Marking Sheet – MSc Business Analytics Consultancy Project/Dissertation 2020-21

Criteria/Weight	Supervisor's comments
Topic, theoretical framework, literature, and methodology (35%): Topic is clearly identified and boundaries are asserted. Knowledge of relevant theories and their limitations. Current and relevant literature coming from reliable sources. Appropriate and adequate methodology for topics. Detailed methodology facilitating replication of project and reproducibility of results.	
Analysis and conclusions /recommendations (35%): Use of primary and/or secondary data. Rigorous analysis and interpretations. Alternative interpretations/arguments are considered. Limitations are identified and justified by reasonable arguments. Conclusions/recommendations are fully consistent with evidence presented.	
Structure, originality and presentation (10%): Provides a concise summary. Demonstrates an understanding of business context. Coherent and appropriate structure. Adequate presentation, language, style, graphs, tables, and referencing. Appropriate use of visualisation. Presents business recommendations.	
Complexity of project scope and progress made towards business goals (10%): Progress made towards overcoming technical and operational challenges encountered during the project. Progress made in overcoming problem framing and theoretical and data related problems encountered during the project.	
Project Management (10%): Good use of project management and communication tools. Use of Kanban board for structuring project work. Evidence of objectives being broken down in appropriate tasks and timely engagement with primary supervisor.	

General marking guidelines

85+ Outstanding work of publishable standard.

70-84 Excellent work showing mastery of the subject matter and excellent analytical skills.

60-69 Very good work. Interesting analysis with original insights. Some minor errors.

50-59 Good work which only covers a basic analysis. Some problems but no major omissions.

40-49 Inadequate work. Not sufficiently analytical. Some major omissions.

39- Work seriously flawed. Lack of clarity and argumentation. Too descriptive.

Mark: _______

UCL SCHOOL OF MANAGEMENT

University College London - School of Management

MSc Business Analytics Consultancy Project/Dissertation

2020-21

Title of Project: **Building an interactive low-cost air quality monitoring dashboard for a pre-revenue startup with Streamlit**

Date: 6th August 2021

Word Count: 11,980 words

Disclaimer:

I hereby declare that this dissertation is my individual work and to the best of my knowledge and confidence, it has not already been accepted in substance for the award of any other degree and is not concurrently submitted in candidature for any degree. It is the end product of my own independent study except where other acknowledgement has been stated in the text.

Abstract

The low-cost air quality monitoring industry is an emerging space, with few competitors in the market providing an air quality monitoring device accompanied by a dashboard. In this project, an interactive web app dashboard will be built for the air quality monitoring startup, CompAir. The process and decisions around the development of the dashboard will be explained, answering the research question: why and how to build an air quality interactive web app dashboard for consumers? The results of the paper demonstrate several hypothetical use cases of consumers using the interactive dashboard to make decisions to improve indoor air quality. Furthermore, the results also recommend using Heroku as the highest performing free deployment method for a Streamlit web application. These results are relevant for businesses and individuals in the air quality industry to understand the process behind developing an interactive dashboard and the reason to increase consumer awareness on air quality.

Table of Contents

List of Figures

1. Introduction

1.1 CompAir

CompAir is a social enterprise that is determined to improve air quality by helping communities and industries reduce their exposure and emission of air pollution. CompAir develops low-cost air quality monitoring devices called Airtrackers. The device measures various air quality metrics such as air pressure, temperature, indoor air quality, eCO2, humidity, PM1, PM2.5, PM10 and VOCs. The company was founded in 2019, is pre-revenue and is backed by Tech for Good venture capitalists, Bethnal Green Ventures. The CompAir team is currently testing five prototype devices in the UK with a goal to begin field-testing, customer trials and carry out a commercial roll out in late 2021/early 2022. CompAir targets two types of users - communities and industry. Industry focuses on helping companies protect their staff and workforce from poor air quality while communities focuses on indoor air quality and measuring air quality in homes, schools and public spaces. The focus of this dissertation and the project will be on the latter, communities.

1.2 Problem Framing & Objectives

After initial meetings with the CompAir founders to frame the problem they were facing, it quickly became apparent that the visualisation and front-end interface was a fundamental element of their product offering they needed help with. As CompAir's company motto is "We are making the invisible visible" having a front-end interface is crucial. Currently, CompAir's devices create a csv file with the device's data on daily basis that is then sent to a google drive, a simple back-end infrastructure with no front-end interface and visualisation of data. The csv format does not offer any value to customers that do not have their own third-party data visualisation interface. This dissertation project will therefore focus on building an interactive web app dashboard for CompAir's live data, specifically for users measuring their own air quality. The research question that will be answered in this paper is: why and how to build an air quality interactive web app dashboard for consumers? In order to create this front-end interface, the back-end data infrastructure will need to be upgraded. This will be improved by a UCL classmate who is also working on his/her dissertation project with CompAir and will be responsible for data engineering and data storage as well as making predictions using machine learning. Together and individually we aim to create value out of CompAir's data.

The business objective of this interactive web app dashboard will be to inform CompAir's customers of the air quality in their living spaces. By interacting and analysing data from their CompAir devices through various visualisations, users will be able to make informed decisions to improve the air quality around them, ensuring a healthy standard of living. The value CompAir will be able to bring to customers will be augmented as instead of receiving a csv file (the current scenario), customers will be able to access an interactive dashboard with access to insightful visualisations. The technical objective of this project will be to create a dashboard in a web app, provide extensive interactivity for the user while keeping loading times below 20 seconds to provide a smooth user experience.

1.3 Project Management

To attain the objectives that have been set, strong project management and the use of project management tools were necessary. To schedule meetings with dissertation supervisors and the CompAir team, Microsoft outlook calendar was used as it best suited everyone involved. For shorter queries and communications, a WhatsApp group with the compare team was used. This was convenient for urgent or informal communications and facilitated fast interactions. The project management tool used to organise tasks and keep meeting notes was Trello. Trello provides a Kanban board, often used in agile software development, which was ideal for this project as the process of creating a dashboard is iterative rather than linear. Kanban boards usually have three columns in which tasks can be organised; To-do, In Progress and Done/Complete. Two additional columns were added to the project's Kanban Board; draft and meetings. Draft was used to track written-up parts that were drafted yet not finalized and the meetings column was used to keep meeting notes as seen in Figure 1. Trello tasks were also labelled using a traffic light system in accordance to the importance and urgency of the task. Important and urgent tasks were labeled in red, important tasks labeled orange and tasks that were not important or urgent were labeled in green as seen in Figure 2 (refer to Appendix for Trello Board link).

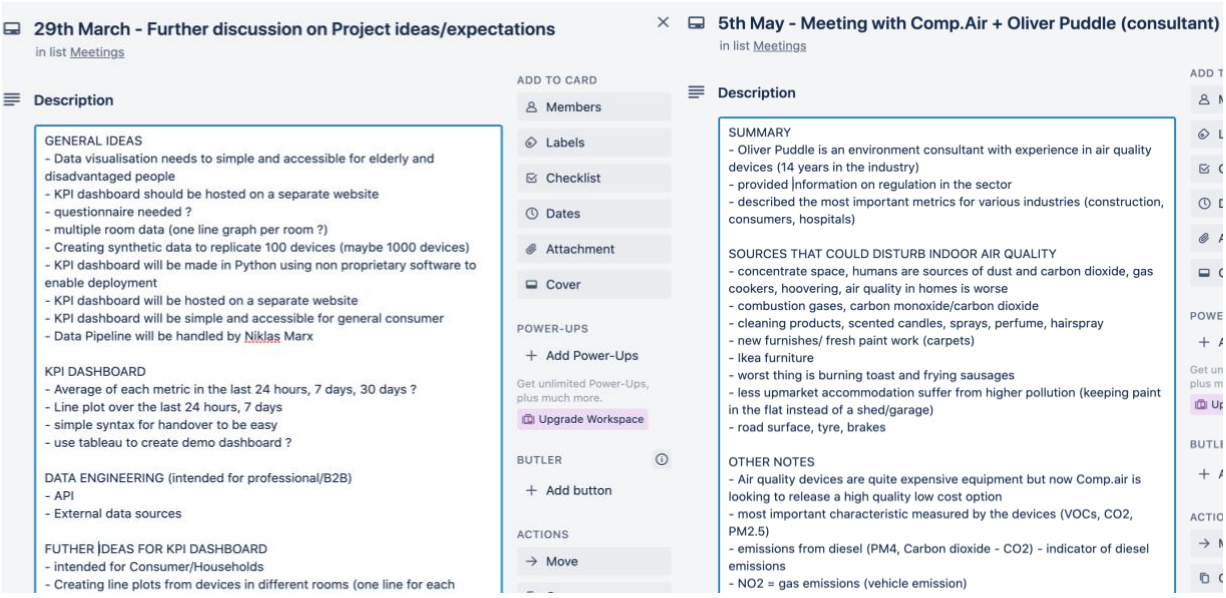

Figure 1. Example of Trello Meeting Notes

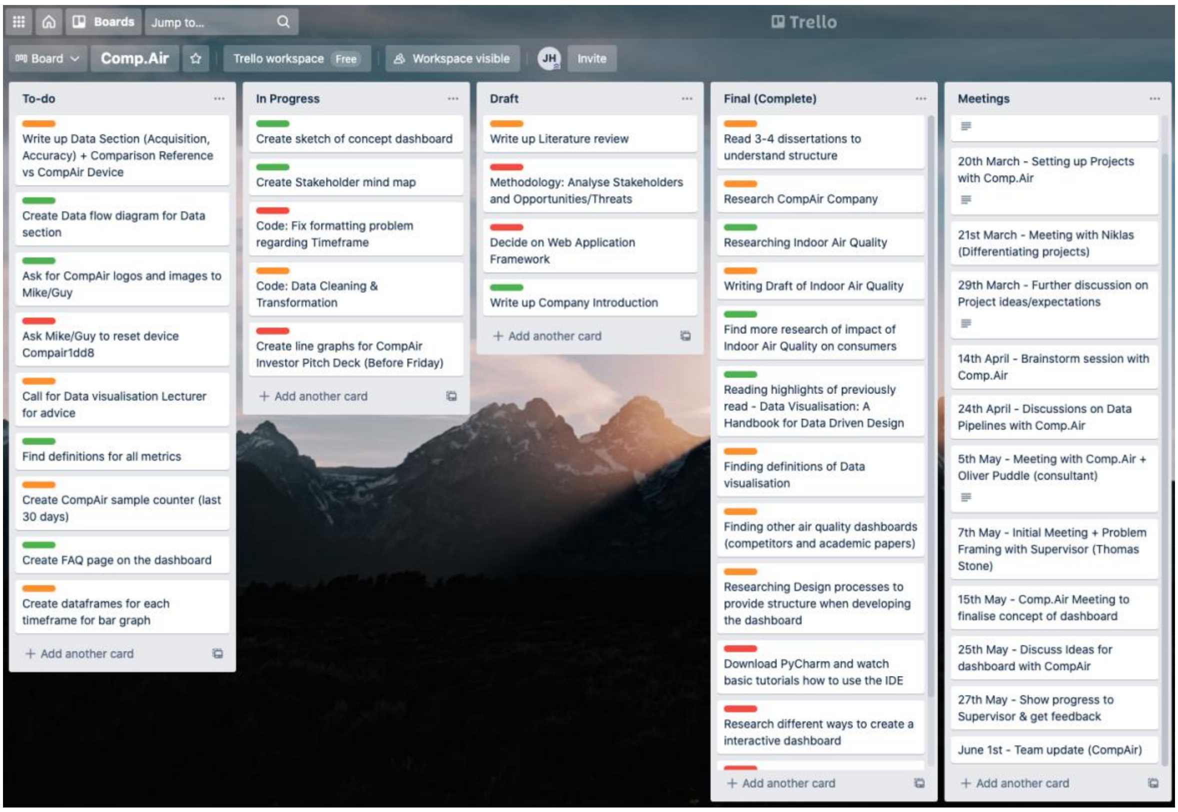

Figure 2. Trello Board

1.4 Outline of the Project

In the Literature Review section, relevant academic work and research will be reviewed and discussed. The literature review will examine current findings on indoor air quality, define data visualisation, explore low-cost air quality dashboards, outline data visualisation design processes/principles, explain relevant statistical models and introduce useful business analytical tools.

The Data section aims to outline the data infrastructure of the project; how the data is acquired, stored and sent from the CompAir devices to the dashboard. Additionally, data reliability and accuracy will be analyzed through comparisons of the CompAir prototype devices to reference grade equipment.

The Method will look into the decisions of the concept and design stage that led to the development of the dashboard. Various important elements that need to be considered before development will be discussed such as stakeholders, a SWOT analysis, the angle/frame of the dashboard and initial concept sketches. Following the concept stage, the evaluation of different web application frameworks will be presented to explain how Streamlit became the framework of choice for this project based on the project requirements and restrictions. The remainder of the method will be a walk-through of the entire web app with Python code and explanations of various development decisions.

In the results section, the final web app dashboard will be analysed and assessed through various use cases and hypothetical scenarios of how a consumer might use the dashboard to make educated decisions to improve indoor air quality. The web app will also be assessed in terms of deployment and performance.

Lastly, limitations and recommendations of the project will be evaluated, providing CompAir with next steps that may be taken to improve, scale and enrich the potential of the dashboard. To conclude, insights gained through the development of the dashboard and the business implication the dashboard may provide CompAir will be summarized.

2. Literature Review

2.1 Indoor Air Quality

To understand why indoor air quality is important and the impact an air quality dashboard can have on consumers, this literature review will explore various studies and papers that have researched the topic at hand.

2.1.1 Consumer Awareness

In developed countries, we spend 90% of our time indoors, with most of our exposure to air pollutants happening indoors rather than outdoors, despite the regulatory focus on the latter (Carslaw et al, 2017). Not only are we more exposed to air pollution indoors but the levels of indoor air pollutants are often 2 to 5 times higher than outdoor levels and occasionally more than 100 times according to the US Environmental Protection Agency (EPA, 2021). Poor indoor air quality caused 3.8 million premature deaths in 2016 with the most common illnesses being pneumonia, lung cancer and cardiovascular stroke (Renard et al., 2020). This is a significant reason why indoor air quality is important and of which most people are unware (Renard et al., 2020). Nevertheless, this lack of awareness of indoor air quality is improving with the COVID-19 pandemic as according to a survey from Carbon Lighthouse, 91% of respondents believe that indoor air quality is important in the prevention of the spread of the COVID-19 virus (Carbon Lighthouse, 2020). The survey also found that 52% of respondents would pay more to stay in a hotel with better indoor air quality and 74% say office indoor air quality data would make employees feel safer in their work environment. Consumers with CompAir devices would be able to track various indoor air quality metrics, helping them become more aware of the air quality around them.

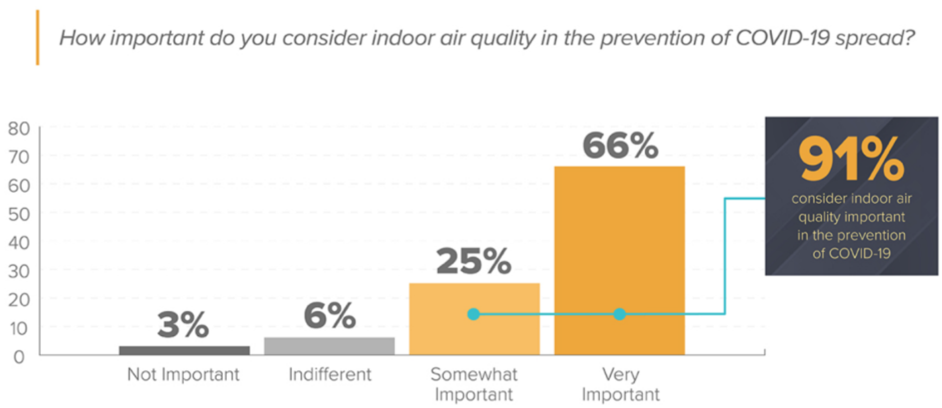

Figure 3. Consumer sentiment of importance of indoor air quality in prevention of COVID-19 (Carbon Lighthouse, 2020)

2.1.2 Metrics

The most important metrics to track indoor air quality are Volatile Organic Compounds (VOCs), Particulate Matter (PM) and Carbon Dioxide (CO2). VOCs are any organic compounds found as a gas in the atmosphere which act as a precursor to the formation of ozone and PM 2.5 (AQEG, 2020). Particulate Matter (PM) according to the Environmental Protection Agency (EPA), is a term used for a mixture of solid particles and liquid droplets found in the air that can be inhaled. The number that follows "PM" is the size of the Particulate Matter in micrometers, with the most common being PM1, PM2.5 and PM10. While Particulate Matter levels are highly impacted by external sources such as engine combustion and factories (King, 2019), PM2.5 concentrations have been found to increase during certain indoor activities such as smoking, grilling and frying from 3 to 90 times above usual levels (He, Morawska, Hitchins & Gilbert, 2004). Additionally, concentrations of VOCs are consistently higher indoors (up to ten times higher) than outdoors due to the variety of organic chemicals that are used as ingredients in household products (EPA, 2021). Paints, varnishes, candles as well as cleaning, disinfecting, cosmetic, and degreasing products all contain organic solvents that emit VOCs when used (Chin et al., 2014). While most VOCs are safe to breath in low concentrations, in higher concentrations they can cause short-term side-effects such as dizziness, headaches and lung irritation. Longer-term exposure levels of VOCs can damage the nervous system, liver, kidneys and lead to some types of cancer (NCAS, 2020). Furthermore, according to the Air Quality Expert Group (AQEG) Report 2020 which used 107 previously published scientific and technical articles on VOCs, emissions of VOCs in the UK have decreased by 72% from 1990 to 2017 mostly from a reduction in emission from road transport. While this is progress in improving outdoor air quality, VOCs emission from solvent use has been increasing, worsening indoor air quality. VOCs emissions from the use of PCPs (personal care products) and HCPs (household cleaning products) are now the most substantial contributor to overall VOCs emissions (Kansal, 2009). These trends can be observed in Figure 4 and 5, showing the increased emissions from Solvents in the UK.

According to the AQEG, to reduce the case of solvent VOC emissions, the levels of solvents in consumer products and/or the demand for these products must be eliminated or reduced. This can be achieved by improving consumer awareness of the causes of VOCs emissions.

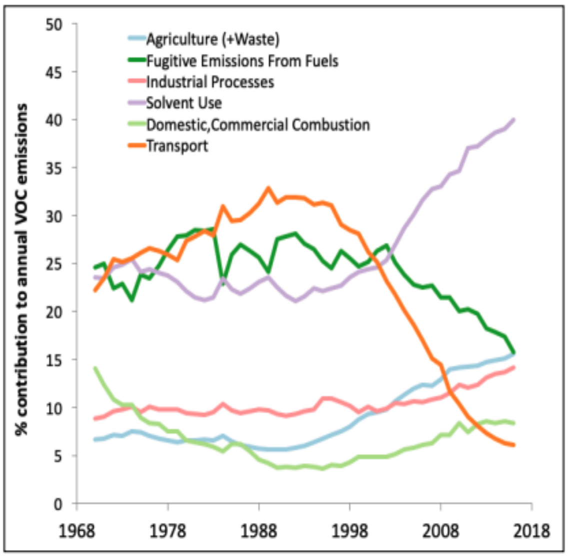

Figure 4. Trends in estimated national emissions of VOCs as a percentage of the overall annual national total, 1970 to 2016. (AQEG, 2020)

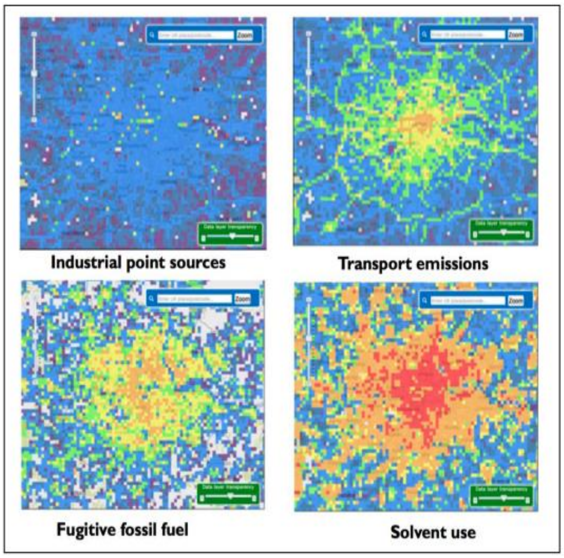

Figure 5. Annual estimate emissions in London for different VOC emission sectors. (AQEG, 2020)

2.1.1 Impact of Indoor air quality on Productivity and The Workplace

As most of people spend a large amount of time working indoors, ensuring that one has a healthy indoor environment to be able to work productively is essential.

In a study about the impact of indoor air quality on sick leave, Milton and Glencross (2000) studied the absence of 600 office workers for one calendar year. The study found that respiratory illness accounts for a large fraction of sick leaves. Furthermore, there are significant correlations between CO2 levels and illness-related absenteeism in the workplace (Erdmann, Steiner & Apte, 2002; Milton & Glencross, 2000) as well as in classrooms (Shendell et al., 2004). This is because CO2 is one of the main elements in an indoor environment due to human respiration and high levels of CO2 decreases cognitive performance (Ahmed, 2017). Now that the impact of indoor air quality on consumers has been explained, this paper will proceed to bring in relevant literature for the creation of the visualisations and dashboard.

2.2 Defining Data Visualisation

To better understand what the project aims to achieve, it is essential to define what is meant by data visualisation. Data visualisation is “the science of visual representation of ‘data’” (Friendly & Truman, 2001). This definition is intriguing as it defines data visualisation as a science rather than an art. While certain data visualisations may be artistic, the purpose of a large majority of visualisations are “the visual representation and presentation of data to facilitate understanding” (Kirk, 2019). Another more dated definition, “the use of computer-supported, interactive, visual representations of data to amplify cognition” (Card, Mackinlay and Shneiderman, 1999), brings importance to the technology used to create data visualisations. Nevertheless, the two latter definitions both emphasize the purpose to strengthen understanding and cognition, a central component of data visualisations. According to Andy Kirk’s definition there are 3 key pillars to facilitate understanding in visualisations:

- Perceiving: What do I see?
- Interpreting: What does it mean?
- Comprehending: What does it mean to me?

The dashboard that will be developed will follow these 3 essential pillars of visualisation to ensure the user understands the data from their CompAir device. The first pillar, perceiving, is important so that the user can easily understand what they see. This is dependent on the appropriate type of chart used, the range of values displayed and what the data represents. Interpreting, the second pillar of understanding, looks to transform the initial observations into meaning whether quantitative or qualitative. During interpretation, the user/viewer uses their knowledge of the subject to derive meaning from the visualisation. It is important to provide context of the subject if the viewer/user does not have a lot of prior knowledge of the subject to allow for interpretation. However, this pillar of understanding “will be significantly determined by factors external to the visualisation itself.” (Kirk, 2019). Lastly, comprehending is the phase where the viewer/user considers what the interpretation means to himself/herself. Comprehension is crucial to reinforce learning and inspire action which creates value for the user/viewer.

2.3 Air Quality Dashboards

In the academic literature there are a few that have studied indoor air quality visualisations using low-cost sensors. inAir, a “tool for measuring, visualising and learning about indoor air quality” developed by Sunyoung Kim and Eric Paulos (2010), was one of the first academic papers that demonstrated how to motivate action to improve indoor air quality with the use of data. Even though, this paper provides data visualisations such as a historical 24-hour line plot and a bar graph of most recent values as shown in Figure 6, the visualisation lacks interaction and visual appeal most likely due to the technological limitations at the time that this data visualisation was developed. The success of the study was in the awareness it brought to users. One user from the user trials explained that they were surprised how poor the air quality became when they deep-fried, knowing this motivated them to turn on the range hood when deep-frying (Kim & Paulos, 2010). The study also found that “users sometimes felt powerless” as they wouldn’t know exactly what to do to control the air quality. Providing contextual information will be a feature in the CompAir dashboard and that may help users to take actions in improving their indoor air quality.

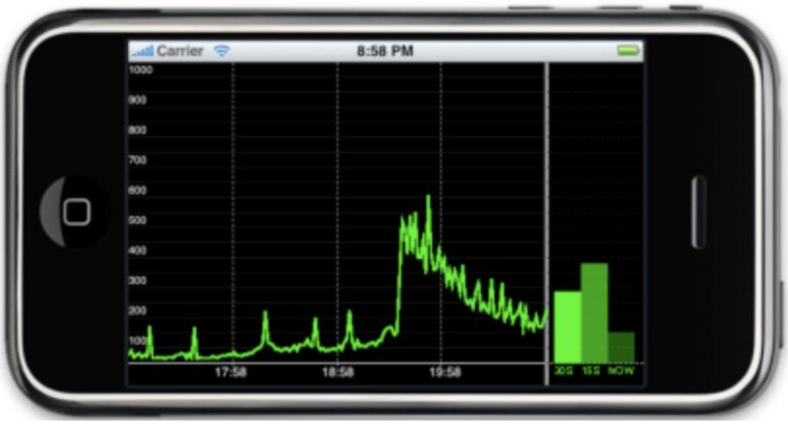

Figure 6. inAir air quality data visualisation on a mobile interface (Kim & Paulos, 2010)

In another paper, Kumar and Jasuja (2017), built a low-cost indoor air quality monitor using a Raspberry Pi minicomputer and displayed data on a dashboard. However, upon looking at their dashboard in Figure 7, the display cannot be interpreted as a visualisation as it does not facilitate understanding or cognition (Kirk, 2019; Card, Card, Mackinlay and Shneiderman, 1999). There is a lack of plots visualizing the data as the dashboard is separated in 5 blocks with the different metrics and their representative numerical value.

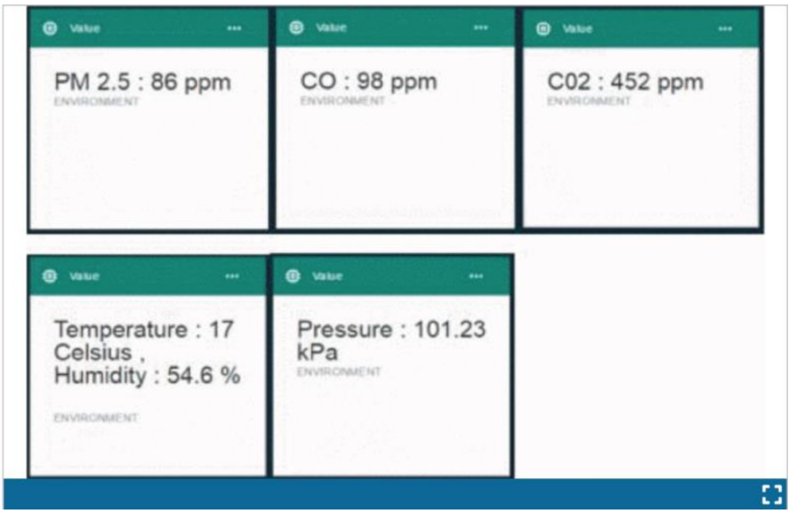

Figure 7. Low-cost indoor air quality monitor Dashboard (Kumar & Jasuja, 2017)

Aside from the academic literature, there are several companies around the world with similar business goals as CompAir; providing low-cost air quality sensors and data visualisation such as Clarity in the US and Atmos in India (Atmos, 2021; Clarity, 2021). The process of data processing, data storage and data visualisation in these companies are similar to what CompAir hopes to achieve after the completion of the UCL consulting/dissertation projects. For example, once Clarity's self-powered air quality devices capture the air quality measurements, the data is uploaded via 3G/4G to the Clarity Cloud, where the data is then processed and stored, after which the data is sent to the Clarity Dashboard and/or Clarity OpenMap web apps (Clarity, 2021). A similar data infrastructure will be achieved for CompAir's devices.

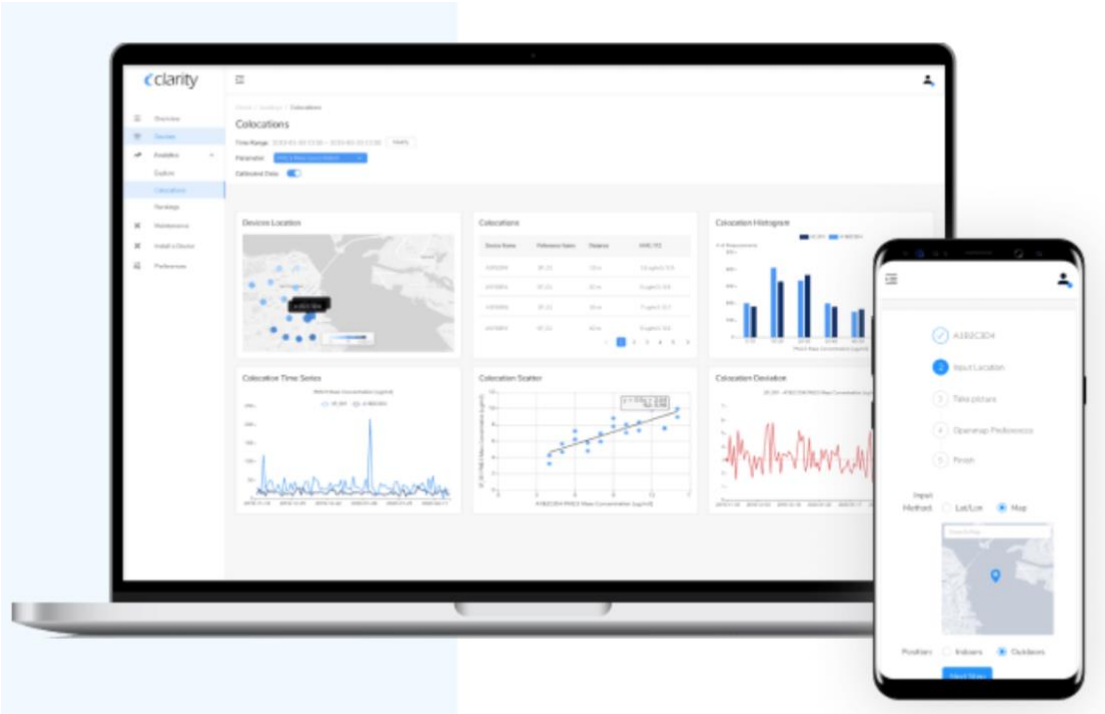

Figure 8. Clarity Dashboard (Clarity, 2021)

Dashboards have long been linked to business management, however, with the rise of smart homes and the internet of things, consumers are becoming more interested in analysing data and making informed decisions based on data. While there are numerous studies and academic papers diving deeply in performance, operational and management dashboards (Eckerson, 2011; Yigitbasioglu & Velcu, 2012 ; Rasmussen, Bansal & Chen, 2009) missing in the academic literature is how interactive air quality dashboards are developed for consumers and how the design and technical decisions are made for these types of dashboards. This gap in the literature will be filled by this paper.

2.4 Data Visualisation: Design Process & Principles

Design processes and principles will be helpful to guide decisions in the development of the dashboard. Kirk (2019) provides 4 stages for the design process of data visualisations:

1) Formulating the brief
2) Working with data
3) Establishing editorial thinking
4) Developing the design solution

The stage of formulating the brief is a way to establish the purpose of the visualisation and identifying the project's curiosity (Kirk, 2019). Formulating in this manner encourages focusing on decisions such as what data to gather, what analysis is required and what features of presentation will be employed. Stakeholders are also identified in this stage, which are people that are involved or that might influence the project. Stakeholders can range from customers, subject-matter experts, employers and investors. Furthermore, formulating the brief involves identifying the project's circumstances such as the restrictions, frictions and constraints (Kirk, 2019). Time and budget constraints are common; however, pressures can come from cultural, political, data and design factors too. Other considerations when formulating the brief are the expected quantities of deliverables, the medium and tone of the visualisation (reading tone or feeling tone). A reading tone, unlike a feeling tone, does not amplify any emotional devices and puts emphasis on optimizing precision and functionality. The goal of a reading tone is to facilitate the understanding of relationships and magnitude between values (Kirk, 2019).

Kirk's second stage: Working with Data consists of the acquisition, examination and transformation of Data. This stage is not only essential in creating data visualisations but also for any data project, and will be discussed in the next chapter of this dissertation.

The third stage: Establishing editorial thinking, is concerned with making informed decisions on the content that will be included in the visualisation. Editorial thinking is characterized by:

1) Angle - What view is most relevant for the data?
2) Framing – What data will be included/excluded?
3) Focus – Are any features of the data going to be emphasized?

The final stage in Kirk's structure to develop data visualisations is Developing the Design Solution which looks at the methods used to create the visualisation. It is important to note that while the design stages are presented as a linear step by step process, Kirk mentions that often projects follow an iterative cycle through these stages. In summary, the process presented by Kirk provides structure and helps to adapt to contextual factors of the project, however, Kirk's design process lacks practical guidance, focusing on theoretical frameworks.

The 10 principles of effective data visualisation (Midway, 2020) takes a more practical approach which is less wide-ranging and flexible yet can still help maximise the success of visualisations. The 10 principles are:

1) Diagram First
2) Use the Right Software
3) Use an Effective Geometry and Show data
4) Color *Always* Mean Something
5) Include Uncertainty
6) Panel, when Possible (Small Multiples)
7) Data and Models are Different Things
8) Simple Visuals, Detailed Captions
9) Consider an Infographic
10) Get an Opinion

There is overlap between both authors such as "Diagram First" principle and "Formulating the Brief" which both point to the importance of prioritizing information, envisioning a design and focusing on the message of the visualisation. Furthermore, the principle "Include Uncertainty" and in the stage "Working with Data", both authors explain the use of standard deviation, confidence intervals and errors to provide additional information to the audience of the visualisation. However, Midway's principles on "Using the Right Software", ignored in Kirk's framework, is an extremely relevant principle nowadays where the possibilities to creating visualisations are endless. For example, a dashboard can be created without code in Power BI, Tableau, or made from scratch using HTML and CSS. It is crucial to use adequate software and technical resources that are able to meet the projects requirements.

The Gestalt Principles developed by German psychologists Max Wertheimer, Kurt Koffka and Wolfgang Kohler, in the early 20th century, provide fundamental principles in how the human brain perceives visuals. The principles are – proximity, similarity, closure, continuity, figure-ground, common fate and connection (Britannica Concise Encyclopedia, 2008). According to Gestalt psychology, our brain needs to organise what we see into patterns to not overwhelm our brain (Opperman, 2020). The principle of proximity refers to perceiving objects that are close to one another as a group. As seen in Figure 9, there is a clear separation between the group of dots on the left and right. The groups are further separated on the right into 3 smaller groups (columns) of dots with the only difference between the group of the dots being the distance (proximity) from each other which our brain interprets as distinct groups. Proximity is a useful way to structure UX design without the use of hard borders.

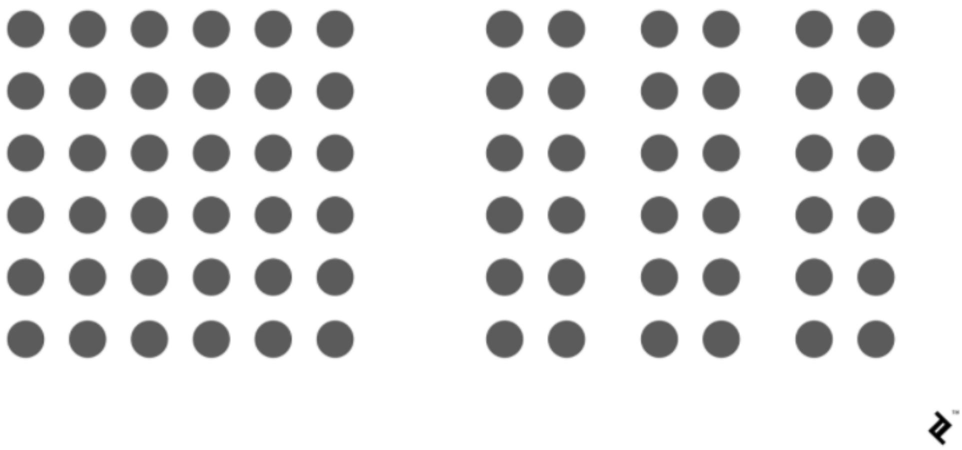

Figure 9. Principle of Proximity (Chapman, 2018)

Edward R. Tufte (1983) introduced the concept of "data-ink ratio" in his classic book, The Visual Display of Quantitative Information. When visualisations are presented in print form some of the ink is used to display data and some of the ink is used to display visual content that is not data. Nowadays, with digital visualisations, the word "ink" can be replaced with "pixel" to modernize the concept (Few, 2006). Tufte believed in maximizing the data-ink ratio. Meaning that most of the "ink" in a visualisation should present data-information and that "every bit of ink on a graphic requires a reason" (Tufte, 1983). As seen in the examples below in Figure 10, the visualisation on the left has a low data ink ratio as there is a lot of unnecessary content that distract from the data being presented. The background colour, labeled x-axis, shadow effects, grid lines, plot borders and legend are all redundant and do not add value to the visualisation. The goal is to reduce the non-data pixels and enhance the data pixels (Few, 2006). From Midway's previously discussed principles of effective visualisation the third principle: Use an Effective Geometry and Show Data, also highlight the importance of high data-ink ratios (Midway, 2020).

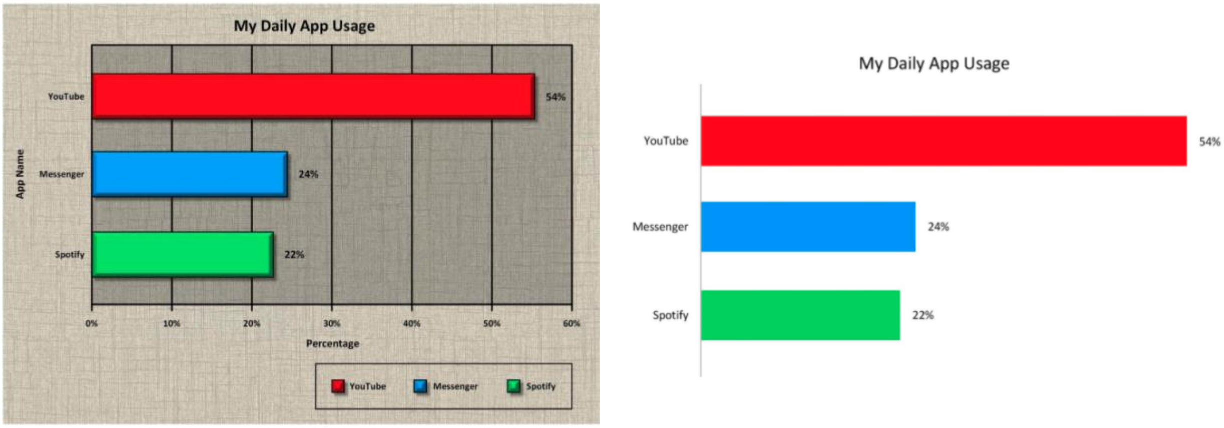

Figure 10. Example of a low data-ink visualisation and a high data-ink visualisation (Blaxell, 2020)

Another important consideration in any web or dashboard design is layout, as it can help put emphasis on some data above the rest (Few, 2006). The top-left and center of dashboards are the focus points and naturally direct the attention of the viewer. The focus to the top left is caused by the conventions of most western languages of reading from left to right while the centre is emphasized due to the "fundamental inclination of visual perception" (Few, 2006).

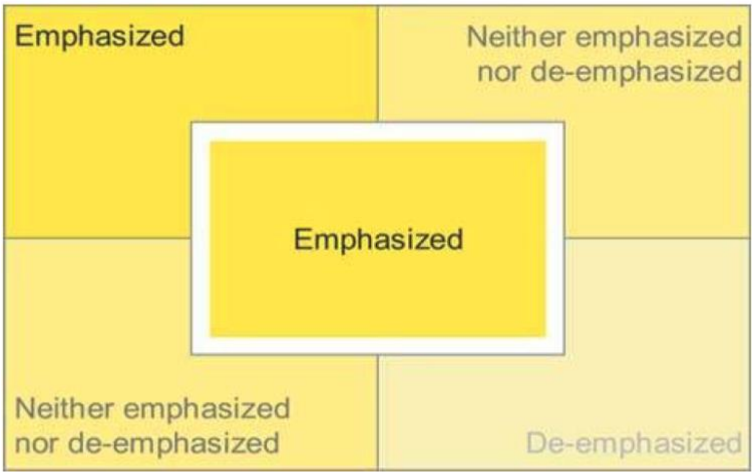

Figure 11. Visual focus points (Few, 2006)

Kirk's design stages, Gestalt's principles of visual perception, Midway's principles of effective visualisations, Tufte's theory of data ink and Few's focus points will help to make design decisions from the concept to the development of the CompAir dashboard.

2.6 Statistics

A common issue with data that is captured at high frequencies, such as 1 min intervals, is the amount of noise the data can create. Smoothing and denoising methods are useful to capture trends of noisy data by removing noise and/or fluctuations in the data. They are several denoising methods in the literature such as Stein's unbiased risk estimate regularization (Stein, 1981), total variation filtering (Rudin, Osher & Fatemi, 1992), wavelet denoising (Luo & Zhang, 2012) , Savitizky-Golay filtering (Sadeghi & Behnia, 2018) , Nadaraya-Watson kernel regression (Nadaraya, 1964) and Triangular (Bartlett) window smoothing (Bartlett, 1950). The Savitzky-Golay (SG) filter is one of the most widely used denoising methods (Sadeghi & Behnia, 2018)[25]. The filter requires three parameters: the data to be filtered, window length and the order of the polynomial function used to fit. The process of filtering and smoothing the data is achieved by convolution which is a mathematical operation that takes two functions to produce a third function. The third function expresses how the shape of one function is modified by the other function. The SG filter uses convolution to fit consecutive subsets of adjacent data points with low-order polynomials with the least squares method (Luo, Bai & Ying . 2005). The least square method aims to find the best fit for a set of data points by minimizing the sum of residuals. One of the greatest benefits of the SG filter is that it keeps features such as the maxima and minima which are often flattened by other denoising/smoothing methods (Siam et al., 2021). This helps maintain the precision of the data while removing unnecessary noise. Furthermore, when data peaks are expressed by only a few points, the SG method flattens peaks less than other methods such as the triangular Bartlett smoothing method even when using the same window width.

2.7 SWOT Analysis

A Strengths, Weakness, Opportunities, Threats (SWOT) Analysis is a framework used in evaluation of an organization, a plan, a project or a business idea. A SWOT analysis is separated into two dimensions, internal factors; strengths and weakness, and external factors; opportunities and threats (Gurel, 2017). A SWOT analysis will help to find a gap in the low-cost air quality monitoring dashboard market and provide an advantage to CompAir by creating a dashboard that is unique and differentiated from competition.

3. Data

3.1 Data Acquisition

The data acquisition flow from CompAir's devices to the dashboard is shown in Figure 12. The sensor in the device is the SPS30 Particular Matter (PM) Optical Sensor. The sensor measurements are captured using laser scattering and are converted using Sensirion's advanced algorithms to provide measurements of different PM Types (PM1, PM2.5 & PM10) as well environmental dust and other particles. The sensor provides readings every second when the device is on. A data pipeline created by a UCL classmate also working with CompAir, accesses the google drive where the data from the device is sent to as csv files. Once the data is assessed the API creates buckets for each device name and sends the csv files to the relevant AWS S3 buckets.

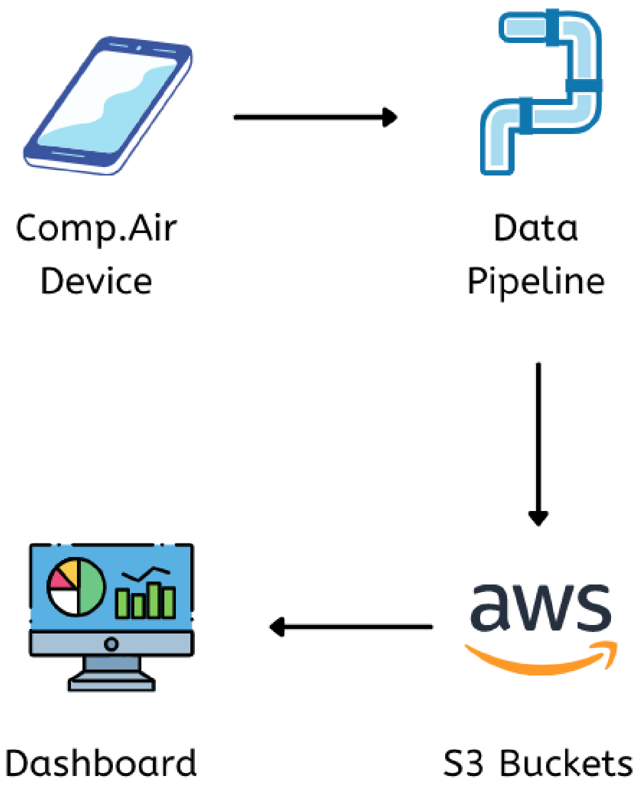

Figure 12. Data flow Diagram created with Canva

The data is stored in AWS S3 buckets, one S3 bucket per CompAir device with one bucket (allsensorsrecent) storing data from the last 30 days of all sensors.

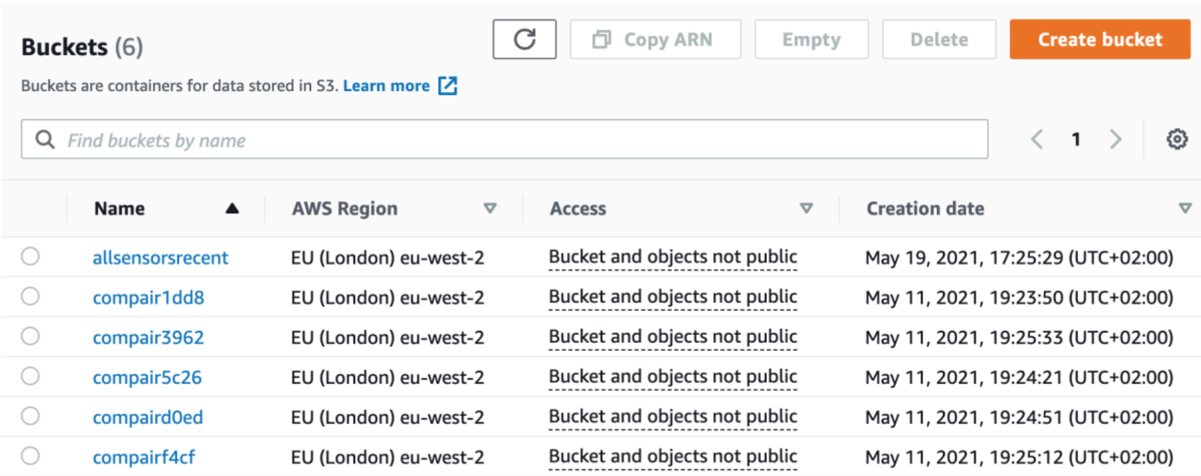

Figure 13. AWS S3 screenshot of data storage buckets

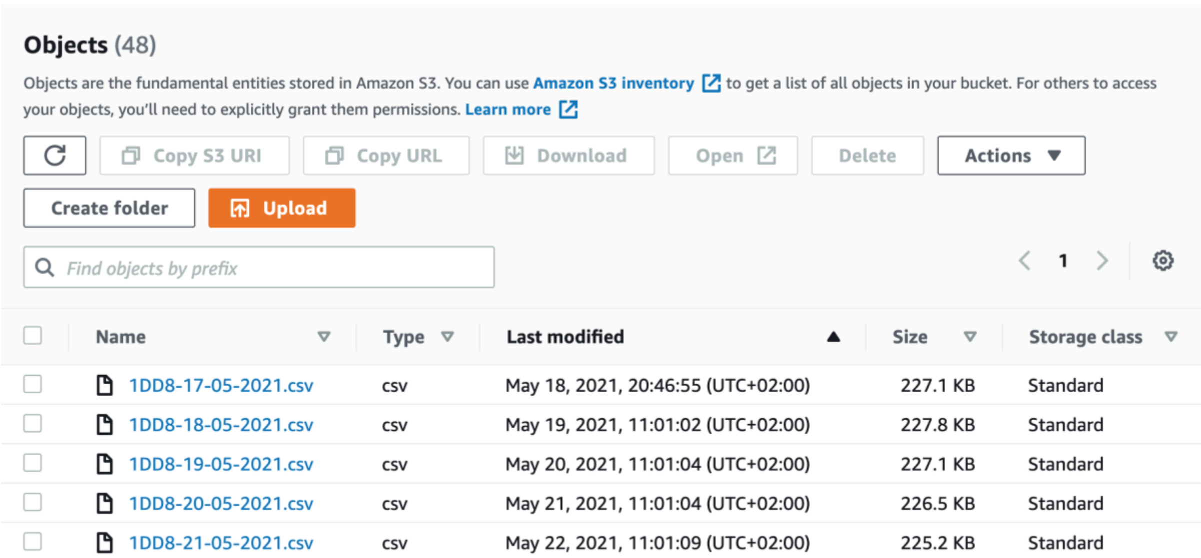

Figure 14. AWS S3 screenshot of objects in a S3 bucket

In each S3 bucket there is a csv file holding data for each day the device is active. How the data is accessed by the dashboard will be outlined in the method of this paper.

3.2 Data Reliability & Accuracy

In a study looking at low-cost air quality monitoring tools, sensors were evaluated in real-life conditions by comparing in the same location low-cost sensors alongside traditional FRM/FEM equipment (Clements et al., 2017). Federal Reference Method (FRM) & Federal Equivalent Method (FEM) are methods of sampling ambient air for air pollutants determined by the US Code of Federal Regulations (Code of Federal Regulations, 1970). Several academic papers have shared data on comparisons between reference (FRM/FEM) equipment and low-cost sensors (Wang, 2015), however, these have been on a limited selection of sensors (Austin et al., 2015; Zikova et al., 2017). These papers found evaluating sensors to reference sensors to be essential to ensure accuracy. Firstly, when deploying sensors, collocation of all sensors to a reference (FRM/FEM) sensor allows for calibration, an essential best practice. Secondly, declining performance can occur when devices age, with loss of accuracy and reliability over time. Evaluating a reference sensor can help determine if a long-term deployment device is still fit for purpose or if it requires replacing. Certain sensors in the studies were already found to have declining performance within the first year of operation (Williams et al., 2014). Comparing CompAir's device with a more expensive, reference sensor will provide an indication of accuracy, answering the essential question: how accurate and true to reality is the data coming from CompAir's devices?

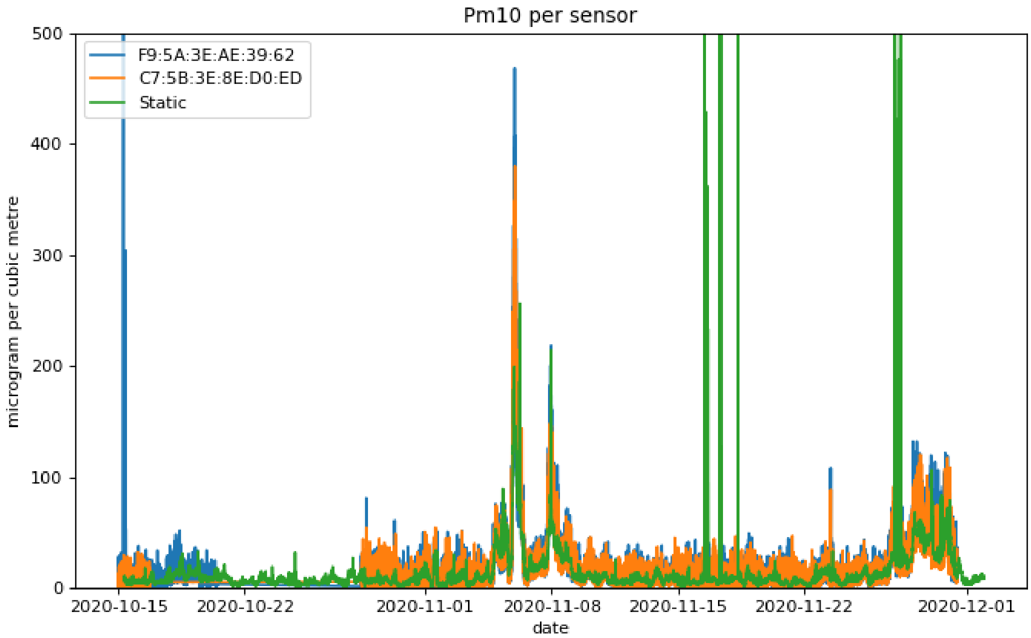

Figure 15. Line graph showing the PM10 of two CompAir devices and a Static device (reference sensor)

When observing Figure 15, we can notice similar readings across devices aside from a few outliers. Nevertheless, the CompAir devices fluctuates more than the static (reference) device indicating less accuracy and reliability which can be expected when considering the price difference of a few thousand pounds (£). Besides evaluating sensors with FRM/FEM, other recommendations for data collection are ensuring uniform and common data structure and format (Williams et al., 2014). Data cleaning and transformation will ensure the data to be uniform.

4. Methodology

The methodology behind the development of the dashboard will answer the “how” of the research question: why and how to build an air quality interactive web app dashboard for consumers?

4.1 Concept Stage & Design Process

4.1.1 Stakeholders

Understanding why and for whom the dashboard should be created for are important questions that need to be explored before starting to design the dashboard. Following Andy Kirk’s first design stage “Formulating the Brief”, stakeholders were identified as they are essential to developing a useful and valuable dashboard. These can be seen in the mind-map in Figure 16. The primary stakeholder for this project is the CompAir founders as the company is pre-revenue and therefore does not have any existing customers, the founders of the company will provide a lot of direction with the concept of the dashboard. Nevertheless, prospective customers, anyone that has a certain interest in air quality, are also key stakeholders as they can be useful to understand a customer’s viewpoint as well as potential user needs and wants. Furthermore, investors are stakeholders that can guide the requirements of the dashboard. However, at the time of writing this dissertation CompAir is still in discussion with potential investors and therefore their influence on the dashboard is limited. Lastly, subject matter experts will be valuable to understand the industry’s trends and best practices. Throughout the development of the dashboard, meetings will be held with an Air Quality Consultant from Dust Scan Ltd to provide useful insights on the industry (meeting notes found on Trello Board).

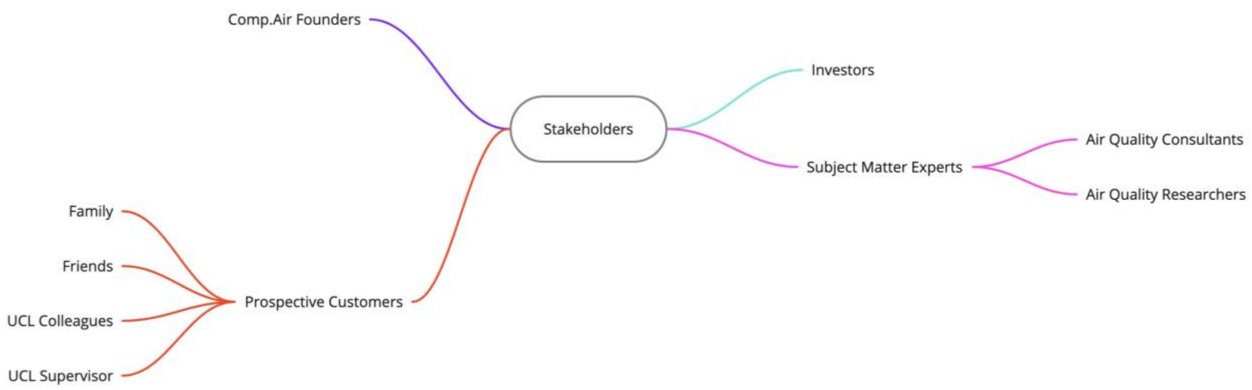

Figure 16. Mind map of Stakeholders created with Mira

4.1.2 Web Application Framework

Taking into consideration Midway's second principle, "Using the Right Software", the choice of web application framework to develop the dashboard was a crucial decision. A web application framework is a set of resources and tools for software developers to build and manage web applications (Luna, 2013). The decision to use a python library rather than creating a website from scratch using HTML and CSS was made by the time restrictions of the project and this thesis' author's personal skill and experience with different programming languages. Streamlit also facilitates creating websites/dashboards that have interactive components which is an important feature the dashboard requires to differentiate itself from competitors. There are currently several open-source Python dashboard web application frameworks in the industry such as Streamlit, Dash, Voila and Panel. After researching all four dashboard frameworks and understanding which suited the project best, Streamlit seemed the most appropriate (refer to Appendix for in-depth comparison of all four ecosystem).

Streamlit provides the simplest framework to code in pure Python and the ability to deploy the dashboard online for free with deployment options such as Streamlit Sharing and Heroku. Another important criterion that helped the decision was the ease of maintenance which is best in Streamlit and Voila. Furthermore, Streamlit offers a concise changelog[1] which helps to monitor all changes ongoing in their ecosystem, which Dash and Panel both lack. When the dissertation is complete a hand over will be required, in which the successor, a CompAir employee/contractor, will need to easily understand and make changes to the dashboard based on evolving needs.

The drawbacks of Streamlit compared to the other dashboard frameworks is the lack of Jupyter Notebook support, which required this thesis' authors to learn how to use PyCharm as an IDE. An integrated development environment (IDE), is a user interface to code and build applications (Fincher, 2018). Secondly, Streamlit does not provide any true authentication mechanisms which may be required in the future should CompAir want greater security for their customers. However, as CompAir is pre-revenue with a prototype product, the dashboard does not currently require authentication.

4.1.3 SWOT Analysis

Carrying out a SWOT analysis of CompAir in the low-cost air quality monitoring device industry will help understand which strengths the dashboard should accentuate, the opportunities it can work towards and the weaknesses and threats it should overcome.

---

[1] https://discuss.streamlit.io/tag/release-notes

Strengths

The Strengths of CompAir's device is the amount of metrics the sensor tracks, as in the industry other low-cost sensors often only measure Particulate Matter (PM1, PM2.5 and PM10). For example, Clarity, a competitor of CompAir, only measures Particulate Matter and Nitrogen Dioxide (NO_2). The various metrics measured by CompAir is a strength that can be utilized in the dashboard by providing the option to analyze separately each metric with a dropdown menu.

Opportunities

As CompAir is pre-revenue and still modifying their prototype at the time of writing this dissertation, CompAir can adapt to user needs by extensive market research. This opportunity requires the web application framework used for this project to be flexible, allowing subsequent employees/contractors of CompAir to alter the code to better suit CompAir's changing business direction and objectives.

Weaknesses

Important to a project's concept stage is understanding the constraints and restrictions (Kirk, 2019), the weakness in a SWOT analysis. The most significant constraints for this project are Time, Budget and Data Storage. In regards to Time, as this was a dissertation/consulting project the time constraint was one of 2-3 months. Budget wise, as CompAir is a pre-revenue company, the costs associated with the dashboard had to be minimal, restricting the selection of dashboard ecosystems and deployment options that could be chosen. Lastly, as CompAir had no data pipeline to automate the process of collecting the data from the devices, a UCL colleague working on his own Data Engineering/Machine Learning dissertation with CompAir had to build APIs, data pipelines and data storage infrastructures before the dashboard could be developed. This delayed the development stage of the dashboard by a month.

Threats

CompAir's primary threat is that they do not have first mover advantage. A first-mover advantage is "a firm's ability to compete against other players in an industry as a result of being first to market in a new product category" (Suarez & Lanzolla, 2005). The dashboard will need to be differentiated from other players in the market. This will be achieved by providing interactive features and recommendation/contextual information on the various metrics.

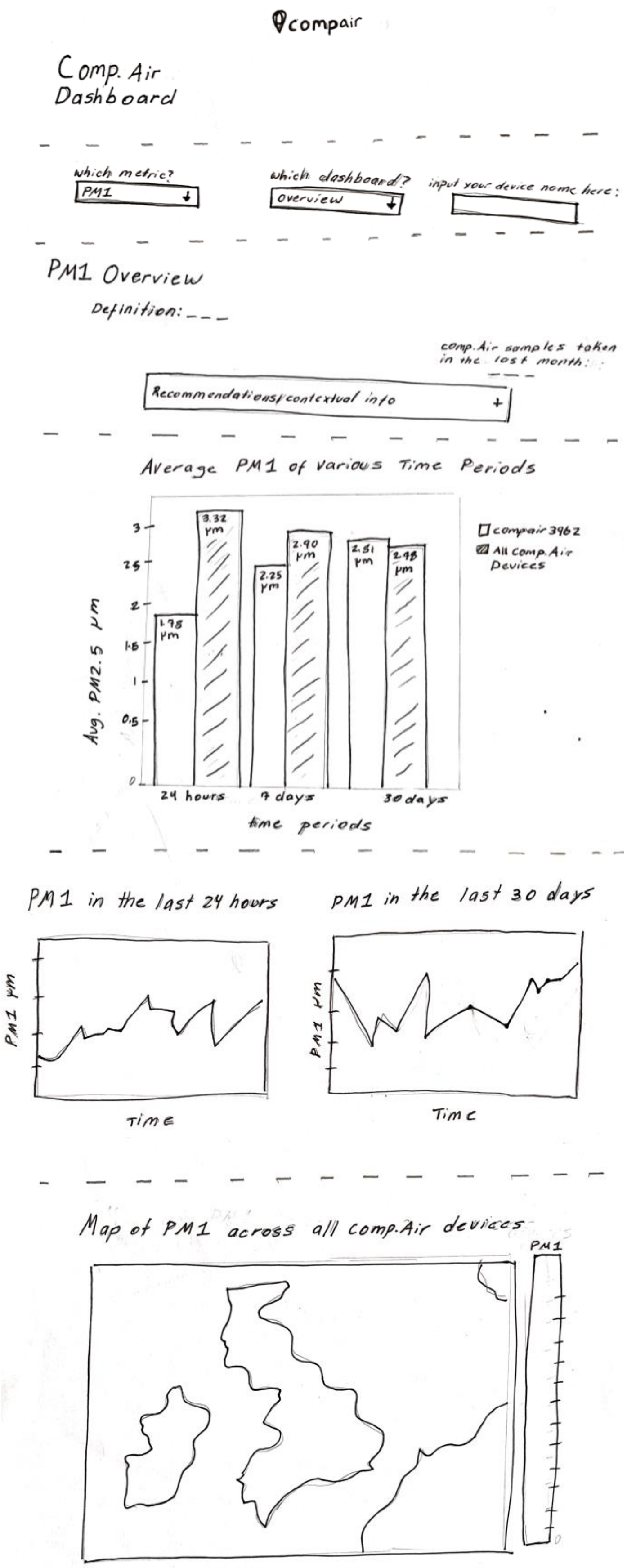

Figure 17. Concept Dashboard sketch

4.1.4 Sketching

Following Midway's (2020) first principle of effective data visualisations: Diagram First, initial sketches were created to understand what the end results should resemble. These sketches will help to break the project into smaller components which will be useful to avoid becoming overwhelmed when programming the dashboard. The decisions concerning the layout of the dashboard are made using Few's theory of focus points (2006). As the top-left and center are the focus points, the metric definition and a recommendation expander are placed in the top right corner as they provide important contextual information while the bar plot is centered to bring focus to the visualisation. The principle of proximity is also used by putting plots close to each other, for the user to perceive them as a group. While dashboards often used hard boarders to group elements, these were avoided by grouping them by row as can be observed in the sketches. The first row provides input information such as dropdowns and input boxes. The second row provides contextual information such as a definition, recommendation dropdown and a sample counter. The third, fourth and fifth row provide different types of plots.

4.1.3 Establishing Editorial Thinking: Angle & Framing

In the 3rd stage of Kirk's design process for data visualisation, establishing editorial thinking, the angle and framing of the visualisation is decided. The angle of the dashboard will look at how the values and the average of those values change over time, how the values compare themselves to other values and how the data looks spatially on a map. These angles will help users understand how the air quality of their device compares to a general standard, helping users understand if their air quality is above or below average. Furthermore, causes of poor air quality can be identified by examining changes over time and pinpointing the time of the activity that caused a change in the air quality metric. Framing, deciding what to include/exclude in the dashboard, will be achieved through the use of dropdowns, allowing the user to interact with the dashboard and deciding what information they wish to see. The framing will be in regards to what device, which metric and what timeframe they wish to analyze.

4.2 Setup

The first stage of development was importing the relevant libraries such as streamlit, pandas, numpy, maplotlib, plotly, seaborn and boto3. This was followed by setting up the dashboard by rounding and formatting all numbers to two decimal places, uploading logo images, and creating containers to make space for the header. The header colour was chosen to remain Black as the CompAir's logo provided is in black and white. The layout was set to wide to allow to fit more on the page, as this created problems with space early on in the process. CompAir's icon logo, an air balloon, was added as the page icon as seen in Figure 19, to personalize the tab. Furthermore, when users bookmark this dashboard, the icon will also be their icon bookmark (Figure 20). Additionally, a Streamlit API was used, st.beta_columns, to create structured space similarly to HTML block elements. Beta columns are used throughout the dashboard and in this case, 3 separate blocks were created; "col1" with a size of 10, "mid" with a size of 6 and "col2" with a size of 20. The images were provided by CompAir and are hosted on ImgBB, a free image hosting and sharing service. Once an image is uploaded to an account, a direct link is provided that was used with Streamlit's image API, st.image, to display the image on the dashboard.

```
# SETUP ------------------------------------------------------------------------------------

#uploading logo images that will be used & config
st.set_page_config(page_title='Comp.Air Dashboard',
                   page_icon='https://i.ibb.co/rbZyb0N/icon3.png',
                   layout="wide")

#creating space between header and image
col1, mid, col2 = st.beta_columns([10,6,20])

#creating header and image
with col1:
    st.title('Comp.Air Dashboard')
with col2:
    st.image('https://i.ibb.co/PwCKwyp/header.png', width=90)
```

Figure 18. Setup code

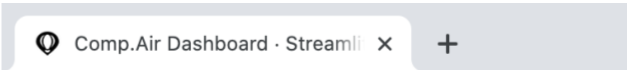

Figure 19. Custom Page Tab

Figure 20. Bookmark bar logo

Figure 21. Header

The setup of the dashboard was continued (Figure 22) by creating more beta columns for the next row below the header, the space length is not specified as the 3 indicates the number of blocks. Dropdown columns were created for the user to select which metric and which dashboard they wish to see as seen in Figure 23. The interactive nature of the dashboard was important for the user to be able to make their own analysis. This enables the user to decide what he/she want to perceive, a key pillar to facilitate understanding in visualisations (Kirk, 2019). The last interactive feature on the dashboard requires users to input their device name in a textbox. Setting the layout to wide with the Streamlit' configuration API in Figure 18 did not get the desired results and therefore the padding of the dashboard was adjusted to further enhance the use of space. Lastly, the developer menu was removed as this is not something users require. Removing excess functionality is important to not confuse and complicate the interface for the user.

```
# creating beta columns to organise the input functions
name_cols = st.beta_columns(3)

#options in alphabetical order
option2 = name_cols[0].selectbox("Which Metric?", ('ALL', 'Air Pressure', 'AQI'

option = name_cols[1].selectbox("Which Dashboard?", ('Overview','Comparison', '

user_input = name_cols[2].text_input('Input your device name here:')

#adjusting the padding of the dashboard to enhance the use of space
padding = 3
st.markdown(f""" <style>
    .reportview-container .main .block-container{{
        padding-top: {padding}rem;
        padding-right: {padding}rem;
        padding-left: {padding}rem;
        padding-bottom: {padding}rem;
    }} </style> """, unsafe_allow_html=True)

#removing the developers menu (should not be there for users)
st.markdown(""" <style>
#MainMenu {visibility: hidden;}
footer {visibility: hidden;}
</style> """, unsafe_allow_html=True)
```

Figure 22. Continuation of setup code

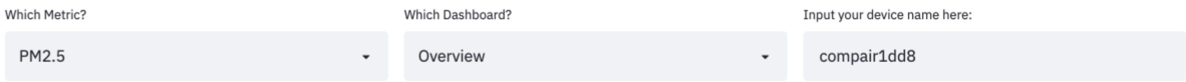

Figure 23. Interactive inputs

4.4 Data Acquisition

```
# GETTING DATA ---------------------------------------------------------------------------

if user_input != "":

#caching the data, will only run the function if it has not been run before
    def get_data():
        s3 = boto3.client("s3", \
                         region_name="eu-west-2", \
                         aws_access_key_id="AKIA3C5IQYEMH7773L7F", \
                         aws_secret_access_key="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")

        response = s3.list_objects(Bucket= user_input)

        df_list = []

        for file in response["Contents"]:
```

Figure 24. Data acquisition code

In Figure 24, an IF statement is used for the function get_data to run only when the user inputs a device name in the input box. The get_data() function uses boto3, a AWS Software Developer Kit for Python. The aws_secret_access_key and the aws_access_key_id directs the boto3 client to the CompAir AWS account to access the data. The boto3 API is then used to access the relevant bucket that has been entered by the user in the user_input variable. Once the device name/bucket name has been set, the function will try to read all csv files in that bucket and insert them into a list called df_list. As seen in Figure 25, error handling is used to avoid any discrepancies in the data causing an error. The output of the function is a dataframe of all the csv files in the relevant bucket (device name).

```
# error handling to check if any files are missing from the bucket (it will ignore missing files)
            try:
                obj = s3.get_object(Bucket= user_input, Key=file["Key"])
                obj_read = obj["Body"].read()
                obj_df = pd.read_csv(io.BytesIO(obj_read))
                df_list.append(obj_df)

            except ClientError as ex:
                if ex.response['Error']['Code'] == 'NoSuchKey':
                    pass
                else:
                    raise

                obj = s3.get_object(Bucket= user_input, Key=file["Key"])
                obj_df = pd.read_csv(obj["Body"], error_bad_lines = False)
                df_list.append(obj_df)


        df_n = pd.concat(df_list)

        return df_n

    df_n = get_data()
```

Figure 25. Continuation of data acquisition code

The get_data function is repeated in the get_data_all function but instead of using the user_input variable to access one specific bucket which the user has inputted, this function accesses the *allsensorrecent* bucket, to get data of all buckets from the last 30 days.

4.5 Data Cleaning & Transformation

Address	Date	Time (UTC)	Humidity	Temperature
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:00:23	26.43	33.65
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:01:23	26.64	33.64
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:02:23	26.43	33.64
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:03:23	26.58	33.63
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:04:23	26.49	33.62
F9:5A:3E:AE:39:62	2021-06-13 00:00:00.000	00:05:23	26.84	33.59

Figure 26. Screenshot from a CSV file of one of the devices

The csv files that are created every day for each device have a few formatting issues which make creating time-series computations and visualisations difficult. As seen in Figure 26, there are two separate columns that pertain to time: a "Date" column and a "Time (UTC)" column. Using a function that has been defined as *clean*, the data will be cleaned and transformed (Figure 27). A function is used as it enables this process to be replicated across different sets of datasets that will be used for the dashboard. The function is cached using Streamlit's cache API, st.cache. Caching helps improve performance by running a function and storing it in local cache. The next time the same cached function is called, and the inputs have not changed, the code will be able to skip the function and return the output previously stored in the cache.

```
@st.cache
def clean(dataf):


    dataf["Date"] = pd.to_datetime(dataf.Date)
    dataf["Date"] = dataf["Date"].astype(str).str[0:10]
    dataf['Time(min)'] = dataf["Time (UTC)"].astype(str).str[:-3]
    dataf['Time(min)'] = dataf['Time(min)'].str[:-2] + "00"

    dataf = dataf.drop(["Unnamed: 0", "Timestamp", "Time (UTC)"], axis=1)

    #concatenating the Date (in datetimeformat) and the Time(min) into one column
    dataf["Timestamp"] = pd.to_datetime(dataf["Date"].astype(str)+" "+dataf["Time(min)"].astype(str))

    #setting this new column as the index
    dataf = dataf.set_index("Timestamp")
    dataf.sort_values("Timestamp", inplace=True)

    #renaming cetain columns to facilitate the use of the input
    dataf.rename(columns={'Pm25': 'PM2.5', 'Pm1': 'PM1','Pm10': 'PM10' }, inplace=True)

    return dataf
```

Figure 27. Data cleaning function code

The *clean* function firstly reformats the "Date" column to a *datetime* format, converts the date to a string and keeps only the first 10 characters. The first 10 characters of the string are the date and this reformatting removes the redundant 00:00:00.000 apparent in every row in that column (as seen in Figure 26) which does not provide any information. Furthermore, the "Time (UTC)" column is reformatted into minutes by removing the 23 seconds of every row and replacing it by 0 seconds. This was done to simplify and round down the minutes. Before these two transformed columns are concatenated a few columns were dropped to avoid overlap of information between columns. The "Date" column and the "Time (min)" column are then concatenated into one column "Timestamp". The "Timestamp" column is then set as the index of the dataset and sorted by time. Lastly, the Particulate Matter metrics are renamed by capitalizing the whole string, making the metrics more visually uniform.

When the first version of the dashboard was created the line-plots were unreadable as seen in Figure 28. While adding a smoothed line helped make the overall trend clearer, the variability of the data was too high especially on the "Last 30 days" line plots. Therefore, minute data points were transformed into hourly averages (Figure 29). This helped improve the user experience by making the line plots clearer and trends recognizable as seen in Figure 30. The SG smoothing line was also removed from the 24-hour line plots as the trends were clearer with the hourly average. Other averages were tested such as 15 min averages and 30 min averages however these still created incomprehensible 30-day line plots.

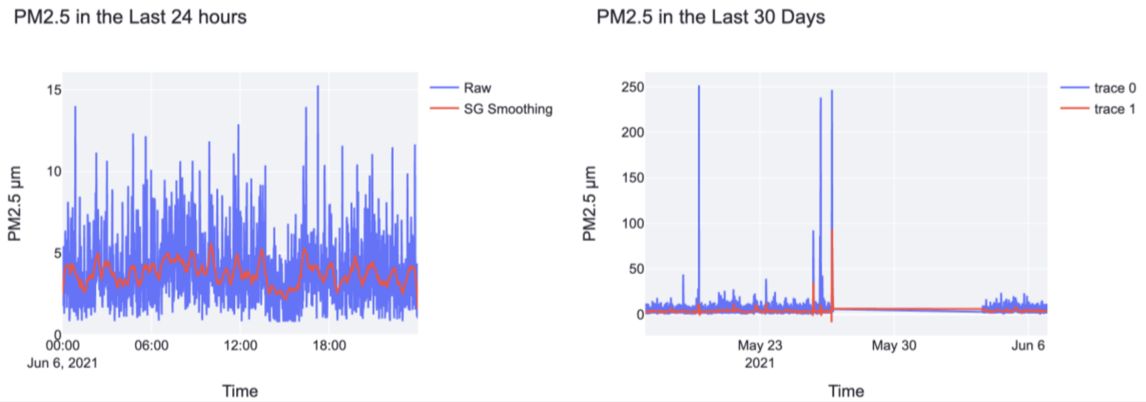

Figure 28. First version of the dashboard with minute data points

```
# DATAFRAME TRANSFORMATION ---------------------------------------------------

    # changing from a frequency of data point per minute to the hourly average
    df_n = df_n.groupby([df_n.index.values.astype('<M8[h]')]).mean()
```

Figure 29. Changing frequency of data point from minute to hourly average

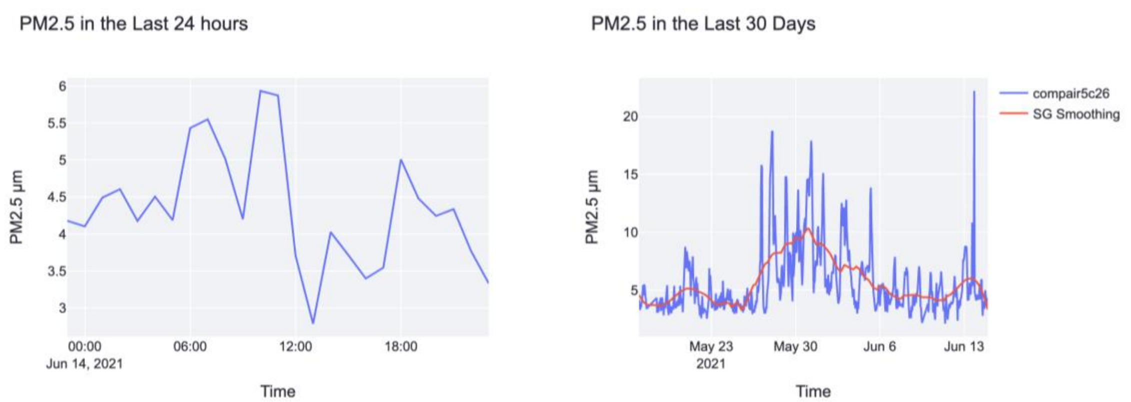

Figure 30. Final version of the dashboard with hourly average data points

Furthermore, 3 separate dataframes were created for different time periods: the last 24 hours, the last 7 days and the last 30 days. These "time period dataframes" were also created for the dataframe containing all devices (*allsensorsrecent*). For each time period the mean was found and assembled into a dataframe. Theses dataframes were used to create a bar graph.

```
#data from the last 24 hours
hour24 = df_n[df_n.index>=(most_recent_date-dt.timedelta(hours=24))]
hour24_all = df_1[df_1.index>=(most_recent_date-dt.timedelta(hours=24))]

#data from the last 7 days
day7 = df_n[df_n.index>=(most_recent_date-dt.timedelta(days=7))]
day7_all = df_1[df_1.index>=(most_recent_date-dt.timedelta(days=7))]

#data from the last 30 days
day30 = df_n[df_n.index>=(most_recent_date-dt.timedelta(days=30))]
day30_all = df_1[df_1.index>=(most_recent_date-dt.timedelta(days=30))]

#grabbing the average of all timeframes
m24 =hour24.mean()
m24_all = hour24_all.mean()
mday30 = day30.mean()
mday30_all = day30_all.mean()
mday7 = day7.mean()
mday7_all = day7_all.mean()

#creating a dataframe of all the averages from the different timeframes
d = {'24 Hours':m24, '7 Days': mday7, '30 Days': mday30, }
dfavg = pd.DataFrame(data=d)

all = {'24 Hours':m24_all, '7 Days': mday7_all, '30 Days': mday30_all, }
dfavg2 = pd.DataFrame(data=all)

# rouding decimals to 2 places
dfavg= dfavg.round(decimals=2)
dfavg2 = dfavg2.round(decimals=2)
```

Figure 31. Time-period Dataframe transformation code

4.6 Dashboards

The header of the dashboard is dependent on the input from the user in the two dropdowns. The PM1 Dashboard will be walked through as an example as all metrics follow a similar code structure.

```
# CREATING HEADERS FOR DIFFERENT DASHBOARDS --------------------------------

    if option == "Overview" or option == "Comparison" :
        st.header(option2 + " " + option)

    if option == "FAQ":
        st.header(option)
```

Figure 32. Creating headers code for dashboards

4.6.1 Overview

The first dashboard that can be selected is called Overview. This dashboard provides an overview of the metric that is selected (in this example PM1). The first feature of the dashboard is a definition and contextual information of the metric to provide comprehension of the data, crucial to reinforce learning and inspire action (Kirk, 2019). After the "PM1" and "Overview" inputs are selected in the dropdowns, the st.write, st.markdown and st.beta_expander Streamlit APIs are used to write the definition of the metric and recommendations/contextual information. Streamlit's beta expander API is useful as it allows to neatly organise and contain large amounts of information into an expander as seen in Figure 33. Recommendations and contextual information are provided as research found that users often feel clueless in the actions they can take to control indoor air quality (Kim & Paulos, 2009). This crucial feature fulfills Andy Kirk's 3rd pillar of facilitating understanding in data visualisations, comprehension, which answers the question of: what does this data mean to me?

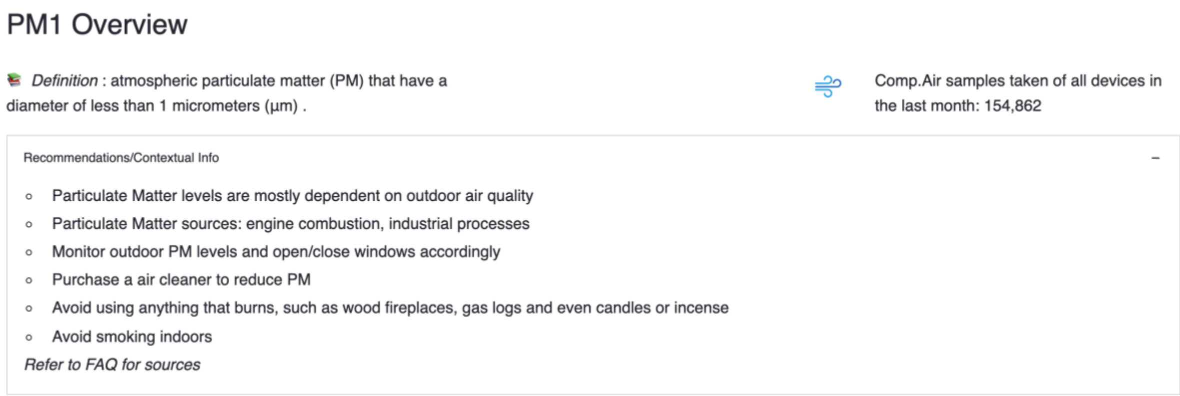

Figure 33. Demo for first section of the overview dashboard

```
# PM1 DASHBOARD -------------------------------------------------------------------------------------------

    if option2 == 'PM1':

        if option == 'Overview':

            st.write('')

            row1_1,space, row1_2, row1_3 = st.beta_columns((1.5,1,0.2,1))

            with row1_1:
                st.markdown(''' :books: * Definition *: atmospheric particulate matter (PM) that have a diameter of less

            with row1_2:
                st.image('https://i.ibb.co/T8887H7/air.png', width=35)

            with row1_3:
                st.markdown('''Comp.Air samples taken of all devices in the last month: ''' + scount)

            about = st.beta_expander(
                'Recommendations/Contextual Info')

            with about:
                '''
                - Particulate Matter levels are mostly dependent on outdoor air quality
                - Particulate Matter sources: engine combustion, industrial processes
                - Monitor outdoor PM levels and open/close windows accordingly
                - Purchase a air cleaner to reduce PM
                - Avoid using anything that burns, such as wood fireplaces, gas logs and even candles or incense
                - Avoid smoking indoors

               _Refer to FAQ for sources_
                '''
```

Figure 34. Overview dashboard code

As part of the data visualisation design process, feedback was gathered continuously through meetings with the CompAir Founders. One request from CompAir's team was to include a sample counter that would record how many samples of air have been taken by all CompAir devices in the last 30 days. This was added by counting the index of the dataframe df_1 which includes the data of all devices from the last 30 days, created from the AWS bucket *allsensorrecent.*

```
# CREATING AIR MEASURED COUNTER -------------------

    count = len(df_1.index)

    scount = '{:,}'.format(count)
```

Figure 35. Comp.Air sample counter code

Before plotting the averages, the dataframes were transposed. This facilitated plotting the dataframes on a bar plot (Figure 36). The dataframe index (Time) was also put into a separate column to be able to access that column for the bar plot.

```
st.write('')

row1_1, row1_2, row1_3 = st.beta_columns((0.75,3,0.75))

temp = dfavg.loc[dfavg.index == 'PM1']
temp2 = temp.T

temp2['Time'] = temp2.index

temp3 = dfavg2.loc[dfavg2.index == 'PM1']
temp4 = temp3.T

temp4['Time'] = temp4.index
```

	24 Hours	7 Days	30 Days
Air Pressure	1,006.7100	1,002.7900	1,009.5200

	Air Pressure	Time
24 Hours	1,006.7100	24 Hours
30 Days	1,009.5200	30 Days
7 Days	1,002.7900	7 Days

Figure 36. Transposing dataframe code and result

For the plots a Python library called Plotly was used. The reason why Plotly was used for the visualisations over other libraries such as Matplotlib and Dash is because Plotly provides built-in interactive features in their plots. This can be seen in Figure 37, of which all interactive features are shown. These can be found in the top right corner of every Plotly graph. The options are to take a screenshot of the plot in PNG format, pan, box select, lasso select, zoom in, zoom out, reset the axes and display the plot in full screen. These interactive features will allow more flexibility in the analysis the users wishes to perform by specifying what timeframe they wish to analyze, facilitating the framing the user desires (Kirk, 2019).

Figure 37. Interactive features of Plotly graphs

When creating the bar plot an idea came to mind to compare the average of the user's device to the average of all CompAir devices. This would provide a relative measurement for the users to understand whether their device was above or below average over several time periods as seen in Figure 39. Additionally, direct value labelling in the bar plots was used to increase precision, emphasizing a reading tone (Kirk, 2019). Providing various timeframes can be useful to understand the data and provides additional context of the data. For example, if the 24-hour PM1 average for the user's device is extremely high relative to the average of all CompAir devices however the 30-day average is below the average of all devices, this may indicate that something has happened in the last 24 hours that has created this increase. Observing increases in the average can help uncover possible reasons for inferior air quality. Furthermore, the design of the bar plot is simplistic as optimizing the data-ink ratio is important to not distract the user with any unnecessary graphics (Tufte, 1983).

```
compareplot = go.Figure(data=[go.Bar(
            name= user_input,
            x=temp2['Time'] , y=temp2['PM1'],
            marker_color='crimson',
            text=temp2['PM1'],
            textposition='auto',
            texttemplate="%{y:.2f} µm"
        ), go.Bar (
        name = 'All Comp.Air Devices',
        x=temp4['Time'], y=temp4['PM1'],
        marker_color='darkblue',
        text=temp4['PM1'],
        textposition='auto',
        texttemplate="%{y:.2f} µm"
)
    ])


templot = go.Figure(data=[go.Bar(
            x=temp2['Time'] , y=temp2['PM1'],
            text=temp2['PM1'],
            textposition='auto',
            texttemplate="%{y:.2f} µm",
)])
```

Figure 38. Comparison bar graph code

Figure 39. Demo for first section of overview dashboard

The decision to make the bar plot the only plot in that row was made by the amount of information that is displayed on this plot by itself and the size it requires to not be disproportionate. Adding more plots on this row would lead to information overload for the user by providing users with too much information (Jacoby, 1984). The second row is made up of two-line plots, a 24-hour line plot and a 30-day line plot which can be seen in Figure 30 in the Data Cleaning & Transformation section of the Methodology. Providing two timeframes allows the users to quickly notice trends in the data without requiring them to interact with the visualisation. As previously explained, the data was transformed to hourly averages from a minute data point frequency to increase readability of the graphs.

```
#LINEPLOTS
line24 = go.Figure(data=go.Scatter(x=hour24.index, y=hour24['PM1'], name = user_input))

line24.update_layout(
            autosize=False,
            width=600,
            height=450,
            title = "PM1 in the Last 24 hours",
            xaxis_title = "Time",
            yaxis_title = "PM1 μm ",
            font=dict(
                family="Arial",
                size=14)
)
```

Figure 40. Line graphs code

After analysing several metrics with different CompAir devices on the 30-day line plots, there were a few unclear trends due to a lot of variability in the data. The decision to include a smoothing line was made to further improve the readability of trends in the data. As outlined in the literature review, SG smoothing provides numerous benefits compared to other denoising/smoothing techniques such as maintaining the maxima and minima of the data and not flattening peaks too strongly. The window size is determined by the number of data points that will be on the SG Smoothed line. After experimentation with different window sizes such as 21, 51 and 401, 121 data points was found to be optimal. This meant the 24 data points per day (one per hour) would be captured by 4 data points per day over the 30-day timeframe of the plot. An additional data point was required as window sizes need to be odd numbers.[2] The 121-window size maintains a certain amount of accuracy while smoothing the trends that were unclear without smoothing/denoising.

```
line30.add_trace(go.Scatter(
    x=day30.index,
    y=signal.savgol_filter(day30['PM1'],
                           121,  # window size used for filtering
                           3), # order of fitted polynomial

    name='SG Smoothing'
))
```

Figure 41. SG smoothing code

Lastly, the final row in the Overview dashboard is a heat map of the selected metric using all CompAir devices. This way the user can examine how their location's values compare to other locations. The

---

[2] 4x30+1 = 121

API used is from Mapbox, a provider of online maps for websites and web applications. The parameters of the map were set using the longitude and latitude to view only the UK, the current target market of CompAir. However, the map has interactive features that allow the user to zoom in and out as well as pan across, which can be useful to specifically view only a city such as London or zoom out to the continent level to view the whole of Europe. The map can also be rotated in 3D for a different perspective (refer to Appendix). The current number of CompAir devices limits the insights that can be gained from this plot as there are only five active CompAir devices at the time of writing this dissertation. CompAir is a pre-revenue company and their device is still in the prototype stage, nonetheless, when eventually the device is on the market and the number of active CompAir devices are in the hundreds or thousands, the map plot will provide extremely useful insights for the users of the dashboard. The inclusion of the map plot was implemented to provide a plot that provides spatial analysis and as a feature that becomes more useful as the company scales.

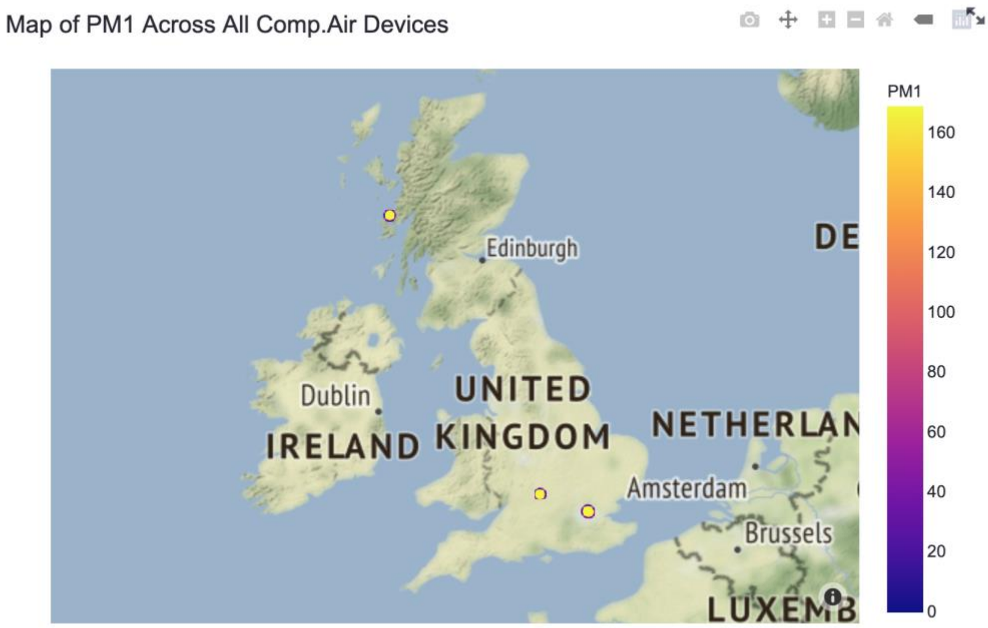

Figure 42. Demo for interactive map plot

4.6.2 Comparison Dashboard

The Comparison dashboard is a useful dashboard for users that will have more than one CompAir device. According to CompAir's founders, users of the device may want to own several devices to place them in various locations. Home owners might want to place devices in different rooms such as the bathroom, kitchen, bedroom etc., while hospitals or elderly homes may want to place a device in each of their occupants' rooms. The Comparison dashboard asks the users how many additional devices they wish to compare, up to 5 devices. Plotting more than 5 lines on a line plot can cause the plot to become cluttered and decrease readability (Krystian, 2018), therefore a limit of 5 devices was set. Depending on how many devices the user wishes to compare, a number of input boxes will appear. The line plot has an extensive amount of interaction possibilities provided by the Plotly library. For example, if a user wantd to analyze a specific timeframe they would be able to zoom in to that timeframe and have a more detailed line plot as seen in Figure 43.

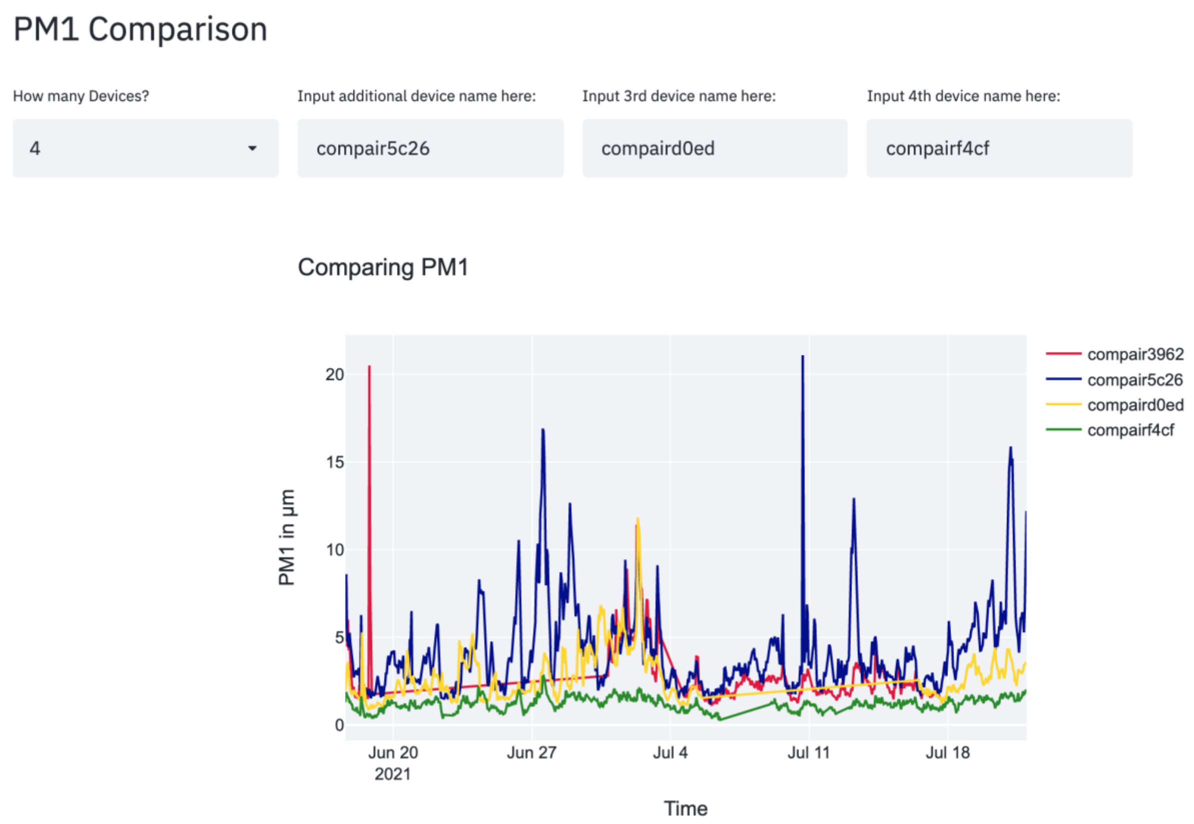

Figure 43. Demo for comparison dashboard

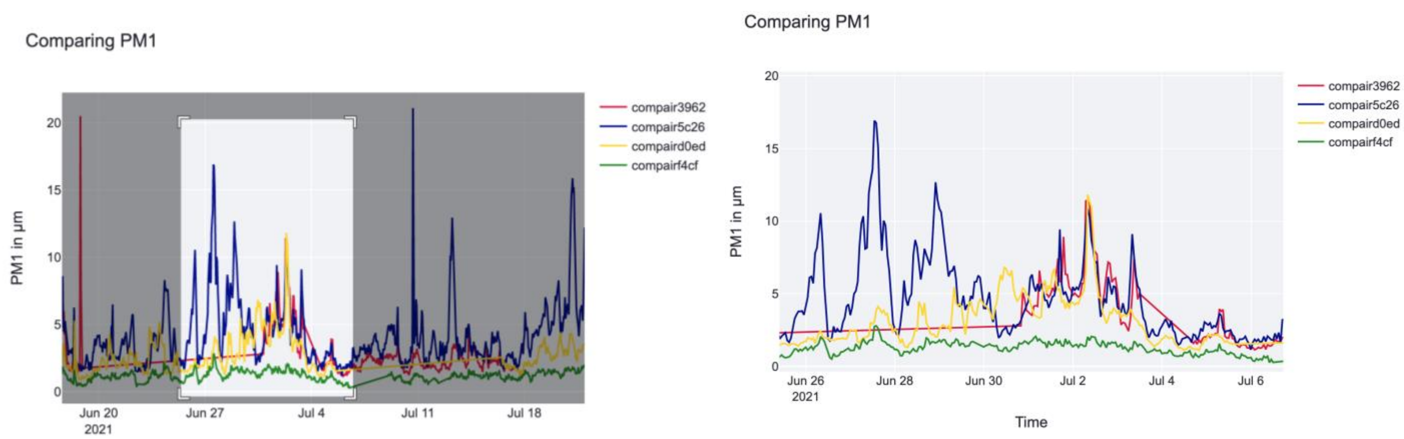

Figure 44. Demo of zoom in functionality with Plotly

4.6.3 FAQ Page

```
if option == "FAQ":

    st.markdown('___')
    about = st.beta_expander('About & Metrics Info')
    with about:
            '''
            Thanks for checking out our Dashboard ! It was built entirely using Comp.Air (https://www.compair.earth/) data.

            This app is a dashboard that runs an analysis on any desired metric captured by Comp.Air devices.
```

Figure 45. FAQ code

The FAQ page provides useful information for users such as answers to various questions on data protection, SG smoothing and recommendations for CompAir device operating conditions. As CompAir grows as a company they may add more information to this page, for any questions they often receive from users. In regards to the development of this page, Streamlit's beta expander was used again similarly to how the recommendations expanders were created.

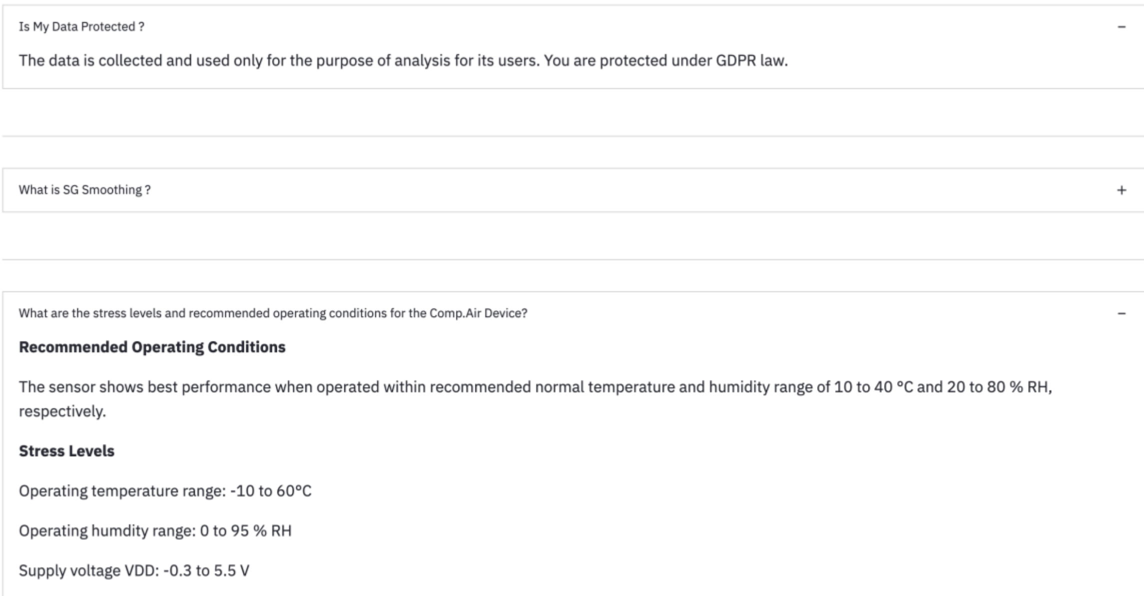

Figure 46. Demo for FAQ page

4.7 Deployment

The deployment of the dashboard application was carried out on two different free deployment platforms to provide performance comparisons in the Results/Analysis section. As CompAir is a pre-revenue company minimizing costs and thus opting for a free deployment platform was important. Heroku and Streamlit Sharing are the two free deployment options for a Streamlit application.

4.7.1 Heroku

After creating an account in Heroku, three files were created in a public Github repository where the code of the dashboard application is held. The first file, requirements.txt, was created by accessing the Sync Python Requirements option under the Tools tab in PyCharm. After using this option in PyCharm, installing a plug-in and connecting PyCharm to my Github account, the requirements file was generated and sent to the selected repository in Github. The generated requirements file is the imported python libraries and their corresponding versions that are in PyCharm. The second file is a setup file (setup.sh) which will handle issues regarding the server side by providing configuration information. The third file, Procfile, will run a command for the setup.sh file and the Streamlit application (refer to Appendix for the files). After the files were created and placed in the relevant Github repository, a new app was set up in Heroku. The deployment method used was Github, requiring to connect a Github account and the relevant project repository to Heroku as seen in Figure 47. The Automatic Deploy option (disabled as default) was enabled for the application to automatically update when a new version of the project file is pushed to Github. This ensures the web app to always be up to date. There were a few errors upon deploying the app as some of the libraries in the requirement.txt file were not used in the code, this required to find the libraries that weren't used and delete them from the requirements file manually.

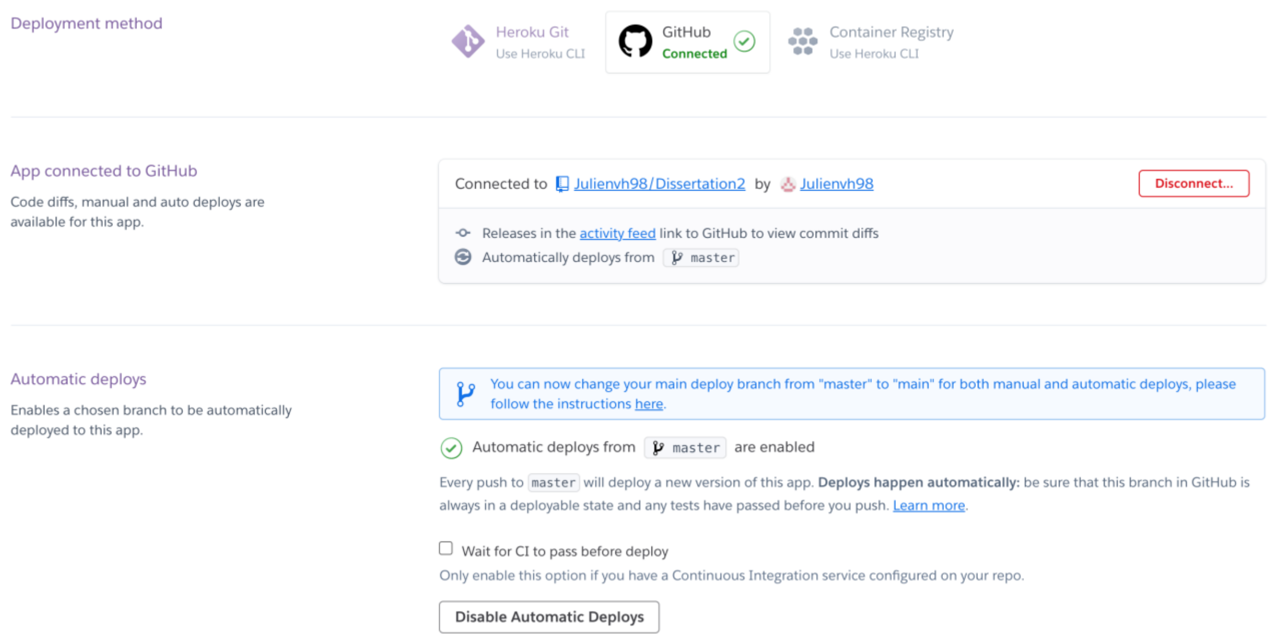

Figure 47. Heroku account view to create a new application

4.7.2 Streamlit Sharing

Streamlit has their own deployment option which requires an invitation to be requested on their website. The invite is received within a week by email and asks to log in using a Github account, selecting the relevant repository of the app and deploying the app. This method was a lot simpler and quicker once the invite was received, as it did not require the creation of the three files that are required for Heroku's deployment.

5 Analysis/Results

5.1 Use Cases

To examine the results of the project and elaborate on the "why" of the research question, the dashboard web app will be analysed through various hypothetical consumer use cases. This analysis will be descriptive as it will be demonstrated the applicability of the dashboard in real life situations. The use cases will demonstrate how users might interpret the visualisations to make decision and change their behavior accordingly with the help of the recommendation/contextual info dropdowns. The amount of air quality metrics that CompAir provides will be useful for any type of user whatever their goal. The dashboard is not limited to everyday consumers as monitoring indoor air quality can be useful for elderly care homes, athlete training camp facilities, luxury hotels, hospitals and more. For example, users living in the city may want to specifically observe the levels of Particulate Matter and CO2 in their homes which may help them understand whether they should open or close their windows or what areas in the city are more highly polluted. A study found that opening windows before going to bed lowered CO2 levels which led to better sleep depth, sleep efficiency, and lesser number of awakenings (Mishra et al., 2017). Improving sleep may be important for highly productive people and athletes, a potential target market for CompAir, as these consumers usually optimize anything related to their health.

Looking into a use case scenario of the integrative map plot, if hypothetically, a user is looking to move to a new neighborhood in London they could use the map plot to examine the spatial distribution of Particulate Matter. In Figure 48, we can notice the purple colored dot indicating a lower PM1 level close to the park while the yellow dots indicate a higher PM1 level and are further away from the park, next to a main road. Research has found that people that live closer to green spaces or parks tend to be happier and healthier (Wolf, 2017). This may be useful for a user looking to move flats and understanding the impact of location, incentivizing them to move in an area closer to a park or greenery.

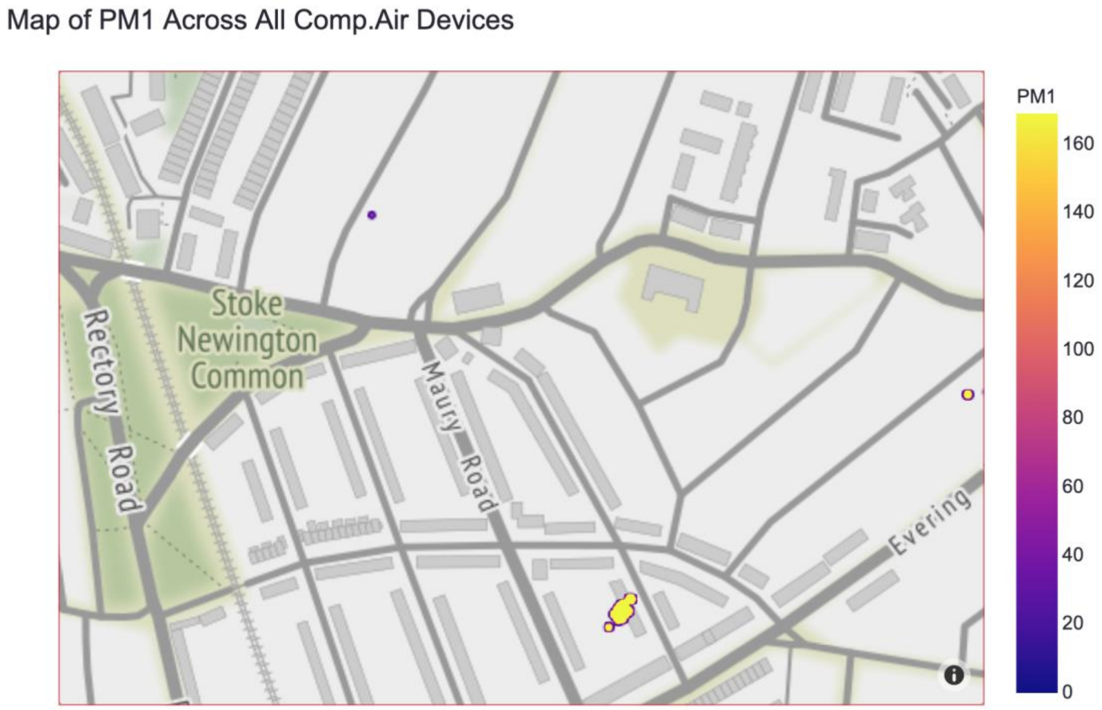

Figure 48. Map plot scenario

Another type of users may want to observe other metrics such as the temperature, humidity and VOCs levels to avoid mold formation and/or toxic pollution from home products such as paints, candles and cosmetics (Kansal, 2009). In this scenario, users may want to look at a bar plot and ensure the humidity to be kept below 60%, as according to the EPA (2021), levels above 60% can lead to mold growth. As seen in Figure 49, it can be understood over the different timeframes the average humidity is never close to the mold growth humidity level of 60%. If the levels are above 60% users may want to investigate what may cause this high level of humidity which can occur as a result of humidifiers, steam radiators, dryers, stoves or improper ventilation (EPA, 2021). These types of recommendations and are provided in the expander on the Overview dashboard and is a useful tool to push users to take actionable steps to improve their indoor air quality.

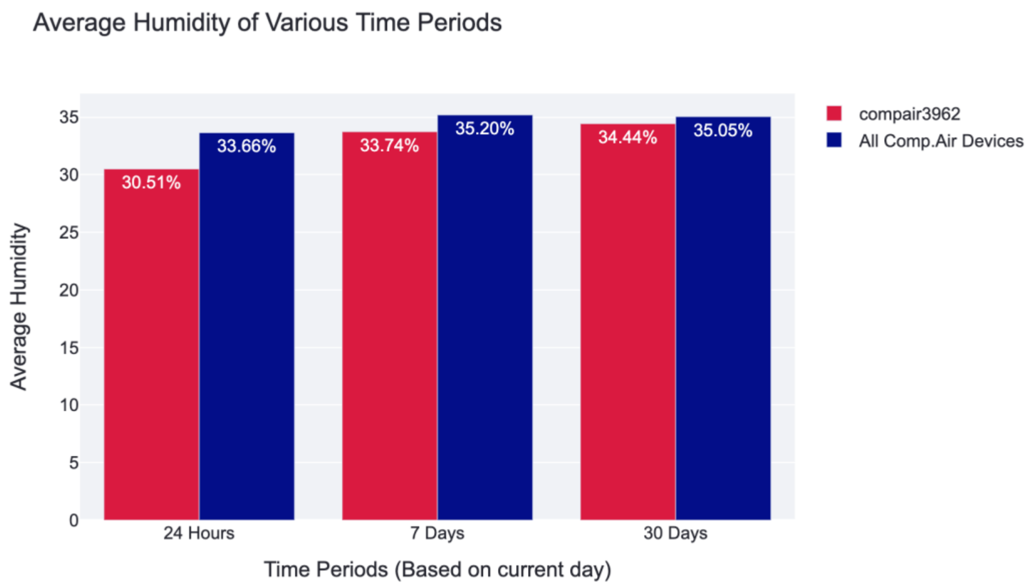

Figure 49. Bar plot scenario

If the user owns a luxury hotel or athlete center with a CompAir device in every room, the Comparison dashboard would be useful to quickly spot the rooms which may need more attention than others to avoid mold formation. Figure 50 shows that the blue line, device compairf4cf, has the highest humidity over a 30-day period and may need regular inspection to ensure no mold is growing.

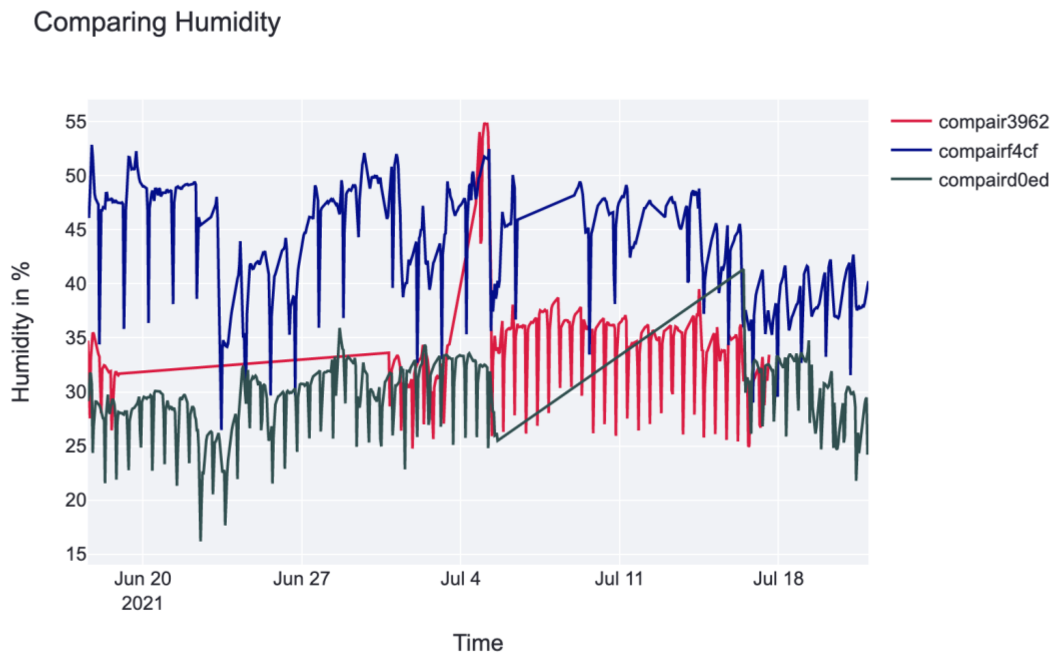

Figure 50. Comparison dashboard line plot scenario

While CO2 and Particulate Matter are hard to influence as they are mostly emitted by external sources such as combustion of engines and factories, VOCs is an air quality metric that is highly impacted by indoor activities (EPA, 2021). Figure 51 shows the 30-day line plot from the Overview dashboard displayed in full screen from device compair5c26. The graph shows that from the 16th of July until the 18th VOCs levels greatly increased, a trend that is clear when looking at the SG Smoothing line as it reduces the noise from the data. It could be that the hypothetical user decided to have a wall painted in their house which could have increased VOCs levels (Kansal, 2009). The user using this information could make sure they are not home the next time when a wall is painted to reduce exposure to harmful VOCs.

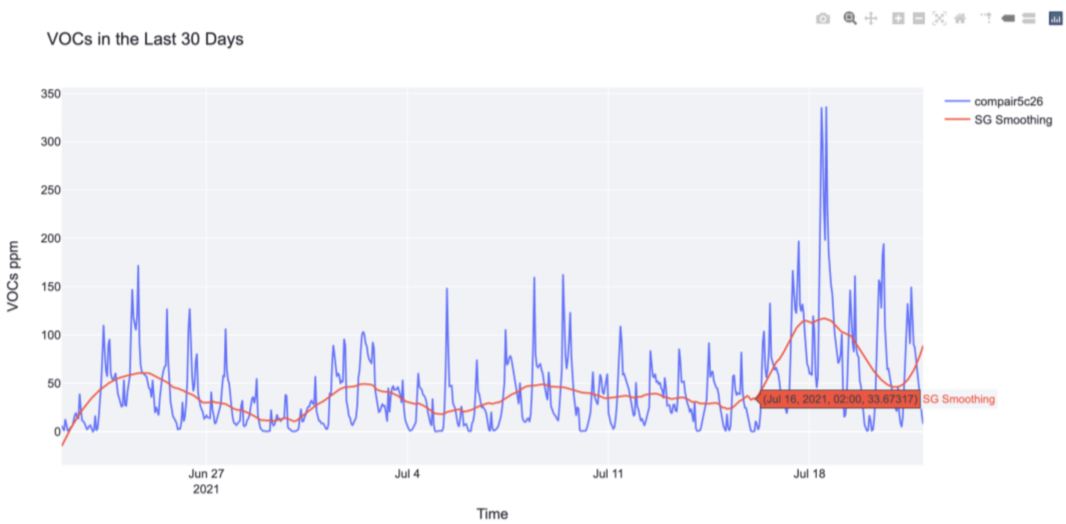

Figure 51. Overview dashboard Line plot scenario in full screen mode

While these scenarios are hypothetical they mimic real life situations and demonstrate how the dashboard can be used by users to make educated decisions based on data to improve indoor air quality.

5.2 Dashboard Performance

Important to a dashboard success is its performance, specifically how quickly pages load. This inferential analysis will answer the question: which free deployment ecosystem provides the fastest loading times for the CompAir dashboard?

TESTS	Streamlit	Heroku
Initial Loading Time (Without Cache)	5.5 seconds	4.0 seconds
Initial Loading Time (with cache)	5.3 seconds	3.4 seconds
Overview Dashboard Loading Time	11.5 seconds	7.0 seconds
Overview dashboard Loading Time (changing device name)	59.2 seconds	46.7 seconds
Comparison Dashboard Initial Loading Time	14.7 seconds	7.4 seconds
Comparison Dashboard Loading Time (2 devices)	26.7 seconds	11.5 seconds
Comparison Dashboard: Loading Time (4 devices)	48.7 seconds	16.4 seconds
FAQ Page Loading Time	14.5 seconds	6.1 seconds
Total Loading Time (Across Tests)	185.8 seconds	102.5 seconds

Figure 52. Table of loading time tests for Streamlit Sharing and Heroku Deployments

To determine which deployment method led to the best performing dashboard web app, several tests were run and collected as shown in Figure 52. Overall, the dashboard web app deployed on Heroku performed 81.2%[3] faster than the web app deployed using Streamlit Sharing. This is interesting as it would be expected that the Streamlit deployment method would perform better as the web app itself was programmed using Streamlit. This could be as Streamlit Sharing is relatively new and is still in beta while Hekoru is an established cloud platform that has widely been used for web app deployments. The recommendation for CompAir is therefore to use the Heroku deployed web app as its performance is extremely faster. Developers using Streamlit should deploy with Heroku for better performance and also ideally run their own loading tests to confirm this conclusion. Another interesting insight is the impact of cache on the app performance. As seen in the table the initial

[3] (185.8-102.5)/102.5

loading time was calculated with and without cached data. While the cached loading time of the dashboard seemed to improve Heroku's web app performance by 17.6%[4] the Streamlit cached version did not have any impact. This may again be caused by Streamlit Sharing being in beta and certain Streamlit APIs not working properly after deployment.

When analysing the loading times of the different sections of the web app, the only surprising loading time is the overview dashboard loading time when changing the device name. What this test entails is, after a device overview dashboard has been loaded, replacing that device name with another device name and running the same overview dashboard again. The 46.7 second load time is a lot longer than the average 8 second load time of the other tests.[5] This might be caused by the code requiring to replace the input rather than providing an input in an empty input box. The replacement of the input is computationally complex slowing down the loading of the web app. This might a limitation in the Streamlit ecosystem. In regards to the comparison dashboard, the increasing loading times for the increasing number of devices that are compared was expected, as each additional device requires additional code to be run.

The technical objective of this project was to create an interactive dashboard that had loading times below 20 seconds. Apart from one loading time test, all other tests succeeded in this objective, an overall accomplishment of the technical objective. In regards to the business impact objective, the objective cannot be verified yet as CompAir's devices remain in the prototype phase and thus the web app dashboard cannot be market tested. However, based on the impressed feedback and gratitude from the CompAir founders, the business value the interactive dashboard will bring to the product offering will be impactful.

[4] (4-3.4)/3.4 = 0.1764

[5] (102.5-46.7)/7 = 7.971

6 Limitation & Recommendations

6.1 Limitations

Working with live data and prototype devices is a challenge and many technical problems were encountered during the development of the dashboard. A study by Chojer et al. (2019), found one of the greatest drawbacks of low-cost devices are calibration issues and weak reliability. A few of CompAir's devices in the development of the dashboard crashed or failed requiring to reboot the AWS API. Furthermore, a few of the csv files in the AWS bucket of certain devices have had formatting errors. This occurred 6-8 times throughout the project and sometimes delayed progress on the dashboard by up to 4 days. These technical problems were mitigated by using older versions of the data to continue progress on the dashboard while the csv files were fixed. The UCL classmate responsible for the data engineering/data storage of CompAir's data was unable to find a long-term solution to this problem as it is a problem of the device/sensor itself. This problem still occurs and requires manually changing the csv files in the AWS bucket and rerunning an AWS Lambda function.

Another limitation are the loading times of the dashboard which remain relatively high for the modern-day impatient consumer that is only satisfied with instant results. According to a report by Google, half of website visitors will leave if they are required to wait longer than 3 seconds (Castro, 2016). While the CompAir web app dashboard's average loading time is around 8 seconds (except for one specific loading tests[6]), a report of 5 million website pages found that the average page speed of websites is 10.3 seconds (Dean, 2019). This suggests that while consumers are impatient, when considering the loading times of other websites, the dashboard has acceptable loading times. The loading time of the dashboard was improved by caching the code and thus saving a few seconds as certain functions did not have to be re-run as previously explained in the methodology.

A technical limitation related to Streamlit's web app framework in handling inputs is that Streamlit runs on every input change. This is a problem for the Comparison dashboard as when some of the input boxes are empty the code will still run automatically. For example, if a user wishes to compare the VOCs of 4 different devices, every time the user types a new device name in one of the input boxes for the 2nd, 3rd and 4th device, the page will run. This takes away from the smooth user experience, as the ideal scenario would be for the user to enter all the device names into the input boxes before the

[6] changing device name on the overview dashboard

code runs. After researching Streamlit's help forums (2020), it was mentioned that this is integral to how Streamlit works and therefore this limitation cannot be eliminated unless a different web app framework is used.

Lastly, as CompAir's device are prototypes and not ready for customer trials, the dashboard was not evaluated and tested on actual users and in real-life scenarios. This would have improved the result/analysis section of this dissertation as the dashboard would be evaluated with real life situations rather than hypothetical situations. The only user feedback that was attained was during the development of the dashboard from friends, family and the CompAir founders. Testing the dashboard and air quality device with customers through customer trails would be a crucial step to improve the effectiveness of the dashboard.

6.2 Recommendations

To conclude this project, a few recommendations and possible next steps for CompAir in regards to the dashboard will be given. The first recommendation would be to calibrate and validate against reference grade equipment by collocation of all Comp.Air's devices to a reference sensor (FRM/FEM or the UK equivalent). This will ensure accuracy of all devices and help measure performance of the sensors as they age, replacing the sensors with poor performance. Competitors are already actively doing this. For example, Clarity uses remote calibration techniques that meet USEPA and EU standards while Atmos devices are validated against FEM grade equipment (Clarity, 2021; Atmos, 2021). While CompAir have compared their devices to reference equipment to showcase to investors in their pitch deck, the CompAir devices should be calibrated by adjusting the devices to closely match the reference equipment.

The second recommendation would be to find existing companies and APIs to collaborate with, adding new and enhanced features to the dashboard. During the SWOT analysis it was found that as CompAir has just entered the market and is still pre-revenue, the direction of the company can still be determined. Finding innovative features whether in the dashboard or the product itself will be key for CompAir to differentiate themselves from competition as they do not have the first mover advantage. An example of an enhanced feature would be an advanced map plot. While the density plot API from Mapbox used in the current dashboard fulfills the requirements of spatially showcasing air quality, it is not innovative as it is an easily accessible API and may therefore not be a Unique Selling Proposition for CompAir. Another, API that may be required in the dashboard when the company goes to market is an authentication API. Authentication will improve security by requiring users to input a username and password. Furthermore, once the user is authenticated, the username could link the user's devices to the dashboard so that they would be able to simply choose which of their own device they would like to analyze without requiring to type in the device name in an input box.

7 Conclusion

The research question: why and how to build an indoor air quality interactive web app dashboard for consumers, was answered in two parts. Firstly, how the interactive dashboard was developed was explained by identifying stakeholders, evaluating the air quality dashboard market through a SWOT analysis, sketching a concept dashboard, selecting an appropriate web app framework, establishing editorial thinking and a line by line Python code walkthrough. Secondly, why the interactive dashboard was built was answered by explaining the threat of poor indoor air quality to human health, wellbeing and productivity. As humans spend over 90% of time indoors where pollution levels are up to five times higher, the importance of understanding how to improve air quality is essential (Carslaw et al, 2017; EPA, 2021). Furthermore, throughout the methodology, the reasoning behind why certain decisions were made was explained using data visualisation processes, principles and theories.

Academically, this paper provides insights into the design and development decisions and processes to build an interactive air quality dashboard, a gap in the current literature. Developers following this paper's methodology, backed with academic evidence and theories, will be able to replicate a similar interactive dashboard. Lastly, the business impact of the dashboard for CompAir is that users will now be able to analyse the data from their CompAir devices through insightful visualisations instead of a simple csv file, providing them with the information required to improve their air quality, health and wellbeing.

Bibliography

Ahmed, R. (2017). The Effects of Indoor Temperature and CO2 levels as an Indicator for Ventilation Rates on Cognitive Performance of Adult Female Students in Saudi Arabia.

Air Quality Expert Group. (2020). Non-methane Volatile Organic Compounds in the UK.

Atmos (2021). Retrieved 3 July 2021, from http://urbansciences.in/.

Austin, E., Novosselov, I., Seto, E., & Yost, M. (2015). Laboratory Evaluation of the Shinyei PPD42NS Low-Cost Particulate Matter Sensor.

Bartlett, M. (1950). Periodogram Analysis and Continuous Spectra. *Biometrika*.

Blaxell, R. (2020). The Manifesto of the Data-Ink Ratio. Retrieved 9 July 2021, from https://speedwell.com.au/en/insights/2019/the-manifesto-of-the-data-ink-ratio#:~:text=Examples%20of%20High%20and%20Low,from%20the%20data%20being%20represented.

Britannica Concise Encyclopedia. (2008).

Card, S., Mackinlay, J. and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision To Think*. Academic Press.

Carslaw, N., Fletcher, L., Heard, D., Ingham, T. and Walker, H., 2017. Significant OH production under surface cleaning and air cleaning conditions: Impact on indoor air quality. *Indoor Air*, 27(6), pp.1091-1100.

Castro, J. (2016). *Increase the speed of your mobile site with this toolkit*. Google Ad Manager. Retrieved 31 July 2021, from. https://www.blog.google/products/admanager/increase-speed-of-your-mobile-site-wi/.

Chapman, C. (2018). Exploring the Gestalt Principles of Design. Retrieved 23 July 2021, from https://www.toptal.com/designers/ui/gestalt-principles-of-design.

Chin, J., Godwin, C., Parker, E., & Lewis, T. (2014). Levels and sources of volatile organic compounds in homes of children with asthma. *Indoor Air*.

Chojer, H., Branco, P., Martins, F., & Sousa, S. (2019). Low-cost portable sensors for air quality monitoring.

Clements, A., Griswold, W., RS, A., Johnston, J., Herting, M., Thorson, J., & Hannigan, M. (2017). Low-Cost Air Quality Monitoring Tools: From Research to Practice (A Workshop Summary). *Sensors*, *17*(11).

Consumers to Drive U.S. Economic Recovery: 91% Say Indoor Air Quality Critical in Fight Against COVID-19 | Carbon Lighthouse. Carbon Lighthouse. (2020). Retrieved 4 June 2021, from https://www.carbonlighthouse.com/consumers-to-drive-u-s-economic-recovery-91-say-indoor-air-quality-critical-in-fight-against-covid-19/.

Dean, B. (2019). *We Analyzed 5.2 Million Desktop And Mobile Pages Here's What We Learned About Page Speed*. Backlink. Retrieved 29 July 2021, from https://backlinko.com/page-speed-stats.

Eckerson, W. (2011). Performance Dashboards: Measuring, Monitoring, and Managing Your Business.

Erdmann, C., Steiner, K. and Apte, M. (2002). Indoor carbon dioxide concentrations and sick building syndrome symptoms in the BASE study revisited: Analyses of the 100-building dataset.

Few, S. (2006). Information Dashboard Design.

Fincher, J. (2018). *Python IDEs and Code Editors (Guide) – Real Python*. Real Python. Retrieved 29 July 2021, from https://realpython.com/python-ides-code-editors-guide/#reader-comments.

Friendly, M., Denis, D., & Truman, H. (2001). Milestones in the history of thematic cartography, statistica l graphics, and data visualisation.

Gurel, E., 2017. SWOT ANALYSIS: A THEORETICAL REVIEW. *Journal of International Social Research*, 10(51), pp.994-1006.

He, C., Morawska, L., Hitchins, J., & Gilbert, D. (2004). Contribution from indoor sources to particle number and mass concentrations in residential houses. *Atmospheric Environment*, *38*(21), 3405-3415.

Jacoby, J. (1984). Perspectives on Information Overload. *Journal Of Consumer Research*, *10*(4), 432-435.

Kansal, A. (2009). Sources and reactivity of NMHCs and VOCs in the atmosphere: A review. *Journal Of Hazardous Materials*, *166*(1), 17-26A.

Kilcommins, S. (2021). Streamlit vs Dash vs Voilà vs Panel — Battle of The Python Dashboarding Giants. Retrieved 2 July 2021, from https://medium.datadriveninvestor.com/streamlit-vs-dash-vs-voil%C3%A0-vs-panel-battle-of-the-python-dashboarding-giants-177c40b9ea57.

Kim, S., & Paulos, E. (2010). inAir: Sharing Indoor Air Quality Measurements and Visualisations.

King, H. (2019). What Is PM 2.5 and How Can You Reduce Your Exposure?. Retrieved 14 June 2021, from https://molekule.science/what-is-pm-2-5-and-how-can-you-reduce-your-exposure/

Kirk, A. (2019). Data Visualisation: a handbook for data driven design (2nd ed.).

Krystian, M. (2018). *Do This, Not That: Line Charts*. Infogram. Retrieved 29 July 2021, from. https://infogram.com/blog/do-this-not-that-line-charts/.

Kumar, S., & Jasuja, A. (2017). Air quality monitoring system based on IoT using Raspberry Pi. *2017 International Conference On Computing, Communication And Automation (ICCCA)*.

Luna, M. (2013). *web development framework (WDF)*. TechTarget. Retrieved 9 July 2021, from https://searchcontentmanagement.techtarget.com/definition/web-development-framework-WDF.

Luo, G., & Zhang, D. (2012). Wavelet Denoising. Advances In Wavelet Theory And Their Applications In Engineering, Physics And Technology.

Luo, J., Ying, K., & Bai, L. (2005). Savitzky–Golay smoothing and differentiation filter for even number data. *Signal Processing*, *85*(7), 1429-1434.

Midway, S. (2020). Principles of Effective Data Visualisation. *Patterns*, *1*(9).

Milton, D., Glencross, P. and Walters, M., 2000. Risk of Sick Leave Associated with Outdoor Air Supply Rate, Humidification, and Occupant Complaints. *Indoor Air*, 10(4), pp.212-221.

Mishra, A., van Ruitenbeek, M., Loomans, C., & Kort, H. (2017). Window/door opening-mediated bedroom ventilation and its impact on sleep quality of healthy, young adults. *International Journal Of Indoor Environment And Health*, *28*(2), 339-351.

Mold Course Chapter 2. United States Environmental Protection Agency (EPA). (2021). Retrieved 2 August 2021, from. https://www.epa.gov/mold/mold-course-chapter-2.

Nadaraya, E. (1964). On Estimating Regression. *Theory Of Probability And Its Applications*, *9*(1), 141-142.

Renard et al., (2020). Indoor Air Quality: Tackling The Challenges Of The Invisible.

Oppermann, L. (2020). How to enhance your design with the gestalt principles of proximity. Retrieved 15 July 2021, from https://vea.re/blog/gestalt-principles-proximity4.

Particulate Matter (PM) Basics. United States Environmental Protection Agency (EPA). (2021). Retrieved 14 July 2021, from https://www.epa.gov/pm-pollution/particulate-matter-pm-basics.

Raatikainen, M., Skön, J., & Johansson, M. (2012). Effects of Weather Conditions and Differential Air Pressure on Indoor Air Quality.

Rasmussen, N., Bansal, M., & Chen, C. (2009). *Business Dashboards: A Visual Catalog for Design and Deployment*. John Wiley & Sons.

Rudin, L., Osher, S., & Fatemi, E. (1992). Nonlinear total variation-based noise removal algorithms. *Physica D*, 259-268.

Sadeghi, M., & Behnia, F. (2018). Optimum window length of Savitzky-Golay filters with arbitrary order.

Shendell, D., Winer, A., Stock, T., & Zhang, L. (2004). Air concentrations of VOCs in portable and traditional classrooms: results of a pilot study in Los Angeles County.

Shower gel, shampoo and chemical fumes: how everyday products affect air quality. (2020). Retrieved 16 July 2021, from https://ncas.ac.uk/shower-gel-shampoo-and-chemical-fumes-how-everyday-products-affect-air-quality/.

Siam, A., Sedik, A., Shafal, W., & Abouelazm, A. (2021). Biosignal classification for human identification based on convolutional neural networks. *International Journal Of Communication Systems*.

Stein, C. (1981). Estimation of the Mean of a Multivariate Normal Distribution.

Streamlit Possible to Not Reload on Changing Input. Streamlit. (2020). Retrieved 1 August 2021, from https://discuss.streamlit.io/t/streamlit-possible-to-not-reload-on-changing-input/8148.

Suarez, F., & Lanzolla, G. (2005). *The Half-Truth of First-Mover Advantage*. Harvard Business Review. Retrieved 6 August 2021, from https://hbr.org/2005/04/the-half-truth-of-first-mover-advantage#:~:text=A%20first%2Dmover%20advantage%20can,in%20a%20new%20product%20category.

Tufte, E. (1983). The Visual Display of Quantitative Information.

U.S. General Services Administration, National Archives and Records Service, Office of the Federal Register. (1970). *Code of Federal Regulations: 1949-1984*.

Wang, Y., Ling, J., Jing, H., Zhang, Q., & Jiang, J. (2015). Laboratory Evaluation and Calibration of Three Low-Cost Particle Sensors for Particulate Matter Measurement. *Aerosol Science And Technology*, *49*(11), 1063-1077.

Why Indoor Air Quality is Important to Schools. United States Environmental Protection Agency (EPA). (2021). Retrieved 5 July 2021, from https://www.epa.gov/iaq-schools/why-indoor-air-quality-important-schools#:~:text=EPA%20studies%20of%20human%20exposure,percent%20of%20their%20time%20indoors.

Williams, R., Kaufman, A., Hanley, T., Rice, J., & Garvey, S. (2014). Evaluation of Field-deployed Low Cost PM Sensors.

Wolf, K. (2017). *The Health Benefits of Small Parks and Green Spaces*. National Recreation and Park Association. Retrieved 2 August 2021, from https://www.nrpa.org/parks-recreation-magazine/2017/april/the-health-benefits-of-small-parks-and-green-spaces/.

Yigitbasioglu, O., & Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal Of Accounting Information Systems*, *13*(1), 41-59.

Zikova, N., Hopke, P., & Ferro, A. (2017). Evaluation of new low-cost particle monitors for PM2.5 concentrations measurements. *Journal Of Aerosol Science*, *105*, 25-34.

Appendix

Appendix 1: Trello Board (https://trello.com/b/uPu6ZfoS/compair)

Appendix 2: Github repository (https://github.com/Julienvh98/Dissertation2.git)

Appendix 3: Deployed Dashboard Links

Heroku: https://compair-2019.herokuapp.com/

Streamlit Sharing: https://share.streamlit.io/julienvh98/dissertation2/main.py

Appendix 4: Evaluation of web application frameworks for dashboards

Framework	Dash	Streamlit	Voilà	Panel
Programming Language Support	Python, R, Julia	Python	Python, C++, Julia	Python
Python Graphing Library Support	Enhanced functionality with Plotly. External libraries for Seaborn/Matplotlib, Altair/Vega-Lite, Bokeh.	All of the main Python plotting libraries. Have their own native line plots, bar plots, area plots, and maps.	All of the main Python plotting libraries	All of the main Python plotting libraries. Enhanced functionality for Bokeh & Plotly.
Multi-Page Application Support	Out-of-the-box	With Workarounds	No Support	Out-of-the-box
Open Source vs Proprietary	Open Source & Enterprise Version	Open Source & Enterprise Version (In Beta)	Fully Open Source	Fully Open Source
Extent of Prerequisite Skills Needed (Excluding Python)	HTML Basics (CSS, JS)	(HTML, CSS, JS)	(HTML, CSS, Jinja)	(HTML, CSS, JS)
Ease of Use	Moderate (Requires Some Learning)	Very Easy	Very Easy	Moderate (Requires Some Learning)
Design Flexibility	Excellent	Limited	Limited	Very Good
Jupyter/IPython Notebook Support	Not explicitly with Dash. Full support with JupyterDash.	No Support	Excellent Support	Excellent Support
Deployment Options	Plentiful	Plentiful	Plentiful	Plentiful
Ease of Deployment	Moderate	Very Easy	Easy/Moderate	Easy/Moderate
Authentication Mechanisms	Out-of-the-box Authentication - HTTP Basic Auth	None	None	Out-of-the-box Authentication - OAuth 2.0
Ease of Maintenance	Easy (May Require Refreshing)	Very Easy	Very Easy	Dependant on API Choice. Mostly Easy.
Level of Support	Very Good	Very Good	Fair	Good
GitHub Community	13.8k Stars, 1.4k Forks, 61 Contributors	12.7k Stars, 1.1k Forks, 80 Contributors	3k Stars, ~290 Forks, 43 Contributors	~950 Stars, ~150 Forks, 60 Contributors
Online Prevalence + Popularity	Popular	Popular	Moderately Popular	Unpopular
Optimal Use Case	Enterprise-grade, one-framework-fits-all solution. Unlimitied design flexibility & great scalability.	Turning Python scripts into interactive dashboards as quickly and painlessly as possible. Ease of use. Rapid prototyping.	Rapid prototyping in Jupyter notebooks. Quickly and reliably reporting data insights across an organisation.	Creating dashboard applications which are not restricted to a single GUI. Working with geospatial data.

(Kilmcommins, 2021)

Appendix 5: requirements.txt file

Dissertation2 〉 requirements.txt

Local Changes | main.py | requirements.txt

Default Changelist 1 file
Unversioned Files 6 files

```
gitdb~=4.0.7
cffi~=1.14.5
pillow~=8.2.0
ipython~=7.22.0
zlib~=1.2.11
olefile~=0.46
pip~=21.0.1
wheel~=0.36.2
openssl~=1.1.1k
cryptography~=3.4.7
tornado~=6.1
toolz~=0.11.1
pyzmq~=20.0.0
pexpect~=4.8.0
attrs~=20.3.0
parso~=0.8.2
jedi~=0.17.0
pytz~=2021.1
toml~=0.10.2
astor~=0.8.1
smmap~=4.0.0
numpy~=1.20.1
setuptools~=52.0.0
six~=1.15.0
pandas~=1.2.4
scipy~=1.6.2
matplotlib~=3.3.4
blas~=1.0
altair~=4.1.0
jsonschema~=3.2.0
jinja2~=2.11.3
entrypoints~=0.3
base58~=2.1.0
webencodings~=0.5.1
packaging~=20.9
bleach~=3.3.0
markupsafe~=1.1.1
openpyxl~=3.0.7
```

23rd July

master

Commit | Amend

Appendix 6: Setup.sh File

master Dissertation2 / setup.sh

Julienvh98 Create setup.sh

1 contributor

8 lines (8 sloc) | 133 Bytes

```
mkdir -p ~/.streamlit/
echo "\
[server]\n\
headless = true\n\
port = $PORT\n\
enableCORS = false\n\
\n\
" > ~/.streamlit/config.toml
```

Appendix 7: Profcile

master Dissertation2 / Procfile

Julienvh98 Rename Procfile.txt to Procfile

1 contributor

1 lines (1 sloc) | 42 Bytes

```
web: sh setup.sh && streamlit run main.py
```

Appendix 8: 3D Map perspective

Map of PM1 Across All Comp.Air Devices