FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Building a City Dashboard using IoT sensors traffic data

João Ricardo Ribeiro Cardoso

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado em Engenharia Informática e Computação

Supervisor: Marta Campos Ferreira

Second Supervisor: António Ramos Silva

July 3, 2022

Building a City Dashboard using IoT sensors traffic data

João Ricardo Ribeiro Cardoso

Mestrado em Engenharia Informática e Computação

July 3, 2022

Abstract

The smart city concept has sprung an increase in data produced daily by many sensors, with the focus of this dissertation being on traffic data. City managers and authorities can use this data to reach various conclusions and solve many problems related to traffic. For instance, traffic data can be used to predict traffic flow and travel times; thus, measures can be taken to avoid traffic congestion and improve citizens' mobility.

However, this type of information is only useful if the authorities and managers of a city have access to it quickly and intuitively. Hence the need to develop more efficient and effective city dashboards capable of conveying information in a simple and integrated manner.

Long traffic jams and inefficient mobility are real problems that big cities face, with solutions hard to devise. Data-assisted decision-making, that is, making decisions based on datasets, helps create these needed solutions, further emphasizing the need for city dashboards.

This dissertation's goal was to analyze traffic data from IoT sensors and develop a city dashboard to support city managers in decision-making. For this purpose, traffic data from Braga, a Portuguese city, was used, and a user-centered design approach was followed to ensure that the solution meets users' needs. The solution design includes analyzing traffic data and applying visualization techniques to ensure the data is easy to read and to understand. The implementation requirements were discussed and defined with a key stakeholder, and frameworks for dashboard implementation were studied and analyzed. In the end, the stakeholder assessed and validated the developed solution to ensure it was suitable and effective.

Although the end-users of a city dashboard in the context of this dissertation are city managers and authorities, the benefits of this solution should impact the entire population through the improvement of issues related to traffic and mobility.

Resumo

O conceito de cidade inteligente propulsionou a produção dos dados de tráfego, foco da presente dissertação, nomeadamente dados de tráfego urbano. Estes dados podem ser utilizados por gestores e autoridades municipais para alcançar diversas conclusões e resolver problemas relacionados com o tráfego. Por exemplo, os dados do tráfego podem ser utilizados para prever o fluxo de trânsito e os tempos de percurso, de modo a controlar o congestionamento do trânsito e a melhorar a mobilidade dos cidadãos.

No entanto, esta informação só se mostra útil se os gestores e autoridades municipais tiverem acesso aos dados rapidamente e de forma compreensível. Consequentemente, surge a necessidade de desenvolver um dashboard da cidade mais eficiente e eficaz, que transmite informação de forma simples e integrada.

Os longos engarrafamentos e mobilidade ineficiente são problemas de difícil solução que as grandes cidades enfrentam. A deliberação assistida por dados apoia a criação de soluções, realçando a necessidade de dashboards das cidades.

O objetivo desta dissertação foi analisar os dados do tráfego de sensores IoT e desenvolver um dashboard para auxiliar o processo decisório dos gestores municipais. Para este efeito, foram utilizados os dados de tráfego de Braga, uma cidade portuguesa, e um design centrado no utilizador para assegurar que a solução satisfaz as necessidades dos seus utilizadores. O design da solução inclui a análise dos dados de tráfego e a utilização de técnicas de visualização para garantir que os dados são fáceis de interpretar. Os requerimentos de implementação foram discutidos e definidos com a principal parte interessada e foram estudadas e analisadas diferentes frameworks para a implementação do dashboard. Por fim, a parte interessada analisou e validou a solução desenvolvida a fim de garantir que a mesma se revela adequada e efetiva.

Embora os utilizadores de um dashboard da cidade sejam gestores e autoridades municipais, os benefícios desta solução devem afetar toda a população através da resolução de problemas relacionados com o trânsito e a mobilidade.

Acknowledgments

I would like to express my deepest gratitude to Dr. Marta Campos Ferreira and Dr. António Ramos Silva for their invaluable guidance, patience, and feedback. Additionally, this endeavor would not have been possible without the Soltráfego management and engineering teams, who provided the data and feedback necessary for developing and validating the dashboard.

I am also grateful to the Faculty of Engineering of the University of Porto and their teachers for developing my academic and personal skills.

Lastly, I'd like to acknowledge my family, especially my parents and my sister, for their unconditional love and support. I would also like to thank my girlfriend for the persistent encouragement she provided during these months.

João Ricardo Ribeiro Cardoso

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
TLS	Technische Lieferbedingungen für Streckenstationen
UI	User Interface

Chapter 1

Introduction

1.1 Context

Smart cities are filled with various sensors that collect different types of data [25]. This notion of monitoring systems has been crucial to society for a while now, possibly since the Industrial Revolution. Being able to measure a quantitative value of performance in a system allows adequate people to take actions for increasing said performance, increasing productivity and profit.

It is essential to highlight that one of the most vital stages in system monitoring is converting raw data created by sensors into performance metrics for extracting valuable information [10]. Moreover, while a simple system like an oven requires little more than a thermometer to exhibit its internal state, complex systems like big cities' traffic networks have respectively complex monitoring needs.

However, in the last decades, data generated by cities has become increasingly higher in number and complexity due to the emergence of new technologies for this purpose [14]. Therefore, automated methods for analyzing this massive amount of data are shown to be ineffective [3].

1.2 Motivation and Objectives

City dashboards can be an effective way to display a target city's condition and performance utilizing charts, maps, and KPIs, thus enabling city managers and decision-makers to devise appropriate and, most importantly, informed solutions for the problems being faced [21].

While there are several different implementations of city dashboards for traffic monitoring, most fail to deliver a good performance in certain aspects. Hence, it would be worth studying the best practices for designing and implementing a successful dashboard and performing a case study to validate them.

The research questions that will drive this work are as follows:

- **RQ1: What information can be extracted from raw sensor data?**
- **RQ2: What are the best techniques for traffic data visualization?**

- **RQ3: How to design a city monitoring dashboard using a user-centered design approach?**

1.3 Problem

All big cities face traffic-related problems, primarily due to a rise in passenger cars on public roads. The increase in the quality of life of the average citizen, coupled with a lack of funding for cheap and efficient alternatives for mobility in the city, explains this phenomenon. While some governments have made some investments in bicycle lanes, good city planning that promotes walking over driving, and other solutions to reduce traffic, these efforts are not enough to solve this ever-growing problem.

Although remote work is becoming increasingly popular, reducing the stress on the road infrastructure of cities, it is undeniable that traffic is and will be a crucial part of a city for a long time.

The smart city concept has sprung an increase in data produced daily by many sensors, including data on urban traffic. However, raw data is hard to understand and analyze. Only after cleaning and transforming this data into valuable and intuitive information can it be beneficial.

City dashboards can organize and depict the status of a city, allowing decision-makers to diagnose traffic problems and propose practical solutions. For instance, traffic data can be used to predict traffic flow and travel times; thus, measures can be taken to avoid traffic congestion and improve citizens' mobility. Hence the need to develop more efficient and effective city dashboards capable of conveying information in a simple and integrated manner.

1.4 Solution

On that account, a method is proposed to develop a dashboard that conveys meaningful and valuable information for decision-making. The proposed solution consists of developing a city dashboard to visualize traffic data that is easy to use and understand. A user-centered design was followed to accomplish that goal by including key stakeholders in the development process. Appropriate data analysis and visualization techniques were heeded to ensure the visualized data was reliable and useful for city managers.

The key stakeholder of the developed dashboard is Soltráfego, SA, a Portuguese tech company. This company is the national leader in the urban traffic and parking business and works with smart cities on urban mobility solutions implementations. Soltráfego offers solutions for smart mobility, off-street and on-street parking, traffic and road safety, bike-sharing, and urban equipment. Being a staple in the traffic market and having access to IoT sensors in different cities of Portugal, this company has the expertise to evaluate and validate a city dashboard. They shared the data for the dashboard and were involved in the entire process, explaining their needs, giving access to real-life traffic data, and providing feedback.

Initially, the stakeholder was inquired about the requirements for the dashboard. After that, an analysis of data from the traffic systems of Braga, a Portuguese city, was performed to extract value from it. This analysis enabled the development of KPIs and metrics related to traffic. Then, a city dashboard was designed and implemented to visualize the analyzed data, following a user-centered design and establishing practices and techniques of data visualization. Ultimately, the finished prototype was handed over to the stakeholder. The stakeholder was involved in the entire development process, continuously assessing and validating the developed solution to ensure it was suitable, effective, and satisfied their needs.

1.5 Document Structure

Chapter 2 outlines the literature review conducted on the concepts of city dashboard, data analysis, and data visualization, predominantly focused on traffic data. Then, Chapter 3 explains the methodology used, with the results and its discussions being present in Chapter 4. This document ends with a conclusion in Chapter 5.

Chapter 2

State of the Art

A systematic literature review was performed to understand the state of the art of the subject of study.

Firstly, a search was done in the reference databases Web of Science and Scopus, with the terms ((Urban Traffic) OR (Urban City) OR (Traffic City)) Data (Dashboard OR Data Visuali?ation). A filter was applied to retrieve only results that were published during and after 2018. Complementary, three references that were deemed of great importance were also included. This procedure resulted in 1219 references after duplicates were removed.

Secondly, to assess and guarantee the quality of the references, two screenings were performed sequentially: one according to the title and one according to the abstract. The first screening removed references that had little to no relationship with the subject of study, easily detected by reading the title.

This screening resulted in 187 references. The second screening ensured that the resulting references were as focused as possible on the study matter to make sure the references were eligible for a full reading. Sixty-seven references passed this stage. A full reading was performed, and notes were taken to retain the paper's general idea. Twenty-seven articles and papers were used as references in this work.

The state of the art was then subdivided into three sections. Section 2.1 regards city dashboards and their story, Section 2.2 concerns data analysis and data processing, and Section 2.3 looks at data visualization and the main visualization techniques used in modern city dashboards.

2.1 City Dashboards

The concept dashboard used to be defined as the panel present in a vehicle or an aircraft, initially associated with mechanical gauges and cogs, and more recently with digital panels. And although this definition is not outdated, a new meaning to the word has been given. A dashboard is also "a diagram that shows important information, typically one that gives an outline of a business" (from the Oxford dictionary).

The first appearance of digital dashboards was at the beginning of the 1990s. These initial dashboards were predominantly used to summarize important information using graphical components. Afterward, the development of online analytical processing (OLAP), key performance indicators (KPIs), and other information technology gave rise to the concept of geospatial dashboards [8, 12].

A solution proposed by A. Kwee et al. uses a congestion cascade clustering algorithm to calculate congestion growth and shrinkage throughout the period corresponding to the data sample being analyzed [15].

The interface, which can be seen in Figure 2.1, doesn't have a lot of different colors, being mainly constituted by tones of black and dark gray. A central map with encompassing charts displays an adequate amount of data, not appearing overwhelming to the user.

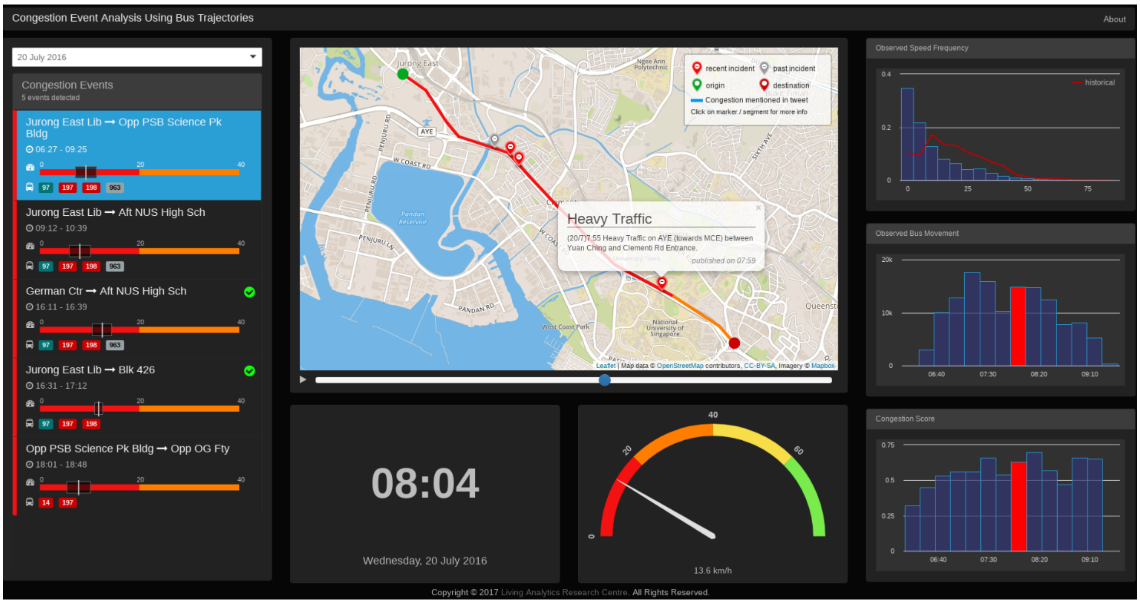


Figure 2.1: Solution for a traffic dashboard displaying traffic congestion. Adapted from [15]

The choice of technologies used, namely Java for data processing and nodeJs for data visualization, allows for greater customization and adaptability to the problem. However, it can lengthen the development time, a trade-off stemming from not using a dedicated framework for dashboard development.

The lacking feature of this solution is support for real-time data analysis and visualization.

A study [26] performed by G. Young et al. proposes a few design guidelines for city dashboards that are especially relevant to this work. They can be seen in Table 2.1.

In [27], V. Zdraveski et al. implement a city dashboard, that can be seen in Figure 2.2, following the ISO 37120:2014 standard that defines methodologies for indicators that measure city performance. While it is not focused on traffic monitoring, it is nonetheless meaningful in presenting and applying concepts of city dashboard design and implementation.

Table 2.1: Design guidelines proposed by G. Young et al. Adapted from [26]

Guidelines	Design Focus
1. Navigation	Implement logical navigation patterns and menus so users can explore data with confidence and quickly trace their progress throughout the dashboard hierarchy.
2. Data Utility	The intended meaning of the data being presented must be explicit and have actionable applications for diverse user types with different data literacies.
3. Style	The overall look and feel should be representative of the city and should be applied consistently to help build familiarity and confidence as well as improving the overall user experience.
4. Visualizations	Data visualizations must be of a suitable type and have further contextual information or metadata attached for clarity of meaning.
5. Veracity	The accuracy, precision, lineage, source, and age of data must accompany all data.
6. Users	Potential user-types for city dashboards are broad; therefore, implement user-centered design methodologies for all system development workflows to build empathy with the different user types of dashboard systems.
7. Data Types	Use both real-time and historic data; arrange them logically and group them thematically.
8. Usability	Usability heuristics should be applied at all stages by all project team members.
9. Communication	Use effective language and appropriate visualizations to communicate meaning across multiple platforms, media, and via multiple modalities if possible.

It is not hard to see a pattern in their architecture from the dashboards studied. It is typical to see at least four modules constituting the dashboard.

The first module is associated with data generation, specifically the traffic sensors infrastructure. The second module comprises the data processing algorithms (data cleaning, data transformation). The third module represents the databases for storing all the processed data. Finally, the last module is the data visualization client, where all the charts and maps are displayed to the user.

Most of the solutions in the papers studied had a module with equivalent functionality to those presented, like [6]. However, some had additional modules required by their specific problems, as can be seen in [19].

2.2 Data Analysis

Raw data collected by sensors isn't very useful in its current state. It requires processing and analysis to extract value from said data.

Generally, there is an initial preprocessing phase aimed at making sure data captured by sensors, which are sometimes faulty, produces valid information. This phase seeks to remove missing,

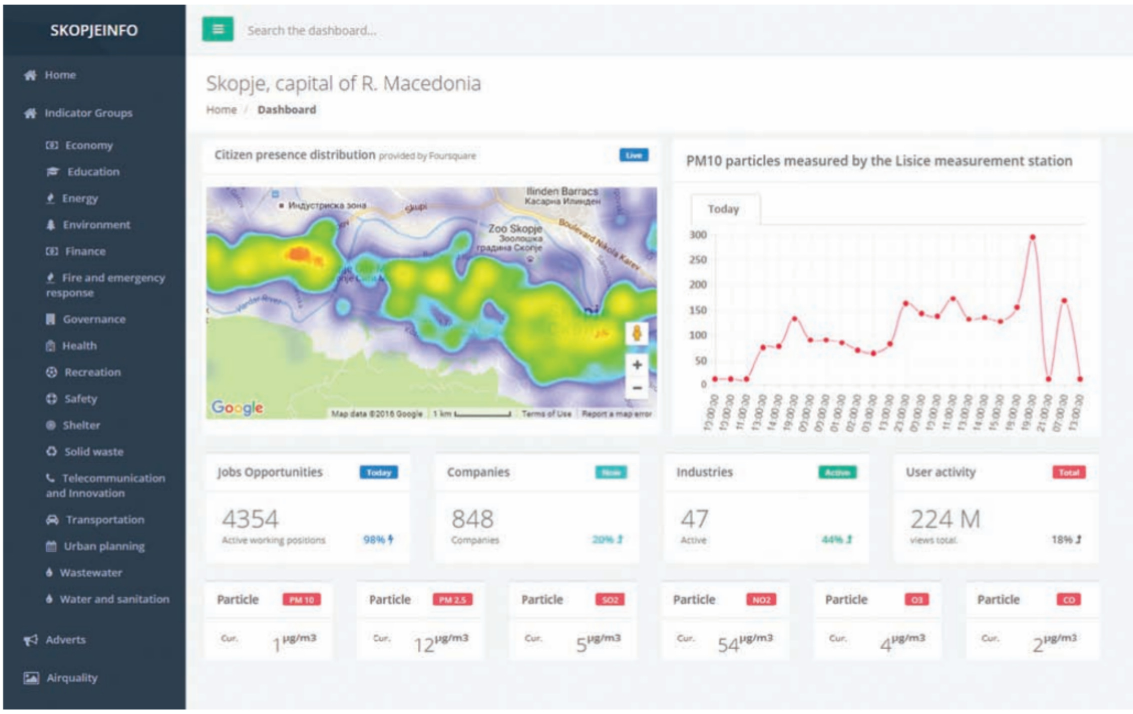


Figure 2.2: Prototype dashboard following the ISO 37120:2014 standard. Adapted from [27]

invalid, or repeated values that can sometimes be present in datasets due to sensor faults or poor environment conditions [9, 22].

In the traffic context, data is inherently spatiotemporal, that is, there is a spatial component (related to some coordinate system) and a temporal component (timestamp of reading or even duration of the event). The measured velocity of a vehicle is collected in a specific place and time. To tackle this property of the data, a solution proposed by L. Liu et al. for traffic data analysis creates a N x N grid for mapping the coordinate of an event to the grid. It also considers the data in time intervals of a meaningful granularity [16].

Another solution by D. Mahajan et al. uses ranking and classification to categorize intersections into four classes based on their ratios of vehicle arrivals on a red traffic light signal over arrivals on green [17]. This metric for traffic performance in intersections allows decision-makers to quickly assess the problems in those zones.

In [4], F. Desimon et al. explore the tool Karma, which takes the data from traffic sensors, present in a database, as a CSV file and transforms it into Linked Data. This Linked Data is then loaded into an endpoint for queries. The data processing architecture described can be seen in Figure 2.3.

Combining various data analysis ontologies can produce a plethora of new information that can be displayed to the data analysts and city managers for decision making.

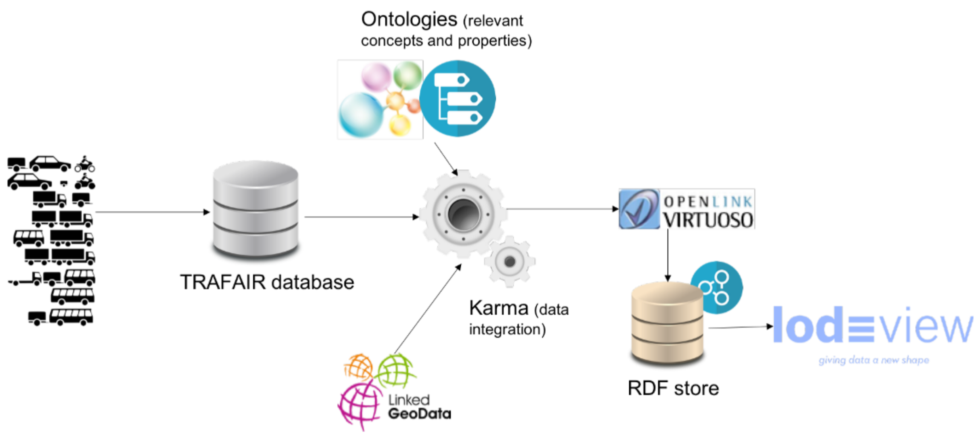


Figure 2.3: Overview of the mappings and tools used for data annotation and publishing. Adapted from [4]

2.3 Data Visualization

After knowledge extraction, appropriate visualization techniques need to be applied to show information in a clear and unbiased way, to facilitate the decision-making process [20].

M. Garcia et al. denotes eleven technologies for visualization of urban data [11]. See Figure 2.4. While this document's work is focused on city dashboards, it is noteworthy to study different visualization techniques and technologies for urban data to improve the clarity and usability of the dashboard to develop.

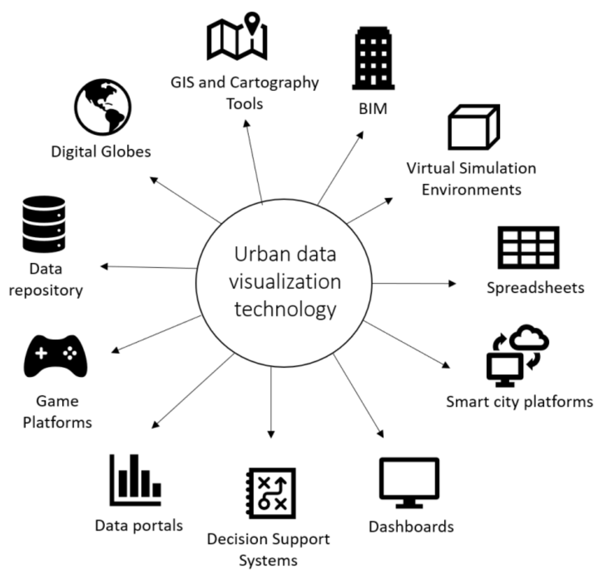


Figure 2.4: Technologies used for urban data visualization. Adapted from [11]

Although the same data can be represented in diverse ways, like graphs or bar charts, it is crucial to choose the appropriate structure for each specific case.

A recent study by T. Sobral et al. explores ways to visualize urban data using heat maps, rose maps, and flow maps [23]. These more complex visualization techniques allow the user to visualize complex relations between datasets that would otherwise be hard to see. I. Caldas et. al. compare different ways to represent traffic on a map in [2], concluding that a line-color metaphor is more effective than a line-thickness metaphor on a larger scale. They also conclude that a heatmap metaphor is especially inadequate at larger scales, whereas in zoomed-in maps it provides a satisfying experience for the user.

In [18], M. Morgan et al. study a set of techniques to improve route clarity in map visualization.

The first technique is named vector aggregation and consists of splitting road segments defined in maps to eliminate overlapping, thus making each segment easier to see. Its main limitation is that it is very memory intensive, which is something to consider when designing a system for real scenarios.

The second technique, rasterization, is used to increase the size of a specific road segment in reference to the values that are being visualized. In the case of traffic speed, this allows the user to quickly see what road segments have a higher average traffic speed by comparing the width of the roads. Alternatively, a color scale can be used instead of widths in a process referred to as tiling, having the advantage of being easier to emphasize a specific interval of values, as seen in Figure 2.5.

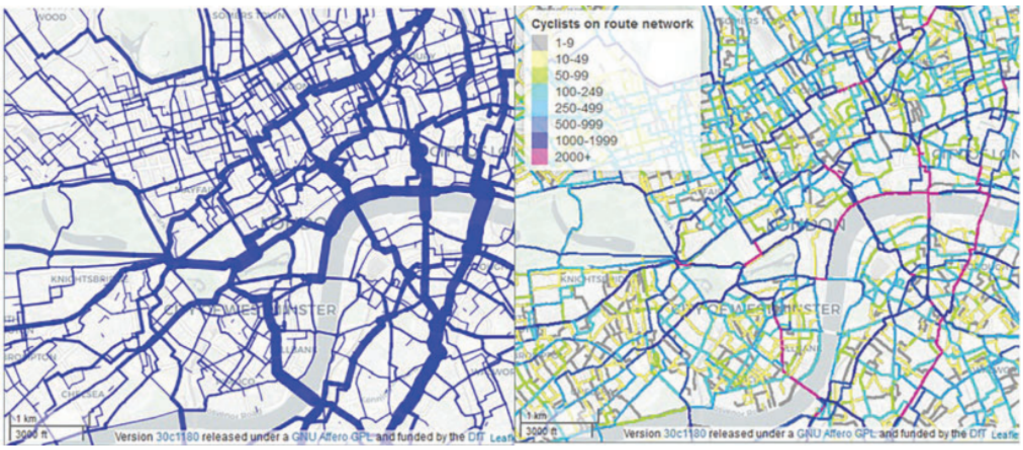


Figure 2.5: Map representing the number of cyclists in central London with line width (left) and color (right). Adapted from [18]

In order to facilitate the visualization of data in a dashboard, various frameworks and libraries can be used. Most systems studied utilized a web application for displaying the data, with a client/server architecture, like in [5], although some used desktop applications.

Front-end frameworks have become very popular for creating good user interfaces. While some solutions, like [17] or [1] used Angular, others like [6] used React. Both these JavaScript-based tools have the advantage of being highly customizable and adjustable to the problem at hand. They also allow for an effortless integration with JavaScript libraries, such as Leaflet, used in [19] or Chart.js, used in [1].

Nevertheless, J. Kazak et. al. in [13] and X. Fan et. al. in [7] made use of the Tableau Desktop application for displaying traffic data. While these solutions were probably more straightforward to implement, desktop applications very often lack the UI customizability that a dashboard requires.

Chapter 3

Methodology

This chapter describes the methodology followed during the development of the dashboard.

3.1 User-Centered Design

The viability and performance of any product requiring user interaction heavily rely on delivering a good user experience. When developing a dashboard, usability and intuitiveness are crucial concerns that continuously influence the design process. In that regard, various paradigms can be adopted to guide the process, one of which is a user-centered design approach.

This design strategy drives the developers to focus on the user in every step of the design process. The users can also be involved during the development, communicate their requirements, and express their concerns. In every phase, the target users' input is considered to create a product more aligned with the their expectations and needs [24].

Access to user feedback is fundamental to achieving a quality dashboard, especially when the use cases are very specialized, as with a traffic monitoring dashboard. For these reasons, a user-centered design approach was followed, relying on the expertise of the stakeholders in this domain.

While a user-centered design is typically applied in the context of user interfaces, this work proposes its use in the entire development process, from the requirements gathering phase to the final validation. This approach creates an even more prominent stakeholder involvement in the conception of the dashboard, leading to a better final result.

3.2 Dashboard Development

The development process of the dashboard was carried out in five distinct phases, each carrying a crucial role in the whole procedure. Soltráfego, the stakeholders of the dashboard, were involved in all phases, explaining their needs, giving access to real traffic data, and providing invaluable feedback. A diagram of the methodology can be seen in Figure 3.1.

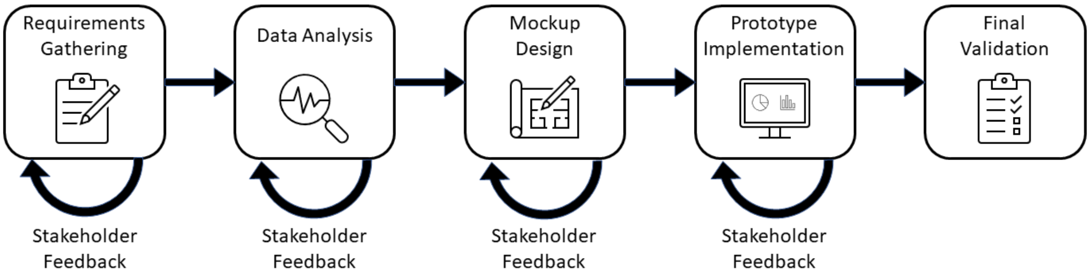


Figure 3.1: Diagram of the methodology followed in this work

3.2.1 Requirements Gathering

The first phase in the development of the dashboard was the requirements gathering. The importance of this phase is evident, as it allows collecting input from a company experienced in the field of traffic solutions.

In this phase, a meeting with Soltráfego took place to define their needs and explain what they were looking for in the dashboard. They also showed their current traffic monitoring solutions as a guideline for the expected layout and features.

The main features present in the solutions were some KPIs such as the count of vehicle passings, a map indicating traffic density, and various tables with data related to each point in the traffic system. All these features were used as inspiration and expanded upon when developing the new dashboard.

The requirements of the dashboard were the following:

- A clear and intuitive user interface to facilitate user interaction;
- Displaying relevant data and metrics using the provided raw data files;
- An interactive map for a general overview of the entire system.

It is essential to point out that these requirements are relatively lenient, so there is a high degree of freedom about what functionalities to implement and how to organize the dashboard's content.

3.2.2 Data Analysis

The information displayed in a dashboard needs to be extracted from data sets. This process entails cleaning the data by removing and correcting duplicated, invalid and unrealistic data, followed by data transformation, which involves calculating and grouping data in a way that can stem new information.

The sensors used for the data collection were a part of a roadside unit, the Cross Counter. This equipment was designed for vehicle counting and classification by utilizing electromagnetic loops installed under the pavement to detect a vehicle's passing. Each sensor is made up of two loops about a meter apart, allowing for the speed of the vehicles to be measured by dividing the distance between the loops by the time it took since the automobile crossed the first loop until it crossed the second one. Calculating its length by multiplying its speed with the time elapsed on the first loop is also possible. A group of sensors constitutes a point in the traffic system.

The traffic data, collected in five different geographical points in a small area of the city of Braga, Portugal, between February 23th and March 25th of 2022, was provided for data analysis, processing, and later visualization in the dashboard.

The data is in CSV format, making it easy to work with. The stakeholder provided five files with traffic data, each generated at a different point. In all these files, there are eight columns of values, each listed and explained in Table 3.1. Additionally, each row represents a vehicle's passing at that point, measured by a particular sensor.

Table 3.1: Columns of the CSV files with their descriptions and type

Column	Description	Type
ID	Identification of the row.	Number
Name	Name of the point where the passing was measured. There are five points: P1, P3, P4, P5 and P6	String
Date	Date and hour when the passing was measured.	String
Category	Type of vehicle measured by the system, following the classification standard described in the TLS guidelines. (Bus, Car, Car with trailer, Lorry, Lorry with trailer, Motorbike, Truck, Van, and Unknown)	String
Lane	Identification of the sensor that measured the passing. Each point can have multiple sensors.	Number
Speed (km/h)	Speed in kilometers per hour of the passing measured.	Number
Length (cm)	Length of the vehicle in centimeters measured.	Number
Gap (seconds)	Time in seconds spent between the last passing measured by that sensor and the current one. The first passing measured of each sensor always has the value 0.	Number

Python was the language chosen for processing the data, as the pandas library is the perfect tool for manipulating the more than 500000 rows of each CSV file. A Jupyter Notebook was created, as that software allows for a good organization and documentation of the data analysis process.

After importing the data into the notebook, some minor adjustments to the data were immediately performed: the "Length (cm)" column was changed from centimeters to meters as that unit of measurement made more sense when referring to vehicles; the "Name" column was renamed "Point" to better understand its meaning. A sample of the data can be seen in Figure 3.2.

3.2.2.1 Data Cleaning

The next step is to verify if the data was realistic and valid. To that end, simple statistical metrics were calculated, such as the average, minimum and maximum values of a few columns. These metrics can be seen in Table 3.2.

ID	Point	Date	Category	Lane	Speed (km/h)	Length (m)	Gap (seconds)
0	P1	2022-03-25 13:52:28.099000+00:00	Car	4	22	3.7	6.432
1	P1	2022-03-25 13:52:27.385000+00:00	Car	5	27	4.2	15.712
2	P1	2022-03-25 13:52:20.737000+00:00	Car	4	25	4.8	1.936
3	P1	2022-03-25 13:52:17.886000+00:00	Car	4	24	4.3	2.288
4	P1	2022-03-25 13:52:16.231000+00:00	Lorry	2	17	11.5	39.848
5	P1	2022-03-25 13:52:14.707000+00:00	Car	4	25	3.7	18.912
6	P1	2022-03-25 13:52:13.476000+00:00	Car	3	23	4.5	3.112
7	P1	2022-03-25 13:52:10.892000+00:00	Car	5	27	2.4	1.296
8	P1	2022-03-25 13:52:09.316000+00:00	Car	3	26	4.1	11.216
9	P1	2022-03-25 13:52:09.057000+00:00	Car	5	25	3.9	15.296

Figure 3.2: Sample of the data imported from the CSV files

Table 3.2: Statistical metrics for the data before cleaning

	Average	Max	Min
Speed (km/h)	19.740	255.000	0.0
Length (m)	10.828	254.000	0.0
Gap (seconds)	2985.191	65528.000	0.0
Gap (minutes)	49.753	1092.133	0.0

Examining the metrics related to the speed column, we can remark that the average value for this metric (19.740 Km/h) is realistic, given that the sensors are located on generally busy metropolitan roadways. On the contrary, the maximum and minimum values do not share the same level of realism. While these extreme values seem like an unpredictable error in the measurement, it is, in fact, a limitation of the sensors, as when a vehicle stops on top of them, its speed can end up being measured as either a zero or the maximum value of 255. According to the stakeholder, removing these entries was undesirable, as having an accurate count of the vehicle passings was more important than realistic measurements.

As for the length column, all the values seem unrealistic. While an average length of 10.828 meters is certainly possible, it seems rather unlikely that the number of long vehicles such as buses or lorries would be enough to sway the average too far away from the length of a car. The maximum value of 254 meters is inconceivable, so further analysis was performed to fix these anomalies.

The data was grouped by day to see if these high values were consistent throughout the measured period; this was not the case. The average values for the length and gap were too high from the start until the middle of March 4th, after which they became acceptable, as seen in Table 3.3.

This problem occurred due to a unit error. After consulting with Soltráfego, it was confirmed

Table 3.3: Excerpt of the average values of the *Speed*, *Length* and *Gap* columns when grouped by day

Date	Speed (km/h)	Length (m)	Gap (seconds)
...	...	...	...
2022-03-02	22.253894	37.777835	13313.953644
2022-03-03	22.253894	37.900257	13596.333295
2022-03-04	21.545115	17.498310	9000.043259
2022-03-05	23.329670	3.738258	15.178673
2022-03-06	22.132238	3.705445	17.252888
...	...	...	...

that the units used for measuring the length and gap values changed around that time. There was a change from millimeters to centimeters for the length and milliseconds to seconds for the gap, which explains the x10 and x1000 magnitude difference, respectively. To fix this error, the data was manually analyzed to determine the exact moments for each point where the problem stopped occurring, and the values were divided by the appropriate factor. After fixing the unit error, both the average (3.834 meters) and the maximum (25.400 meters) values for the length column appeared more realistic, as can in Table 3.4.

Table 3.4: Statistical metrics for the data after fixing unit error

	Average	Max	Min
Speed (km/h)	19.740	255.000	0.0
Length (m)	3.834	25.400	0.0
Gap (seconds)	12.362	65.535	0.0
Gap (minutes)	0.206	1.092	0.0

The sensors' limitation (explained above) that affected the speed also explains the minimum value of zero for the length. Once again, removing these entries was avoided to preserve the accurate count of vehicle passings.

Finally, examining the gap column's metrics (after fixing the unit error), the following observations were made:

- Its average value, which can be interpreted to mean "on average, one vehicle is detected every 12.362 seconds", seemed plausible;
- Its maximum value is not only unrealistic, as there must surely be times, at least during the night, when no vehicle passes for more than 65.535 seconds, but it is also not a random number, being exactly 2^{16}-1 ms.
- Its minimum value is zero, once again on account of the sensors' limitation.

After discussing the gap's maximum of 65535 ms with the stakeholder, a conclusion was reached that this was the highest value measurable by the sensor. Although the timer used to measure this metric was remarkably precise, this limitation resulted in inaccurate readings. Fortunately, the solution was to calculate the difference in dates between two consecutive passings to attain an approximation. Notably, this calculation was only conducted in entries where the gap was exactly 65535, as only in these situations could there have been an inaccurate measurement. The new value of the maximum gap (seen in Table 3.5) registered was now around 6 hours and 43 minutes, a time that seems reasonably achievable during a non-busy night.

Table 3.5: Statistical metrics for the data after recalculating the gap

	Average	Max	Min
Speed (km/h)	19.740	255.000	0.0
Length (m)	3.834	25.400	0.0
Gap (seconds)	26.153	24209.102	0.0
Gap (minutes)	0.436	403.485 (6h43min29s)	0.0

The data cleaning step's importance is unequivocal since a dashboard's effectiveness in aiding decision-makers in solving problems is tightly coupled with the validity of the data it displays.

3.2.2.2 Data Transformation

After the data cleaning process, the data needed to be transformed and processed so that the dashboard could display more interesting information. While Table 3.5 shows statistical metrics for the entire data, it was also possible to group the data by point before computing such values, as seen in Figure 3.3 and Table 3.6. It was also possible to follow the same logic to group the data by sensor for each point, as exemplified by Figure 3.4 and Table 3.7 for point P1.

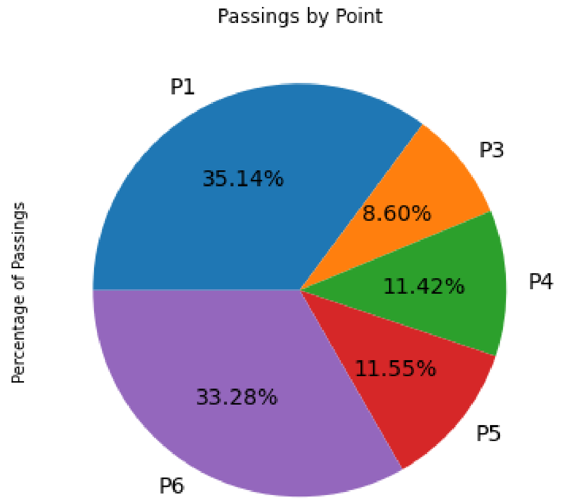


Figure 3.3: Pie chart with relative frequency of passings detected in each point

Table 3.6: Statistical metrics for the data when grouped by point

Point	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
P1	533358	35.142%	18.919	4.096	23.063
P3	130518	8.600%	31.186	3.749	39.456
P4	173365	11.423%	26.858	3.970	44.602
P5	505112	11.554%	18.542	2.502	29.256
P6	505112	33.281%	15.621	3.995	18.568

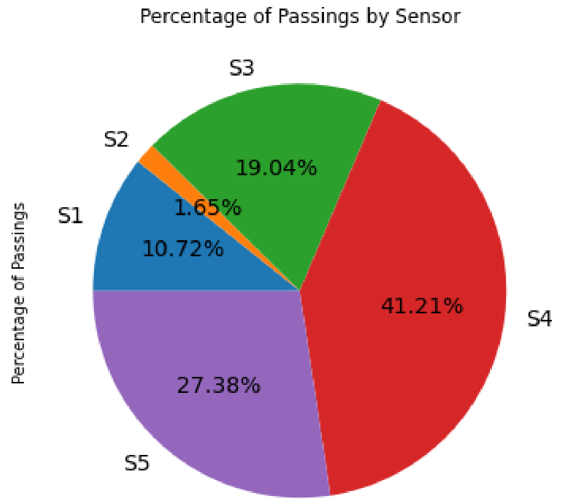


Figure 3.4: Pie chart with relative frequency of passings detected in each sensor, for point P1

Table 3.7: Statistical metrics for the data in point P1 when grouped by sensor

Sensor	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
S1	57190	10.723%	18.738	3.214	44.769
S2	8782	1.647%	20.040	5.588	299.362
S3	101557	19.041%	17.150	4.119	23.610
S4	219813	41.213%	23.710	4.229	10.819
S5	146016	27.377%	12.941	4.137	15.996

Another way the data could be transformed to increase its value was by analyzing it over time. The data was grouped in three distinct ways: by the hour of the day, by the day of the week, and by the day of the month. For each method of grouping the data, the average values of a specific column could be plotted over time, as illustrated by figures 3.5, 3.6, and 3.7.

Looking at Figure 3.5, one can easily note that the average speed is lowest at 8 a.m. and 5 p.m., which coincides with the usual timestamps of entering and leaving work. Similarly, glancing at Figure 3.6 reveals that the average time between passings is higher on Sundays, likely due to a

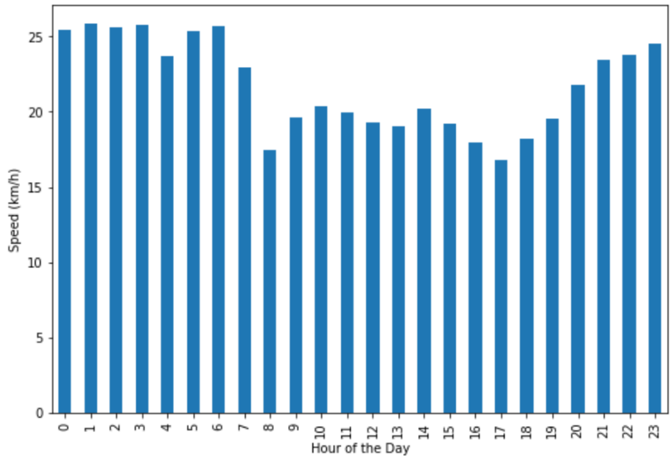


Figure 3.5: Bar chart with the average speed by hour of the day, for all the data

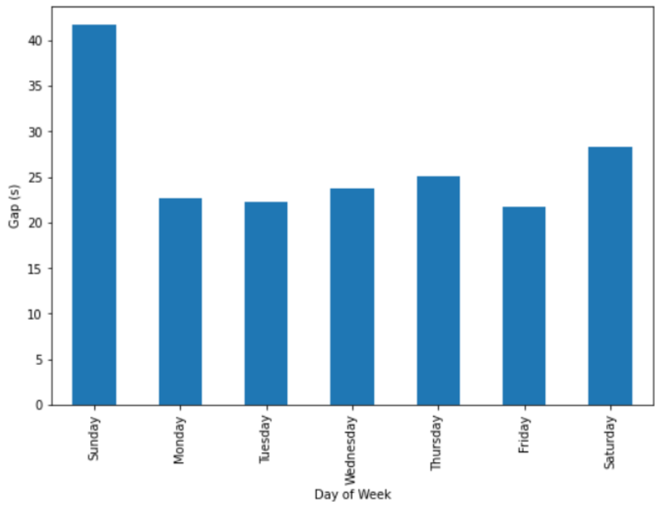


Figure 3.6: Bar chart with the average gap by day of the week, for the second week of March, 2022

decrease in commuting on those days. Furthermore, Figure 3.7 depicts a decline in the number of passings on February 27th, March 6th, March 13th, March 20th (Sundays), and March 1st (Portuguese Carnival).

Another way data was grouped was via the category assigned to each vehicle by the sensors. This data representation allows analysts to make conclusions related to specific categories, such as 68.42% of all registered passings being assigned to cars, as seen in Table 3.8.

The stakeholders were pleased with the data analysis conducted, so no further work was required in this phase.

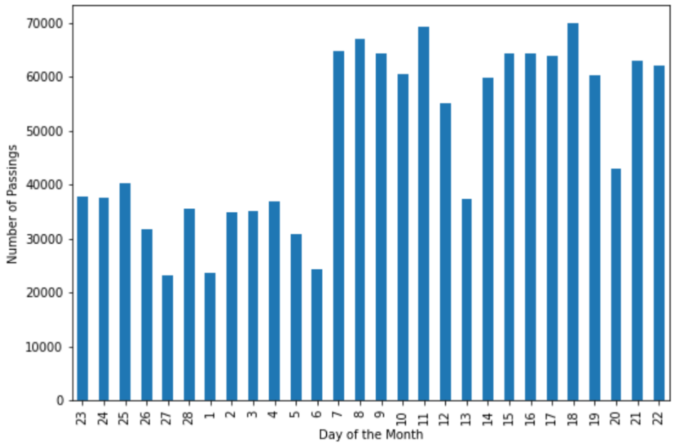


Figure 3.7: Bar chart with the total number of passings by day of the month, from February 23th until March 22th of 2022

Table 3.8: Statistical metrics for the data when grouped by category

Category	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
Bus	2867	0.19%	22.158	12.470	66.774
Car	1038347	68.42%	24.832	3.662	27.769
Car with trailer	1720	0.11%	21.957	9.052	102.355
Lorry	15119	1.00%	24.319	7.952	34.883
Lorry with trailer	2895	0.19%	36.758	22.733	25.741
Motorbike	13792	0.91%	27.990	1.125	34.765
Truck	3706	0.24%	26.520	14.654	78.041
Unknown	412549	27.18%	5.943	3.792	18.584
Van	26706	1.76%	24.827	5.407	54.405

3.2.3 Mockup Design

After knowing what information was available as of the previous phase, choices had to be made on how to display them. These choices include selecting the layout and aesthetic of the dashboard, as well as the best charts for exhibiting the processed data. As expected, the end-users and practical applications were considered in this process.

Firstly, a color palette had to be chosen. Since Soltráfego already had its palette, using it allowed the stakeholder to easily picture the dashboard as a solution they would own. The chosen colors were the company's signature blue (#007CBB) and white (#FFFFFF).

Secondly, two views were devised to satisfy the requirements of having an interactive map and displaying relevant data using appropriate visualization methods: a Map View and a Chart View. The Map View consists of a scene with an interactive map of the sensors' area, where the user can gather a general idea of the system's state. The Chart View delivers a collection of diverse and intuitive charts with varied information.

Ultimately, a filter feature was envisioned to increase user control and interactivity. It was important for data analysts to change the information presented based on a few constraints, such as the time interval, the vehicle category, the point, and the sensor.

Because the final product's look largely depended on the technologies that would be used in the implementation process, the mockup served as a guideline, not a final compromise; figures 3.8 and 3.9 show screenshots of the designed mockup.

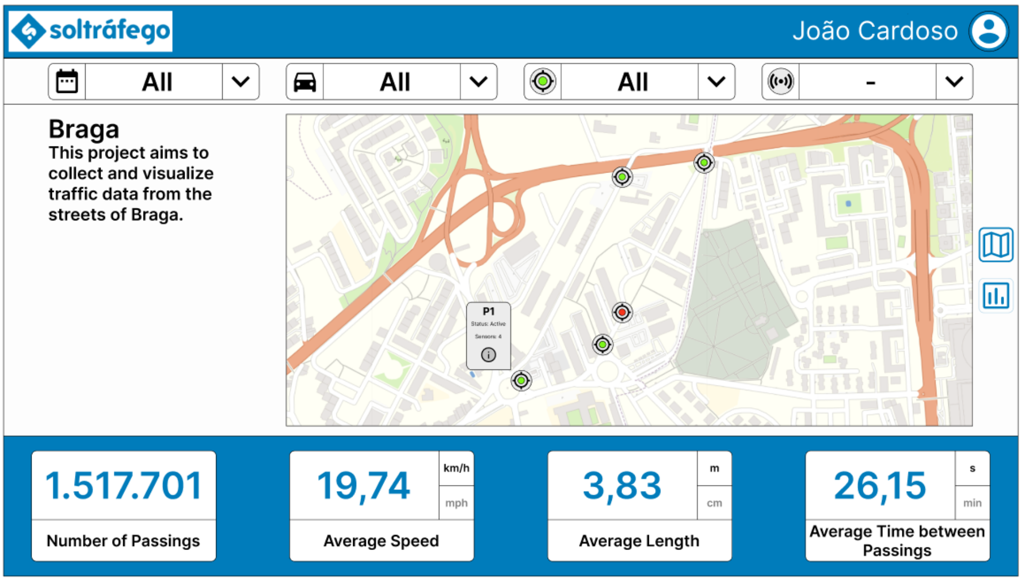


Figure 3.8: Mockup of the Map View

Figure 3.9: Mockup of the Chart View

After evaluating the design, the stakeholders proposed the inclusion of a third view with some data tables. While having engaging and appealing visual elements was undoubtedly welcomed, having classic tables allowed for a more complete solution. Tables enable data analysts to look at data more comprehensively.

3.2.4 Prototype Implementation

This section describes the dashboard implementation process, from the design choices to the functionalities implemented.

Implementing the dashboard requires determining the target device of the solution. As the end-users are data analysts and decision-makers, the dashboard was developed for desktop and laptop devices. As for what kind of application to create, the choice was between an application software to be installed on the target device or a web application to be used in a web browser. The latter was selected for two reasons: the previous solutions shown by Soltráfego were web apps; web apps are far easier and faster to implement due to an increase in accessible JavaScript/CSS libraries and frameworks in the last decade.

Picking the appropriate architecture and the programming languages, libraries, and frameworks for the implementation was another step that couldn't be overlooked. The charts in the application were required to support updating the data in runtime while exhibiting acceptable performance given the size of the dataset.

The most logical and simple architecture for a web application was a simple Server-Client model. The role of the server (or back-end) is to receive the data generated from the sensors and transform it according to the procedures described in the Data Analysis phase. The server should also provide an API for the transformed data to be used by the client. The client (or front-end) should constitute the application interface while communicating with the server to receive the data in real-time. An architecture diagram is presented in Figure 3.10.

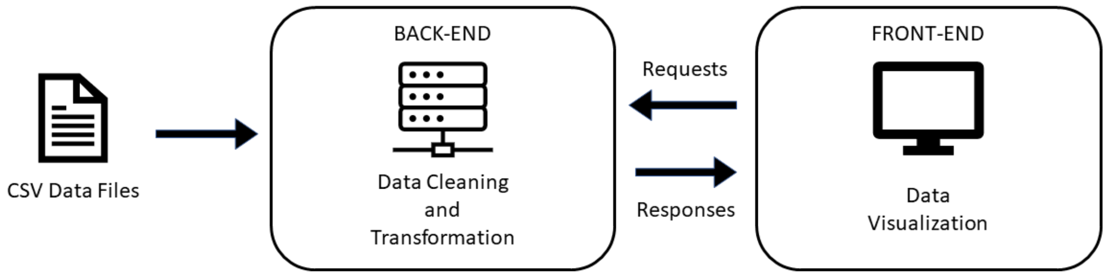


Figure 3.10: Client-Server architecture used

3.2.4.1 Front-End

The development started on the front-end side, as it would be easy to work with template data initially and later transition to using the data requested from the back-end. The front-end required a JavaScript framework that was easy to use and had a lot of supported features, so React was chosen for having these characteristics. Using JavaScript/CSS libraries and frameworks to facilitate the implementation process allowed more focus on having a clear, intuitive design and a good user experience. Since plain CSS is tedious and hard to work with, React-Bootstrap was used, offering a set of ready-to-use general-purpose components, such as navigation bars and buttons. As envisioned in the Dashboard Mockup phase, the web app includes three views: The Chart View, which was updated to the Home View, the Tables View, and the Map View.

For the Home View, three tools were considered for implementing the charts: Apache ECharts, Chart.js, and Recharts. While all of them offered esthetically pleasing React components, Recharts was chosen for being highly customizable and easy to use. Since this view is the dashboard's home page, it was essential to offer the users an overview of the entire system. As such, three intuitive types of charts were implemented:

- **A pie chart** — With this component, it is possible to check the absolute and relative amounts of passings at each point of the traffic system; See figure 3.11.

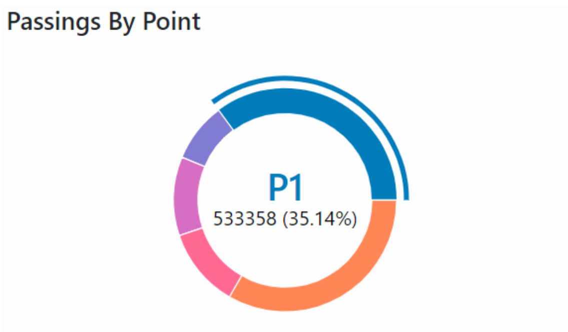


Figure 3.11: Pie chart component with passings by point

- **A biaxial bar chart** — This component displays the average speed and length of each vehicle category; See figure 3.12.

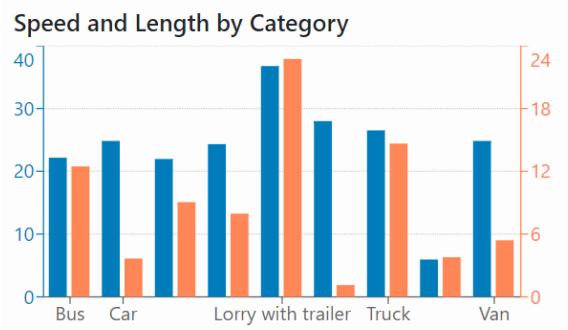


Figure 3.12: Biaxial bar chart component with speed and length by vehicle category

- **Three classic bar charts** — These charts show the average of three parameters (speed, time between passings, and the number of passings) over time. They also allow the user to choose the temporal granularity (between day, week, or month). See figure 3.13.

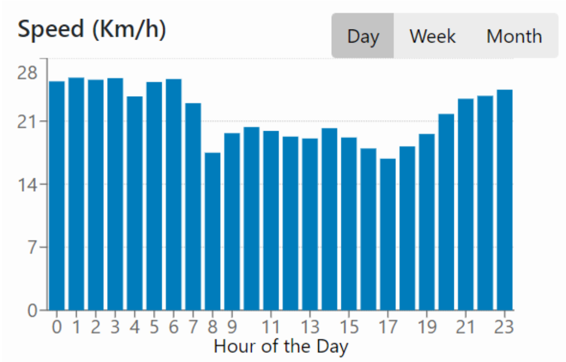


Figure 3.13: Bar chart component with average speed for each hour of a day

Additionally, a filter function was implemented in the Home View to give more control to the users over the data presented. The data parameters that can be adjusted are the date, the vehicle category, the point, and the individual sensor. A sliding sidebar was added to provide this functionality.

For the Tables View, the Material UI library allowed the implementation of data tables with built-in filters and column visibility toggles. Presenting the user with all the data in the CSV files was simply not feasible due to the high amount of information. Therefore, two tables were implemented, with the data grouped by point and category. While some of the information in this view is already present in the charts on the home page, it's essential to mention that the Tables View's function is to offer a more detailed look at the data, even at the cost of visual appeal. Figure 3.14 shows the implementation of one of the tables.

Passings By Point

COLUMNS FILTERS

Point	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
P1	533 358	35,142	18,919	4,096	23,063
P3	130 518	8,6	31,186	3,749	39,456
P4	173 365	11,423	26,858	3,97	44,602
P5	175 348	11,554	18,542	2,502	29,256
P6	505 112	33,281	15,621	3,995	18,568

Figure 3.14: Table with metrics of passings grouped by point

The Map View was the most challenging to implement, mainly because existing mapping frameworks are either restricted to paid subscriptions or lacking in the functionalities required.

Initially, experiments were done with the kepler.gl framework to evaluate its viability in this project, as pictured in Figure 3.15. This tool offers a highly engaging interface with a collection of interesting functionalities, such as a dynamic timeline with fluid animations. Unfortunately, the tool's installation using npm failed due to unsolvable package compatibility errors, making it impossible to incorporate into the dashboard.

The other mapping library that was experimented with was Leaflet. Even though this tool alone does not contain a lot of functionalities, it benefits from a list of plugins that extends it. Utilizing

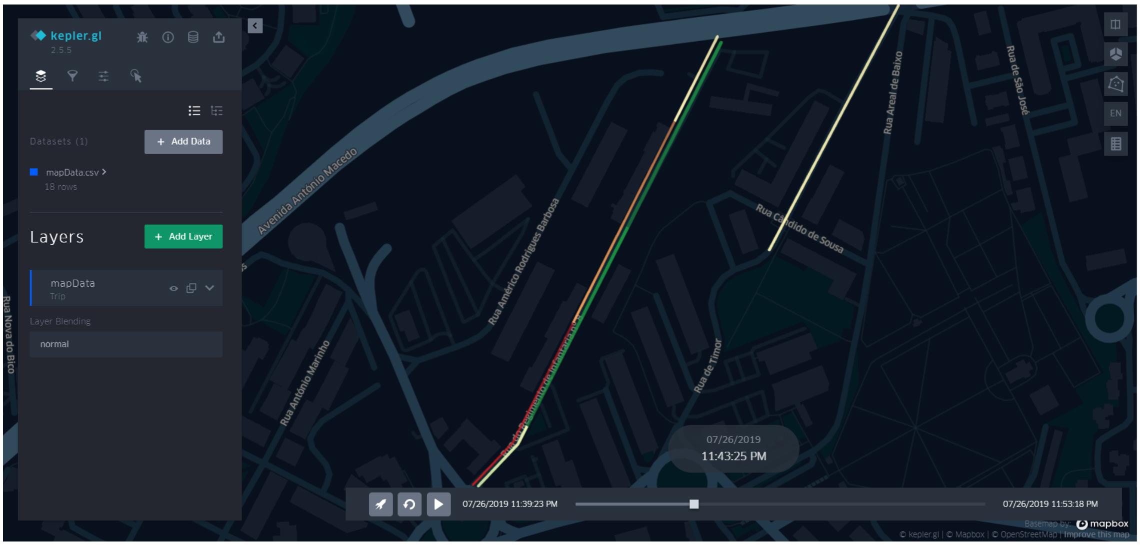


Figure 3.15: Screenshot of a kepler.gl demo, showing an animation in a timeline

the timeline plugin, it was possible to implement a simple timeline slider to display time-dependent data on the map. Since this tool included the required functionality for the project, it was selected for use.

Defining how to visualize the data in the map was crucial for a design that is easy to use and understand. For each road segment, the two most logical metrics for visualization were the average speed and traffic volume of the corresponding road segment and time interval. As concluded in [2] and [18], the best way to represent data on a map is by displaying lines representing each road while varying their size and color. A correlation was designed between these parameters and the metrics indicated above.

After some experimentation, it became apparent that whether color or size was used for average speed or traffic volume would not result in a significant difference in the final design, given that when the traffic volume was low, the average speed was high and vice-versa. Nevertheless, the traffic volume is conveyed in the map by the color of the line, with a red-green gradient, where red symbolizes a higher value and green a lower one. This gradient is frequently used in traffic-related visualizations, being Google Maps a notorious example. Conversely, the average speed is represented by the width of the line on the map, with a wider line being correlated to higher average speeds and vice-versa. A depiction of these colored lines on the map can be seen in Figure 3.16, along with markers representing the points of the system that comprise a group of sensors.

The data used by Leaflet is represented in GeoJSON, a subset of JSON utilized for encoding geographic data structures. As exemplified in Figure 3.17, a GeoJSON feature is defined by a LineString geometry with a set of coordinates and a group of properties. The "speed" and "numberPassings" represent the data associated with that feature, and the "start" and "end" properties represent the time boundaries of the data.

Combining multiple features with the same coordinates and consecutive timestamps makes it

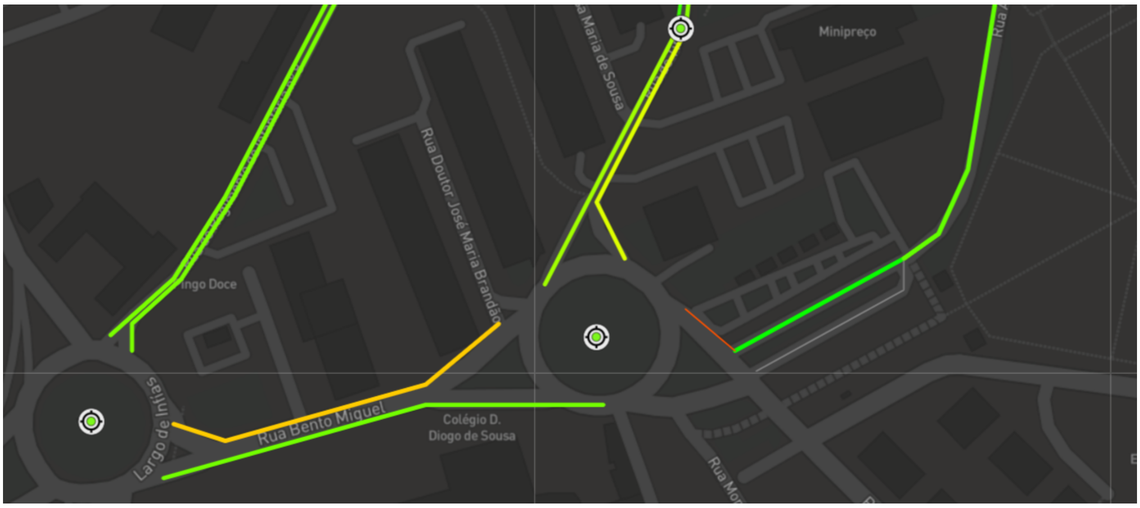


Figure 3.16: Representation of the average speed and traffic volume on the map, using lines with different color and width

```
{
    "type": "Feature",
    "geometry": {
        "type": "LineString",
        "coordinates": [
            [-8.41745, 41.55800, 0],
            [-8.41745, 41.55810, 0],
            [-8.41721, 41.55826, 0],
            [-8.41696, 41.55855, 0],
            [-8.41483, 41.56161, 0]
        ]
    },
    "properties": {
        "speed": 19.929,
        "numberPassings": 14.0,
        "start": 1648213200.0,
        "end": 1648213500.0
    }
}
```

Figure 3.17: GeoJSON feature representation of data associated with a road segment

possible to create a timeline of a specific road segment. Similarly, following this process for all the roads where the sensors are located extended the timestamp to the whole system.

It's important to clarify that the data associated with each road segment was collected by a sensor located at the start of the segment (considering driving directions). This means that the data for each road segment was extrapolated by assuming vehicles that passed over a sensor kept their speed over the entire stretch. This assumption is reasonable since the traffic sensors are not positioned too far apart and are almost all installed in intersections.

While at first the data presented in the web application was static for ease of implementation, a back-end was developed to provide the front-end with data dynamically.

3.2.4.2 Back-End

The choice of programming language to use in the back-end was evident. Python was used, as it would be easy to reuse the code implemented in the Data Analysis phase contained in the Jupyter Notebook.

A simple API with a few routes was developed using the Flask framework. Upon startup, the server imports the CSV data files and executes the data cleaning process. The defined routes receive the filters established in the front-end as parameters in HTTP GET requests so that they can apply these filters to the data before responding with the filtered data in JSON format.

While CSV files made the implementation more manageable and were an acceptable choice for a prototype, a solution ready for a production environment should have access to the sensors' API instead.

As a company that offers traffic solutions, Soltráfego is familiar with the data analysis field. In that regard, they expressed concern about the lack of an export functionality in the tables featured in the Table View. It is a convenient feature in this field, allowing users to easily extract the information in question for reports and other documentation. For this reason, that functionality was implemented, allowing users to export the data in the tables to a CSV file (with filters applied).

3.2.5 Final Validation

The last phase of this work was a final validation of the implemented solution with the stakeholder. Conducting another meeting with Sóltrafego, the product was thoroughly presented, the architecture was explained, and the functionalities and use cases were demonstrated.

The stakeholder expressed their overall content and approval of the implemented dashboard. As a stakeholder and expert in the traffic monitoring area, Soltráfego's evaluation validates the work developed and the methodology followed.

Some future functionality was also discussed. As stated before, using CSV files as input for a dashboard only made sense during the development phase, but upon deployment of a real-life solution, a connection between the back-end and the sensor's API should be established. Additionally, implementing a login feature with proper authentication solutions would allow access only to intended users.

Chapter 4

Dashboard Implementation, Validation and Discussion

This chapter describes the implemented functionalities of the dashboard and challenges faced and discusses stakeholder validation.

4.1 Functionalities

A navigation bar on the top of the page allows the user to switch between different views, all of which will now be explained in further detail.

4.1.1 Home View

The Home View contains a system description and various charts, as seen in Figure 4.1. A pie chart allows the user to watch the number of passings registered by each point of the sensor system. A biaxial bar chart displays the average speed and length of the vehicles, grouped by their category. Three simple bar charts show the hourly average values of speed, the time between passings, and their count over a day. Additionally, these charts can be individually changed to display the daily average values over a week or a month. All the bar charts include tooltips so users can inspect specific values.

All the data in this view can be filtered using the filter menu (depicted in Figure 4.2), which is accessible by pressing the corresponding button. This menu allows the user to filter the data by time, vehicle category, point the data was measured in, and individual sensor. The charts change automatically when filters are modified.

4.1.2 Tables View

The Tables View provides a more detailed look at the data than the charts, even though they are not as interactive and visually appealing. The tables have a filter functionality, giving the data analysts and decision-makers more control over the data they can see and a feature that allows users to

Figure 4.1: Screenshot of the Home View of the dashboard

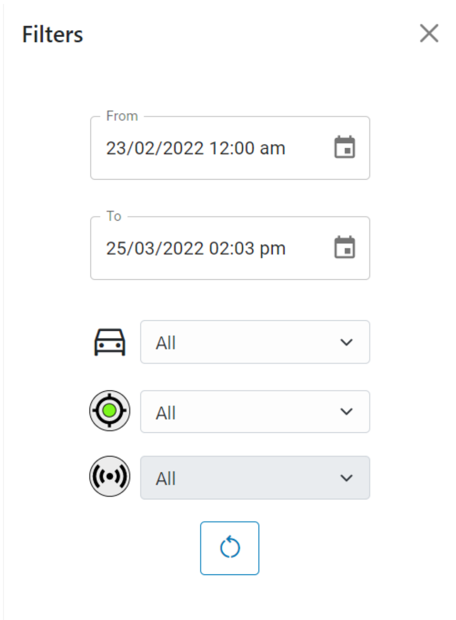
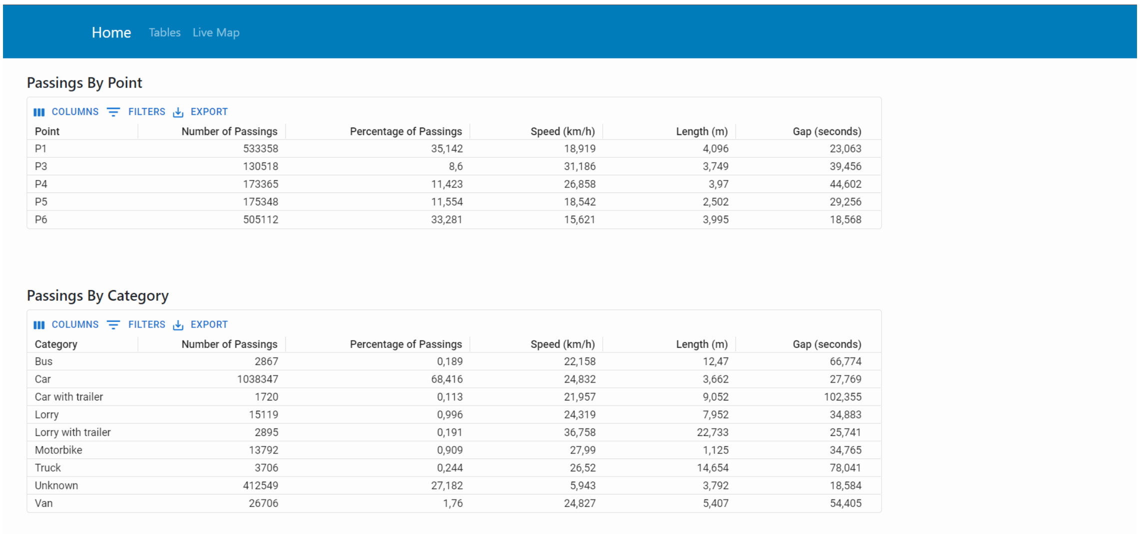


Figure 4.2: Screenshot of the filters sidebar of the dashboard

export a table's data to a CSV file for later inspection or documentation. As seen in Figure 4.3, this view has a table with various metrics for each point and an identical table for the category of vehicles.

4.1.3 Map View

The Map View delivers a more interactive experience for the user. A map shows the current state of traffic in the area of the sensors with different colored lines with varying widths. Green



Passings By Point

Point	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
P1	533358	35,142	18,919	4,096	23,063
P3	130518	8,6	31,186	3,749	39,456
P4	173365	11,423	26,858	3,97	44,602
P5	175348	11,554	18,542	2,502	29,256
P6	505112	33,281	15,621	3,995	18,568

Passings By Category

Category	Number of Passings	Percentage of Passings	Speed (km/h)	Length (m)	Gap (seconds)
Bus	2867	0,189	22,158	12,47	66,774
Car	1038347	68,416	24,832	3,662	27,769
Car with trailer	1720	0,113	21,957	9,052	102,355
Lorry	15119	0,996	24,319	7,952	34,883
Lorry with trailer	2895	0,191	36,758	22,733	25,741
Motorbike	13792	0,909	27,99	1,125	34,765
Truck	3706	0,244	26,52	14,654	78,041
Unknown	412549	27,182	5,943	3,792	18,584
Van	26706	1,76	24,827	5,407	54,405

Figure 4.3: Screenshot of the Tables View of the dashboard

lines represent lighter traffic, and red lines represent heavier traffic. The width of the lines is directly proportional to the traffic speed in that road segment. This view also contains a timeline functionality: pressing the play button shows the traffic state from the last hour, with each update representing five minutes. Other implemented map features include markers for each point of sensors in the system and tooltips so the user can inspect the actual values being displayed. This view can be seen in Figure 4.4.

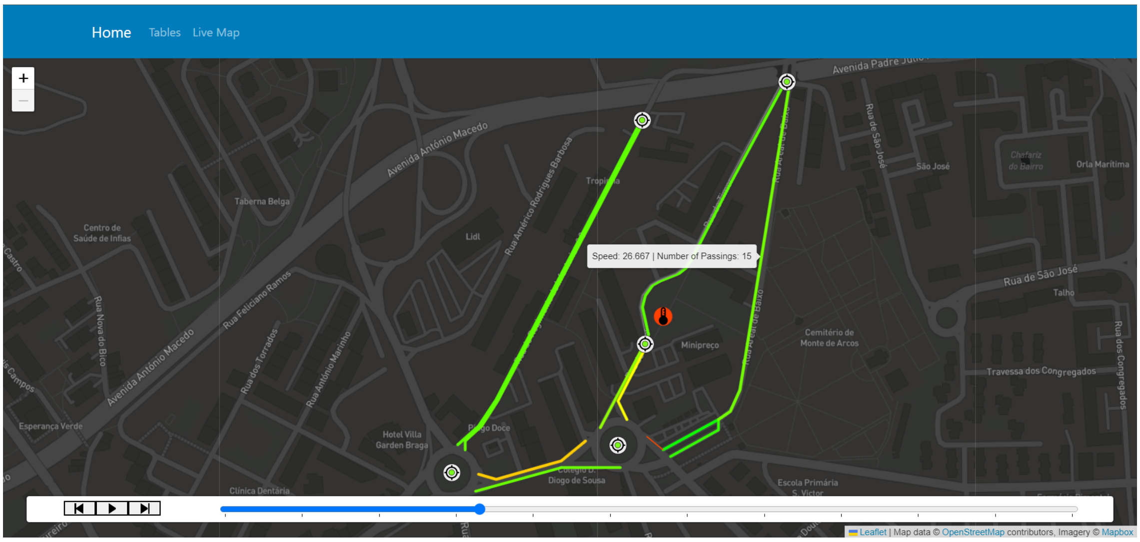


Figure 4.4: Screenshot of the Map View of the dashboard

4.2 Stakeholder Validation

This work had the crucial contribution of a traffic monitoring expert and the Portuguese national leader in the mobility market, Soltráfego. Their help in the requirements phase provided clear goals and objectives for developing the dashboard and introduced valid concerns that were considered in the following stages of development. The mockup design was improved due to their recommendation to add an additional view from which data analysis would benefit. The stakeholder's contribution to the prototype implementation resulted in the addition of relevant features that improved the final solution.

Regarding the final prototype, Soltráfego was pleased with the functionalities implemented and the dashboard layout. According to them, the system architecture was adequate for the problem. The missing export functionality in the tables was implemented as a result of their feedback, leading to a better traffic monitoring and analysis tool.

Some future functionality was also suggested by the company. As stated before, using CSV files as input for a dashboard only made sense during the development phase, but upon deployment of a real-life solution, a connection between the back-end and the sensor's API should be established. Additionally, implementing a login feature with proper authentication solutions would allow access only to intended users.

The stakeholder expressed their overall content and approval of the implemented dashboard. As a stakeholder and expert in the traffic monitoring area, Soltráfego's evaluation validates the work developed and the methodology followed. With this in mind, it's possible to conclude that the final dashboard is an adequate traffic data visualization solution. Following a user-centered design allowed the solution to fulfill the requirements of a suitable dashboard more effectively.

4.3 Discussion

After designing and implementing a city dashboard using traffic data, some observations can be made about the methodology proposed and the work done.

The data analysis phase confirms the importance of cleaning raw data. In this phase, steps were taken to ensure the validity and exactness of the data, contributing to a more reliable dashboard. Although the data was not very diverse, it was still possible to extract a fair amount of information to visualize. An origin-destination matrix could have been constructed with more complex datasets, like vehicle identification for each sensor detection.

The data visualization phase presented many challenges, from the most appropriate layout to the type charts to utilize. Data analysts need an intuitive interface with reliable and readable data. These goals were achieved by employing a minimalistic design with a soothable color palette. The presence of interactable charts and tables enables the users to filter and adjust the interface to specific needs and use cases.

Including a map in the implementation offered a unique opportunity to explore more complex visualization techniques as the timeline feature. While there was a yearning for more visually appealing graphics, the available mapping tools proved to be lacking. The existing tools for map visualization either offer limited functionality like simple markers and heatmaps, are very complex to use and work with, or are paid solutions that cannot be easily used as a study target. Developing such tools can be challenging, but an open-source alternative to subscription-based frameworks would allow for higher-quality dashboards to be built.

4.4 Research Questions

RQ1: What information can be extracted from raw sensor data?

The present work explores data analysis and processing approaches of raw traffic data measured by IoT sensors, enabling the extraction of new information essential for problem-solving. Grouping datasets by a particular parameter provides a different view of the data, allowing data analysts to make informed decisions in city management.

Looking at the raw data, little was possible to conclude about the traffic situation in the area. However, plotting the data over time made it possible to deduce that traffic intensity was higher or lower during specific day periods, specific days of the week, and even on some days of the month. Similarly, analyzing the relative frequency of passings of specific vehicle categories can exhibit the level of adoption of public transportation by the public. Speed can also be monitored when transforming raw data, thus aiding city managers in mitigating traffic accidents in dangerous areas.

RQ2: What are the best techniques for traffic data visualization?

Data visualization challenges are intrinsic to dashboard development, and the choices taken define the solution's efficacy. As expected, pie and bar charts provide a simple and intuitive way to display a considerable amount of information to users. As such, featuring them on the home page of any application is an appropriate choice. Nonetheless, using tables proved to be very effective for a more detailed data analysis.

However, the traffic domain brings forth more visualization opportunities, such as utilizing mapping tools to display the road system's status. Representing data on a map using lines with varying colors and widths resulted in an intuitive illustration of traffic that allows users to quickly diagnose specific road segments. Furthermore, adding a timeline functionality enhances a map's capabilities, enabling the representation of spatiotemporal data.

RQ3: How to design a city monitoring dashboard using a user-centered design approach?

One of the most critical attributes of a dashboard is the quality of user experience it provides. Failing to deliver this trait can invalidate a technically impressive implementation. As such, this work explored the application of a user-centered design philosophy in the entire development process. Soltráfego served the role of the users due to their expertise in the traffic monitoring field and experience in traffic solutions.

By involving the company in all phases, the work produced was better aligned with their needs and expectations, creating a solution better suited for its purpose. In addition, the iterative incorporation of feedback resulted in a more complete dashboard, with features that data analysts greatly value.

Chapter 5

Conclusions and Future Work

This dissertation studied the development of a city dashboard using traffic data following a user-centered design. A methodology comprised of five distinct phased was defined and followed, with involvement from Soltráfego, a traffic monitoring company that provided their feedback and expertise in every phase.

Requirements were gathered with Soltráfego, as their expertise and experience in the domain proved valuable attributes in designing a dashboard.

Data cleaning was performed using the data provided by the stakeholder, ensuring the data used for the following phases were not invalid, followed by data transformation, allowing interesting representations to be analyzed.

A mockup was designed to define the solution's layout, aesthetics, and features. This phase turned ideas and concepts into drafts to ensure the stakeholder's vision was upheld.

A prototype was implemented using the mockup as a guideline and expanding on it, focusing on interactivity, intuitiveness, and visual clarity. Many interesting visualization techniques were employed, such as charts, tables, and a timeline on a map.

Finally, a final assessment of the stakeholder validated the work accomplished and the methodology conducted. Their contribution and guidance throughout the project proved crucial in creating a proper solution for traffic monitoring.

Future improvements to the dashboard would involve making it ready for production by replacing the data source from CSV files with a sensors' API. Additionally, a login functionality could be implemented to restrict access to authorized personnel.

References

[1] C. Bachechi, L. Po, and F. Rollo. Big data analytics and visualization in traffic monitoring. *Big Data Research*, 27, 2022.

[2] I. Caldas, J. Moreira, J. Rebelo, and R. J. F. Rossetti. Exploring visualization metaphors in macroscopic traffic simulation. In *2018 IEEE International Smart Cities Conference, ISC2 2018*.

[3] A. Clarinval and B. Dumas. Intra-city traffic data visualization: A systematic literature review. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[4] F. Desimoni, S. Ilarri, L. Po, F. Rollo, and R. Trillo-Lado. Semantic traffic sensor data: The trafair experience. *Applied Sciences-Basel*, 10(17), 2020.

[5] H. Ding, M. Cai, X. F. Lin, T. Chen, L. Li, and Y. H. Liu. Rtvemvs: Real-time modeling and visualization system for vehicle emissions on an urban road network. *Journal of Cleaner Production*, 309, 2021.

[6] H. Ebrahim, T. Santhanavanich, P. Wuerstle, and V. Coors. Concept and evaluation of an urban platform for interactive visual analytics. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 8, pages 33–40.

[7] X. L. Fan, B. Q. He, and P. Brezillon. Context-aware big data analytics and visualization for city-wide traffic accidents. *Modeling and Using Context (Context 2017)*, 10257:395–405, 2017.

[8] M. Farmanbar and C. Rong. Triangulum city dashboard: An interactive data analytic platform for visualizing smart city performance. *Processes*, 8(2), 2020.

[9] E. Felemban, F. U. Rehman, A. A. Biabani, A. Naseer, O. Hussain, and E. U. Warriach. An interactive system for analyzing movement of buses in hajj. *Journal of Theoretical and Applied Information Technology*, 98(21):3468–3481, 2020.

[10] C. Gao, L. Xu, and W. Li. Research on visualization method of congestion pattern in urban transportation hub. In *ACM International Conference Proceeding Series*, 2021.

[11] M. T. C. Garcia and L. G. Montane-Jimenez. Visualization to support decision-making in cities: advances, technology, challenges, and opportunities. *2020 8th Edition of the International Conference in Software Engineering Research and Innovation (Conisoft 2020)*, pages 198–207, 2020.

[12] C. Jing, M. Du, S. Li, and S. Liu. Geospatial dashboards for monitoring smart city performance. *Sustainability (Switzerland)*, 11(20), 2019.

[13] J. Kazak, M. Chalfen, J. Kaminska, S. Szewranski, and M. Swiader. Geo-dynamic decision support system for urban traffic management. *Dynamics in Giscience*, pages 195–207, 2018.

[14] K. Kourtit and P. Nijkamp. Big data dashboards as smart decision support tools for i-cities - an experiment on stockholm. *Land Use Policy*, 71:24–35, 2018.

[15] A. T. Kwee, M. F. Chiang, P. K. Prasetyo, and E. P. Lim. Traffic-cascade: Mining and visualizing lifecycles of traffic congestion events using public bus trajectories. *Cikm'18: Proceedings of the 27th Acm International Conference on Information and Knowledge Management*, pages 1955–1958, 2018.

[16] L. Liu, H. Zhan, J. Liu, and J. Man. Visual analysis of traffic data via spatio-temporal graphs and interactive topic modeling. *Journal of Visualization*, 22(1):141–160, 2019.

[17] D. Mahajan, Y. Karnati, T. Banerjee, V. R. Regalla, R. Reddy, A. Rangarajan, and S. Ranka. A scalable data analytics and visualization system for city-wide traffic signal data-sets. *2020 Ieee 23rd International Conference on Intelligent Transportation Systems (Itsc)*, 2020.

[18] M. Morgan and R. Lovelace. Travel flow aggregation: Nationally scalable methods for interactive and online visualisation of transport behaviour at the road network level. *Environment and Planning B-Urban Analytics and City Science*, 48(6):1684–1696, 2021.

[19] J. Pereira, S. Sargento, and J. M. Fernandes. Decision support dashboard for traffic and environment analysis of a smart city. In *VEHITS 2018 - Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, volume 2018-March, pages 387–394.

[20] F. Qian. Visualization of traffic data: A survey of methods and datasets. In *2020 IEEE 6th International Conference on Computer and Communications, ICCC 2020*, pages 1990–1994, 2020.

[21] A. Sakib, S. A. Ismail, H. Sarkan, A. Azmi, and O. Mohd Yusop. Analyzing traffic accident and casualty trend using data visualization. In *Advances in Intelligent Systems and Computing*, volume 843, pages 84–94. 2019.

[22] Z. Shao, Z. Lin, and W. Hao. Research on visual analysis technology and application of urban road data. In *IOP Conference Series: Materials Science and Engineering*, volume 688.

[23] T. Sobral, T. Galvão, and J. Borges. Visualization of urban mobility data from intelligent transportation systems. *Sensors (Switzerland)*, 19(2), 2019.

[24] A. Twomlow, S. Grainger, K. Cieslik, J. D. Paul, and W. Buytaert. A user-centred design framework for disaster risk visualisation. *International Journal of Disaster Risk Reduction*, 77, 2022.

[25] R. A. Vila, E. Estevez, and P. R. Fillottrani. The design and use of dashboards for driving decision-making in the public sector. *Proceedings of the 11th International Conference on Theory and Practice of Electronic Governance (Icegov2018)*, pages 382–388, 2018.

[26] G. W. Young and R. Kitchin. Creating design guidelines for building city dashboards from a user's perspectives. *International Journal of Human-Computer Studies*, 140, 2020.

[27] V. Zdraveski, K. Mishev, D. Trajanov, and L. Kocarev. Iso-standardized smart city platform architecture and dashboard. *IEEE Pervasive Computing*, 16(2):35–43, 2017.