**DUNDALK INSTITUTE OF TECHNOLOGY**

**A logo with a bird and text

Description automatically generated**

**Technical Documentation On**

**WRISTBAND AIR QUALITY MONITOR**

Project Carried Out

**By**

**D00251785 PATRICK ORJIEH**

**D00244618 ROBERT NUGENT**

**D00261104 CONOR MC GINN**

**D00240060 HANNAH MC ELROY**

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**

In the context of our Universal Design Project, we - the LifeSync team - embarked on an innovative journey that led to the birth of "Aerosense." Imagine a wristband that does more than tell time; it gives you a real-time snapshot of the air you breathe. With the surge of pollution in our cities, it`s high time we have something that keeps us informed. That's where Aerosense comes into play.

This isn't just about tech for the sake of tech. By leveraging the latest IoT, we've merged the handiness of everyday wearables with cutting-edge sensors that monitor air quality with finesse. It's all rooted in Universal Design's philosophy, aiming to make air quality data not just available but effortlessly integrated into our routines.

Dive into this report, and you'll discover the ins and outs of Aerosense's design, the strategies we employed, and the hurdles we overcame. For us at LifeSync, this wristband isn't just a gadget; it encapsulates our dedication to creating solutions that genuinely enrich our lives.

Table of Contents

[**GLOSSARY** 4](#_Toc149513638)

[**LIST OF TABLES AND DIAGRAMS** 5](#_Toc149513639)

[**List Of Tables** 5](#_Toc149513640)

[**List Of Diagrams** 5](#_Toc149513641)

[**1. INTRODUCTION** 6](#_Toc149513642)

[**1.1 Purpose** 6](#_Toc149513643)

[**1.2 Significance** 6](#_Toc149513644)

[**2. HARDWARE** 7](#_Toc149513645)

[**2.1 Hardware Components Required for Aerosense Development:** 7](#_Toc149513646)

[**2.2 Hardware Overview** 8](#_Toc149513647)

[**2.3 Hardware Connectivity** 8](#_Toc149513648)

[**2.4 Power And Internet Connectivity** 10](#_Toc149513649)

[**3. DATA** 11](#_Toc149513650)

[**3.1 Data Gathering** 11](#_Toc149513651)

[**3.2 Data Storage** 12](#_Toc149513652)

[**3.3 Data Processing** 13](#_Toc149513653)

[**4. SECURITY AND PRIVACY** 15](#_Toc149513654)

[4.1 **Device Security** 15](#_Toc149513655)

[4.2 **Data in transfer security** 15](#_Toc149513656)

[4.3 **Data in storage** 16](#_Toc149513657)

[**5. UI, USER AND TESTING** 17](#_Toc149513658)

[**5.1 Registration & Login:** 17](#_Toc149513659)

[5.2 **Home Screen & Menu** 18](#_Toc149513660)

[5.3 **Location & History** 19](#_Toc149513661)

[**5.4 Settings**: 20](#_Toc149513662)

[**6 USER & TESTING:** 22](#_Toc149513663)

## **GLOSSARY**

* Aerosense: The brand name of the air quality wristband.
* IoT: Internet of Things - Refers to the network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices over the Internet.
* API: Application Programming Interface - A set of tools and protocols that allow different software applications to communicate with each other.
* Asthma: A respiratory condition marked by spasms in the bronchi of the lungs, causing difficulty in breathing. It is often connected to allergic reactions or other forms of hypersensitivity.
* GPIO (General Purpose Input/Output) pins: Pins on the Raspberry Pi used for connecting and communicating with external devices.
* ADC (Analog to Digital Converter): A device that converts analogue signals (continuous values) to digital values.
* Particulate Matter: Tiny particles or droplets in the air that are two and one-half microns or less in width.
* AQI (Air Quality Index): A measure used to show how polluted the air currently is or how polluted it is forecast to become.
* VOC (Volatile Organic Compounds): Organic chemicals that have a high vapor pressure at room temperature. Their high vapor pressure results from a low boiling point, which causes large numbers of molecules to evaporate from the liquid or solid form of the compound and enter the surrounding air.
* PMS7003: A sensor that measures particulate matter of different sizes: PM10, PM2.5, PM1, and PM0.3.
* BME680: A gas sensor array used to measure VOCs, which impact air quality.
* Air530: A GPS sensor used to capture latitude and longitude for correlation with air quality measurements.
* Cron: A time-based job scheduler in Linux operating systems. Cron is used to schedule jobs (commands or scripts) to run periodically at fixed times, dates, or intervals.

## **LIST OF TABLES AND DIAGRAMS**

### **List Of Tables**

Table 1: Hardware Components Required for Aerosense Development .......................... 7

Table 2: User Information and Details ............................................................................ 12

Table 3: Wristband Specifications and Details ............................................................... 12

Table 4: Air Quality Measurement Data .......................................................................... 13

Table 5: User Location.................................................................................................... 13

Table 6: Notification Details ............................................................................................ 13

### **List Of Diagrams**

Diagram 1: Fritzing Diagram: Sensor Connections with Pi .................................................... 9

Diagram 2: Database Schema for Aerosense ...................................................................... 12

Diagram 3: User Registration and Login Page Design ........................................................ 17

Diagram 4: Reset Password Page Design ........................................................................... 18

Diagram 5: Home Screen Page Design ............................................................................... 18

Diagram 6: Menu Navigation Page Design .......................................................................... 19

Diagram 7: Location Display Page Design ........................................................................... 19

Diagram 8: History and Past Data Page Design .................................................................. 20

Diagram 9: User Settings and Configuration Page Design ………………………................ 21

## **1. INTRODUCTION**

### **1.1 Purpose**

In today's rapidly urbanizing world, the quality of air we breathe is a growing concern, particularly for vulnerable groups such as asthma sufferers. To address this challenge, we've initiated the development of the "Aerosense Wristband." This innovative wristband is specially designed to monitor air quality, catering specifically to those with asthma. With it, users can access immediate updates on the surrounding air quality, enabling them to make decisions that prioritize their health.

### **1.2 Significance**

The idea behind the Aerosense Wristband was not just to create a new tech gadget. There's a genuine, urgent need for such a device. Several research studies have shown a clear correlation between declining air quality and an increase in asthma-related issues. With the Aerosense Wristband, we aim to empower users with the knowledge they need to avoid potential asthma attacks, ensuring they're not caught off guard. On a broader scale, by collecting data from numerous Aerosense devices, we might be able to identify specific areas with high pollution levels, as well as detect patterns over time. This could prove invaluable for those in roles ranging from city planning to healthcare policy development.

As we worked on the Aerosense Wristband, we focused on integrating the latest air quality sensing technology while keeping the user interface simple and intuitive. We strived for a sleek design, making sure that whether you're a tech enthusiast or just someone looking for a solution, the device is easy to use. This report will delve deeper into our development process, discussing everything from our initial feasibility studies to the depth of data management.

To make this report as reader-friendly as possible, we've avoided unnecessary technical jargon. Where complex ideas arise, we've tried to simplify them with relatable analogies and, when needed, included diagrams for a clearer understanding.

## **2. HARDWARE**

### **2.1 Hardware Components Required for Aerosense Development:**

|  |  |
| --- | --- |
| Component | Description |
| Raspberry Pi 400 | Central processing and connectivity module. |
| Female/Female Jumper Wires | Connects sensors to the Raspberry Pi. |
| PMS7003 | Particulate Matter sensor to measure air quality. |
| BME680 | Measures air quality, temperature, humidity, and pressure. |
| Air530 | GPS Module to provide location data for air quality readings. |
| Vibrating mini motor disc | Alerts users to significant changes in air quality. |
| LDR Light dependent resistor | Measures light intensity for sleep mode. |
| MCP 3008 ADC | Analog to Digital Converter; works with the LDR. |
| Resistor kit | Various resistors to ensure proper voltage across circuits. |
| Full size breadboard | For testing and connecting various components. |
| Female/Male Jumper Wires | For flexible connections during prototyping. |
| Power bank | Portable power source for Aerosense when unplugged. |

Table 1: Hardware Components Required for Aerosense Development

### **2.2 Hardware Overview**

The primary objective of the Aerosense Wristband project is to develop an advanced air quality monitoring system made specifically for asthma patients. The primary component of this undertaking is the precise and efficient hardware design.

At the core of our hardware parts is the Raspberry Pi 400. This minicomputer acts as the central processing unit, interpreting and analysing data from various sensors. It’s very essential for our needs.

For the accurate assessment of air quality, we have added two specialized sensors: the PMS7003 and BME680. The combination of these sensors ensures comprehensive and precise data on particulate matter, volatile organic compounds, and other air quality metrics pertinent to asthma patients.

Geographical location is essential in understanding air quality patterns. The inclusion of the Air530 GPS module provides users with real-time location data, allowing them to know specific zones with bad air quality. This feature is useful for users to make informed decisions about their daily routes and routines.

User feedback is equally important. Rather than giving the user raw numerical data, the Vibrating mini motor disc and coloured LED serves as a tactile alert mechanism. In instances of deteriorating air quality, the device will vibrate.

Connection between the Raspberry Pi and the sensors is achieved using jumper wires, breadboards, and connectors. Power considerations led us to select a power bank. For real-time data retrieval and cloud-based analyses with AWS, internet connectivity is provided via the Raspberry Pi's inbuilt Wi-Fi capabilities.

### **2.3 Hardware Connectivity**

To illustrate the connectivity of the components, the Fritzing tool would be used to generate the diagram.

A diagram of a circuit board

Description automatically generated

Diagram 1: Fritzing Diagram: Sensor Connections with Pi

* The **Raspberry Pi 400** serves as the central unit connecting all sensors and modules.
* **PMS7003** and **BME680** sensors connect to the Raspberry Pi via GPIO pins using Female/Female Jumper Wires.
* **Air530** GPS Module connects to the Raspberry Pi to fetch and log location data, helping users identify pollution hotspots.
* **Vibrating mini motor disc** gets activated when the sensors detect air quality levels beyond a predefined threshold.
* The **LDR** works in tandem with the **MCP 3008 ADC** to measure light intensity, which could help in activating a sleep mode for the wristband during low light conditions, thus conserving battery.
* The **Resistor kit** will be used to ensure that the correct voltage is applied across circuits, especially when connecting LEDs or buzzers.

### **2.4 Power And Internet Connectivity**

The Aerosense Wristband will primarily be powered by a **power bank**, making it portable and user-friendly. This ensures that the device can continuously monitor air quality without being connected to a fixed power source. The Raspberry Pi 400's built-in Wi-Fi capabilities will be leveraged for internet connectivity. This allows the device to upload air quality data to a server and fetch updates if necessary.

## **3. DATA**

### **3.1 Data Gathering**

**a. Particulate Matter Measurements:**

**Sensor Utilized:** PMS7003  
**Measurements Captured:**

* PM0.3
* PM1
* PM2.5
* PM10

The measurements for PM2.5 and PM10 will be critical since there are standard AQI calculations for these. The PM2.5 and PM10 values provide insights into smaller particles that can be inhaled and pose health risks.

**b. Volatile Organic Compounds (VOC) Measurements:**

**Sensor Utilized:** BME680  
The BME680 sensor measures the concentration of VOCs in the air. High levels of VOCs can indicate polluted air, often resulting from chemicals or other pollutants.

**c. GPS Measurements:**

**Sensor Utilized:** Air530  
This sensor captures the geographical location (latitude and longitude) where air quality measurements are taken. This allows users to identify specific locations with good or poor air quality.

### **3.2 Data Storage**

A black background with white rectangles

Description automatically generated

Diagram 2: Database Schema for Aerosense

- An SQL file is added in the git repo that will have the create statements for the Database

For storing the data gathered, MySQL relational database will be utilized. The following are the tables with their associated attributes:

**a. User Table:**

|  |
| --- |
| * userID: Unique identifier for the user. |
| * name: Name of the user. |
| * email: Email address of the user. |
| * password: Hashed password for authentication. |
| * wristbandID: Foreign key referencing the Wristband table. |

Table 2: User Information and Details

**b. Wristband Table:**

|  |
| --- |
| * wristbandID: Unique identifier for the wristband. |
| * model: Model name or number of the wristband. |
| * batteryLevel: Current battery level percentage. |

Table 3: Wristband Specifications and Details

**c. Air Quality Measurement Table:**

|  |
| --- |
| * measurementID: Unique identifier for the measurement. |
| * wristbandID: Foreign key referencing the Wristband table. |
| * PM1: Measurement for PM1 particles. |
| * PM2.5: Measurement for PM2.5 particles. |
| * PM10: Measurement for PM10 particles. |
| * VOC: Measurement for VOCs. |
| * timestamp: Time when the measurement was taken. |

Table 4: Air Quality Measurement Data

**d. Location Table:**

|  |
| --- |
| * locationID: Unique identifier for the location entry. |
| * wristbandID: Foreign key referencing the Wristband table. |
| * latitude: Latitude of the location. |
| * longitude: Longitude of the location. |
| * timestamp: Time when the location was recorded. |

Table 5: User Location

**e. Notification Table:**

|  |
| --- |
| * notificationID: Unique identifier for the notification. |
| * measurementID: Foreign key referencing the Air Quality Measurement table. |
| * message: Content of the notification. |
| * timestamp: Time when the notification was triggered. |

Table 6: Notification Details

### **3.3 Data Processing**

The data processing section shows how frequently the data will be collected, then processed and converted into meaning data for the user.  
  
- Frequency of Data Collection and Processing:

Given the dynamic nature of air quality, data from sensors will be collected and processed every minute.

2. Using Cron for Automated Data Collection and Processing:

Cron will be set up to execute scripts that handle data collection and processing tasks.

* 1. \* \* \* \* \* sensor\_data\_collection.py
  2. \* \* \* \* \* data\_processing.py

3. After Data Is collected during the processing we have a detailed way we will convert the data into bad or good air quality, Using method in the **data\_processing.py** in the git link.  
- Extract the latest readings of PM10, PM2.5, and VOC from the database.

- Compute the AQI value for each pollutant type (PM10, PM2.5, VOC) based on predefined AQI conversions.

- Choose the highest AQI value among the measurements as the overall AQI for that minute.

- After processing, save the computed AQI and the percentage score back into the database. This transformed data will then be accessed by the user interface to provide real-time feedback.

4. Alerting Mechanism:

If the computed AQI reaches a threshold that indicates poor air quality, an alerting mechanism will be activated, notifying users to take appropriate action.

## **4. SECURITY AND PRIVACY**

### 4.1 Device Security

The first feature that we will be implementing in order to support device security is a 4-digit pin for the device. When first using the wristband, the user will be prompted to setup a 4-digit PIN which will be used in the future to gain access to the device when attempting to connect to it. This will prevent anyone from connecting to the device and tampering with it.

To build on the security of this PIN, we will also implement a device lockout system. Whenever the user enters the PIN incorrectly 5 times then the device will be locked and not allow any access to it until the lock is lifted. When the device becomes locked, an email will be sent to the email address associated with the device with an option to unlock the device. By using this system. It will prevent unauthorized users to have an infinite number of tries at bypassing the PIN to access the device.

Another feature that we are designing that will increase the level of security for the device is a way for the user to remotely wipe or lock the device if they lose it. Once the device becomes lost, it will be vulnerable if found by someone else. To counter this, we will allow the user to lockdown their lost device from their current location using the app so that no one will be able to access it and tamper with it. As well as this, the user could wipe all of their data from the device and essentially reset it so that if someone would gain access to it, they wouldn’t be able to see any of their data.

### 4.2 Data in transfer security

The main way that we are going to keep data safe in transfer is through end-to-end encryption. This means that the data will be encrypted before it is transmitted and will only be decrypted once it has reached its destination. There are a lot of different ways to do this but the way which we plan on doing it is through the use of a strong encryption algorithm called AES, which stands for Advanced Encryption Standard. We have decided to use this specific algorithm as it is considered highly secure and efficient which will highly contribute to the overall security and performance of the system.

Another way to improve the security of the data is by carrying out data integrity checks. To ensure that the data was not messed with during transfer, we must check the integrity of the data once it is received by the server. We can do this by using cryptographic hashes. When the data is hashed using a hash function, a hash value is produced. This value will be sent with the data to the server so that the server can compare the received hash value with the one that it gets from carrying out the hash function itself. If the hash values match, then the data has not been messed with, otherwise, the data isn’t the same as what was originally sent.

Finally, we will make use of tokens during data transfer for authentication reasons. Tokens are important for authenticating the users who will be carrying out these data transfers. By using tokens, we can ensure that the user has appropriate permissions to carry out what they are trying to do as well as determine if they are a valid user or not. If the token hasn’t expired and is has been verified, then the action relating to data transfer will be performed.

### 4.3 **Data in storage**

While in storage, we will be encrypting data to make it more secure. It’s important that we encrypt personal data such as emails and passwords as we don’t want such information to end up leaked or stolen. Other than specifically encrypting, we can keep some of the data secure using methods such as password hashing and salting.

Another way of keeping the data safe while in storage is by enforcing strict access control. By using access control, we only allow certain authenticated users access to the data and decide which of these users will be able to modify this data. We will only allow each user the exact permissions that they will need to carry out their task and nothing more than that. We will be basing the access control on user roles so that each user has the appropriate permissions for their role and responsibilities.

We will also be backing up the data frequently to ensure that we maintain data integrity. There is always a chance that data can become corrupted, or a problem will arise with it and we want to have a solution for this. By backing up the data, we will easily be able to restore it when something like this happens. This will make data recovery easier and will stop us from losing data in these events while also improving the integrity of the data.

## **5. UI, USER AND TESTING**

### **5.1 Registration & Login:**

When a user first opens the application, they will be presented with the log in page. If the user has an account and knows their password then they can simply input their details and click the ‘LOGIN’ button, which will take them to the main functionality of the app.

If it is there first time using the app they will need to create an account, they can do this by clicking the ‘Sign Up’ button at the bottom of the page. Doing this will take them to the Create Account page where they can fill in their personal details and then submit them using the ‘Register’ button.

The buttons on these pages are easy for the user to understand they are clickable by making them blue. The icons to the left when entering details will be helpful to make it obvious what needs to be input even if the user does not know English.

**A screenshot of a login form

Description automatically generated**A screenshot of a login form

Description automatically generated

Diagram 3: User Registration and Login Page Design

A screenshot of a login box

Description automatically generatedIf the user has account but cannot remember their password, they can then click the ‘Forgot Password’ button on the login page. This will take them to the Reset password page where they can enter the email they previously signed up with and hit submit. This will send an email to the account where they can set a new password for the account.

Diagram 4: Reset Password Page Design

### 5.2 **Home Screen & Menu**

Diagram 5: Home Screen Page Design

A screenshot of a phone

Description automatically generatedWhenever a user is signed in and opens the app they will be shown the home screen. This page displays all the important data gathered by the wristband device in a neat and clear manner that is easy for any user to understand. At the top there is a circle that fills up with a certain colour depending on how bad the air quality is in the user’s current location. If the air quality is bad it will display red, if it is moderate it will display yellow and if the air is clean it will be green. This colour coding along with the air quality being displayed as a whole number percentage makes it simple for the user to instantly understand if the air quality is bad or not.

Underneath the circle is more technical data such as Particle Matter data and VOC Level. The colour coding on the values of these data means the user does not need to actually know what they mean but can tell if the levels are dangerous or not by the colour being red or orange. The icons also help illustrate what the data is conveying. At the bottom of the page the user can also click to view their location on a map.

A screenshot of a phone

Description automatically generated

In the top left of the app there is a menu beutton, this is displayed on all main pages of the app. The hamburger menu icon is a uiversaly understood so any user will know what the button is for. When clicked a dropdown menu will appear with a list of the different pages on the app, this allows the user to quickly switch between different pages for all their information.

Diagram 6: Menu Navigation Page Design

### 5.3 **Location & HistoryA map with red dots Description automatically generated**

On the location page, the user is presented with a map of their current location. From here, they can view nearby areas where air quality is at a hazardous level for them. They can tell where they are from the pin on the map and where the hazardous areas are from the red circles, indicating danger.

On this page you will also be able to swipe around and zoom out to see a wider area, user’s will be able to plan ahead on what route they take and can avoid the hazardous air quality, when they are for example going for a run or walk.

Diagram 7: Location Display Page Design

A screen shot of a graph

Description automatically generatedOn the history page, the user will be able to view detailed statistics regarding the data they have collected while wearing their wristband. There is a dropdown menu to select the duration of time they want to view data from, such as past week, month or day.

The graph below this displays the average air quality the user has been exposed to whilst wearing the band over this period of time. On the left of the graph is the percentages of air quality cleanliness and along the bottom are periods in time, in this example we have each day of the week.

This graph simply outlines how clean the air quality they have been surrounded with is and lets the user know if their percentage is high on a certain day or time, they should try to avoid the location they were at that time

Below the graph, we also have average particle matter as well as highest and lowest VOC readings for the period selected.

Diagram 8: History and Past Data Page Design

### **5.4 Settings**:

The settings page displays numerous settings in a clear and concise matter. Icons are used to signify to the user what they are for. The pair device button is used to connect the device to the wristband and once connected the user will be able to tell from the connection status to the right of it.

The battery level is shown for the wristband device so the user knows if it needs to be charged or not. If it does go low during use however, an alert will be sent to the user.

The vibration setting can be clicked on and off and the volume level for alerts can be selected by dragging the sound bar higher or lower. When these options are selected, they will also be tested to let the user know what they have chosen, for example when vibration is turned on the band will vibrate and when the sound is selected an alert will play at that volume.

The notification frequency setting includes a dropdown menu for a list of multiple options such as always, only when critical and never.

The user can also change their password from this page in case they are signed in but are not sure what their password is.

There are also colour-coded buttons on the page. The red buttons are for logging out of the application and resetting the changed settings. While the green save button is for saving the changed settings. These colours help the user understand what the buttons might be for.

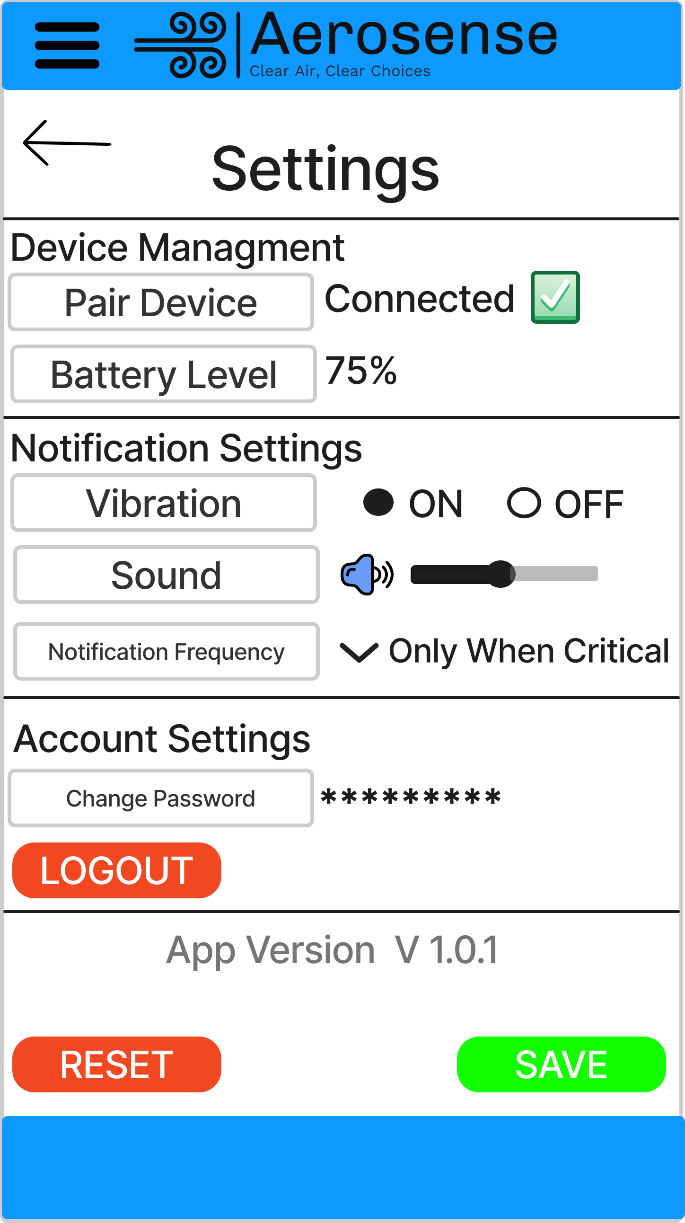


Diagram 9: User Settings and Configuration Page Design

## **6 USER & TESTING:**

This device is for people with asthma, as such this means the user interface needs to be simple and intuitive to use. This way even if the user is panicked or just need to quickly look at the app to understand the data displayed on it they should instantly be able to.

What will determine if this device is successful or not is if people with asthma can use it to track the air quality in their current location and use that information to tell if it is dangerous or not for them to stay or if they should move. They should also be able to view nearby locations and see where the air quality is bad so they know to avoid these areas.

We will test this device by bringing it to areas of varying air quality to test that it properly alerts the user of the air quality in their current location and whether or not it is hazardous for them.

**References:**

* Information on the PMS7003 sensor [Accessed**:** 21/10/23 14:57] :<https://download.kamami.pl/p564008-PMS7003%20series%20data%20manua_English_V2.5.pdf>
* Information on the BME680 sensor[Accessed**:** 21/10/23 15:34]: <https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/>
* Information on the Air530 sensor[Accessed**:** 22/10/23 11:29]: <https://wiki.seeedstudio.com/Grove-GPS-Air530/>
* Data Protection, in transit & at rest [Accessed: 22/10/23 12:24]: <https://www.digitalguardian.com/blog/data-protection-data-in-transit-vs-data-at-rest#:~:text=For%20protecting%20data%20in%20transit,contents%20of%20data%20in%20transit>.
* Data Protection Information [Accessed: 22/10/23 13:43]: <https://www.dataprotection.ie/en/organisations/know-your-obligations/data-security-guidance>