**DUNDALK INSTITUTE OF TECHNOLOGY**

**A logo with a bird and text

Description automatically generated**

**Beta Release Technical Documentation On**

**AEROSENSE – SMART HUB FOR AIR QUALITY MONITORING**

Project Carried Out

**By**

**D00251785 PATRICK ORJIEH**

**D00244618 ROBERT NUGENT**

Under the Supervision of

**DR. JOHN LOANE**

DEPARTMENT OF COMPUTING AND MATHEMATICS

**SCHOOL OF INFORMATICS AND CREATIVE ARTS**

**DEPARTMENT OF VISUAL AND HUMAN-CENTRED COMPUTING**

**EXECUTIVE SUMMARY**

The Beta release of the Aerosense project shows an advancement from our Alpha prototype, showing good progress in the Smart Indoor Air Quality Hub's development. This release includes the successful integration of an additional USB-to-UART converter, enhancing the PMS7003 sensor's functionality within the system. Also, the inclusion of the L76K GPS (new GPS sensor) module.

A significant difference between the alpha release and beta release in this phase is the transition to the PubNub Access Manager, making our system's security framework stronger. This upgrade ensures a more controlled communication environment between the IoT devices and our cloud infrastructure the AWS server.

With these enhancements, the Beta release strengthens the core attributes of Aerosense. It demonstrates our system's capability to support asthma patients effectively by providing vital, real-time insights into air quality dynamics. This Beta Release documentation will detail these improvements.

In conclusion, the Beta release signifies more about the final product, showing a position improvement from the Alpha release and the effect of the feedback received from the lecturers. It is a good step to our ongoing project to enhancing indoor air quality management, especially for individuals with severe asthma.

Table of Contents

[**GLOSSARY** 4](#_Toc151936077)

[**LIST OF TABLES AND DIAGRAMS** 7](#_Toc151936078)

[**List Of Tables** 7](#_Toc151936079)

[**List Of Diagrams** 7](#_Toc151936080)

[**1. INTRODUCTION** 7](#_Toc151936081)

[**2. SYSTEM ARCHITECTURE** 8](#_Toc151936082)

[**Overview** 8](#_Toc151936083)

[**IoT Elements** 9](#_Toc151936084)

[**Data Communication via PubNub** 9](#_Toc151936085)

[**Cloud Server and Database** 9](#_Toc151936086)

[**Firebase Authentication** 10](#_Toc151936087)

[**Mobile Application** 10](#_Toc151936088)

[**Community Data Sharing** 10](#_Toc151936089)

[**Conclusion** 10](#_Toc151936090)

[**Fritzing** 11](#_Toc151936091)

[**3. Alpha Prototype Documentation** 13](#_Toc151936092)

[**Introduction** 13](#_Toc151936093)

[**Hardware** 13](#_Toc151936094)

[**Webserver** 13](#_Toc151936095)

[**Hosted on AWS** 14](#_Toc151936096)

[**PubNub Communication** 14](#_Toc151936097)

[**Database** 14](#_Toc151936098)

[**Security** 15](#_Toc151936099)

[**3. Cloud Server Description** 15](#_Toc151936100)

[**Endpoints** 15](#_Toc151936101)

[**4.PubNub** 16](#_Toc151936102)

[**5.Security Implementation in the Aerosense Project** 16](#_Toc151936103)

[**IoT Device Security:** 16](#_Toc151936104)

[**Access to Communication Channels:** 16](#_Toc151936105)

[**Database Security:** 17](#_Toc151936106)

[**Webserver Security:** 17](#_Toc151936107)

[**Data in Transit Security:** 17](#_Toc151936108)

[**Firebase Security:** 17](#_Toc151936109)

[**6.Future Enhancements for the Aerosense Project** 17](#_Toc151936110)

[**CONCLUSION** 18](#_Toc151936111)

[**REFERENCES:** 18](#_Toc151936112)

## **GLOSSARY**

* **Aerosense**: A project aimed at developing a portable device (Smart Indoor Air Quality Hub) for monitoring indoor air quality, particularly beneficial for individuals with asthma.
* **Smart Indoor Air Quality Hub**: A portable device designed to provide real-time data and insights about indoor air quality, helping individuals, especially those with asthma, to understand and manage their environment better.
* **PubNub**: A cloud-based service that provides real-time data streaming and messaging solutions, used in Aerosense for real-time communication between the device and the server.
* **AWS (Amazon Web Services)**: A comprehensive and widely adopted cloud platform that offers various services such as computing power, database storage, and content delivery. In Aerosense, it's used for data processing and storage.
* **Alpha Prototype**: An early version of a product that is functional enough to demonstrate the concept and design but may not have all the final features and polish of the final product.
* **Cloud Server**: Remote servers accessed over the internet used to store, manage, and process data, as opposed to a local server or personal computer.
* **User Interaction**: The process and experience of a person engaging with the Aerosense system, particularly through its mobile application.
* **Sensor Data Acquisition**: The process of collecting data from various sensors (like those measuring air quality) used in the Aerosense system.
* **Raspberry Pi**: A small, affordable computer used for various programming and electronics projects. In Aerosense, it serves as the central unit for collecting sensor data.
* **PMS7003 Particle Sensor**: A sensor that measures particulate matter (PM) in the air. It's crucial for providing data on air quality, especially for asthma patients.
* **BME680 Sensor**: This sensor provides a range of environmental data, including temperature, humidity, and volatile organic compounds (VOCs), which are chemicals that can affect air quality.
* **Grove Air 530 GPS Sensor**: A sensor used for determining geographical location, enabling the Aerosense system to tag air quality readings with specific locations.
* **PubNub Access Keys**: Unique identifiers used for secure access to PubNub's publish and subscribe channels, ensuring that only authorized devices and servers can communicate through the platform.
* **AWS Cloud**: Amazon Web Services Cloud, a cloud computing platform that hosts the server for the Aerosense system. It's responsible for data processing and storage.
* **AES-256 Encryption**: A method of encrypting data to protect it from unauthorized access. It's used in Aerosense to secure data at rest in the database.
* **TLS Protocol**: Transport Layer Security protocol, a method for encrypting data during transmission to prevent interception by unauthorized parties. Used in Aerosense for secure data transfer.
* **Firebase Authentication**: A Google service that provides user authentication for applications. In Aerosense, it manages user login processes and secures access to user data.
* **Google OAuth**: An authentication method provided by Google that allows users to log in to applications securely using their Google account.
* **Mobile Application**: The user interface of the Aerosense system, available on smartphones. It displays air quality data and allows users to interact with the system.
* **Community Data Sharing**: A feature in Aerosense where users can share anonymized environmental data to contribute to broader air quality studies.
* **USB to UART Converter**: A tool that lets the Raspberry Pi talk to the sensors using USB ports, which are very common on computers (TX & RX).
* **Wiring**: Cables that connect different parts of the electrical circuit in the diagram, each with a different color for easy identification.
* **Flask**: A lightweight web application framework written in Python, used for creating the web server in the Aerosense project.
* **Cron Job**: A scheduled task in Unix-like operating systems. In Aerosense, it's used to periodically trigger data collection from the sensors.
* **Encryption**: The process of converting information or data into a code to prevent unauthorized access. Used in Aerosense to secure data transmission.
* **HTTPS (Hypertext Transfer Protocol Secure):** An extension of HTTP used for secure communication over a computer network. In Aerosense, it's used for secure communication between the client application and the server.
* **Database Schema**: The structure of a database system, described in a formal language. In Aerosense, it defines the structure of the MySQL database.
* **Real-time Data Transfer**: The process of continuously transferring data as it's collected without delay. In Aerosense, this is facilitated by PubNub.
* **API (Application Programming Interface):** A set of protocols for building and interacting with software applications. Aerosense uses APIs to enable communication between different components of the system.
* **Volatile Organic Compounds (VOCs):** Organic chemicals that have high vapor pressure at room temperature and can affect air quality. Measured by the BME680 sensor in Aerosense.
* **MySQL:** An open-source relational database management system. In Aerosense, it's used to store and manage data.
* **AWS EC2 (Amazon Web Services Elastic Compute Cloud**): A web service that provides resizable compute capacity in the cloud, used to host the server for Aerosense.
* **L76K GPS Module:** A new GPS sensor module implemented in the Aerosense system for enhanced geolocation capabilities.
* **PubNub Access Manager:** A security feature in PubNub that provides fine-grained access control to channels for improved security in real-time communication.
* **Beta Prototype:** The subsequent version of the Aerosense product after the alpha phase, which includes new features, improvements, and integrations based on
* **User Feedback Integration:** The process of incorporating suggestions and critiques from users into the development of the Aerosense system to enhance its functionality and user experience.
* **Enhanced User Interface:** Improvements made to the system's interface to make it more intuitive and user-friendly, often as a result of user feedback.
* **Firmware Updates:** Software updates for the embedded system within the Aerosense device, which can improve functionality or fix issues.

## **LIST OF TABLES AND DIAGRAMS**

### **List Of Tables**

### **List Of Diagrams**

Diagram 1: System Architecture Diagram of Aerosense Application (Revised for Beta Release) ……………………………………………………………………................................... 8

Diagram 2: Updated Fritzing Diagram: Sensor Connections with Pi .....................................11

## **1. INTRODUCTION**

The Beta release of the Aerosense project marks a significant step forward from our initial Alpha prototype, showing the improved parts in monitoring indoor air quality, important for people with asthma. In this release, we've integrated new hardware components and made good improvements, and for the universal design principles caters to innovation and user-centric design. This document will go through the advanced features and architectural enhancements, telling on the integration of real-time data acquisition and user interaction. With a focus on accessibility and functionality.

## **2. SYSTEM ARCHITECTURE**

A diagram of a cloud computing system

Description automatically generated

Diagram 1: System Architecture Diagram of Aerosense Application (Revised for Beta Release)

### **Overview**

The Beta release of Aerosense maintains the foundational system architecture established in the Alpha release, with just little changes made to the PubNub communication protocol and GPS module integration for functionality and improved accuracy.

### **IoT Elements**

The core of the Aerosense system is based around a Raspberry Pi, which acts as the central data collection unit. Attached to the Pi are three crucial sensors:

1. **PMS7003 Particle Sensor**: This sensor is responsible for measuring particulate matter in the air, providing data on air quality that is particularly relevant to asthma sufferers.
2. **BME680 Environmental Sensor**: Offers a broader range of environmental readings including temperature, humidity, and volatile organic compounds (VOCs).
3. **L76K GPS Module**: Used to provide location data, enabling tagged air quality readings, which is essential for mapping and analysing environmental conditions in different areas.

### **Data Communication via PubNub**

Data from these sensors is transmitted using PubNub, a real-time communication platform. The Raspberry Pi publishes the sensor data to a dedicated PubNub channel, which is then subscribed to by the cloud server. This setup ensures the delivery of data from the sensors to the server for processing. We have advanced our data communication setup by implementing PubNub's Access Manager. This makes the security protocols better, ensuring admin control over data streams:

To secure this communication:

* **PubNub Access Manager:** Replacing simple access keys, the Access Manager introduces a permission layer, safeguarding channel communication.
* **Bidirectional Communication**: Certain scenarios allow for two-way communication between the cloud server and the Raspberry Pi, facilitating real-time updates and commands.

### **Cloud Server and Database**

The server hosted on AWS Cloud forms the backbone of data processing and storage. It subscribes to the PubNub channel to receive sensor data, processes this data according to predefined processing algorithms that will be made by us, and stores it in a secure database. The database maintains records of air quality readings, user profiles, and other relevant data.

For securing data:

* **At Rest**: The database employs encryption mechanisms like AES-256 to secure data at rest.
* **In Transit**: Data transferred between the IoT devices, server, and the mobile application is encrypted using TLS protocols by PubNub, ensuring secure data transmission.

### **Firebase Authentication**

User authentication is managed by Firebase, which integrates Google OAuth for a secure and convenient login process. Firebase provides authentication tokens that are used to verify user identity and secure access to the mobile application and personal data.

### **Mobile Application**

The client-side of the Aerosense system is a mobile application that serves as the user interface. It allows users to view real-time air quality data, receive alerts, and manage their profiles. The app fetches data from the cloud server, user interactions with the app are also sent back to the server for processing and response.

### **Community Data Sharing**

One of the innovative features of Aerosense is its community data sharing aspect. Users can choose to share anonymized environmental data, contributing to a broader understanding of air quality trends. This data is coupled by the server and can be accessed for public awareness purposes.

### **Conclusion**

The Aerosense system architecture has been designed to ensure a flow of data from the sensors to the end-user. Security protocols are in place at every step to protect sensitive information, especially considering the health-related nature of the data (GDPR).