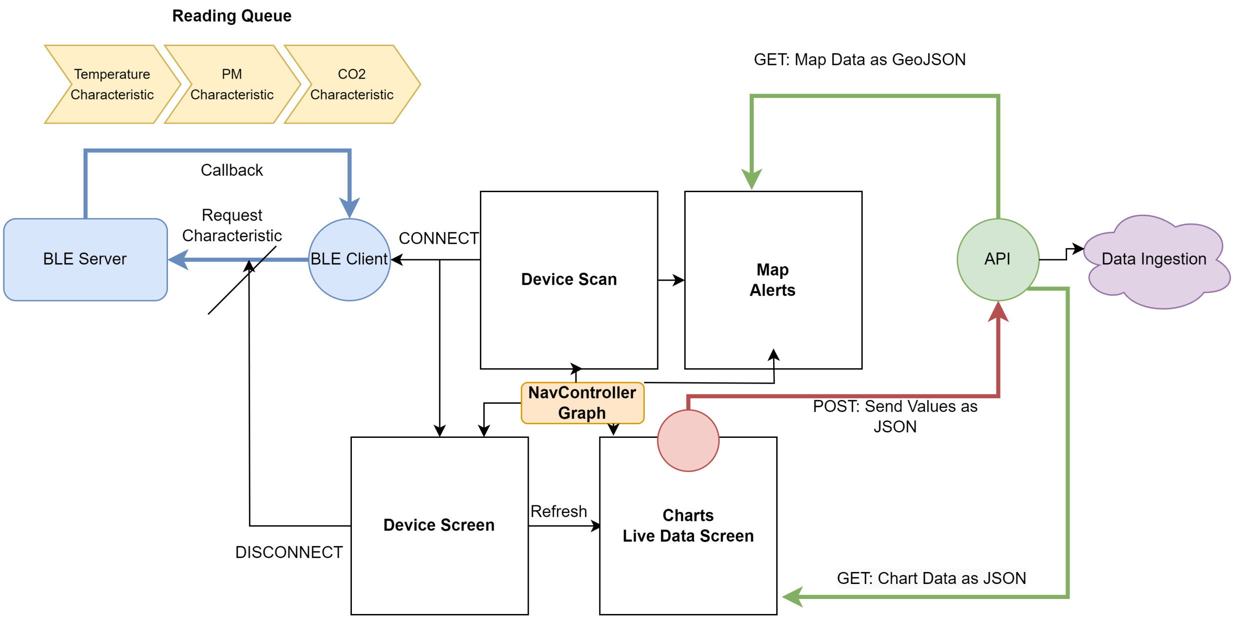

Figure 9. CityAirQ Application Architecture.

5.2. Communication with the Hardware Device

Once the device, the first essential source of data in the air monitoring flow, is powered, values are collected and sent through BLE, a low-power, low-bandwidth protocol, widely used in IoT data transfer. Although BLE has some well known limitations, such as short-range connectivity [23], it is definitely appropriate for our use case, not only for its sleep mode mechanism during no connection, which preserves the battery life, but also due to its support for Android Operating System (OS).

The hardware device plays the role of BLE Server, which, in low power mode, advertises its availability to connect through parameters, called services (one per BLE session) and characteristics (one per air pollution parameter), under a common standardization, called Generic Attribute Profile [23] (GATT).

As Figure 10 shows, the mobile application portrays the BLE Client, which has the responsibility to scan for all available BLE devices in the close area, differentiated by the same particular parameters. Both client and server share a series of unique attributes, known as Universally Unique Identifiers (UUIDs), which are useful to identify services and characteristics and, if a match is found, establish connection and listen for input values. For CityAirQ, UUIDs were selected from the documentation [24], eight UUIDs for the parameters presented in Table 1, one for timestamp (to differentiate live data from older values) and one for battery.

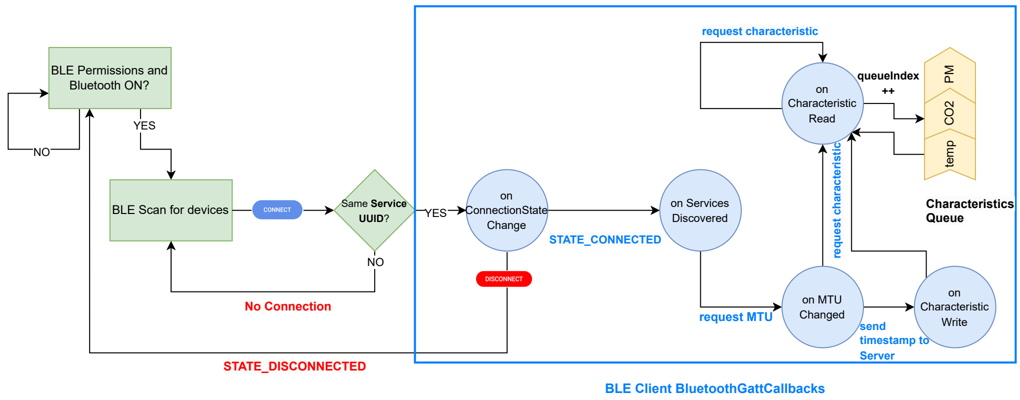

Figure 10. BLE GATT Client Android Implementation.

One limitation of this library, which may not be clarified in the documentation, is that the client is only able to read one characteristic at a time, followed by waiting for a callback before being able to read again, which makes the process sequential. The optimization described in [25] utilizes a Queue mechanism, that definitely improves the reading speed in the application. The application iterates through a list of characteristics to be read using a queueIndex. To prevent blocking the communication channel, the queueIndex is incremented regardless of whether the previous reading was successful or not (Figure 10).

5.3. Communication with the Cloud System

The Android architecture was designed in a way to ensure a fast communication interface between the presented endpoints, therefore it shall not take responsibility to manage large volumes of live sensor readings or historical values. That method would inevitably cause unnecessary processing overhead and inefficient use of memory. This is the motivation for the following architectural decision: raw values are locally stored and periodically updated in the UI on new BLE readings, then immediately sent to the Ingestion API through Hypertext Transfer Protocol (HTTP) POST methods (right of Figure 9).

As presented in Section 6, ingested data is processed, resulting in relevant outputs for the application. Listing 1 shows the GET request format, which requests pollution values within an area centered in coordinates [latitude, longitude] and a specific radius. On the other hand, the response body contains resulting values in both Cartesian coordinates and H3 spatial system.

Listing 1. Request URL for Results API.

```
'http://{domain}/measurements?
location=44.435022,26.046144&
radius=1000&
layers=TEMP,HUM,PRESS,ALT,CO\textsubscript{2},PM1,PM2.5,PM10,AQ'
```

5.4. Implementation and Libraries

A global leader in the mobile OS world [26], Android was the obvious choice for CityAirQ, together with Kotlin and Jetpack Compose for development. This modern UI toolkit is declarative, based on UI Composables, which are reusable, responsive and can be redrawn on the screen based on events. When it comes to performance, tests [27] have confirmed that, compared to traditional XML, Jetpack Compose has faster launching times on application startup and overall better Central Processing Unit (CPU) and memory usage on user navigation.

The front-end was implemented using Material Design, with a user-friendly light theme and UI components such as expandable buttons, clickable menu and text to speech for accessibility. For navigation, by using subclasses and functions from the official Android Navigation Library [28], an interface was built to manage which UI Composable is rendered at all times. The most important element is the Navigation Controller (center of Figure 9), which is the fundamental API and allows defining navigation under arborescent form, respectively a graph with routes and destinations, handled by a Navigation BackStack and a Navigation Host.

Multiple interfaces and ViewModels were necessary when it comes to state management: Bluetooth, Location and Network Connectivity services need to be monitored, together with necessary permissions. The Android version installed must exceed at least minSDK 26, Android 8.0 Oreo, for BLE support. For the data communication flow with the Cloud Server, the most straightforward library for the HTTP protocol was Retrofit, supported by Moshi for JSON serialization. After building a Retrofit HTTP service to the desired endpoint URL, serialized sensor data is sent every 30 s, through HTTP POST requests, followed by receiving responses with HTTP GET.

Once the response is parsed, data visualization becomes the key objective (Figure 11). On one hand, live values are rendered, together with historical graphs (MPAndroid Chart), animated circular charts scaled to the corresponding range, using a Navigation Rail for observing each pollution index. On the other hand, alerts are set to inform users through Push Notifications about different events, rendered on screen as Cards that are swipeable and can be arranged at the top of the notification stack.

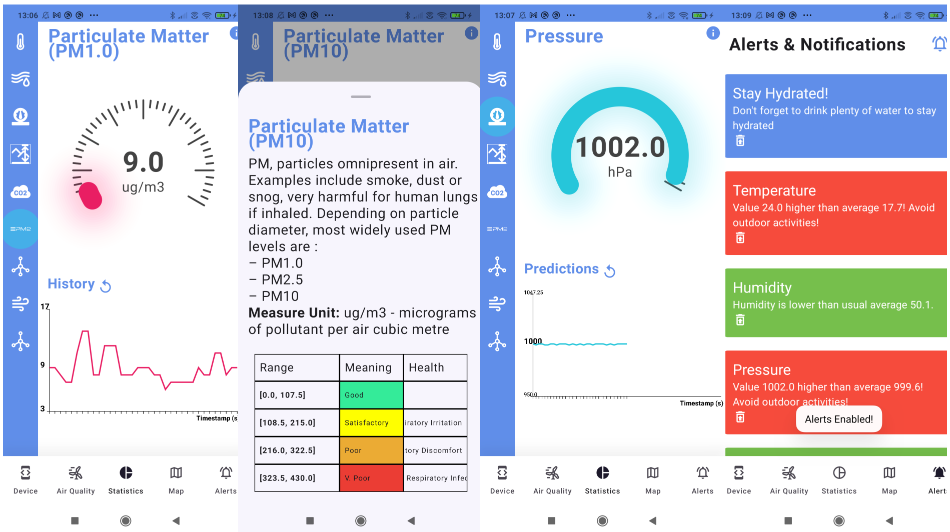

Figure 11. CityAirQ Data Visualization.

Most importantly, for the mapping system, the decision was to to explore the Jetpack Compose MapBox extension [29], which is quite new and experimental, but offers functionalities that were put into practice in the form of a map Composable with zooming, scale of dimensions, compass, 2D location puck, trailing routes (Points made of location latitude and longitude, transformed into LineString [29], a specific layering GeoJSON format) and computing the total distance covered.

The HeatMap is potentially the most important and dynamic visual part of the mapping system, which describes pollution levels the best. After analyzing multiple heatmap styles and formats (including the classic H3 style), the best decision was to stick with the HeatmapLayer from the extension [29]. In the first layer, location tuples are combined with an intensity property and transformed in GeoJSON format. By matching intensity, opacity ratios and other characteristics, separate layers can be rendered on the the HeatMap for each pollutant.

6. Cloud-Based Data Pipeline

6.1. Pipeline Architecture

The proposed data pipeline is designed to efficiently manage and process large volumes of IoT data as represented in Figure 12. We have successfully implemented the core backend components, laying a solid foundation for future automation and integration.

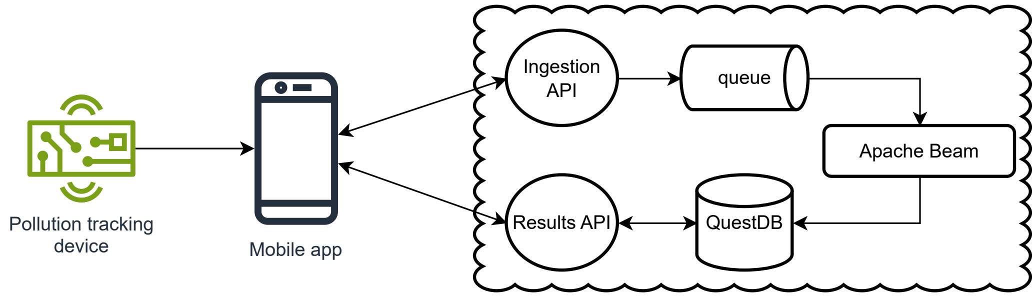

Figure 12. Pipeline Architecture.

The high-level architecture of our project is designed with the following flow:

1. A REST API connects edge devices to the back-end infrastructure, transferring data received from the mobile application into a message queue (Google Pub/Sub) for further processing.
2. All data from the queue is then processed using the Apache Beam framework.
3. The processed data is stored in a QuestDB database.
4. In the final step, an additional API delivers the processed data from QuestDB to the mobile application.

The data pipeline includes several steps, such as: deduplication, validation, transformation, and storage.

- Deduplication: Streamlining deduplication using an open-source Apache Beam sink for QuestDB data storage.
- Validation: Ensuring that the data for each dimension falls within the correct boundaries.
- Transformation: Using the H3 algorithm, a geospatial indexing system with hexagons of varying resolutions to uniquely identify Earth's surface zones. The transformation step includes normalizing data (e.g., setting correct data types) and converting GPS coordinates into H3 zones.
- Storage: Persisting the processed data in QuestDB.

6.2. Data Processing

The data processing component of the pipeline is crucial as it transforms raw data into valuable information. Our primary requirements that we want to achieve for this part of the architecture were: the ability to accumulate data for rolling time windows, scalability with a large volume of data and flexibility to enable easy code writing for actual data processing.

We have chosen Apache Beam as it performed exceptionally well for our requirements. The programming model of Apache Beam includes transformations that manage the processing of data stream, making it easy for us to define, deploy, and manage complex data processing pipelines. The most important step that needs to be integrated into the current flow in the future is using trained ML algorithms to make predictions and visualise the pollution map on a larger scale. Moreover, Apache Beam's dedicated functionality enables us to integrate ML predictions into the pipeline.

Apache Beam is responsible for communicating between the code written in its API and the runner that executes it, performing heavy operations. After evaluating several options, we chose Google Dataflow as our preferred runner. This fully-managed service eliminates the need for server management and scaling, making it easier to maintain. Additionally, since our cloud provider is Google Cloud, it is already present there as a managed service, making integration with the rest of the architecture components easier. We also considered Apache Flink but ultimately chose the more straightforward option.

6.3. Data Storage

We considered two types of data storage: frequently used data storage and archived raw data storage. To date, we have implemented the frequently accessed data solution.

For fast, frequently accessed data, we have incorporated QuestDB, a high-performance columnar database designed explicitly for time-series data. QuestDB is fast and efficient, handling massive volumes of time-series data with low latency. This makes it a perfect candidate for IoT applications, where sensor data is produced continuously and in large volumes. We also introduced an instance of Apache Superset into our infrastructure to facilitate data exploration and allow a more complex visualisation. It can be directly connected to the database thanks to the existing Apache Superset - QuestDB connector.

7. Experimental Evaluation

This section presents the experimental evaluation of our pollution tracking system, outlining the comprehensive testing procedures and data collection methods employed.

7.1. Hardware Device Testing

Both laboratory and field tests were conducted on the physical pollution tracking device to validate its functionality.

7.1.1. Estimated Power Consumption

Comparing the measured current consumption of the final prototype with the theoretical values from component datasheets (Table 2), we observed slightly higher real-world values. This discrepancy may be due to inaccurate datasheet estimates from less reliable sources and unaccounted power consumers like the battery charging module, diodes, and RTC. However, the differences remain small. To improve accuracy, the theoretical model included duty cycling—e.g., the PM sensor operates at 100 mA only during readings (5–7 s every 20 s), averaging 1200 s of active use per hour, while consuming just 2 mA in sleep mode the rest of the time.

Table 2. Theoretical power consumption.

Component	Standby Current (mA)	Active Current (mA)	Standby Time in 1h (s)	Active Time in 1h (s)	Consumption (mAh)
Screen	0.035	10	3000	600	1.69
BME680	0.1	17	2460	1140	5.45
microSD	1	80	3350	250	6.48
ESP32	1	24	0	3600	24
PM sensor	2	100	2400	1200	34.66
CO_2 sensor	60	150	3000	600	75
GPIO	1	1	3570	30	1
3.3 V source	5	10	1800	1800	7.5
5 V source	0.1	2	0	3600	2
Voltage divider	0.2	0.2	0	3600	0.2
Total					157.8

7.1.2. Actual Power Consumption

One of the most relevant aspects, if not the first, of a battery powered device is to evaluate its power consumption to obtain an expected estimate of its running time. For this reason, we tested it against the Nordic Power Profiler Kit II for power profiling. Our initial findings showcased an average power consumption of 265 mA, with constant voltage spikes between 180 and 420 mA due to electrical noise generated by PM's sensor fan, as seen in Figure 13; these values lead to an estimated usage time of about 3 and a half hours.

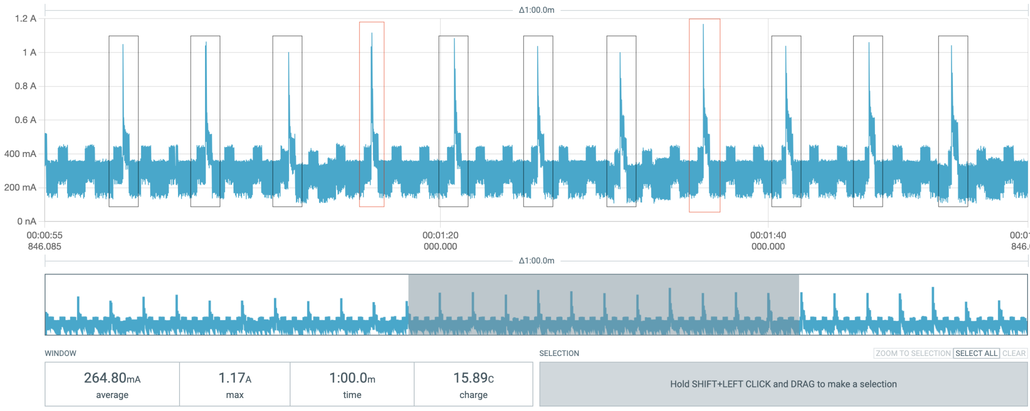

Figure 13. Live mode not optimised.

Software improvements, specifically applied duty-cycling techniques on the PM sensor, resulted in a 30% reduction in overall power consumption, averaging 186 mA, leading to an expected continuous usage time of about 5 h. Moreover, the constant noise was reduced by disabling the fan in between reads, as seen in Figure 14.

Figure 14. Live mode optimised.

In Figures 13 and 14, the moments when all sensors are read sequentially are indicated by the red rectangles. These events occur at regular intervals of 20 seconds, corresponding to the selected sampling rate of the sensors. The black rectangles highlight other periodic voltage spikes, which are associated with the heating process of the CO_2 sensor. This heating event can be observed through the sensor's intake vent, where the resistor filament visibly glows red for approximately one second.

7.1.3. CO_2 Accuracy

To evaluate the impact of calibration on CO_2 measurement accuracy, I tested two non-calibrated MH-Z19B sensors against a third sensor that had undergone a 30-h calibration process. All devices were allowed to warm up for approximately three minutes before measuring outdoor CO_2 levels in proximity to uRADMonitor stations, which reported an average concentration of 522 ppm. Notably, Airly and IQAir do not provide CO_2 tracking, making uRADMonitor the primary reference for this test.

The results (Figure 15) confirm the benefits of calibration:

- The calibrated sensor (right) provided the most accurate reading at 516 ppm, closely matching the reference value.
- The two non-calibrated sensors (left and center) reported higher CO_2 levels, at 634 ppm and 609 ppm, respectively. Despite this offset, both remained within an acceptable deviation of +− 100 ppm from the expected range.

Results state that after calibration we achieved a 1% off precision rate, compared to a 20% error on the non-calibrated ones.

Additional Notes:

- The rightmost sensor was running a different firmware version for testing purposes, which led to variations in temperature and PM readings. The temperature measurement appears more realistic, but PM data was inaccurate (this issue was later resolved).
- The battery level reading on the left sensor was arbitrary, as the reference pin was not connected to the ESP32 ADC.

Figure 15. Comparison of calibrated and non calibrated sensors.

7.2. Application Deployment and Performance

The mobile application was deployed on more than 5 Android mobile devices, both smartphones and tablets, proving compatibility with devices of minSDK higher than Level 29, which means Android 10.

Figure 16 summarizes application performance, measured with the Android Studio Profiler tool. As expected, due to scanning and location optimizations, the application performs best when it runs in the background with the screen off, with a CPU of 17–20%, 170 MB of memory and light energy level. Due to the large volume of data points, constant location approximation, and rendering provided by the MapBox Extension, the system experienced peak resource usage of 43% CPU and 470 MB of memory. An additional evaluation method is through objective user feedback, as described in Section 9.5.

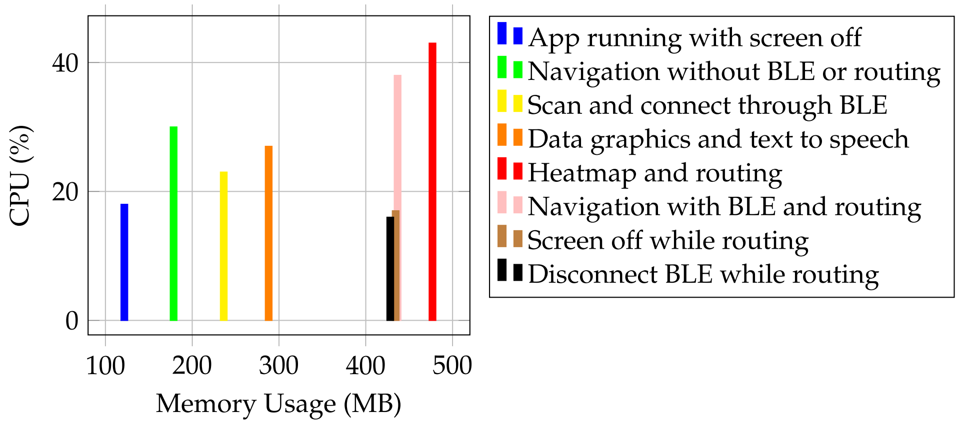

Figure 16. Performance Parameters for Mobile Application.

7.3. Field Trial Evaluation

Having successfully deployed and evaluated both the device and application, the next phase is to prove the performance of the whole system by running tests for the connected infrastructure.

In the first phase, we conducted a field trial for demonstration purposes, involving the whole CityAirQ infrastructure: hardware device and the application deployed on a Xiaomi Redmi Note 8 Pro with Android 10. The area of interest was 1 km around a crowded residential area in southern Bucharest, with busy traffic.

The first step, before going on the field, is to establish a connection to the hardware device. The application is launched with both mobile data and Bluetooth enabled, followed by scanning for nearby devices. As shown in Figure 17, the application successfully distinguishes between different BLE services, connecting to CityAirQ pollution tracker.

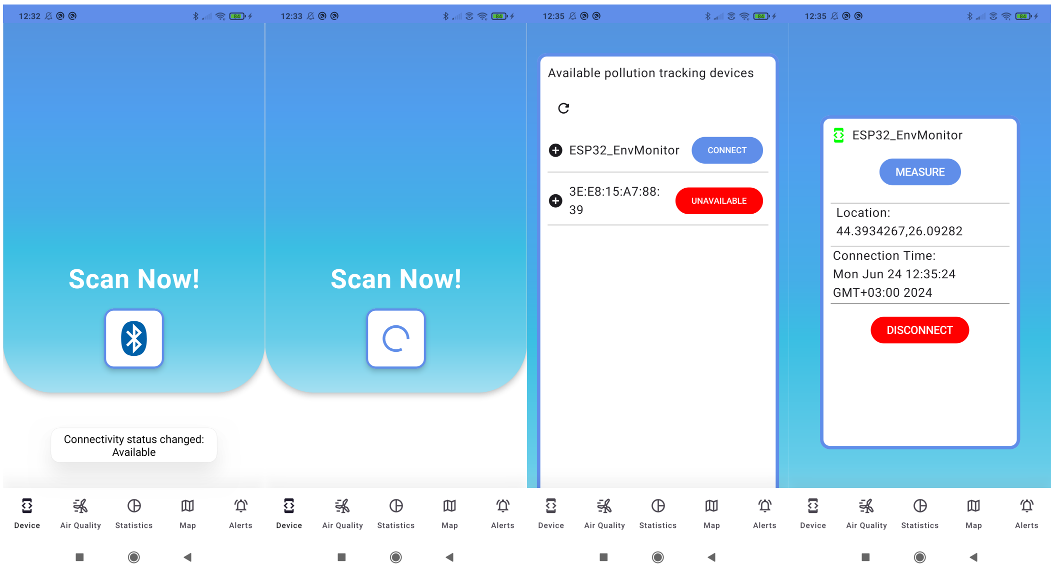

Figure 17. BLE Scanning and Profiles.

Once the BLE connection is validated, moving on to the next screen by pressing the 'MEASURE' button, Figure 18 provides a continuous live data flow, as air pollution parameters update in real-time in sync with the hardware device. Clicking on a parameter button triggers an audio description of the pollutant value, while the 'AUDIO' button activates Text-to-Speech for all parameters, useful while cycling.

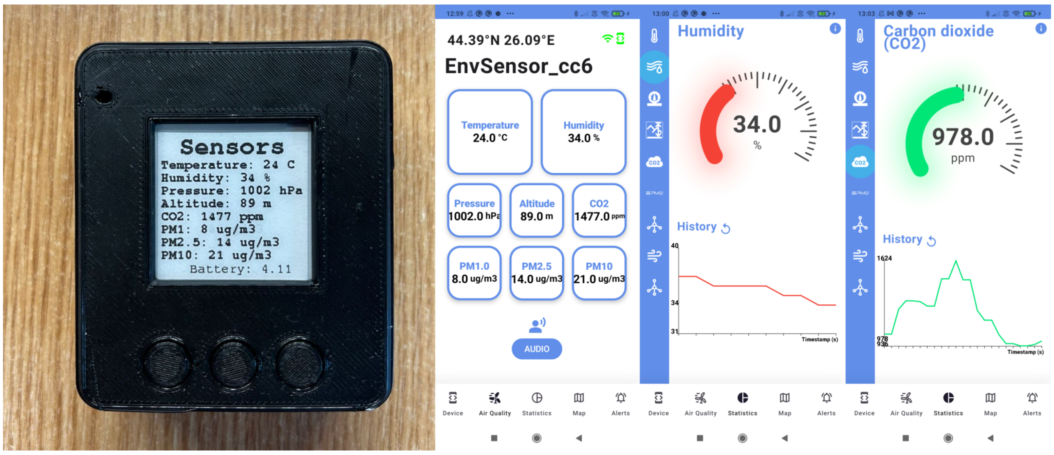

Figure 18. Data Transfer and Visualization.

The system remains resilient to changes, as disabling Wi-Fi updates the state while maintaining a persistent BLE connection. Live values are also reflected on visualization screens, alongside historical charts. Multiple pollutants can be selected for visualization, with updated values, distinct chart designs, additional information and getting push notifications or alerts based on severity (Figure 11).

Figure 19 shows an example of successful data collection by several pollution tracking prototypes, pursuing correctly the entire data path: from BLE server (hardware device) to client (mobile application) and, finally, transmission of air quality parameters to the database through the API endpoints, for ingestion and processing to begin.

client symbol	timestamp timestamp	latitude double	longitude double	dimension symbol	value double
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	Temperature	39
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	Humidity	22
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	Pressure	993
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	Altitude	167
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	CO2	410
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	PM1.0	9
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	PM2.5	12
clientId-ec85695bd6a1fdaa	2024-06-10T16:23:22.598000Z	44.436375	26.047916	PM10	15

Figure 19. Field Demo Data Collection.

The mapping system highlights responses from the data pipeline and involves pollution renderings that are very relevant to users. With the infrastructure up and running, the actual field test consists of activating the 'START' button on the map, which initiates real-time route generation, dynamically updating with each step in the location puck (as latitude and longitude play a very important role in the data ingestion). The dynamic map and routing, which can easily be reset by pressing 'STOP', keeps a persistent state across navigation and even screen off. Accessing the Expandable Menu and selecting the 'HMap' button enables visualization of pollution HeatMaps for various air quality parameters, as depicted in Figure 20.

The HeatMap structure and intensity vary based on pollutant levels, represented by color-coded points, with ranges explained on the upper right corner for the chosen parameter. Figure 21 displays pollution (AQI) levels in Bucharest, Romania, within a 10 km radius of the current location. As the user zooms in, the point opacity smoothly transitions, revealing distinct patterns and clustering based on various environmental factors.

Figure 20. Field Demo Data Collection.

Figure 21. HeatMap Pollution Layers.

When zooming in even more while walking around (Figure 20), the renderings become an overlay to the route. The mapping system is dynamic, updating on a timer with new values, as we can see on the right side of Figure 20.

By the time we have reached the end of the street, some nearby points have maintained a constant value, while others have either improved to healthier levels or worsened to more detrimental ones. Nevertheless, this serves as a pertinent example through which users can prioritize one route over another in real time, based solely on air quality parameters.

In the last phase, multiple practical evaluations were conducted in different areas of Bucharest, Romania, throughout several days, in order to collect as much data as possible, with varying values depending on the pollution and congestion levels of each particular zone. Figure 22 displays examples of different routes for data collection, ranging between 0.5 and 5.2 km, involving commuting zones, green areas and even the Politehnica University campus.

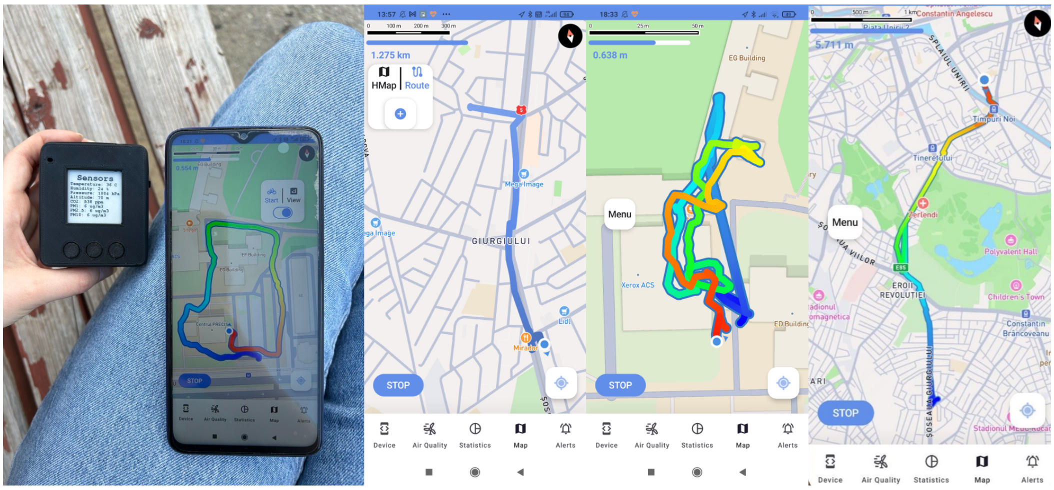

Figure 22. Routes for Data Collection in Bucharest, Romania.

7.4. Monitored Pollution Impact

While using the prototype to collect pollution data across various areas in Bucharest, an interesting observation was made: pedestrian exposure to pollution varies significantly based on sidewalk and road types. For instance, the sidewalk adjacent to our University directly borders the main avenue, separated only by a curb (right side in Figure 23); in contrast, the sidewalk along Drumul Taberei avenue (a busy neighborhood just 15 min away) features a buffer of over 5 m of green space with grass and trees before connecting to the main road (left side in Figure 23). This difference significantly impacted pollution levels: even under similar traffic conditions (moderate), pollution levels were more than 35% lower on the sidewalk with greater separation in regards to CO_2 levels, and more than three times lower in relation to PM levels, as Figure 24 reflects.

Figure 23. Two different sidewalks where pollution data was collected.

Several key environmental and physical factors contribute to these variations in pollutant levels:

- Proximity to Traffic Emissions:

  Vehicles are the primary source of CO_2 and particulate matter in urban areas.

  Sidewalks immediately adjacent to the roadway are directly exposed to exhaust emissions, tire wear particles, and resuspended road dust, leading to higher pollutant concentrations.
- Airflow and Dispersion:

  The absence of a buffer zone next to the university sidewalk limits air dispersion, trapping pollutants within the pedestrian zone.

  In contrast, the Drumul Taberei sidewalk, with its green buffer, allows for natural air dilution and pollutant dispersion, leading to significantly lower readings.
- Role of Green Space and Vegetation:

  Trees and grass act as natural air filters, capturing airborne PM and absorbing CO_2 through photosynthesis, as vegetation intercepts and deposits PM while also influencing local air circulation.
- Street Canyon Effect vs. Open Space:

  The university sidewalk is part of a denser urban setting, where buildings and traffic lanes form a "street canyon" effect, trapping pollutants at pedestrian level.

  The Drumul Taberei avenue features a more open layout, which facilitates natural air exchange and pollutant dispersion, preventing pollutant accumulation near pedestrians.

Figure 24. Pollution comparison between different sidewalks.

7.5. Comparative Evaluation

By integrating the three key components and several innovative elements described in Section 4.6, we have developed a compact, ready-to-use system that demonstrates superior cost efficiency compared to existing market alternatives. Table 3 illustrates its comparative positioning, while Table 4 depicts its advantages over the competitors. Additionally, our design prioritizes portability and ease of deployment, making it suitable for both individual users and community-driven air quality monitoring initiatives. The combination of low power consumption, real-time data accessibility, and accurate sensor calibration ensures reliable long-term operation. Furthermore, the modular nature of the system allows for future upgrades, enabling the integration of additional sensors and improved analytical capabilities.

Table 3. Summary of Products compared to CityAirQ

Product	Parameters	Device	Visualisation	Price Range
CityAirQ	temperature, humidity, pressure, altitude, C02, PM1.0, PM2.5, PM10	small, low power BLE embedded gadgets attached to bikes	Android App with live data, historical charts, dynamic maps and alerts	86$
Sodaq Air	humidity, pressure, PM1.0, PM2.5, PM10 (reported overestimation)	similar device with super capacitor, lower data bandwidth	recorded data sent to "Know Your AIR" global air quality platform for mapping	174€
Oizom Polludrone	temperature, humidity, pressure, PM1.0, PM2.5, PM10, PM100, CO_2, SO2, NO_2, O3, H2S, ambient noise and light (data calibration)	7.2 kg static, cutting edge device, weather resistant and solar powered	Separate Product (mobile and web) for real-time data, mapping and alerts	3.9–9.0k€
IQAir AirVisual Pro	temperature, humidity, pressure, altitude, C02, PM1.0, PM2.5, PM10, very reliable data	medium-sized device with LCD screen and Wi-Fi connectivity	#1 Rated Air Quality user-friendly App with real-time, historical, and forcasted air quality insights and alerts	319€
AirBeam3	temperature, humidity, PM1.0, PM2.5, PM10	palm-sized device with onboard GPS, SD card, can communicate via BLE, WiFi, or cellular 4G	BLE connection to AirCasting mobile app and web server, graphs and maps	249€
Atmotube Pro	temperature, humidity, pressure, PM1.0, PM2.5, PM10	similar to AirBeam3, but smaller in size and equipped with LEDs	connected to mobile app, renders graphs and maps	189€

Table 4. Advantages of CityAirQ over competing solutions.

Compared Solution	Advantages of CityAirQ
Sodaq Air	• Lower cost • More pollutants tracked • Real-time view on device • Dedicated mobile application
Oizom Polludrone	• Highly portable • Significantly lower cost • Designed for individual users rather than large-scale static monitoring
IQAir AirVisual Pro	• Smaller • Lighter • Lower cost • Optimized for mobile air quality monitoring on bikes rather than static indoor use

Table 4. *Cont.*

Compared Solution	Advantages of CityAirQ
AirBeam3	• Lower cost • BLE connectivity optimized for low-power operation • Better adaptability for cycling use cases
Atmotube Pro	• Lower cost • BLE-based real-time data transmission • Optimized form factor for wearable/mobile applications

7.6. Scientific Evaluation

Up until this point, we have conducted an evaluation of the system's performance and functionality. Moving forward, we are focusing on the scientific evaluation and the value it brings.

From a scientific research perspective, CityAirQ contributes by filling a significant gap in urban pollution monitoring, especially in Bucharest, which definitely lacks solutions to this imminent problem. By collecting data, we are creating a detailed database that tracks pollution levels across different areas and timestamps, helping to better understand air quality patterns in the city. Following the data collection through this functional system, we can analyze the gathered information.

In this context, it is essential to evaluate and validate the dataset to ensure the accuracy and reliability of the results. One way to achieve this is by minimizing errors in the data collection and dataset. Values that fall outside the acceptable range can be identified, allowing for the detection of hardware sensor errors (Section 4.7.1) or further filtered within the mobile application based on predefined ranges, presented in Table 1. The erroneous data points are discarded, therefore not transmitted to the backend for further processing. To assess and validate the positioning of our data, the stored values underwent a comparison with existing pollution data sourced from public APIs for Bucharest, such as Open-Meteo Weather API.

Figures 25–28 outline examples of scientific evaluation for a selection of air quality parameters such as PM2.5, PM10, pressure and CO_2. The figure shows a dataset collected on 6 June 2024, from the area situated at coordinates 44.43609° N, 26.04775° E and a radius of 1 km around our university campus in Bucharest. The data collectors walked around the nearby mentioned area, which includes both green spaces and regions with heavy traffic from the nearby road. Data was gathered constantly for about an hour in the afternoon, within 15:00 and 16:00 p.m., using two different CityAirQ devices, with client applications installed on mobile phones for continuous monitoring.

Due to the fact that the data was collected in a confined geographic area, around the same conditions and at the same time, one first important observation is that the recorded values are similar for both CityAirQ devices. Looking at the graphs, there is definitely a strong correlation between data points, which consistently remain within the predefined acceptable range for each air quality parameter due to the minimizing errors techniques both on the hardware and mobile sides. The points are most correlated for pressure and CO_2, as shown in Figures 27 and 28, while for PM2.5 and PM10, small differences can be observed, likely due to external influences or local variations in weather conditions. However, the general trend is a convergence of values, indicating that the system yields consistent results under identical conditions and with identical devices.

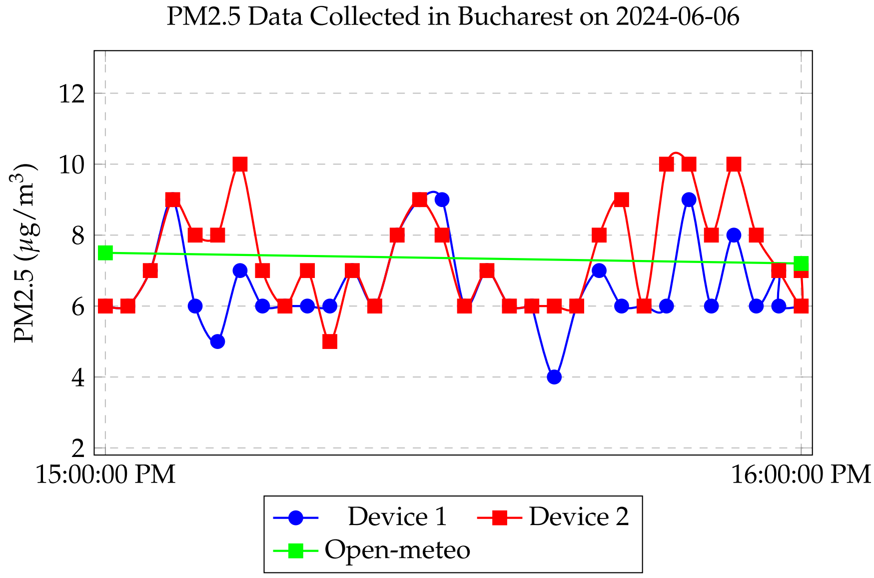

Figure 25. PM2.5 Data Evaluation.

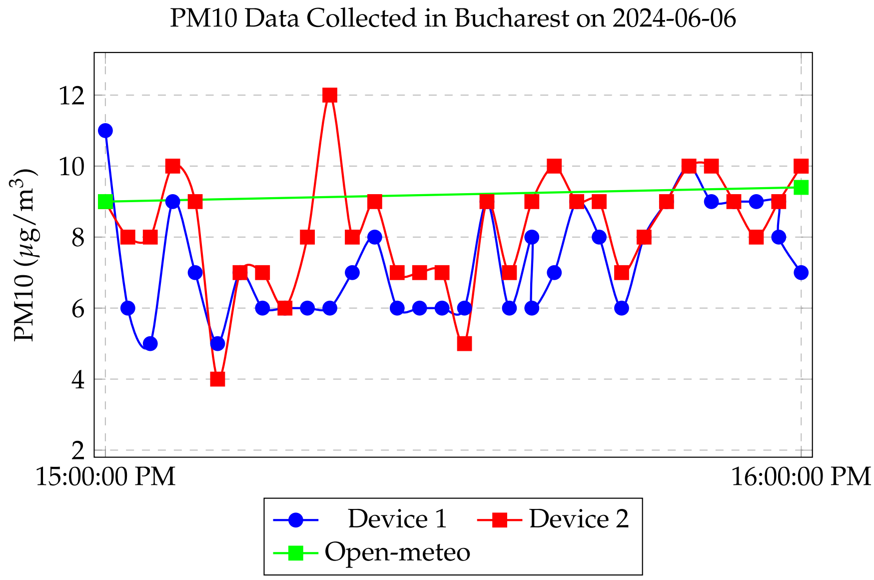

Figure 26. PM10 Data Evaluation.

Furthermore, to evaluate the dataset, pollution data was requested for the same indicators from Open-Meteo, from a station near the coordinates in the specified time interval. For this specific example, we were only able to receive historical reference values collected by Open-Meteo stations on June 6, 2024 for PM2.5, PM10, and pressure, which are displayed on Figures 25–27 with green.

As observed, especially for PM2.5 and PM10, the collected values approach the reference value, and the range is narrow, which validates the accuracy of the measurements and demonstrates that the dataset is reliable. Naturally, there may be slight measurement discrepancies, as seen in the case of pressure, due to the fact that the location is nearby but not exactly the same. Additionally, the detailed conditions of the measurements from Open-Meteo are not available for us to assess and as mentioned, Bucharest lacks pollution data providers for validation, particularly historical data.

Nevertheless, this specific example is part of a broader series of validations of the dataset, which demonstrates that we have collected real data with scientific value. This data will assist us in achieving many of our future goals. By further developing the system, we will enhance its capability to provide more accurate and reliable data, supporting continuous improvements in air quality monitoring and analysis.

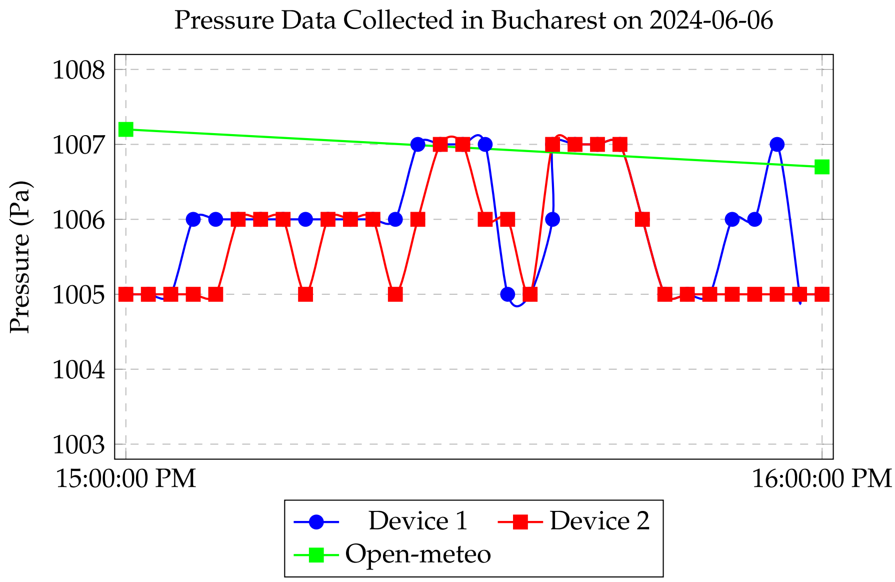

Figure 27. Pressure Data Evaluation.

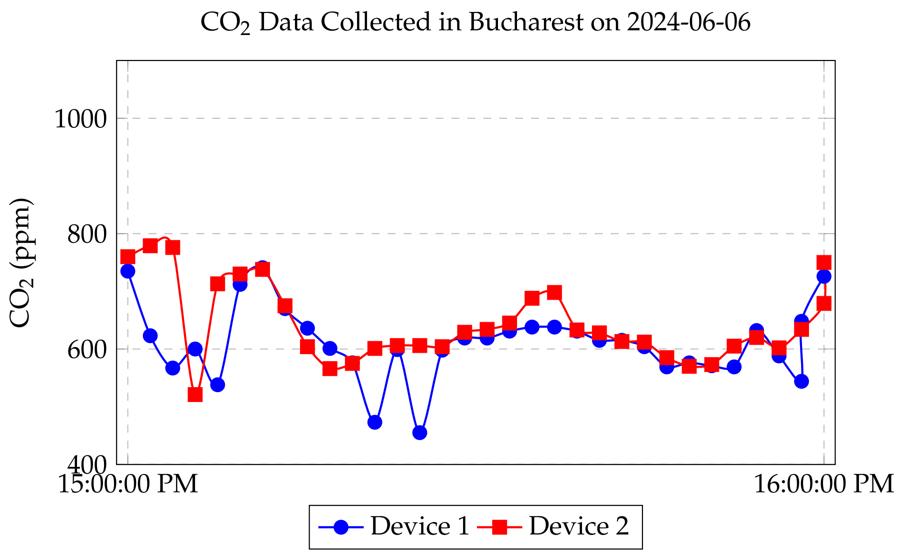

Figure 28. CO_2 Data Evaluation.

8. Conclusions and Further Work

CityAirQ is an air quality monitoring system that addresses the dangers of air pollution by raising public awareness and enabling informed decision-making, potentially leading to improved air quality and healthier communities.

8.1. Conclusions

On the whole, the CityAirQ project objectives were very well accomplished. The real and most important result was obtaining a valuable, deployable product that holds scientific value.

- Solving the Stated Problem: Based on results and user feedback, CityAirQ turns out to be a reliable system that fights air pollution through information. While collecting reliable data, cyclists have the possibility to be informed and alerted at all times, by visualizing the graphics, air pollutants and maps in the application and gives them the power to inform others, leading to a more cautious community.

- Generated a pollution dataset: By using the CityAirQ infrastructure, we managed to create a reliable pollution dataset. We collected by using more than 3 different hardware devices connected to mobile clients on different Android mobile phones. The main areas of interest were western, southern and north-central parts of Bucharest, Romania, on routes with distances raging from 0.5 to 7 km. Not only does this offer a relevant dataset for pollution renderings, but it also lays the foundation for future developments.
- CityAirQ as a Product: By combining the three main elements of the system together, we managed to create a reliable product, ready for use.
- Functionality: All wanted features work as expected and the communication stream successfully transports data from source to destination, with a minimal networking overhead.
- Performance: The performance metrics revealed by the hardware and software profilers are overall good and they prove that the system is low-power.
- Price: Compared to other competitors on the market, our device is by far the cheapest option, being 2 times less expensive than the next option in line. Most importantly, the application was fully created by using open source and public Android libraries, with no additional subscriptions needed, which makes CityAirQ overall low-cost compared to other products.

8.2. Technical Contributions

Key technical innovations include:

- Ultra-Compact Custom PCB Design: The development of a custom PCB that enables the integration of multiple pollution and environmental sensors into a compact, portable device, allowing for real-time tracking of a broad range of environmental parameters.
- Sensor Calibration and Data Validation: A hybrid calibration approach was applied to the CO_2 sensor, combining rich CO_2 mixture exposure, fresh air calibration, and auto-calibration to ensure long-term measurement stability. Additionally, data validation tests were performed by comparing readings against reference stations, confirming the accuracy and reliability of the collected pollution data
- Energy Efficiency and Portability: The system's energy-efficient design ensures that the tracking devices can operate for extended periods without frequent recharging or maintenance, making them practical for continuous monitoring in urban environments.
- Wearable and Bike-Mounted Flexibility: The compact enclosure was designed for versatile deployment, making CityAirQ suitable for both personal exposure monitoring and urban pollution mapping.
- Air Quality Dynamic Maps: The mobile application includes dynamic maps that transform raw sensor data into actionable insights. By combining real-time data integration, high-resolution visualization, cloud-based processing, and user-friendly design, these maps provide a powerful tool for understanding and addressing urban air pollution.

8.3. Practical Application Value

The CityAirQ system offers significant practical application value, directly addressing the critical need for real-time air quality monitoring in urban environments:

- Public Health Awareness and Decision-Making: CityAirQ provides real-time air quality data to citizens, enabling them to make informed decisions about their daily

activities, such as choosing less polluted routes. Especially individuals with respiratory and cardiovascular diseases can benefit from our system in reducing their health risks.
- Community Engagement and Advocacy: Local communities and environmental groups can use the system to collect air quality data, in order to raise awareness and advocate for cleaner air policies.
- Supporting Urban Planning and Policy: CityAirQ's detailed pollution maps allow officials to pinpoint pollution hotspots, prioritize interventions and evaluate the effectiveness of existing air quality policies.
- Enhancing Environmental Monitoring and Research: The system provides a high-resolution, real-time dataset that can be used by researchers to study the dynamics of urban air pollution.

8.4. Further Improvements

Naturally, even if CityAirQ seems to be working as expected, there is always room for improvement and future directions:

- BLE Improvements: Although it works at the moment, the BLE connection states are still sometimes unpredictable and can easily send the BLE Client to an unstable state, which is obviously dangerous to the system. The same can be said about the Bluetooth scanning logic, especially shortly after device disconnect. A more persistent and safe solution should be found in this direction.
- Memory Usage: Operations such as HeatMap renderings require a large volume of data, high CPU and can lead to higher memory values. As readings are necessary for data visualization, a more efficient way to store this big number of points needs to be studied, in order to avoid loading the application unnecessarily.
- New Functionalities: More exciting functionalities can further be incorporated. One interesting idea would be including sources of pollution in the map. Another take here would be comparing the pollution level on the map with number of smoked cigarettes per day, based on the study [30].
- Accessibility: One final improvement is to keep on making the application more suitable for usage while cycling, when the attention needs to be put into another direction. The charting system needs to offer more accessibility as well [20].
- Power consumption: While the device currently provides approximately 5 h of continuous battery operation, improvements are possible. Implementing a more power-efficient version of the CO_2 sensor, such as the MH-Z1311A, could reduce its average power consumption from 60 mA to 1 mA.
- Pollution prediction: To further enhance our system, we will integrate trained machine learning algorithms into our data pipeline, through the use of Apache Beam, in order to create high-resolution predictive pollution maps, at a much larger scale than is currently available.
- Archived raw data: Future implementation will include cloud standard object storage, via Google Cloud Storage, for the archiving of raw data. This data will then be leveraged for the refinement of our predictive models and algorithms.
- Backend: Future efforts will focus on achieving complete automation and seamless integration of the backend components.

9. Supplementary Information

9.1. Air Pollution

Emissions of smoke and pollutants such as Sulfur dioxide (SO_2), Nitrogen dioxide (NO_2), Hydrogen nitride (NH3) substantially increased due to coal ignition [2]. New findings [31] suggest that nowadays the situation has unfortunately gotten even worse, with Carbon dioxide (CO_2) emissions now 189 times higher than they were in 1850. It goes without saying that sources of pollution are varied, especially in the current industrial economy, where large-scale production comes at a significant environmental cost. Among the relevant causes of pollution are both human-made factors such as fuel oil consumption, emissions, agricultural runoff, power plants, and industrial manufacturing, as well as natural occurrences like volcanic eruptions and wildfires [9].

9.2. Air Pollution Monitoring

On a global scale, the most relevant unit measure for air quality is the Air Quality Index (AQI) [32]. The AQI transforms pollution as an ongoing phenomenon into a numerical, comparable quality value [32], in the range 0–500 [18]. It is a value that characterizes air pollution levels in a specific area at a particular moment in time. It is calculated by combining a predefined model with different historical and real-time air quality parameters, called pollutants. Examples of different pollutants measured by sensors include Particulate Matter (PM), Carbon Dioxide (CO_2), Ozone (O_3), Nitrogen Dioxide (NO_2), Sulfur Dioxide (SO_2), Carbon Monoxide (CO).

9.3. Other Monitoring Solutions

Oizom is a modern monitoring system provider that offers a high-tech and precise system primarily designed for industrial applications, lacking the mobility required to facilitate dynamic use cases such as integration with biking activities. Additionally, its visualization capabilities are limited to web pages, which may not align with the intended use case.

AirBeam3 and Atmotube Pro are both advanced air quality monitoring hardware devices which communicate with mobile applications via Bluetooth and Wi-Fi, where users can see AQI values and maps. Disadvantages include limited sensor diversity, limited parameter measurements, and a high price (189–249€).

9.4. Hardware Device

9.4.1. RTC Module Backup Power

Choosing the right backup power source for the RTC module (which only needs 40 nA for time-keeping) implied making a decision on whether to opt for a supercapacitor or a coin cell battery, each having advantages and disadvantages.

In order to assess the supercapacitor running time, the following formula was used:

$$\Delta T = C \cdot \frac{\Delta V}{I} \tag{1}$$

After thorough calculations, we found out that a 220 mF capacitor would theoretically support around 3 months of off-battery timekeeping, but if internal discharge current is taken into account (around 5 µA), the time estimate is reduced to only 1 day, making the method impractical. The coin cell battery is the preferred choice due to its significantly longer lifespan, despite its larger space consumption.

9.4.2. Section Analysis

Performing section analysis on the 3D design is the proper way to check the fitment and tolerances used in the modeling phase. A reference is provided in Figure 29. This confirms that the battery (bottom left) does not touch the PM sensor and the same applies to CO_2 with the SD card module. In addition, for the components mounted on the top side, there is enough clearance to the lid assembly to house the flexible screen connector.

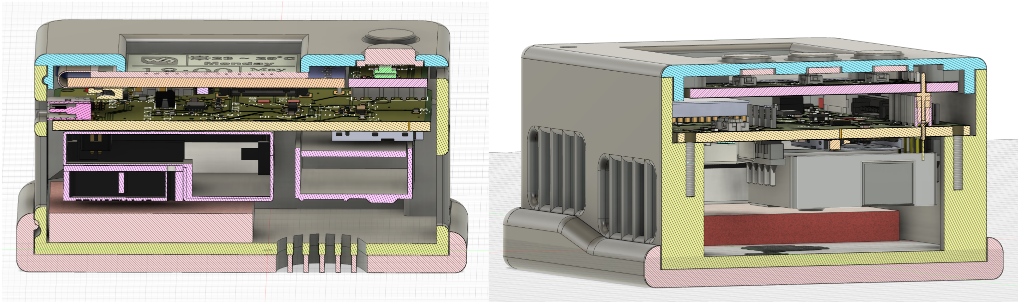

Figure 29. Section Analysis.

9.5. Application User Feedback

An additional evaluation method is through objective user feedback. Overall, the Google Forms Analytics [33] ratings are positive, particularly in terms of usefulness, user experience, and design. Regarding privacy, users are asked for permissions in a clear and detailed manner, with explanations, because fine and coarse location access (necessary for BLE and maps) can rather have a significant impact on data safety [34].

Figures 30 and 31 present the responses gathered from the survey participants, providing illustrative examples of the feedback. On short, the ratings were very good when it comes to performance and the most useful functionalities turned out to be the Real Time MeasureScreen, the MapScreen and the AirQualityScreen.

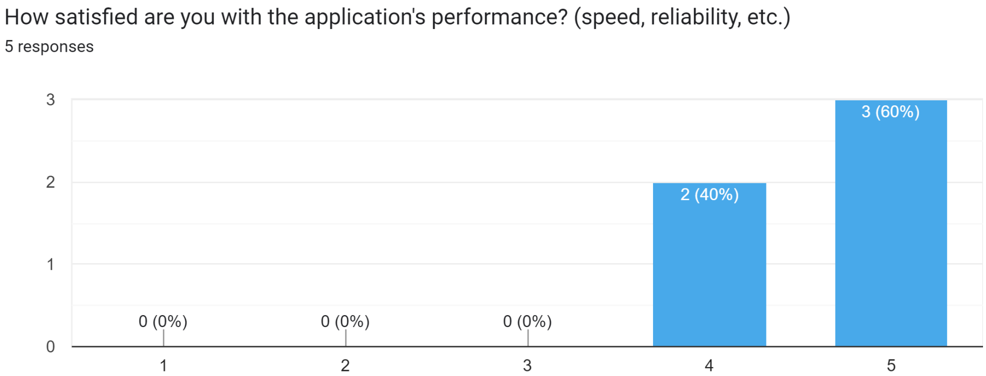

Figure 30. CityAirQ Feedback (1).

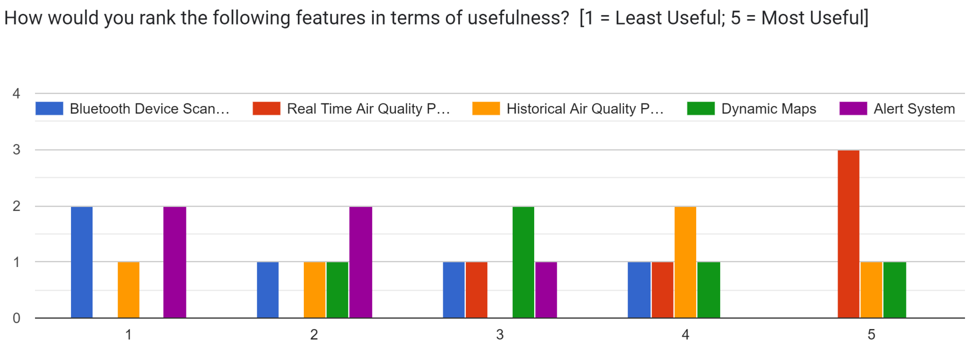

Figure 31. CityAirQ Feedback (2).

Author Contributions: Conceptualization, M.D., D.P., D.T., L.R. and A.P.; software, M.D., D.P., B.D. and M.P.; validation, M.D., D.P., B.D. and M.P.; writing—original draft preparation, M.D., D.P., B.D., L.R. and M.P.; writing—review and editing, M.D., D.P. and L.R.; supervision, D.T., L.R. and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: Author Abhinuv Pitale was employed by the company Google. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Walters, R., Air Pollution and Invisible Violence. In *Invisible Crimes and Social Harms*; Davies, P., Francis, P., Wyatt, T., Eds.; Palgrave Macmillan UK: London, UK, 2014; pp. 142–160.
2. Fowler, D.; Brimblecombe, P.; Burrows, J.; Heal, M.R.; Grennfelt, P.; Stevenson, D.S.; Jowett, A.; Nemitz, E.; Coyle, M.; Liu, X.; et al. A chronology of global air quality. *Philos. Trans. A Math. Phys. Eng. Sci.* **2020**, *378*, 20190314.
3. Xu, X.; Zhong, Y.; Cai, S.; Lei, L.; Peng, J. Does Air Pollution Aggravate Health Problems in Low-Income Countries? Verification from Countries Along the Belt and Road. *Sustainability* **2025**, *17*, 1796. [CrossRef]
4. Abadi, A.R.S.; Hamzeh, N.H.; Kaskaoutis, D.G.; Vuillaume, J.F.; Shukurov, K.A.; Gharibzadeh, M. Spatio-Temporal Distribution of PM2.5 and PM10 Concentrations and Assessment of Public Health Risk in the Three Most Polluted Provinces of Iran. *Sustainability* **2025**, *17*, 44. [CrossRef]
5. Liu, L.; Zhao, J.; Li, Z. Sustainable Urban Development: A Comprehensive Model for Assessing Health Risks from Compounded Pollution in Xi'an. *Sustainability* **2024**, *16*, 9799. [CrossRef]
6. Manisalidis, I.; Stavropoulou, E.; Stavropoulos, A.; Bezirtzoglou, E. Environmental and Health Impacts of Air Pollution: A Review. *Front. Public Health* **2020**, *8*, 13. [CrossRef] [PubMed]
7. Li, Q.; Zheng, D.; Wang, Y.; Li, R.; Wu, H.; Xu, S.; Kang, Y.; Cao, Y.; Chen, X.; Zhu, Y.; et al. Association between exposure to airborne particulate matter less than 2.5 μm and human fecundity in CHINA. *Environ. Int.* **2021**, *146*, 106231. [CrossRef] [PubMed]
8. Gauderman, W.; Avol, E.; Lurmann, F.; Kuenzli, N.; Gilliland, F.; Peters, J.; McConnell, R. Childhood asthma and exposure to traffic and nitrogen dioxide. *Epidemiology* **2005**, *16*, 737–743. [CrossRef] [PubMed]
9. National Institute of Environmental Health Sciences. Air Pollution and Your Health. Available online: https://www.niehs.nih.gov/health/topics/agents/air-pollution (accessed on 21 February 2025).
10. Bogdan, E. Promoting the Health and Social Inclusion of the Urban Population by endowing and improving the Quality of Leisure Facilities in the Parks in BUCHAREST. *Qual. Access Success* **2018**, *19*, 80–86.
11. Rosen, Z.; Channing, B.; Scott, S. Visualizing the invisible: Visual-based design and efficacy in air quality messaging. *Int. J. Environ. Res. Public Health* **2021**, *18*, 18. [CrossRef] [PubMed]
12. Tonne, C.; Adair, L.; Adlakha, D.; Anguelovski, I.; Belesova, K.; Berger, M.; Brelsford, C.; Dadvand, P.; Dimitrova, A.; Giles-Corti, B.; et al. Defining pathways to healthy sustainable urban development. *Environ. Int.* **2021**, *146*, 106236. [CrossRef] [PubMed]
13. Levy Zamora, M.; Xiong, F.; Gentner, D.; Kerkez, B.; Kohrman-Glaser, J.; Koehler, K. Field and Laboratory Evaluations of the Low-Cost Plantower Particulate Matter Sensor. *Environ. Sci. Technol.* **2018**, *53*, 838–849.
14. Particulate Matter (PM) Sensors of Two Different Principles. Available online: https://www.winsen-sensor.com/knowledge/particulate-matter-sensors-of-two-different-principles.html (accessed on 21 February 2025).
15. Kazdaridis, G.; Sidiropoulos, N.; Zografopoulos, I.; Symeonidis, P.; Korakis, T. Nano-things: Pushing sleep current consumption to the limits in IoT platforms. In Proceedings of the 10th International Conference on the Internet of Things, Malmö, Sweden, 6–9 October 2020. [CrossRef]
16. Carlo, Z.; Carraro, M.; Marchi, M.; Pappalardo, S. Volunteered Geographic Information for Mapping Urban Climate and AIR Quality: Testing and Assessing 'Sniffer Bikes' with Low-Cost Sensors. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 555–561. [CrossRef]
17. Random Nerd Tutorials. ESP32 Web Bluetooth (BLE): Getting Started Guide. Available online: https://randomnerdtutorials.com/esp32-web-bluetooth/ (accessed on 21 February 2025).
18. Yadav, S.; Jain, M. Variation in concentrations of particulate matter with various sizes in different weather conditions in mining zone. *Int. J. Environ. Sci. Technol.* **2020**, *17*, 695–708. [CrossRef]
19. Zusman, M.; Schumacher, C.S.; Gassett, A.J.; Spalt, E.W.; Austin, E.; Larson, T.V.; Carvlin, G.; Seto, E.; Kaufman, J.D.; Sheppard, L. Calibration of low-cost particulate matter sensors: Model development for a multi-city epidemiological study. *Environ. Int.* **2020**, *134*, 105329. [CrossRef] [PubMed]
20. Joshi, I.A. Unblind the charts: Towards Making Interactive Charts Accessible in Android Applications. *arXiv* **2021**, arXiv:2109.12442.
21. Daniel Lozano Valdés. iOS Architecture: MVVM-C, Introduction. Available online: https://medium.com/sudo-by-icalia-labs/ios-architecture-mvvm-c-introduction-1-6-815204248518 (accessed on 21 February 2025).

22. Lou, T. A Comparison of Android Native App Architecture—MVC, MVP and MVVM. Master's Thesis, Aalto University, Espoo, Finland, 2016.
23. Random Nerd Tutorials. ESP32 BLE Peripheral (Server): Environmental Sensing Service. Available online: https://randomnerdtutorials.com/esp32-ble-server-environmental-sensing-service/#demonstration (accessed on 21 February 2025).
24. Bluetooth SIG. Bluetooth Document Assigned Numbers. 2024. Available online: https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Assigned_Numbers/out/en/Assigned_Numbers.pdf?v=1718639667001 (accessed on 21 February 2025).
25. Stuart Kent. Android Bluetooth Low Energy (BLE) Resources, 2018–2024. Available online: https://gist.github.com/stkent/a7f0d6b868e805da326b112d60a9f59b?permalink_comment_id=3592424 (accessed on 21 February 2025).
26. Howarth, J. iPHONE vs ANDROID USER STATS (2024 Data). 2024. Available online: https://explodingtopics.com/blog/iphone-android-users (accessed on 21 February 2025).
27. Noori, Z.; Eriksson, C. UI Performance Comparison of Jetpack Compose and XML in Native Android Applications. Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2023.
28. Android Developers. Navigation. 2024. Available online: https://developer.android.com/guide/navigation (accessed on 21 February 2025).
29. Mapbox. Mapbox Maps Compose Extension. 2024. Available online: https://github.com/mapbox/mapbox-maps-android/blob/main/extension-compose/README.md (accessed on 21 February 2025).
30. Korunoski, M.; Stojkoska, B.R.; Trivodaliev, K. Internet of things solution for intelligent air pollution prediction and visualization. In Proceedings of the IEEE EUROCON 2019-18th International Conference on Smart Technologies, Novi Sad, Serbia, 1–4 July 2019; pp. 1–6.
31. Leandro Vigna, J.F.; Damassa, T. The History of Carbon Dioxide Emissions. 2024. Available online: https://www.wri.org/insights/history-carbon-dioxide-emissions (accessed on 21 February 2025).
32. Janarthanan, R.; Partheeban, P.; Somasundaram, K.; Elamparithi, P.N. A deep learning approach for prediction of air quality index in a metropolitan city. *Sustain. Cities Soc.* **2021**, *67*, 102720. [CrossRef]
33. CityAirQ. CityAirQ Application Feedback. 2024. Available online: https://docs.google.com/forms/d/e/1FAIpQLSdpK44fr4BuOazFPr6ApJPl3RNfPHbSVvF_xr2Os_7TjzHZyw/viewform?usp=sharing (accessed on 21 February 2025).
34. Khatoon, A.; Corcoran, P. Android permission system and user privacy—A review of concept and approaches. In Proceedings of the 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 3–6 September 2017; pp. 153–158.