SUP'COM

المدرسة العليا للمواصلات بتونس

Ecole Supérieure des Communications de Tunis

Higher School of Communications of Tunis

Scope Statement
AIR-AWARE: Indoor Air Quality Intelligent Surveillance System

Realised by :

Fatma Abid & Ichrak Aouadni

Ahmed Guermazi & Adem Bakey

Faculty Advisor :

Mohamed Becha Kaaniche

2025/2026

Contents

1 Concept

1.1 Project Context

Indoor air quality (IAQ) has become a growing public health concern as individuals spend nearly 90% of their time in enclosed environments such as homes, offices, and educational institutions. Numerous studies conducted by the World Health Organization (WHO) and environmental agencies have highlighted that the concentration of air pollutants indoors is often two to five times higher than outdoors. These pollutants—originating from building materials, cleaning products, combustion processes, and human activity—can have serious effects on health, leading to respiratory, cardiovascular, and allergic diseases.

In addition to health implications, poor indoor air quality negatively impacts comfort, productivity, and overall well-being. Conventional air quality monitoring solutions are typically expensive, limited in functionality, or unable to provide real-time, data-driven insights.

In this context, the AIR-AWARE project proposes the design and implementation of an Intelligent Indoor Air Quality Surveillance System that combines IoT technologies, artificial intelligence, and cloud-based analytics to monitor, analyze, and predict variations in indoor air quality. This system aims to empower individuals, institutions, and organizations with real-time information and actionable recommendations to maintain healthier indoor environments.

1.2 Problem Statement

Despite the increasing awareness of environmental health, indoor air pollution remains an underestimated threat. Many residential, educational, and professional spaces suffer from high levels of airborne pollutants such as fine particulate matter ($PM_{2.5}/PM_{10}$), carbon dioxide (CO_2), and volatile organic compounds (VOCs), as well as unfavorable temperature and humidity conditions. These factors collectively degrade indoor air quality and directly contribute to a range of health issues, including asthma, allergies, respiratory irritation, and fatigue.

Existing monitoring solutions are often costly, technically complex, or limited in scope, making them inaccessible to the general public. Moreover, most current systems are reactive rather than proactive, offering basic data collection without intelligent analysis or predictive capabilities. This lack of automation prevents users from receiving early warnings, adaptive recommendations, or long-term insights about air quality trends.

There is therefore a critical need for an integrated, intelligent, and affordable system capable of continuously monitoring indoor air quality, detecting anomalies in real time, and providing personalized, data-driven recommendations to improve environmental health. The AIR-AWARE system is designed to fill this gap by combining IoT-based sensing, secure cloud connectivity, and AI-powered analytics within a user-friendly platform.

1.3 Objectives

1.3.1 Principal Objective

The main objective of the AIR-AWARE project is to design and develop an intelligent ecosystem for real-time monitoring, analysis, and improvement of indoor air quality. This ecosystem integrates IoT sensors, cloud-based data management, and artificial intelligence models to ensure continuous surveillance, predictive analytics, and actionable recommendations aimed at maintaining healthy and safe indoor environments.

1.3.2 Specific Objectives

To achieve this overarching goal, the project defines the following specific objectives:

1. **Continuous Measurement:** Implement a network of IoT sensors capable of accurately measuring key indoor air parameters such as particulate matter ($PM_{2.5}/PM_{10}$), carbon dioxide (CO_2), volatile organic compounds (VOCs), temperature, and humidity.

2. **Real-Time Analysis and Alerts:** Develop a data processing pipeline that performs real-time validation, anomaly detection, and triggers instant alerts when pollutant levels exceed predefined thresholds.

3. **Predictive Modeling:** Employ machine learning algorithms (e.g., ARIMA, LSTM, Random Forest) to forecast short-term air quality variations and identify long-term pollution patterns.

4. **User-Centric Visualization:** Design a Progressive Web Application (PWA) providing intuitive dashboards, historical data visualization, and personalized recommendations based on user preferences and environmental context.

5. **Data Management and Scalability:** Implement a secure cloud backend using a scalable NoSQL database (MongoDB) to handle large volumes of sensor data, ensuring reliability, performance, and fault tolerance.

6. **Location Tracking via LBS:** Integrate Location-Based Services (LBS) to enable dynamic mapping of indoor air quality data, geolocation of devices, and contextual analysis across multiple monitored areas, facilitating spatial correlation between environment and air quality metrics.

7. **Automation and Maintenance:** Introduce automatic sensor calibration, drift detection, and self-diagnosis mechanisms to maintain data accuracy and reduce manual intervention.

8. **Sustainability and Accessibility:** Deliver a cost-effective and energy-efficient solution accessible to both residential and professional users, contributing to healthier indoor spaces and sustainable smart building environments.

2 Scope and Target Markets

The AIR-AWARE project falls within the domain of intelligent indoor air quality monitoring systems, combining Internet of Things (IoT) technologies, Artificial Intelligence (AI), and Location-Based Services (LBS) to provide a comprehensive, connected, and predictive solution.

The project scope covers the entire cycle of data collection, processing, visualization, and analysis of environmental parameters obtained from sensors deployed in various indoor environments.

The system targets residential, professional, educational, and industrial spaces where air quality directly impacts health, comfort, and productivity. Thanks to its modular and scalable architecture, AIR-AWARE can be deployed in both small domestic areas and large multi-site infrastructures.

2.1 Primary Markets

The primary markets include sectors where indoor air quality monitoring is a critical concern for public health, comfort, and operational efficiency:

2.1.1 Residential Sector

Homes, apartments, and residential buildings requiring continuous air quality monitoring. Families, elderly individuals, or people with respiratory conditions seeking personalized, real-time insights through a mobile interface.

2.1.2 Corporate and Office Environments

Offices, coworking spaces, and administrative buildings where air quality directly affects employee productivity and well-being. Companies committed to sustainability and workplace health initiatives.

2.1.3 Educational Institutions

Classrooms, laboratories, and lecture halls requiring optimal air conditions to ensure a healthy and productive learning environment.

2.1.4 Healthcare and Medical Facilities

Hospitals, clinics, laboratories, and retirement homes where maintaining clean air is essential for infection control and patient health.

2.2 Secondary Markets

The secondary markets represent areas where the system can be adapted or extended to deliver additional value beyond its core applications:

2.2.1 Industrial and Logistics Sector

Factories, warehouses, and production zones requiring precise monitoring of gas emissions, dust, and chemical pollutants. Integration with energy management or industrial safety systems to ensure compliance and operational efficiency.

2.2.2 Hospitality and Commercial Sector

Hotels, restaurants, shopping malls, and fitness centers where air quality directly influences customer satisfaction and experience.

2.2.3 Public Institutions and Municipalities

City halls, public schools, libraries, and cultural centers aiming to deploy large-scale environmental monitoring networks for community well-being.

2.2.4 Research and Innovation Sector

Universities and research laboratories focusing on environmental modeling, urban health, and smart cities, using AIR-AWARE as a research and experimentation platform.

3 Functional Needs

The AIR-AWARE system is designed to ensure real-time monitoring, analysis, and control of indoor air quality through an intelligent, connected network of IoT devices and data-driven software components.

The functional needs are organized into three main categories: data acquisition, data processing and analysis, and user interaction and alerting.

3.1 Data Acquisition and Sensing Layer

This layer is responsible for collecting environmental data through multiple sensors integrated into the AIR-AWARE device.

Key functionalities:

- Continuous measurement of air quality parameters, including:
    - Temperature
    - Humidity
    - Carbon Dioxide (CO_2) concentration
    - Carbon Monoxide (CO) concentration
    - Volatile Organic Compounds (VOCs)
    - Particulate Matter ($PM_{2.5}$ / PM_{10})

- Automatic calibration and synchronization of sensors to ensure data accuracy and reliability.
- Data collection at configurable time intervals, depending on the use case (e.g., every 30 seconds or every 5 minutes).
- Integration with Location-Based Services (LBS) for dynamic tracking of sensor positions in large buildings or multi-zone installations.

3.2 Data Processing and Intelligence Layer

This layer performs data aggregation, cleaning, and analysis using intelligent algorithms to interpret environmental conditions.

Key functionalities:

- Local preprocessing on the edge device (microcontroller or Raspberry Pi) to reduce latency and bandwidth use.
- Centralized data storage in the cloud for long-term historical analysis.
- Application of AI and Machine Learning models to:
  - Detect anomalies and identify pollution peaks.
  - Predict air quality degradation trends.
  - Recommend corrective actions such as ventilation or purification.
- Real-time alert generation based on threshold violations or predictive risk patterns.
- Automatic synchronization between the edge and cloud components to maintain data consistency.

3.3 User Interaction and Visualization Layer

This layer ensures that users can easily access, understand, and act upon air quality data through an intuitive interface.

Key functionalities:

- A PWA displaying:
  - Real-time sensor readings.
  - Historical graphs and trend visualizations.
  - Air quality index (AQI) indicators with color-coded severity levels.
- Push notifications and SMS/email alerts when critical thresholds are exceeded.
- Location-based visualization, showing air quality variations across different rooms or buildings.

- Personalized user profiles allowing threshold customization and alert preferences.
- Administrative access for system management, device registration, and user control.

3.4 Communication and Integration Layer

To ensure a seamless data flow, the AIR-AWARE system integrates multiple communication protocols.

Key functionalities:

- Support for Wi-Fi, Bluetooth Low Energy (BLE), and LoRaWAN for long-range communication.
- Secure MQTT or HTTP-based APIs for transmitting data between devices and the cloud.
- Compatibility with third-party platforms (e.g., Google Cloud IoT, AWS IoT Core) for scalability and interoperability.

3.5 Maintenance and System Reliability

Ensuring reliability and continuity of service is a key functional requirement.

Key functionalities:

- Self-diagnostic procedures to detect sensor malfunction or data loss.
- Over-the-air (OTA) firmware updates for easy maintenance.
- Encrypted data transmission and user authentication to guarantee security and privacy.
- Redundancy mechanisms and backup storage to prevent data loss.

4 Non-Functional Requirements

The AIR-AWARE system must not only fulfill its functional objectives but also ensure efficiency, reliability, security, and user satisfaction across various operational contexts.

These non-functional requirements define the qualitative attributes that guarantee the system's overall performance, robustness, and usability.

4.1 Performance and Efficiency

- The system must ensure real-time data collection and processing, with a maximum latency of less than 2 seconds between sensor measurement and cloud synchronization.

- The sensors should operate efficiently, consuming minimal power to support long-term deployment (battery life exceeding 6 months for low-power devices).
- The data processing and AI modules must handle large data volumes (up to thousands of sensor inputs simultaneously) without degradation in performance.
- The dashboard must refresh and display updated values instantly for smooth user experience.

4.2 Reliability and Availability

- The system must maintain an availability rate of at least 99% to ensure continuous monitoring.
- In case of network or sensor failure, the device should store data locally and automatically resynchronize once connectivity is restored.
- Each hardware component should undergo periodic self-testing to detect malfunction or calibration drift.
- The cloud infrastructure must support redundant backups to prevent data loss.

4.3 Security and Privacy

- All communications between sensors, gateways, and servers must use end-to-end encryption (e.g., TLS 1.3).
- User credentials and sensitive data must be hashed and securely stored in compliance with best practices.
- Access to administrative features must be restricted to authenticated and authorized users only.
- The system must comply with GDPR and relevant data protection regulations for user privacy.
- Location-Based Services (LBS) data should be anonymized to prevent unauthorized tracking or misuse.

4.4 Scalability and Extensibility

- The system architecture should be modular and scalable, capable of integrating new sensors, communication protocols, or analytics modules without major redesign.
- The platform must support horizontal scalability, allowing the addition of new nodes or devices to the network dynamically.
- APIs and microservices should be loosely coupled to facilitate integration with external platforms (e.g., smart buildings or environmental control systems).

4.5 Maintainability and Upgradability

- The system should allow remote updates (Over-the-Air firmware) for sensors and gateways.
- The software should follow clean architecture principles to facilitate code maintenance and feature evolution.
- Comprehensive logging and diagnostic mechanisms must be included to simplify troubleshooting and performance monitoring.
- The platform should be documented with clear technical manuals and user guides for both administrators and end users.

4.6 Usability and Accessibility

- The interface must be intuitive, responsive, and user-friendly, designed with accessibility standards (e.g., WCAG 2.1).
- Users should be able to configure thresholds, alerts, and visualizations with minimal technical knowledge.
- Mobile and desktop versions must provide consistent user experiences.
- Use of color coding, icons, and graphs to simplify air quality interpretation.
- Support for multi-language options to widen accessibility for international users.

4.7 Interoperability

- AIR-AWARE must support standard IoT communication protocols (MQTT, CoAP, HTTP, LoRaWAN).
- The system should be interoperable with third-party services (e.g., Google Cloud, AWS IoT, or municipal smart city platforms).
- Data exports should be available in standard formats (JSON, CSV) for external analysis or reporting.

5 Machine Learning Integration

AIR-AWARE leverages machine learning (ML) techniques to enhance air quality monitoring, provide predictive insights, and trigger proactive alerts based on environmental conditions.
The ML integration ensures that the system can learn from sensor data, adapt to changing environments, and deliver accurate predictions for potential air quality hazards.

5.1 Model Development

The objective of model development is to build an ML model capable of detecting air quality anomalies and predicting potential health risks.

Key features include:

- **Input variables:** CO_2 levels, VOC concentrations, particulate matter ($PM_{2.5}$ / PM_{10}), temperature, humidity, and location-based data.
- **Predictive output:** Anomaly detection score or air quality index (AQI) indicating the likelihood of unsafe conditions.
- **Algorithm choice:** Ensemble methods, regression models, or neural networks optimized for real-time inference.
- **Training data:** Historical sensor readings, environmental data, and labeled events (e.g., detected air hazards).

The model continuously learns patterns from new sensor data, improving accuracy over time while minimizing false alarms.

5.2 MLOps Implementation

MLOps practices ensure that the ML model is robust, reliable, and continuously updated in production.

Key components of the MLOps framework include:

- **Automated Training and Deployment:** New sensor data is periodically used to retrain and redeploy models automatically.
- **Continuous Monitoring:** Track model performance metrics (accuracy, recall, false positives) to detect drift and maintain prediction quality.
- **Alerts and Notifications:** When the model detects potential air quality hazards, it triggers real-time alerts to users.
- **Data Versioning and Management:** Sensor data and model versions are version-controlled for reproducibility and historical analysis.
- **Security and Compliance:** Ensure secure handling of environmental and user data, maintaining privacy standards.
- **Performance Optimization:** Hyperparameter tuning and resource management ensure efficient and accurate predictions.
- **Feedback Loop:** User feedback is incorporated to refine predictions and improve the system's relevance in different environments.

The MLOps implementation is critical for sustaining reliable, high-quality predictions, adapting to environmental changes, and continuously improving user safety.

6 Equipment

To ensure precise air quality monitoring and real-time alerting, the following components will be used in AIR-AWARE:

6.1 Raspberry Pi 4 Model B

A versatile single-board computer with multiple GPIO pins, supporting Python, Java, and C/C++. Built-in Wi-Fi and Bluetooth make it ideal for IoT projects. It provides sufficient processing power for real-time sensor data collection, processing, and cloud communication.

6.2 BME680 Environmental Sensor

Measures temperature, humidity, pressure, and indoor air quality (IAQ) by detecting volatile organic compounds (VOCs). Communicates via I2C or SPI and is suitable for monitoring environmental conditions that may affect air quality.

6.3 SGP30 Gas Sensor

Detects indoor air quality, including CO_2 and total VOCs. Provides additional data to identify potential air hazards, enhancing anomaly detection in combination with other sensors.

6.4 Particulate Matter Sensor (PMS5003 / SDS011)

Measures $PM_{2.5}$ and PM_{10} particles in the air. Critical for monitoring fine particulate pollution and assessing health risks.

6.5 CO_2 Sensor (MH-Z19 or equivalent)

Monitors carbon dioxide concentrations in indoor environments, providing early warning for poor air quality conditions.

6.6 Cables and Connectors

High-quality wiring to ensure stable communication between sensors and the Raspberry Pi, reducing noise and data loss.

6.7 Power Supply and Housing

Reliable power adapters and protective enclosures to maintain system stability and durability, ensuring continuous operation in different environments.

These components collectively form the core hardware infrastructure of AIR-AWARE, enabling precise environmental monitoring, seamless IoT integration, and accurate real-time data analysis.

7 Technologies Choice

7.1 Back-end

- **MongoDB:** A NoSQL document-oriented database for storing sensor readings, user profiles, and historical air quality data. Offers flexibility, scalability, and easy integration with IoT devices.

- **MQTT:** Lightweight publish-subscribe protocol for transmitting real-time sensor data from AIR-AWARE devices to the cloud server efficiently and reliably.

7.2 Middleware

- **JAX-RS:** Java API for RESTful web services, enabling secure and scalable APIs to fetch or push air quality data to applications.

- **WildFly:** Java EE application server for hosting middleware services, providing high reliability, load balancing, and enterprise-grade performance.

- **Mosquitto:** MQTT broker facilitating communication between IoT sensors and back-end systems.

7.3 Front-End

- **Progressive Web App (PWA):** A web-based interface for users to monitor real-time air quality, receive alerts, and access historical data from any device without installing a native app.

- **Interactive Dashboards:** Visualizes sensor readings, air quality indices, and trends over time for better decision-making.

7.4 IoT Integration

- **Flogo:** Lightweight, event-driven framework to orchestrate data flows, preprocess sensor data, and automate alerts in IoT environments.

- **ESP32/Raspberry Pi Integration:** Connects sensors to the IoT network, enabling real-time data collection, processing, and cloud synchronization.

7.5 MLOps Integration

- **CI/CD Pipeline (Jenkins):** Automates the deployment and updates of machine learning models for predicting air quality hazards.

- **Testing Frameworks (PyTest, Selenium):** Ensures reliable model performance and accurate alert generation.

- **Monitoring (Prometheus & Grafana):** Tracks system health, model predictions, and sensor anomalies.

- **Containerization (Docker):** Facilitates consistent deployment across multiple devices and servers.
- **Stream Processing (Apache Kafka):** Handles real-time sensor data streams for timely alerts and analytics.
- **Security (HashiCorp Vault):** Protects user data and sensor information with encryption and access control.

8 Architecture

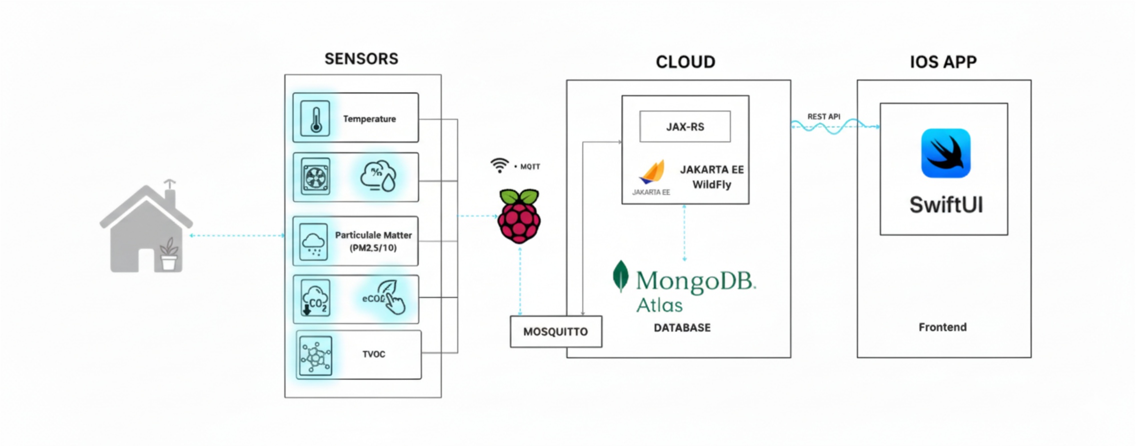


Figure 1: Architecture

The diagram above describes the main architecture of the Aire-Aware monitoring system which is mainly composed by Backend, Middleware and Frontend.

9 Timeline and Tasks (8 Weeks)

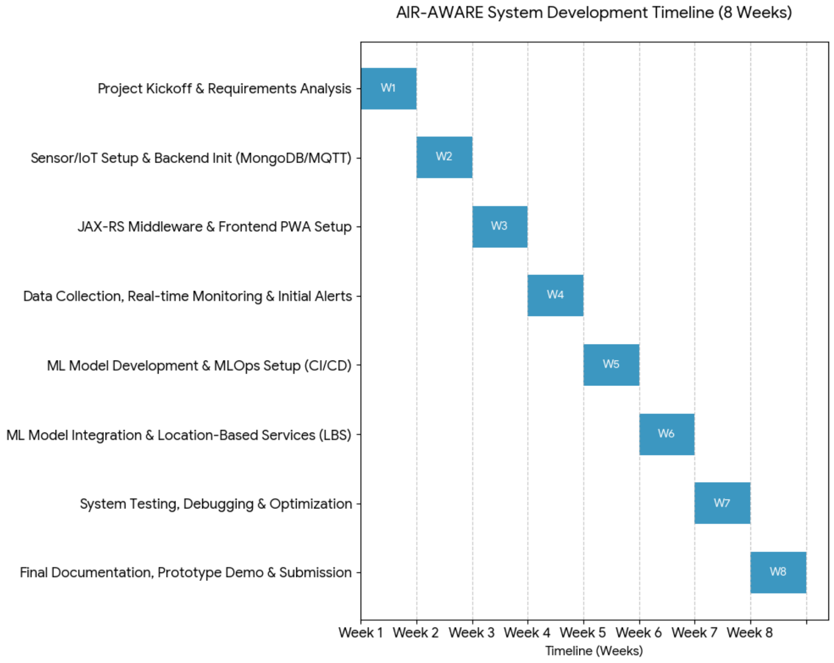


Figure 2: Timeline and Tasks

Week	Tasks
1	Project kickoff, requirement analysis, and defining user stories. Research existing air quality monitoring systems and finalize hardware and software components.
2	Sensor setup and initial IoT network configuration (ESP32/Raspberry Pi). Start backend setup (MongoDB & MQTT broker).
3	Develop middleware services using JAX-RS and integrate MQTT communication. Begin basic front-end PWA setup.
4	Implement data collection from sensors, real-time monitoring, and storage. Develop initial alert notification system.
5	Machine learning model development for air quality prediction. Begin MLOps setup including CI/CD pipeline, testing, and monitoring tools.
6	Integrate ML model with backend and front-end dashboards. Implement Location-Based Services (LBS) for alerts.
7	System testing: sensor accuracy, real-time alerts, dashboard functionality, and ML predictions. Debug and optimize system performance.
8	Finalize documentation, create prototype demonstration, and prepare presentation/video of the AIR-AWARE system. Deliverables submission.

10 Methodology

For the development of the AIR-AWARE system, we will employ Extreme Programming (XP), an Agile software development framework renowned for its adaptability, high-quality output, and focus on customer satisfaction. XP is particularly suitable for short-duration projects where requirements may evolve during the development process.

10.1 Principles

The methodology is guided by the following principles:

- **High responsiveness to changing needs:** Quickly adapt to any modifications in user requirements or environmental considerations.
- **Team collaboration:** Pair programming and frequent communication to ensure a shared understanding of project goals.
- **Quality-focused development:** Emphasis on clean, maintainable code and early, frequent testing.
- **Continuous feedback:** Incorporate feedback from stakeholders, sensor testing, and ML predictions to improve the system.

10.2 Process

The XP framework normally involves 5 phases or stages of the development process that iterate continuously:

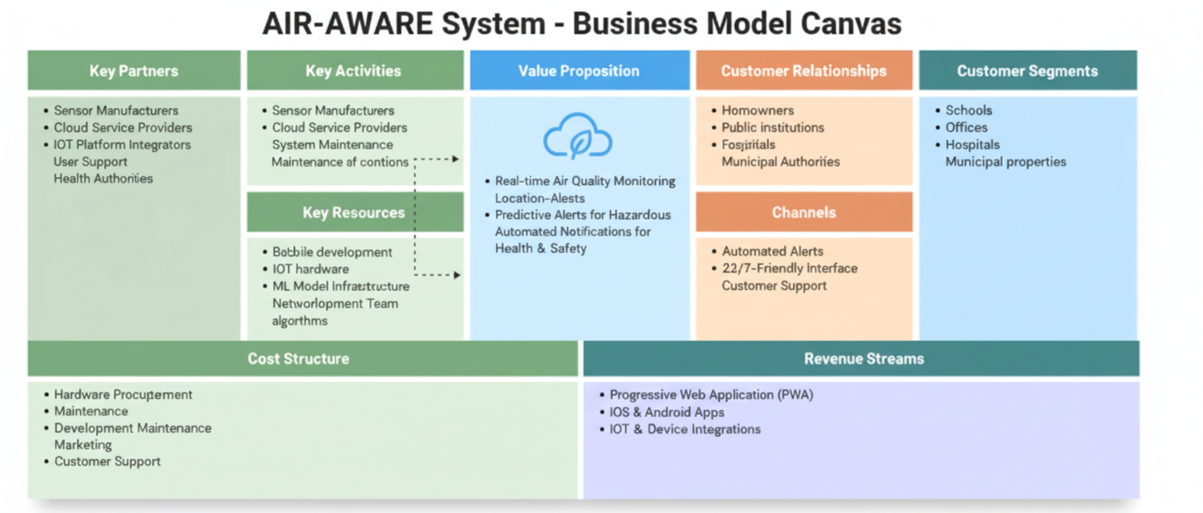
- **Planning:** Define user stories, functional and non-functional requirements, and desired outcomes. Create an 8-week release plan breaking down the system into manageable iterations.
- **Design:** Develop a simple, logical system architecture for IoT integration, backend, frontend, and ML components. Focus on clarity and maintainability.
- **Coding:** Implement the system using best practices, continuous integration, pair programming, and collective code ownership. Develop backend services, IoT communication, ML model integration, and front-end interfaces.
- **Testing:** Conduct ongoing testing, including unit tests for individual components, integration tests for IoT-backend-frontend communication, and acceptance tests with users. Validate ML model predictions and alert accuracy.
- **Listening:** Maintain continuous communication with stakeholders to incorporate feedback, clarify requirements, and ensure alignment with project objectives.

11 Business Study

The Business Study section provides an overall view of the AIR-AWARE system's business model and marketing strategy.

11.1 Business Model Canvas (BMC)

The Business Model Canvas visually represents the main aspects of AIR-AWARE, including value proposition, customer segments, channels, cost structure, and revenue streams.



AIR-AWARE System - Business Model Canvas

Key Partners

- Sensor Manufacturers
- Cloud Service Providers
- IOT Platform Integrators
  User Support
  Health Authorities

Key Activities

- Sensor Manufacturers
- Cloud Service Providers
  System Maintenance
  Maintenance af contions

Key Resources

- Batbile development
- IOT hardware
- ML Model Infrastructure
  Networlopment Team
  algorthms

Value Proposition

- Real-time Air Quality Monitoring
  Location-Alests
- Predictive Alerts for Hazardous
  Automated Notifications for
  Health & Safety

Customer Relationships

- Homowners
- Public institutions
- Fosyritals
  Municipal Authoriies

Channels

- Automated Alerts
- 22/7-Friendly Interface
  Customer Support

Customer Segments

- Schools
- Offices
- Hospitals
  Municipal properties

Cost Structure

- Hardware Procupement
- Maintenance
- Development Maintenance
  Marketing
- Customer Support

Revenue Streams

- Progressive Web Application (PWA)
- IOS & Android Apps
- IOT & Device Integrations

Figure 3: Business Model Canvas

11.2 Marketing Policy

The marketing strategy for AIR-AWARE focuses on positioning it as a reliable, high-value air quality monitoring solution:

11.2.1 Product

- A smart air monitoring system equipped with high-precision air quality sensors for real-time detection of pollutants and hazardous gases.
- A Progressive Web Application (PWA) for immediate alerts, historical data visualization, and location-based monitoring.
- Continuous enhancement of ML algorithms for accurate predictive alerts and reduced false positives.

11.2.2 Price

- Competitive pricing with flexible plans for individual, commercial, and institutional customers.
- Discounts for bulk installations, long-term service agreements, and educational institutions.

11.2.3 Promotion

- Multi-channel promotion through social media, digital marketing campaigns, and participation in health and environmental technology trade shows.
- Free trials, webinars, and educational content to raise awareness of air quality monitoring benefits.

11.2.4 Place

- Accessible via web, iOS, and Android apps, providing users with convenient access to air quality data and alerts from anywhere.

12 Limitations

While AIR-AWARE aims to provide accurate and real-time air quality monitoring, several limitations should be considered:

- **Internet Connectivity Dependence:** The system relies on internet connectivity for real-time monitoring and alerts. Outages may prevent timely notifications.
- **Sensor Accuracy:** Environmental sensors may occasionally produce false readings due to calibration errors, interference, or extreme conditions, potentially triggering unnecessary alerts.
- **Device Placement:** The precision of air quality detection depends on optimal placement of sensors. Incorrect placement may lead to inaccurate readings or missed hazardous events.
- **Data Volume and Processing:** Managing and analyzing large volumes of sensor data in real time can be challenging, requiring efficient data processing pipelines to maintain system performance.
- **Environmental Interference:** Factors such as dust, humidity, and temperature fluctuations may affect sensor readings, which could influence the reliability of alerts.
- **User Response Dependency:** While automated alerts are sent, user action may still be necessary for mitigating risks, and delays in response can impact effectiveness.

13 Deliverables

The AIR-AWARE project will produce the following key outputs:

- **Conceptual Document:** A detailed document outlining the system specifications, features, and services provided, including sensor network architecture, IoT integration, and user interface design.

- **Source Code:** Complete source code for the backend, middleware, frontend, and IoT integration components, hosted on GitHub for transparency and collaboration.
- **Prototype/Simulation:** A functional prototype or simulation demonstrating real-time air quality monitoring, notifications, and location tracking via LBS.
- **Demonstration Video:** A video (mp4 format) showcasing the system's functionalities, including real-time alerts, dashboard interface, and sensor data visualization.
- **User Documentation:** Guides for installation, setup, and operation of the AIR-AWARE system for end-users, including troubleshooting tips.
- **Technical Documentation:** Comprehensive documentation for developers, detailing architecture, APIs, and system workflows for maintenance and future upgrades.