BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

ANALYSIS AND VISUALIZATION OF TRAFFIC ACCIDENT DATA

ANALÝZA A VIZUALIZACE DAT DOPRAVNÍCH NEHOD

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **VERONIKA ŠIMKOVÁ**
AUTOR PRÁCE

SUPERVISOR **Ing. MAGDALÉNA ONDRUŠKOVÁ**
VEDOUCÍ PRÁCE

BRNO 2025

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

164963

Institut:	Department of Information Systems (DIFS)
Student:	**Šimková Veronika**
Programme:	Information Technology
Title:	**Analysis and Visualization of Traffic Accident Data**
Category:	Information Systems
Academic year:	2024/25

Assignment:

1. Study the types of data intended for evaluation of urban automobile traffic, methods of their acquisition, intrepretation and visualization.
2. Study technologies and tools for data processing and evaluation.
3. Analyse datasets about traffic accidents. Evaluate the potential use of these data for urban transportation improvement.
4. Propose an extension of the system analyzing Waze data to analyze and visualize the data from item 3.
5. Implement the proposed solution.
6. Perform user testing of the implemented solution.

Literature:

- Haining, M. (2010). *Spatial data analysis: Theory and practice*. Cambridge University Press. ISBN: 9780511754944. Available at: https://doi.org/10.1017/CBO9780511754944.
- Rehborn, H., Koller, M., Kaufmann, S. *Data-driven traffic engineering*. Elsevier. ISBN: 9780128191385.
- Ondrušková, M. (2024). Analysis and Visualization of Brno Traffic Data. Master Thesis. Brno University of Technology. Faculty of Information Technology. Available at: https://www.vut.cz/studenti/zav-prace/detail/153666

Requirements for the semestral defence:
Points 1 to 4.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Ondrušková Magdaléna, Ing.**
Head of Department:	Kolář Dušan, doc. Dr. Ing.
Beginning of work:	1.11.2024
Submission deadline:	14.5.2025
Approval date:	22.10.2024

Abstract

This bachelor's thesis extends an existing web tool for analysing Waze congestion data by integrating police accident reports and crowdsourced Waze crash alerts for the city of Brno. After preparing the datasets and removing duplicate user reports, the datasets are matched based on spatial and temporal proximity to enhance official records and fill in gaps. The resulting data is presented on an interactive map and in the form of charts, offering a set of filters for customizing both views. Similarly to the original work, the extension was developed using Python and React. User testing was conducted to confirm that the new functionality clarifies accident hotspots and supports data-driven planning.

Abstrakt

Táto bakalárska práca rozširuje existujúci webový nástroj na analýzu dát o dopravných zápchach z aplikácie Waze tým, že prepája policajné záznamy o nehodách s hláseniami nehôd z aplikácie Waze. Po spracovaní datasetov a odstránení duplicitných hlásení sú záznamy spárované na základe priestorovej a časovej blízkosti, aby doplnili oficiálne štatistiky a vyplnili informačné medzery. Výsledné údaje sa zobrazujú na interaktívnej mape a vo forme grafov, pričom oba pohľady možno prispôsobiť pomocou filtrov. Rovnako ako pôvodná aplikácia je aj toto rozšírenie vyvinuté v Pythone a Reacte. Užívateľské testovanie potvrdilo, že nová funkcionalita zreteľne vizualizuje nehodové lokality a uľahčuje plánovanie založené na dátach.

Keywords

traffic, traffic data, data analysis, data visualization, traffic accidents, Waze, Brno, road safety

Klíčová slova

doprava, dopravné dáta, dátová analýza, vizualizácia dát, dopravné nehody, Waze, Brno, bezpečnosť na cestách

Reference

ŠIMKOVÁ, Veronika. *Analysis and Visualization of Traffic Accident Data.* Brno, 2025. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Magdaléna Ondrušková

Rozšířený abstrakt

Bezpečnosť cestnej premávky patrí medzi dlhodobo sledované spoločenské témy, ale aj napriek zlepšujúcim sa štatistikám zostáva počet dopravných nehôd vysoký a ich dopady na zdravie a majetok sú neprehliadnuteľné. Úradné databázy, ktoré publikuje Policajný zbor Českej republiky, ponúkajú detailný pohľad na vážne incidenty, zapríčinené škody a príčiny nehôd, avšak pri menej závažných udalostiach či krátkodobých prekážkach strácajú pokrytie. Navigačná aplikácia Waze poskytuje komplementárnu dátovú sadu hlásení od užívateľov tejto aplikácie s presným časovaním, ktorý pomáha rozšíriť obzory ale aj vypĺňať časové a medzery v oficiálnej dátovej sade. Bakalárska práca sa zameriava na spojenie oboch zdrojov a na vytvorenie rozšírenia webovej aplikácie, pôvodne využívajúcej dáta o dopravných zápchach z aplikácie Waze, ktoré umožňuje spoločnú analýzu a vizualizáciu oboch nových zdrojov v prostredí mesta Brno s ambíciou rozšírenia na ďalšie územia. Rovnako ako pôvodná aplikácia, rozšírenie je primárne určené pre dve skupiny užívateľov a to širokú verejnosť, ale aj odborníkov na mestské plánovanie a dopravu. Kým pre vývoj aplikácie boli dostupné len dáta mesta Brno, aplikácia bola vyvíjaná tak, aby bola veľmi jednoducho rozšíritelná pre celé územie Českej Republiky, ak budú dodané dáta v kompatibilnom formáte.

Údaje z policajných protokolov obsahujú desiatky atribútov vrátane kategorizácie príčin, poveternostných podmienok, stavu vozovky či následkov na zdraví a majetku. Lokalizované sú v národnom súradnicovom systéme S-JTSK. Waze udáva polohu v globálnom referenčnom rámci WGS 84, hlásenia generuje v dvojminútových intervaloch a priraďuje im spoľahlivosť na základe reputácie používateľa. Prvým krokom riešenia bol preto jednotný prevod súradníc, očistenie dát od zjavne chybných záznamov a dekódovanie policajných atribútov na čitateľný formát. Z dôvodu rozdielnych zdrojov dát a absencie spoločného identifikátora medzi policajnými záznamami a hláseniami z aplikácie Waze bolo potrebné vyvinúť metodiku pre ich vzájomné spárovanie. Tento proces bol postavený na porovnaní časovej a priestorovej blízkosti jednotlivých udalostí. Z dôvodu opakovaného nahlasovania tých istých udalostí boli Waze hlásenia najprv spracované zhlukovaním, aby sa odstránili duplicity pri vizualizácii, no samotné párovanie s policajnými dátami prebiehalo s pôvodnými hláseniami. Ku každej nehode sa vyhľadávali všetky relevantné Waze hlásenia v okolí, pričom sa hodnotila ich vzdialenosť a časová súvislosť. Pre policajné záznamy bez presného času sa časový údaj pároval na základe dátumu a následne bol pre tieto nehody prevzatý čas zo súvisiaceho Waze hlásenia s najskorším časovým údajom.

Celá aplikácia je postavená na trojvrstvovej architektúre. V dátovej vrstve, ktorá bola poskytnutá vrámci pôvodnej aplikácie, je použitý datbázový systém PostgreSQL. Vyvíjaná aplikačná vrstva využíva Python a FastAPI; poskytuje REST rozhranie pre selektívne dotazovanie nehôd, zhlukov Waze hlásení a počítanie štatistík. Pre vývoj prezentačnej vrstvy bol zvolený React s TypeScriptom a Tailwindom, pričom mapovú vizualizáciu zabezpečuje knižnica React–Leaflet. Stav filtrov a používateľských volieb drží knižnica Zustand, sieťovú komunikáciu manažuje React Query.

Mapové rozhranie je možné prepínať medzi viacerými možnosťami zobrazenia. Po priblížení sa zobrazujú jednotlivé nehody a incidenty identifikované z Waze hlásení, pri oddialení dochádza ku zhlukovaniu, čo výrazne zlepšuje prehľadnosť. Panel detailu nehody sumarizuje všetky kľúčové atribúty s možnosťou zobraziť kompletný list dostupných atribútov a možnosťou kopírovať presné GPS súradnice. Pri polacajných záznamoch sú uvedené korešpondujúce Waze hlásenia.

V časti dashboard sú prehľadové ukazovatele umiestnené v hornej časti, ktoré zobrazujú celkový počet nehôd, počet obetí na životoch, počet ťažko zranených osôb a odhadovanú

hodnotu škôd na majetku. Pod týmito ukazovateľmi nasledujú časové vizualizácie, ktoré zobrazujú denný počet hlásených nehôd kombinovaním policajných a Waze dát, ako aj graf trendu závažnosti nehôd s počtami obetí, ťažkých a ľahkých zranení. Nižšie sa nachádzajú kategorizované grafy, ktoré rozkladajú typy nehôd, príčiny a charakteristiky dopravnej infraštruktúry vrátane klasifikácie ciest, geometrie úsekov a pozície nehôd na vozovke.

Nasledovalo užívateľské testovanie, do ktorého sa zapojila vzorka užívateľov z oboch skupín. Respondenti dostali sadu vopred definovaných úloh, ktoré splnili bez významných chýb. Následne im bolo umožnené samostatne preskúmať systém. Na základe ich spätnej väzby boli pridané napríklad farebné odlíšenia symbolov na mape na základe závážnosti nehody a zhlukovanie nedôd pri oddialení.

Výsledkom práce je rozšírenie webovej aplikácie, ktoré umožňuje vizualizáciu a analýzu dopravných nehôd spojením oficiálnych policajných dát s hláseniami z Waze. Vďaka tomu sa podarilo zlepšiť pokrytie incidentov a sprístupniť prehľadné štatistiky pre odborníkov aj verejnosť. Implementované riešenie bolo overené užívateľským testovaním a vytvára základ pre ďalší vývoj a rozšírenie pre celé územie Českej Republiky.

Analysis and Visualization of Traffic Accident Data

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Magdaléna Ondrušková. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Veronika Šimková
May 14, 2025

Acknowledgements

I would like to thank my boyfriend for supporting me throughout working on this thesis and to my thesis supervisor, Ing. Magdaléna Ondrušková, for her help and guidance.

Contents

List of Figures

Chapter 1

Introduction

Transportation is an inevitable part of our daily lives. Whether one chooses to drive, use public transport, bike or walk, there is always a risk of being involved in a traffic accident. These accidents happen unexpectedly, without warning, and can cause a wide range of problems, from minor inconveniences, like being late for work, to the worst-case scenario, loss of life. According to the World Health Organization, vehicle accidents are responsible for the deaths of approximately 1.19 million people and 20 to 50 million non-fatal injuries globally each year. Vulnerable traffic participants like pedestrians and cyclists are at the highest risk, accounting for more than half of the deaths [20]. In urban areas, the concentration of vehicles, bikers and pedestrians is greater, and the likelihood of traffic accidents naturally increases. Thus, effective analysis and visualization of traffic incident data is crucial for improving road safety and urban infrastructure planning.

The aim of this thesis is to create an extension of an existing web application titled *Waze Data Analysis*[1] [19]. The extension will add analysis and visualizations of available data about traffic accidents. Access to data, especially visualised in a clear, intuitive way, can increase awareness among citizens, making them adjust their behaviour when approaching hotspots, potentially preventing further accidents. This extension can also help city officials target problematic areas, redesign traffic rules, or even guide their decisions when assigning limited resources to rebuild parts of the infrastructure. The functionality of the project is illustrated using data from the city of Brno, but the algorithms are applicable to other cities within the Czech Republic.

The thesis begins with Chapter 2, introducing key concepts such as vector and raster data, coordinate systems, and the role of Geographic Information Systems in traffic analysis. Chapter 3 explores data processing, integration, and visualisation techniques to reveal trends and insights. Chapter 4 describes the datasets used in this thesis, their origins and their role in addressing user needs. It also investigates potential users and their requirements for the application.

Chapter 5 outlines the design concept of the application, emphasizing its objectives and proposed functionalities, which are further detailed in Chapter 6, focusing on the technical realization and development process. Chapter 7 describes the methods used to test the system, including feedback from users and explains the changes made as a result. Lastly, Chapter 8 reflects on the project outcomes, identifies limitations, and discusses opportunities for future enhancements.

[1] Waze Data Analysis: https://analyticity.github.io/waze-data-analysis/

Chapter 2

Traffic and Geographic Data: An Overview

This chapter provides an overview of geographic and traffic data, focusing on their types, collection methods, sources, and uses. It introduces the key features of geographic data and the role of coordinate systems and Geographic Information Systems (GIS) in providing spatial context. Traffic data is further examined, highlighting its use and comparing methods of data collection. Finally, the chapter discusses how collecting and analyzing traffic data contributes to safer and more efficient transportation systems, exploring existing applications.

2.1 Types of Geographic Data and Traffic Data

To make sense of traffic data, it is essential to first understand geographic data as location is one of the most important aspects of traffic accidents. Without the geographic component, it would lack the context needed for a meaningful analysis, such as identification of hotspots, temporal-spatial analysis or correlation of accidents with road structures.

2.1.1 Geographic Data

Historically, geographic data was primarily represented and passed on through hand-drawn maps, but with the rise of digital computing in the 1960s, it began to transform. Today, there are two primary types of structures for storing and displaying geographic data: vector and raster, each with its own benefits and drawbacks. Data in these formats can be further used by Geographic Information Systems (GIS), further discussed in 2.1.3.

Vector Data

Vector data is made up of vertices and paths. There are three types of symbols: points, lines and polygons [7].

- **Points** are XY coordinate pairs, representing latitude and longitude within a spatial reference frame. While latitude and longitude enable pinpointing a precise location, they are often used to represent larger objects that would be too small to be mapped as polygons at larger scales.

- **Lines** are collections of ordered points connected by vertices, with each point representing a vertex. They are used to represent objects linear in nature, objects that have length but no significant area, like roads and rivers.
- **Polygons**, similarly to lines, are formed by a collection of vertices connected in a particular order, where the first and last coordinate pairs are the same, closing the shape. Polygons represent features that have an area, such as buildings, lakes or boundaries.

Using vector data allows for precision when representing discrete features. The features retain their detail at any zoom level. As shown in Figure 2.1 using vector format allows to represent the complex shapes accurately, which makes it crucial for urban planning and management [6]. Vector data comes in multiple formats. The most common ones are listed below.

- **GeoJSON**

  GeoJSON[1] is a format based on JavaScript Object Notation (JSON). The format is versatile, easy to read and edit manually, and is widely supported by programming languages, libraries, and GIS platforms.

- **Shapefile**

  A shapefile[2] is a format that stores nontopological geometry and attribute information of spatial features. By omitting topology (relationships between spatial features), the format is kept simple and gives the advantage of faster rendering and easier editing.

  A shapefile is not just a single file, but a set of several files that work together to store the data. A complete shapefile must consist of at least three mandatory files: `.shp` (geometry), `.shx` (shape index), and `.dbf` (attribute data). All files must be in one folder and have the same name and the same order of records [2].

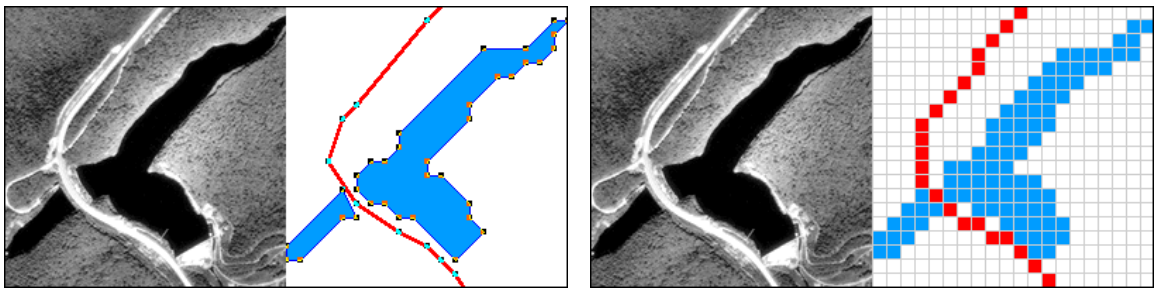

Figure 2.1: Comparison of vector (left) and raster (right) representations of a reservoir and highway. Adapted from [7].

[1]RFC 7946: https://datatracker.ietf.org/doc/html/rfc7946, accessed on 21 December 2024.
[2]Shapefile: https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf, accessed on 21 December 2024.

Raster Data

Raster data is stored in a grid-like matrix where each cell is identical in shape and size and contains a value. The amount of land a cell represents is known as spatial resolution [12].

Raster format is ideal for representing continuous data, such as elevation or temperature, where the change in adjacent values is gradual. Using a grid of cells, each with its own value, creates a smooth transition. The raster data structure is efficient for operations over continuous data, such as calculating distances or averages. The data structures are simple and easy to work with, but the graphical output depends on the grid size, limiting the precision of the represented features. Higher spatial resolution provides more precise details but can significantly increase the storage needed, making raster not particularly scalable but rather suitable for large-scale images [4]. As shown in Figure 2.1, the features are not particularly accurate when pictured in the raster format at this scale. Raster data can be stored in various formats.

- **GeoTIFF** is one of the most widely supported in GIS software, including ArcGIS, QGIS and others. It is an extension of the TIFF (Tagged Image File Format) format that includes metadata to connect the image with the geographic information, allowing it to be accurately located on Earth's surface within a geographic coordinate system. A GeoTIFF supports multiple data types and can store multiple data bands, each representing a different layer of information. It also supports many compression methods, making it useful for working with large datasets [5].
- Raster formats like **BMP**, **PNG** and **JPEG** are sometimes used for aerial and satellite imagery, but they typically require an accompanying world file for georeferencing. A world file is a separate, plaintext data file that specifies the locations and transformations that allow the image to be projected into a standard coordinate system. Because they lack the embedded spatial metadata, these formats are not commonly used in complex GIS [3].

2.1.2 Coordinate Systems

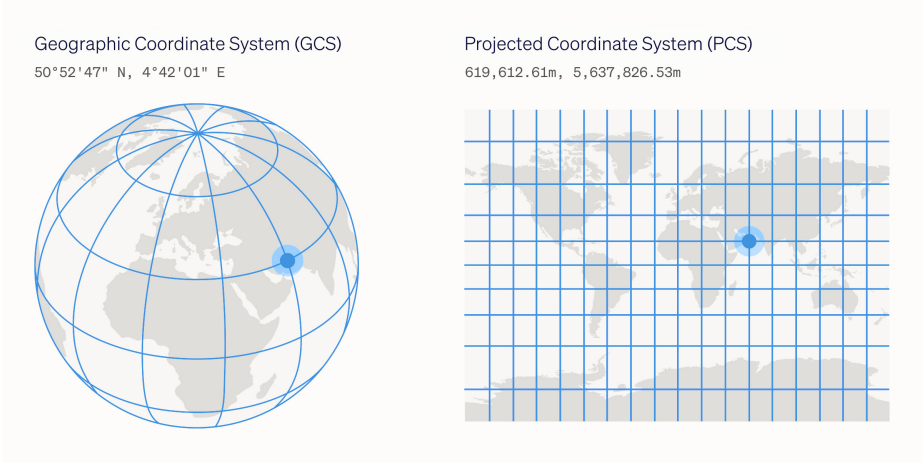

Figure 2.2: A location represented in geographic and projected coordinate systems. Adapted from [9].

Geographic data relies on coordinate systems and projections to represent locations on Earth accurately. These systems allow spatial data to be standardized, enabling analysis, integration, and visualization in Geographic Information Systems. A coordinate system defines how locations are identified on the Earth's surface using numerical values, such as latitude and longitude. There are two main types, as shown in Figure 2.2 [9].

- **Geographic Coordinate Systems** use a spherical model of the Earth and are based on angles measured in degrees. The most commonly used is WGS84 (World Geodetic System 1984), which is the standard for GPS and many GIS applications.
- **Projected Coordinate Systems** convert the Earth's curved surface into a flat, two-dimensional map using mathematical formulas. They are typically measured in linear units, such as meters, and are ideal for mapping smaller areas or projects that require a high level of precision.

Using datasets with different coordinate systems can lead to misalignment or errors in analysis. To resolve this, all datasets need to be projected into one common coordinate system.

2.1.3 Geographic Information Systems

Geographic Information Systems (GIS) are software systems for capturing, storing, managing and displaying spatial data. GIS provide a platform for integrating different types of information, regardless of their source or format, by overlaying them on a single map. Utilizing location as the main reference point, they connect these unrelated datasets to uncover patterns, relationships, and insights. Geographically referencing data gives it a whole new dimension and opens up new opportunities for analysis [14].

2.1.4 Traffic Data

Because road traffic is a complex system, it offers many aspects to measure and analyze. This includes vehicle counts, speeds, travel times, congestion levels, patterns of movement, or traffic accidents. While geographic data represent permanent or semi-permanent features, traffic data is highly dynamic, reflecting constant changes over time. Generally, the different aspects can be categorized either as event-based data or as time series data [1].

Event-based traffic data, such as traffic accidents or road closures, provide information about specific incidents. Besides the geographic location, traffic datasets contain additional information like the time of the event, causes, severity, etc. This combination of spatial data and other attributes makes formats like GeoJSON suitable for their storage.

Time-series traffic data, which consists of a sequence of observations recorded at specified times [22], tracks constant changes over time and requires frequent updates. From a storage perspective, this poses additional challenges compared to geographic data, which is mainly static, but also compared to event-based traffic data, as far more datapoint need to be stored.

Time-series traffic data requires systems that can handle high-velocity information streams, with fast ingestion, time-based indexing, and fast data retrieval for real-time applications like congestion monitoring. Regarding data retention, traffic data often uses a tiered approach, where recent data is stored in high-performance systems for real-time use, while older data is archived for historical analysis [8]. Specialized databases such as InfluxDB or TimescaleDB are commonly used for this purpose.

2.2 Traffic Data Collection

For data to be statistically significant, it must be collected in the real world. This can prove challenging for traffic data, especially as real-life driver behaviours and road conditions are complex and dynamic, making them difficult to standardise and turn into a dataset. To address these challenges effectively, it is crucial to follow a structured process that ensures the data collected is both accurate and meaningful. Each step builds on the previous one to ensure meaningful results [14].

1. **Understanding the Real World**

   Before the data collection process is started, it is necessary to identify the key elements and relationships within a transportation system that are critical for analysis. This will ensure the dataset is relevant, comprehensive, and capable of addressing specific questions or challenges. Selecting the wrong attributes or overlooking important factors can skew the results, leading to incorrect conclusions.

2. **Making Representation Choices**

   Decisions about how to represent data, such as selecting the spatial scale, level of aggregation, and geometric representation (points, lines, or areas), are critical. These choices shape how well the dataset reflects the real-world transportation system and supports specific analytical objectives.

3. **Measuring Attributes and Spatial Relationships**

   The process of capturing data involves measuring key attributes, such as traffic volume and speed, and spatial relationships like proximity or connectivity. These measurements must be accurate and precise to ensure the dataset's reliability for analysis.

2.2.1 Traffic Data Collection Methods

Both traditional and modern methods can be used to collect meaningful traffic data. Depending on the intended use, merging data from different sources and methods may be necessary to ensure the traffic conditions are represented with utmost accuracy [13].

Manual observation, where data is collected on-site by people is the oldest and most straightforward method in practice today. While this method is simple, it can be very labour-intensive and prone to human errors.

Conducting **surveys** is another traditional method that collects data from people. It is helpful for collecting data that automated processes cannot effectively capture, such as vehicle occupancy, driver behaviours or temporary road hazards. This method is not the most reliable and it is not well suited for collecting all sorts of data as respondents may be biased and sampling errors can occur. Surveys also can't be too detailed, as too many questions may affect the willingness of participants to respond or lead them to provide less accurate or incomplete answers [25]. This means it may not be possible to collect all the data necessary solely through surveys. Additionally, obtaining standardized responses through this method can be challenging, making the data difficult and time-consuming to work with.

The drawbacks of the traditional methods, alongside the advancing technology, led to the development of new, innovative strategies for data collection. A popular, widely used method for collecting traffic data is using **inductive loops**. It has been a reliable tool for data collection since its introduction in 1960s. An inductive loop sensor is a wire loop

built into the road that generates a magnetic field when powered by a detector. The detector monitors the loop's frequency, which changes when a vehicle passes over due to the interaction with the vehicle's metallic components. Single-loop sensors detect and count vehicles, while dual-loop sensors (two loops placed sequentially in a lane) can also measure a vehicle's length and speed [15]. While installing inductive loops requires high initial investment and can temporarily disrupt traffic flow, the technology demands minimal maintenance and can collect data over long periods of time thanks to its durability.

LiDAR (Light Detection and Ranging) is a high-resolution sensing technology widely used in traffic data collection for its ability to create precise 3D models of the environment. Unlike cameras, LiDAR is not affected by weather, time of day, or lighting conditions, ensuring reliable performance in diverse environments. However, its effectiveness can be limited by occlusion, where desired objects are blocked by others in the sensor's line of sight. To overcome this, multiple LiDAR sensors are often used in combination to provide full coverage. Despite its advantages, advanced LiDAR systems with multiple lasers offering superior detail and accuracy, are expensive, making cost a consideration for large-scale deployments.

While using these technologies cuts labour costs they can be costly to upkeep. Additionally, it can be restricting that they are limited by their stationary nature. This has led to increasing interest in solutions that can operate beyond static locations.

Global Positioning System (further referred to as GPS) technologies offer precise location tracking and a dynamic look into vehicle movements. They are widely used for monitoring traffic flow, analyzing travel behaviour, and enhancing transportation planning. However, tracking individual vehicles raises privacy concerns, requiring the data to be strictly anonymous and secure. Another potential drawback is signal obstruction. In [11], researchers tested data collection using GPS technology in forestry settings and found that rain and increased forest density adversely affect GPS performance. Similarly, in urban areas, tall buildings can disrupt signals due to a phenomenon known as urban canyoning.

Although all these methods have limitations and cannot guarantee absolute accuracy, they are the most reliable approaches for data collection available and are used widely. However, new technologies are emerging that will hopefully allow for even better, more accurate data collection and might change the way traffic data is collected in the future.

2.3 Data Sources and Uses

For a meaningful analysis, it is essential to work with quality data. If direct data collection is impossible due to a lack of equipment or resources, it is necessary to obtain the data from external sources. However, the data is not always publicly available and is often only used internally by these institutions, subject to paid access, or available only under strict licensing terms.

2.3.1 Official Sources

Official traffic data comes from sources such as government agencies, transportation departments or other formal institutions. They collect the data using standardized methods, ensuring it is a reliable, well-structured source of information.

Police reports can provide detailed records of road accidents and traffic violations. Following regulatory guidelines, these reports are gathered with high accuracy and comprehensive information about each occurrence, resulting in a feature-rich dataset that may be

used for a variety of analytical purposes. However, they are limited to incidents involving law enforcement. While all serious crashes should be accounted for, less severe incidents can go undocumented.

Traffic sensors, such as inductive loops, radars and cameras, offer continuous data on vehicle flow, speed and congestion. These sensors are typically managed by government agencies and transportation authorities. This technology provides highly reliable, objective measurements, but it does not provide complete spatial coverage, as sensors are usually only deployed in a limited number of locations. Installing sensors in all locations would be impractical and costly, so authorities select their placement carefully to maximize effectiveness. This means they are suitable for monitoring the traffic situation in places already considered critical but not so suitable for identifying such places.

Public transit vehicles are often equipped with GPS to monitor transit operations, allowing them to collect data on speed, locations or delays. These data are limited to public transit vehicles, and although they make up a sizeable portion of urban traffic, they do not provide a complete picture of overall traffic conditions.

Cities and organizations sometimes deploy specially equipped vehicles, known as probe vehicles, to gather data. The most well-known example is probably the Google car, which captures highly detailed maps of its surroundings. More locally, the AdMaS centre (BUT) operates a specialized vehicle fitted with advanced sensors that can map road infrastructure with remarkable precision, down to 5mm. Using laser technology, it can scan its surroundings in real-time, providing detailed scans even capturing potholes and surface irregularities with high accuracy [27].

2.3.2 Unofficial Sources

Unofficial sources are data collected by private entities or crowdsourced from individuals. These sources are dynamic and often provide real-time information, although their accuracy can vary.

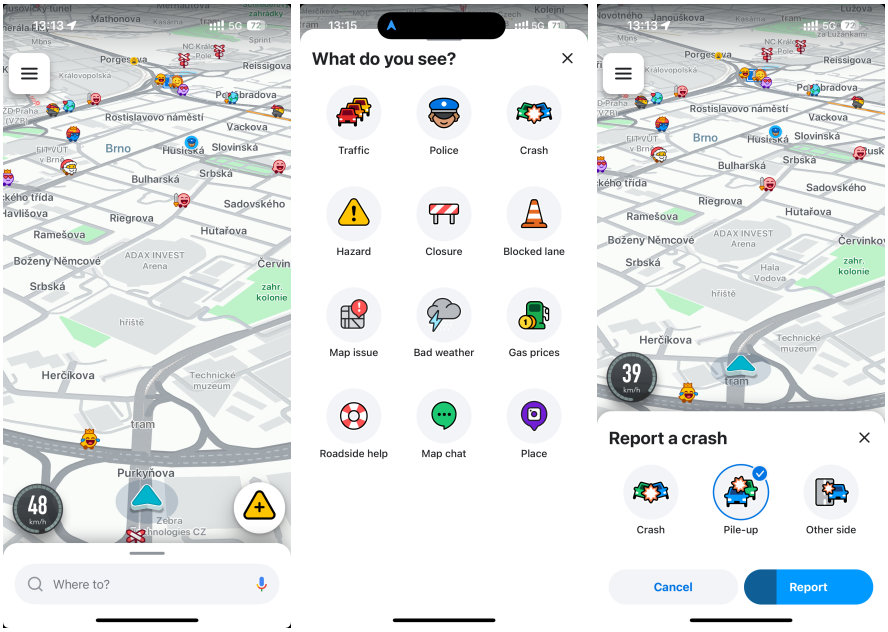

Figure 2.3: The process of reporting an accident in the Waze application.

Crowdsourced platforms like Google Maps or Waze gather data from user reports, offering a dynamic insight into the traffic conditions. Waze shares its data through the Waze for Cities program[3]. Although the program is not publicly accessible, data for this project was obtained through a collaboration between Faculty of Information Technology at Brno University of Technology and the Waze for Cities initiative.

However, the data can be inconsistent and lacking in areas with few Waze users. A study of traffic accidents in Brazil [21] found that users of Waze have only reported 7% of accidents reported by official sources. Accidents reported only by official sources were concentrated in the central region, while those recorded by Waze are primarily on major roads all over the city.

Waze relies on its users to report road incidents, so the reporting tool needs to be quick and easy to use, even while driving. While this enables real-time reporting, detailed accident information is left out to keep the process simple and intuitive. Instead, users can report an accident in just three quick steps, as illustrated in Figure 2.3.

2.3.3 Uses of Traffic Data

The usefulness of traffic data depends heavily on its quality, structure, and coverage. Ideally, datasets should include both quantitative and contextual information, such as vehicle counts, speeds, time, location, road types, and environmental conditions. The broader and more accurate the data, the more meaningful its application becomes. Traffic data serves a wide range of purposes, from reactive safety measures to long-term strategic planning.

Existing Systems for Traffic Accident Data Analysis

Traffic accident data analysis plays a crucial role in understanding road safety trends, identifying hazardous areas, and supporting decision-making processes for traffic management.

Several systems in the Czech Republic utilize accident data to provide analytical insights and visualizations, as it is publicly available through the Police of the Czech Republic's official website.

- **Nehody**[4]

  The system uses Microsoft Power BI to generate a variety of visualisations utilising the traffic accident dataset provided by the Police of the Czech Republic, updated on a monthly basis. The application offers two views. The homepage is a dashboard of charts with the most important attributes plotted. This view allows for some filtering but not all attributes can be combined to create a more complex filter. The time can only be filtered by years, not specific dates. In the second view, accidents are visualised on an interactive map view with multiple base maps to choose from, such as OpenStreetMap, Google map, Mapy.cz and more. The user can set the beginning and end date of the time period they wish to see, as well as pick the region in the suggestion box or choose it manually by drawing a polygon on the map. In this view, users can also filter the data by defining up to 64 types of conditions based on the traffic accident protocol. Defining one condition multiple times with different parameters results in combining the conditions with a logical OR operator. The system allows users to export the filtered data in the form of graphs, allowing them

[3] https://www.waze.com/cs/wazeforcities

[4] nehody: https://nehody.cdv.cz/, accessed on 10 November 2024.

to choose the graph type and time unit used. When viewing charts, it is only possible to combine a small set of different attributes, other attributes cant be combined in the filter. Figure 2.4 presents a representative screenshot of the application, intended as an illustrative example.

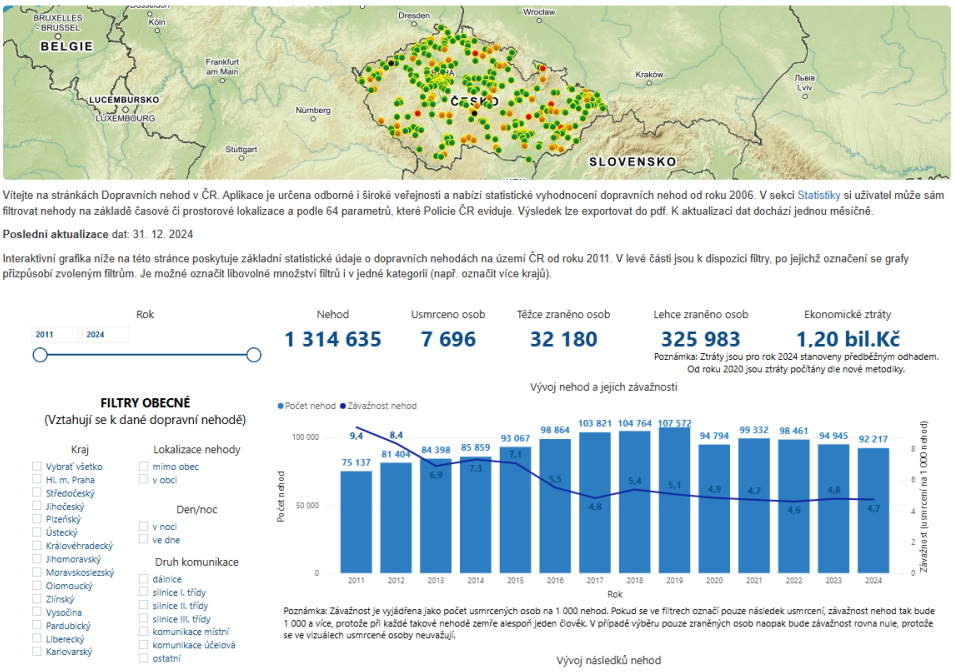

Figure 2.4: The web application **Nehody** provides a map and dashboard with filtering options.

- **Kde Bouráme**[5]

  Kde Bouráme is an advanced web-based application developed by the Transport Research Centre to identify and visualise high-risk accident locations across the Czech Republic. It raises road-safety awareness and to supports professionals engaged in traffic-safety research and infrastructure planning. The core functionality is based on the KDE,+ method, which processes multi-year police accident records and flags spots where the observed crash frequency exceeds statistical expectations.

  The platform offers an interactive map with side control panels for filtering and analytics. Users can zoom from a nationwide overview down to individual road segments. Filters allow to focus on accidents with specific injury severities, collision types, weather conditions, or road classes. Temporal sliders let analysts select year ranges and thematic filters allow viewing only particular accident types, weather conditions, or road classes. Detected hotspots appear as colour-graded polygons; clicking on one reveals detailed statistics and links to individual police reports. Beyond the spatial view, Kde Bouráme provides an option to export shapefiles and CSV tables for deeper evaluation, making it useful for expert users.

[5]kdebourame: https://www.kdebourame.cz/cz/, accessed on 9 November 2024.

Although powerful, the platform currently refreshes its analyses on an annual basis that can limit responsiveness to very recent changes. Figure 2.5 shows a screenshot of the interface.

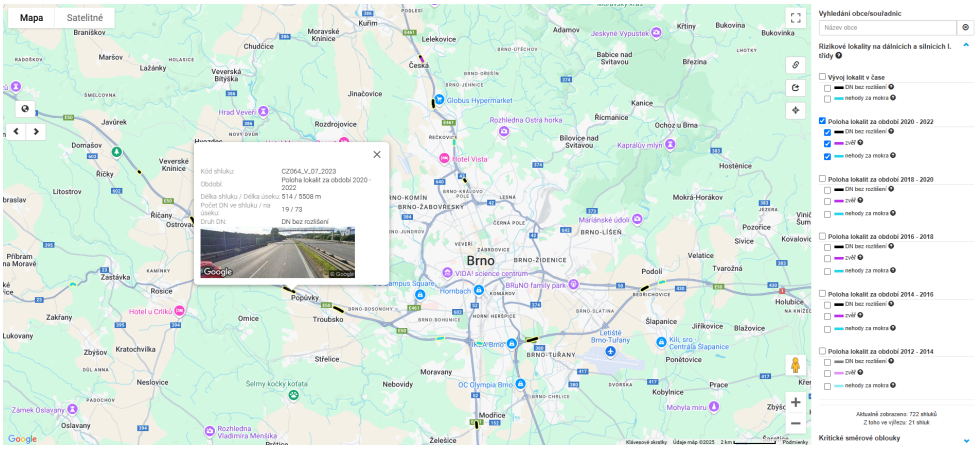

Figure 2.5: The **Kdebourame** web application shows an accident heatmap with basic filters.

- **TrafficAcc**[6]

  TrafficAcc is a demo version of an application initially developed for the police of the Czech Republic, only featuring selected data that is publicly available. The application is intended to be a tool for traffic experts, who are able to interpret the data and use it alongside other resources. The application covers the entire road network of the Czech Republic. Notably, it includes intersections, which are often overlooked for their complexity, despite accounting for a significant portion of traffic accidents. It also differentiates between accidents according to their severity, making identifying the dangerous areas more precise.

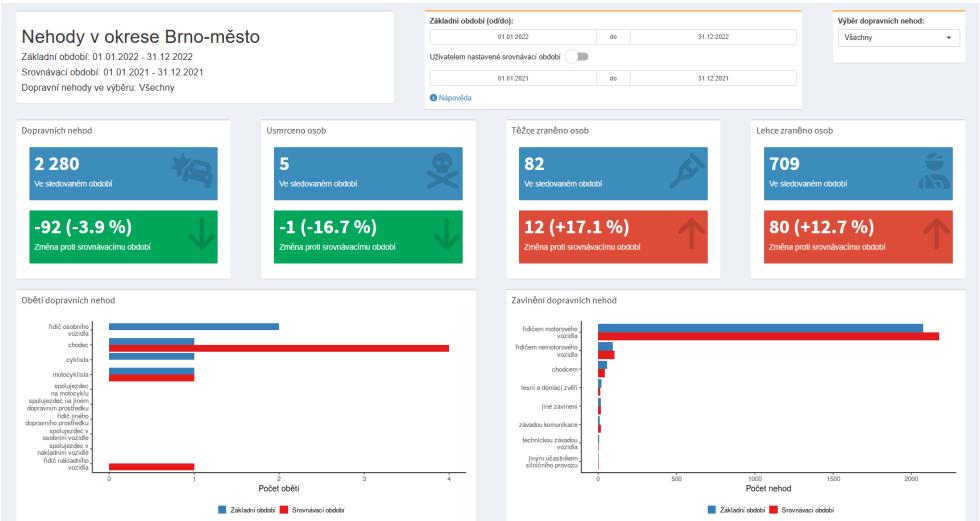

Figure 2.6: The **Trafficacc** web application provides a dashboard and a map view.

[6]trafficacc: https://trafficacc.econ.muni.cz/, accessed on 9 November 2024.

Chapter 3

Traffic Data Analysis and Visualization

Traffic data analysis and visualization are essential for understanding traffic patterns, identifying problematic areas, and making informed decisions for urban planning and traffic management. This chapter explores the methodologies used to process and integrate geographic data, the challenges encountered during analysis, and the techniques employed for effective visualization.

3.1 Geographic Data Preprocessing and Integration

Data preprocessing is a crucial stage in data mining, designed to prepare raw data for effective analysis and integration. This phase addresses common challenges, such as incompleteness, inconsistency, and redundancy, ensuring that data is high quality and ready for analysis. Raw data often contains noise, missing values, and inconsistencies that can hinder accurate analysis. Proper preprocessing not only enhances data quality but also improves the efficiency and reliability of analytical methods.

3.1.1 Data Preprocessing Workflow

The data collected from various sources undergoes a processing workflow commonly referred to as ETL (Extraction, Transformation, Loading) [16]:

- **Extraction:** Data is retrieved from multiple sources, such as police records, Waze events, and geographic databases.
- **Transformation:** The extracted data is cleaned, normalised, and formatted to resolve inconsistencies and ensure compatibility and usability for analysis. This process may include filling in missing values, converting to uniform formats and aggregating or restructuring data. Visual tools help verify data quality and identify patterns or anomalies.
- **Loading:** The transformed data is then stored in system or database for further analysis and visualisation. This ensures that the data can be accessed efficiently and consistently.

3.1.2 Data Transformation Techniques

According to [26], there are three characteristics that suggest a low quality of data; incompleteness, noisiness and inconsistency. These can be caused by various reasons like equipment malfunctions, human error or merging of multiple datasets. They are, however, to some extent, present in most raw datasets. To address these issues, the data transformation phase employs numerous techniques aimed at improving data quality and reliability.

Data cleaning focuses on correcting inaccuracies, filling in missing values through statistical imputation, and smoothing out random errors or outliers. Records with insufficient data are often excluded to maintain dataset integrity.

Data integration resolves discrepancies between different sources, unifying schemas, naming conventions, and measurement units to create a coherent dataset.

In the **transformation** phase, categorical data is encoded numerically, values are normalized to a consistent scale, and granular records are aggregated into summary metrics for efficient analysis.

Finally, **data reduction** minimizes dataset size without losing critical information by selecting relevant attributes, applying dimensionality reduction methods, sampling representative subsets, or discretizing continuous values into intervals [26].

3.2 Visualisation

Data visualisation plays a crucial role in the analysis and interpretation of traffic data. For datasets containing large volumes of complex, multidimensional information, as traffic datasets often do, visualisation serves as a bridge between raw data and human understanding. Translating data into visual formats allows for more intuitive exploration, helping to uncover patterns, detect anomalies and identify trends or issues that would be difficult to detect from raw data alone. Visualizations support effective data interpretation for both expert and non-expert audiences.

While aesthetic appeal can enhance clarity and user engagement, the main purpose of data visualisation is to accurately convey information to the user and it must not come at the cost of accuracy or misrepresentation. A well-designed graphic should improve understanding, not distort or distract from the underlying data [24].

Core design principles for effective visualisations are clarity, precision and efficiency [23]. Clarity ensures that the visual message is easily understood without confusion, precision guarantees that the data is represented truthfully and without distortion and efficiency enables the viewer to grasp the intended information quickly, without unnecessary visual noise or distractions. Colours, shapes and sizes should be applied meaningfully to highlight relevant differences but also avoid unnecessary complexity.

3.2.1 Graphs and Charts

Graphs and charts are essential tools for summarising and comparing data across numerous dimensions. The choice of chart type must reflect the structure of the data and intention of the visualization. Some visual formats like 3D charts or overly segmented pie charts can be visually appealing but often distort proportions or conceal insights, and should be used with care or avoided altogether.

- **Bar charts** – Useful for comparing categorical data, such as the number of accidents by region or accident type.

- **Line charts** – Ideal for showing trends over time, such as accident counts across months or years.
- **Histograms** – Effective for displaying distributions, like the number of accidents across different days of the week.
- **Pie charts** – Best used sparingly for showing proportions, should be limited to a small number of categories for clarity.
- **Heatmaps** – Suitable for visualising intensity or density over space or time.
- **Scatter plots** – Useful for exploring correlations or clustering between two numerical variables.

Coordinate systems and scales must also be selected appropriately. When the same units are plotted on both axes, such as distance versus distance, maintaining equal scale is essential to preserve spatial accuracy. When plotting different types of variables like time and count, proportions of the axes can be adjusted to create different visual emphases. For instance, extending the time axis in a line graph can highlight gradual changes, whereas compressing it may draw attention to short-term spikes or anomalies [24].

If any interactive features are applied, they should enhance user experience by supporting filtering, zooming, or highlighting relevant data points. These tools are most effective when they help guide users toward meaningful insights rather than overwhelming them with options.

Effective use of charts enables users to spot trends, detect anomalies, and compare values at a glance. To maintain clarity, visual clutter should be avoided, and labels, axis ticks, and legends should be used consistently and thoughtfully.

3.2.2 Maps

Maps are a fundamental tool for visualising the spatial aspects of traffic data. Visual layers can greatly improve the readability of geographic data, which would otherwise be difficult to interpret in its original format. They allow viewers to explore where incidents occur, observe spatial distributions, and identify areas of frequent congestion or risk. Interactive map features such as zooming and clustering improve usability and help users engage with data at different levels of detail.

3.2.3 Dashboards

Dashboards are visual displays of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the data can be monitored at a glance [10]. Rather than presenting all available data, dashboards focus on the most critical metrics and trends, enabling users to quickly assess the current state of situations or systems.

By integrating multiple charts and metrics into a cohesive layout, dashboards support situational awareness and enable informed decision-making. When well-designed, they reduce cognitive load, prioritize clarity, and allow users to interact with the data without being overwhelmed.

3.2.4 Target Users

Understanding the target audience is vital in visualisation design as the effectiveness of a visual representation heavily depends on the viewer's level of expertise, expectations, and goals. Experts may require detailed, data-rich visuals, while general audiences benefit from simpler, more intuitive representations. Visualisations aimed at this group should prioritize clarity, avoid being overly detailed, and highlight only the most relevant insights. Visualisations should therefore be planned with the user base in mind and adapted accordingly.

Chapter 4

Analysis

This chapter outlines the datasets used in the project and defines key requirements for extending the existing application. The following sections examine the strengths and limitations of each dataset, describe the challenges of integrating them, and identify how this data can support traffic safety improvements. Finally, the chapter defines the application's functional and non-functional requirements based on the needs of its primary user groups.

4.1 Datasets Used

In this bachelor's thesis, two datasets from two different sources were used and combined to provide a more comprehensive view of traffic incidents. The existing applications for traffic accident analysis in the Czech Republic mentioned in 2.3.3 rely exclusively on police records, which may be incomplete due to under-reporting. By integrating both official police records and user-reported traffic events from Waze, this project aims to address these gaps and identify patterns that may otherwise go unnoticed.

4.1.1 Police Traffic Accidents Dataset

The police dataset contains records of traffic accidents that have occurred within the city of Brno since 2016. The data is sourced from the Police of the Czech Republic, making it an official record of reported accidents. Anonymised data is published on their website [1] and updated on a monthly basis, providing a regularly maintained source of reliable traffic incident information. The long time span and frequent updates of the dataset make it useful for long-term trend analysis.

The dataset includes a wide range of attributes related to the location, cause, severity, and circumstances of accidents, making it particularly useful for in-depth analysis. Selected attributes can be seen in Table 4.1, with a full list of attributes available in Appendix A.1.

However, not all accidents are officially reported to the police. According to local regulations[2], traffic accidents must be reported if there are injuries, fatalities, damage to public property, or significant private property damage exceeding a defined threshold (CZK 100,000 as of 2024). However, drivers often handle the situation privately for minor collisions to avoid police involvement or potential increases in insurance premiums. This

[1] Traffic Accident Data on Offical Websites of Police of the Czech Republic: https://nehody.policie.gov.cz/

[2] Investigation of Traffic Accidents: https://www.policie.cz/clanek/zverejnene-informace-2024-setreni-silnicnich-dopravnich-nehod.aspx, accessed on 10 December 2024.

Attribute	Description	Data Type
p1	Accident ID	Integer
p2a	Date of accident	Date
p2b	Time of accident	String (optional)
p9	Accident consequences	Integer
p13a	Number of fatalities	Integer
p13b	Number of severely injured persons	Integer
x	Longitude (geographical)	Float
y	Latitude (geographical)	Float
...	...	...

Table 4.1: Selected key attributes from the police accident dataset.

underreporting may create gaps in official traffic accident data, which can be partially filled in by the second dataset used in the thesis.

4.1.2 Waze Events Dataset

The first record in this dataset is dated 25.4.2024. The data is sourced from Waze Mobile Ltd. (further referred to as Waze), which collects information from users of its navigation app. Waze shares this data through the Waze for Cities program. The coordinate system used for this data is the Geographic Coordinate System WGS 84, which allows for precise geospatial mapping and integration with other datasets.

Events are recorded in real time and updated every two minutes, providing near real-time information about road conditions. Still, the inability to verify the correctness of the user-submitted data could reduce the reliability of this dataset. Waze allows for a single event to be reported multiple times by different users. Multiple reports of the same event can make it more credible but also cause redundancy within the dataset. It is important to note that most Waze users who have already seen that an accident was reported in their path will likely take a different route or avoid reporting the same accident again [21].

Selected attributes of the dataset can be seen in Table 4.2, with full list of attributes available in Appendix A.2.

Attribute	Description	Data Type
uuid	Unique report ID	String
location	Geographical location	String
published_at	Report publication time	DateTime
reliability	Reliability score (0–10)	Integer
type	Report type	String
subtype	Detailed report type	String
...	...	...

Table 4.2: Selected attributes from the Waze dataset.

Like the police dataset, the Waze dataset also suffers from underreporting. Accidents occurring at times or in areas with minimal traffic impact or low Waze user activity are unlikely to be reported. Because of this, the datasets should be complementary, helping to fill in the holes [21].

4.1.3 Identified Problems

No real-world dataset is perfectly clean or analysis-ready. The ones used in this thesis are no exception. Several challenges have to be addressed to ensure the data could be meaningfully analyzed and visualized. The most significant issues are described below.

Encoding and Formats

Many attributes are stored are stored using encoded formats or in formats not suitable for further work. Police attributes are stored under names like `p1`, `p2a`, and `p6`, and values are represented by numerical codes. While this encoding is efficient for storage, it is not suitable for direct analysis or visualisation and working with these encoded values when programming adds an extra layer of complexity.

Location attributes in both police and Waze datasets are in Well-Known Binary format for efficient querying in the PostgerSQL database, but this format is not ideal for further work with the data. Additionally, the police dataset uses the Krovak East North coordinate system (EPSG:5514), a projection commonly used in the Czech Republic and Slovakia. Although this project focuses on data from the Czech Republic, this coordinate system can lead to compatibility issues with modern web mapping libraries such as Leaflet. Furthermore, it does not align with the coordinate system used by Waze, which relies on the globally recognized WGS84 system (EPSG:4326).

Data Redundancy

The Waze dataset often contains duplicate reports of the same incident submitted by different users. This redundancy introduces noise into the data and can distort analyses if not handled properly. There is no straightforward solution such as a unique accident ID, so the problem must be addressed by estimating spatial and temporal proximity between reports to identify and merge duplicates.

Matching the Datasets

There is no shared identifier or direct attribute that links Waze alerts with police accident records. Therefore, a custom matching method based on spatial and temporal proximity must be used to estimate relationships between records. However, accidents may be reported with varying delays or even location inaccuracies. Because of this, some true matches may be missed, and some unrelated events may be mistakenly linked. No matching approach based on indirect comparison can guarantee complete accuracy, but the method must be designed to be as accurate as possible, with well-defined distance and time thresholds.

Missing Timestamps

Approximately 30% of the police accident dataset lacks a timestamp. Temporal information is one of the most critical attributes for both data matching and analysis. Without an accurate record of when an incident occurred, it becomes significantly more difficult to correlate police reports with crowdsourced Waze alerts, as matching will rely heavily on proximity in both space and time.

Moreover, time plays a fundamental role in the visualization and interpretation of traffic accident data. Temporal patterns, such as peak accident hours, daily trends, or seasonal

variations, provide valuable insights into the behavior of traffic and the underlying causes of accidents. Missing timestamps limit the ability to identify temporal patterns and trends and reduce the overall effectiveness of data-driven safety analysis. Without reliable timestamps, a significant portion of the data must either be excluded or handled with approximations, which can introduce bias or reduce the accuracy of conclusions drawn from the data.

In addition, even for accident reports that do include a timestamp, the temporal information remains incomplete. The dataset does not specify whether the recorded time reflects the actual occurrence of the accident or the time it was reported. Furthermore, many timestamps appear rounded to the nearest hour (e.g., 9:00, 11:00), suggesting they may not capture the precise moment of the incident. In contrast, Waze alerts are typically associated with exact timestamps reflecting real-time user reports. Additionally, the police dataset provides no indication of how long an accident impacted the road, for example, how long it took for traffic to clear or for the scene to be resolved, which is critical for understanding the severity and real-world consequences of incidents. These gaps further complicate the process of merging the datasets and undermine the precision of any temporal analysis or event matching.

4.2 Possibilities of Using Accident Data for Urban Traffic Improvement

By visualizing police reported accident locations together with Waze reports, users and traffic experts can quickly identify spatial patterns, such as accident hotspots or high-risk road segments. Cross-referencing both sources makes it possible to verify incidents, assess their visibility to the public, and estimate the level of disruption or danger they pose. This integration also allows for retrospective estimation of missing information—such as accident times not recorded in official reports—based on related Waze alerts. Information about the type of accident, surface and weather conditions, and road types allows for basic analysis of contributing factors to incidents. For example, repeated occurrences of accidents at a specific intersection, combined with Waze-reported hazards like obstacles or road surface problems, may indicate areas where infrastructure improvements or better signage could be considered.

The use of charts based on filtered accident characteristics (e.g., accident types, time of day, weather conditions) helps to reveal general trends and seasonal variations, which can inform planning of safety measures such as increased monitoring during high-risk periods or proposing changes in road infrastructure and traffic signs.

4.3 Potential Users

Since this thesis builds upon an existing application, it is designed to meet the needs of a similar user base. Two primary user groups have been identified for the enhanced application, each with different needs and levels of expertise:

The first group is the general public, who is assumed not to possess any background in traffic management or data analysis. These users are likely interested in easily accessible statistics and visual insights regarding traffic accidents and overall road safety. The extension should provide these users with intuitive, interactive features that allow them to explore accident data visually, without requiring technical expertise.

The second user group consists of professionals such as urban planners, policymakers, and traffic engineers. For these experts, access to detailed and accurate accident data can greatly assist in decision-making, infrastructure planning, and policy development.

4.4 User Requirements

The requirements for the extension of the existing application are informed by the needs of both user groups as well as the functionality already present in the base application. They can be divided into functional and non-functional requirements.

4.4.1 Functional Requirements

- **Integration of Accident Data:** The extension should enable the import and processing of traffic accident data, ensuring it is handled alongside existing datasets.
- **Accurate and Relevant Insights:** The system should provide the most accurate and up-to-date accident data available, and highlight particularly significant events, such as those with high severity.
- **Map-Based Accident Visualisation:** Accident locations should be visualised in the application on an interactive map. This allows users to explore specific incidents in detail and understand their geographic context within the transportation network.
- **Heatmap of Problematic Areas:** In addition to individual accident markers, the extension provide a heatmap layer to help identify and highlight areas with high concentrations of accidents.
- **Statistical Dashboards and Charts:** The extension should provide additional dashboards and charts focused on traffic accident statistics.
- **Temporal Filtering:** The application should allow users to filter accident data based on time-related parameters such as specific dates or days of the week. This enables users to explore temporal trends, analyse seasonal variations or focus on a particular timeframe relevant to their analysis.
- **Attribute-Based Filtering:** Users should be able to apply filters based on various accident attributes such as severity, weather conditions, or location characteristics. An advanced filter builder should be implemented to support complex queries and provide customised data views.

4.4.2 Non-Functional Requirements

- **Consistent User Experience:** The user interface and interaction style of the extension should remain consistent with the existing application to ensure a seamless experience for users and preserve the application's overall coherence.
- **Performance and Responsiveness:** The integration of accident data and new analytical features should not negatively impact the responsiveness or performance of the application.
- **Maintainability and Scalability:** The extension should be designed with maintainability and future scalability in mind, allowing for potential further enhancements.

Chapter 5

Proposed Solution

The proposed solution builds on an existing web application focused on visualising traffic congestion data in the city of Brno. The objective of this extension is to incorporate the analysis and visualisation of traffic accident data by leveraging both official police reports and user-submitted alerts from the Waze platform. This chapter outlines the technical design, data handling, system architecture, and integration strategy used to achieve this goal.

5.1 System Architecture

The final system is composed of three main layers:

- **data layer**, which supplies raw data for further processing;
- **application layer**, which processes the raw data, performs analysis, and prepares it for the next stage;
- **presentation layer**, which displays the processed data in a user-friendly format.

This thesis focuses on the development of new **application** and **presentation** layers. The data layer, was already established and has been modified by the thesis supervisor to include the new datasets required for this project. A diagram of the final system can be seen in Figure 5.1

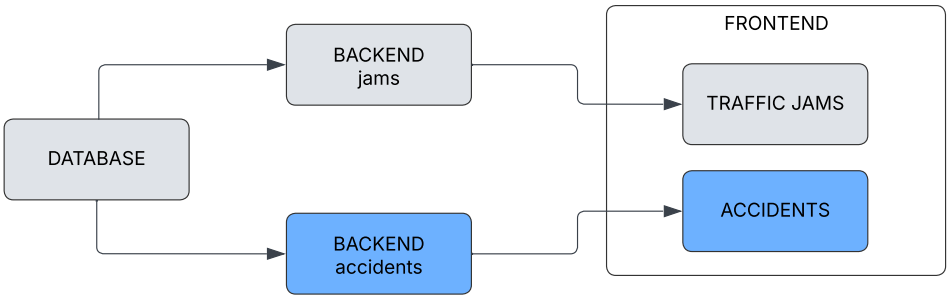

Figure 5.1: Overview of the complete system. The components developed in this thesis are shown in blue.

5.1.1 Data Layer

The data layer is responsible for providing the raw input used by the application layer for processing and analysis. The database infrastructure is provided by the thesis supervisor. It consists of several data sources used by the web application, with two primary datasets relevant to this thesis: official police accident records and crowdsourced alerts from the Waze platform. These datasets are stored in a PostgreSQL database equipped with the `PostGIS` extension for spatial data and the `TimescaleDB` extension for time-series data. Data will be retrieved from the database using SQL queries that allow filtering based on temporal parameters and attributes.

5.1.2 Application Layer

The application layer will serve as a core between raw data and a user-friendly interface. It will process raw data, transform it, perform various analyses and expose endpoints for communication with the web application using a REST interface with two main kinds of endpoints: endpoints for getting a filtered list of police-reported accidents and Waze reports, and endpoints for generating chart-specific data.

The application will be implemented in the Python programming language, selected primarily for the fact that the original system is implemented in Python and for its extensive ecosystem of data analysis libraries, such as `Pandas`, `GeoPandas`, and `GeoPy`, which provide robust functionality for handling and analyzing geographic data, streamlining the process of getting valuable insights from the data. The `FastAPI` library will be used to create the API interface, with many built-in features, such as the automatically created Swagger documentation. The documented REST API allows for integration of the backend with different applications, or even using it in a raw JSON form.

5.1.3 Presentation Layer

In line with the original work, the presentation layer will be implemented as a single-page web application using the `React.js` framework. Its primary function will be to display data obtained from the backend in an intuitive and user-friendly way. The application will enable users to visualise traffic accidents on an interactive map, apply filters based on various attributes, and constrain results to a specified time window. Additionally, it will offer a comprehensive dashboard that presents the obtained information through multiple types of charts.

5.2 Working with Datasets

The extension will work with datasets introduced in Chapter 4. The dataset of police-reported traffic accidents is described in detail in subsection 4.1.1, and the dataset of accidents reported by users of the Waze application is presented in subsection 4.1.2.

Before meaningful analysis and visualisation can be performed, the raw data must undergo several transformation steps. In this project, both the police accident dataset and the crowdsourced Waze alerts require preprocessing to ensure consistency and usability. This includes mapping coded fields to human-readable labels, retroactively calculating missing data, and filtering records based on spatial and temporal relevance. Data transformation will be essential to match and unify the two datasets, allowing accurate matching of events and enabling comprehensive traffic accident analysis.

5.2.1 Police Reports

The police accident dataset serves as the primary data source in this project, as it contains the most detailed and structured information about each traffic incident. It includes a wide range of attributes capturing various aspects of each incident, which are described in detail in Appendix A.1.

Encoded Values and Incompatible Formats

Many attributes in the police dataset are stored using encoded formats, such as names like `p1`, `p2a`, and `p6`, and values are represented by numerical codes. This encoding is not suitable for analysis and visualisation intended for users and encoded values are unnecessarily difficult to work with when programming. Each variable would have to be manually translated or referenced upon every use, which increases the risk of error and slows development. To address this, each encoded field must be translated into a human-readable format using a mapping guide provided by the dataset documentation.

The `location` attribute must be converted from the Well-Known Binary (WKB) format into standard latitude and longitude coordinates, which are more suitable for further processing and visualization. Additionally, the Krovak East North coordinate system (EPSG:5514) used in the police dataset will be converted to the widely adopted WGS84 system (EPSG:4326) to ensure compatibility with web mapping libraries such as Leaflet and consistency with the Waze dataset.

Missing Timestamps

To address the issue of missing timestamps in the police accident dataset, a fallback approach will be implemented during the matching process. The methodology of matching the datasets will rely on spatial and temporal proximity, and is further described in Subsection 5.2.3. If an accident report lacks a specific time but includes a valid date, matching will be performed based solely on geographic proximity and date. When matching Waze alerts are found, the earliest timestamp among those alerts will be assigned to the accident record as an estimated occurrence time. This allows for estimating some of the incomplete records and including them in time-based analyses.

However, if no Waze matches are found, the accident record remains without a precise timestamp and will be excluded from analyses that require exact time information. This approach helps retain valuable spatial data while minimizing bias in temporal analyses where precise timing is essential.

5.2.2 Waze Alerts

The Waze dataset serves as a complementary source of traffic incident data. Although it is less detailed and not as reliable as official records, it can help identify underreported accidents and enhance the context provided by police data. Waze alerts will also be mapped to police-reported accidents when possible, enabling enhanced analysis. Several actions have to be taken to prepare the data for analysis.

Format Conversion

The Waze alerts include a `location` attribute, which stores the geographic position of each report in Well-Known Binary format. This encoding is natively supported by PostGIS

and allows for efficient querying, but for the purposes of this thesis, the location will be converted to latitude and longitude coordinates. This conversion is necessary for direct use in web-based mapping tools and spatial matching based on geographic proximity.

Data Redundancy

Because the dataset includes user-reported accidents, it is common for a single incident to be reported by multiple users, resulting in data redundancy. To overcome this problem, data points will be clustered based on spatial and temporal proximity. A new, unique accident will then be generated for each cluster, with a timestamp of the earliest alert in the cluster and its location recalculated based on the reliability scores of the individual reports within the group, as illustrated in Figure 5.2. The data aggregation logic is shown in a diagram in Figure 5.3.

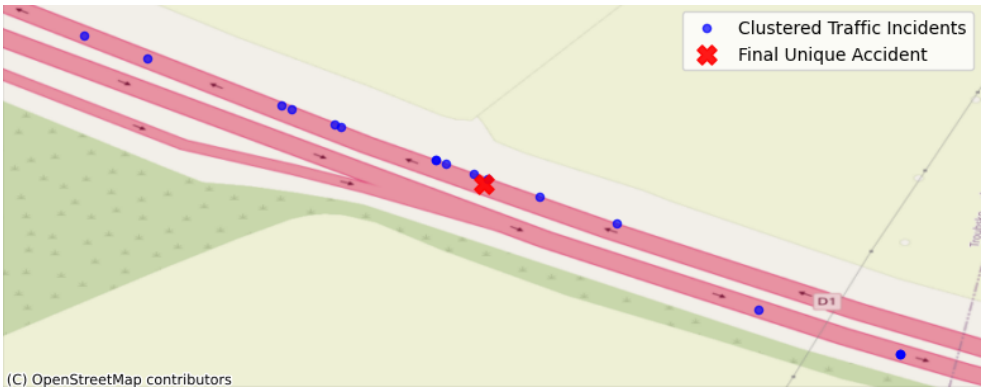

Figure 5.2: Final location of the unique incident is recaltuled from all reports in the cluster.

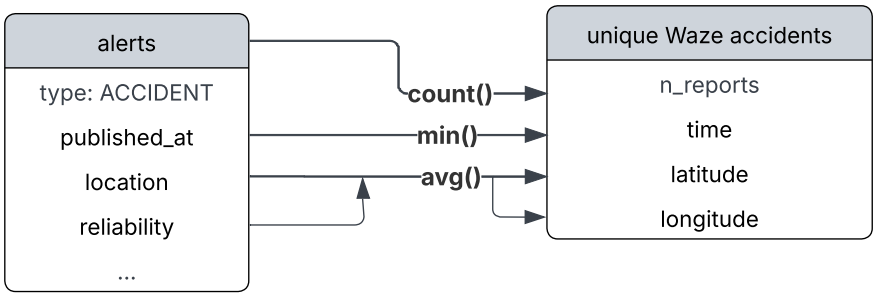

Figure 5.3: Data aggregation for recalculating unique accidents from Waze alerts using time, location and reliability score.

5.2.3 Integration of Datasets

As mentioned in the previous chapter, combining Waze alerts with police accident reports can improve understanding of each incident. Linking both sources adds context, helps verify events, and provides the user with a complete overview of accidents.

Proposed Matching Methodology

To reliably associate police accident reports with crowdsourced Waze alerts, a proximity-based matching approach is proposed. The goal is to associate incidents that occured close to each other both spatially and temporally.

Matching will be performed directly on individual Waze alerts, without prior aggregation, ensuring that the relationship between multiple alerts and a single police accident is preserved. For each police-reported accident, Waze alerts falling within a specified distance and time window will be considered candidates. Geographic coordinates will be used to assess spatial proximity, while temporal proximity will be evaluated based on the reported timestamps. A scoring mechanism will prioritize matches that are closer in distance and time, assigning each Waze alert to the most relevant police accident based on the highest match score. This approach enables a detailed analysis of how many user reports correspond to a single official record.

For police records without a timestamp, temporal proximity will be assessed based solely on the date of the record, under the assumption that it is unlikely for two distinct accidents to occur at the exact same location on the same day. Although this approach may be less precise, it significantly increases coverage, as approximately 30% of police reports lack an exact timestamp and would otherwise be excluded. Additionally, this method enables the retroactive estimation of more precise timestamps based on matched Waze reports. In the absence of a valid timestamp for a police-reported accident, the timestamp of the earliest corresponding Waze report is used for further analysis. Although accidents occurring shortly before midnight could potentially be missed, traffic volume during these hours is among the lowest of the day, according to the police and Waze datasets, as can be seen in figure 5.4, suggesting that any resulting data loss should be minimal. The detailed implementation of this methodology is presented in the subsequent chapter.

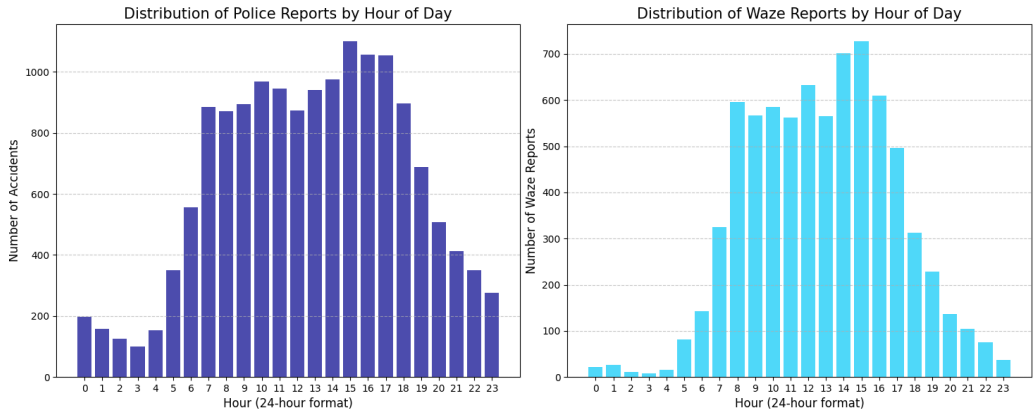

Figure 5.4: Hourly distribution of police and Waze reports, based on data available through February 22, 2025.

5.3 User Interface Design

Similarly to the initial system, the extension will consist of two main screens, a map that visualizes the locations of accidents and a dashboard, both with interactive components for filtering and exploring accident data.

5.3.1 Map

The map view will provide a geospatial representation of reported accidents. Users will be able to explore the geographic distribution of both police-reported and Waze-reported incidents, each type with a visually distinct marker. Map view will have te ability to be customized by choosing whether to display police accidents, Waze reports, or both, but both datasets are shown by default. In addition to point-based markers, a heatmap visualization will be available to highlight areas with a high concentration of incidents, making it easier to identify accident hotspots.

To avoid duplication, Waze reports will be clustered based on spatial and temporal proximity, and only a single marker will be shown for each group at a position recalculated from all reports from the cluster. Both police and Waze markers will be interactive. By clicking on the markers, users will be able to access detailed information. An early design of the map view can be seen in Figure 5.5.

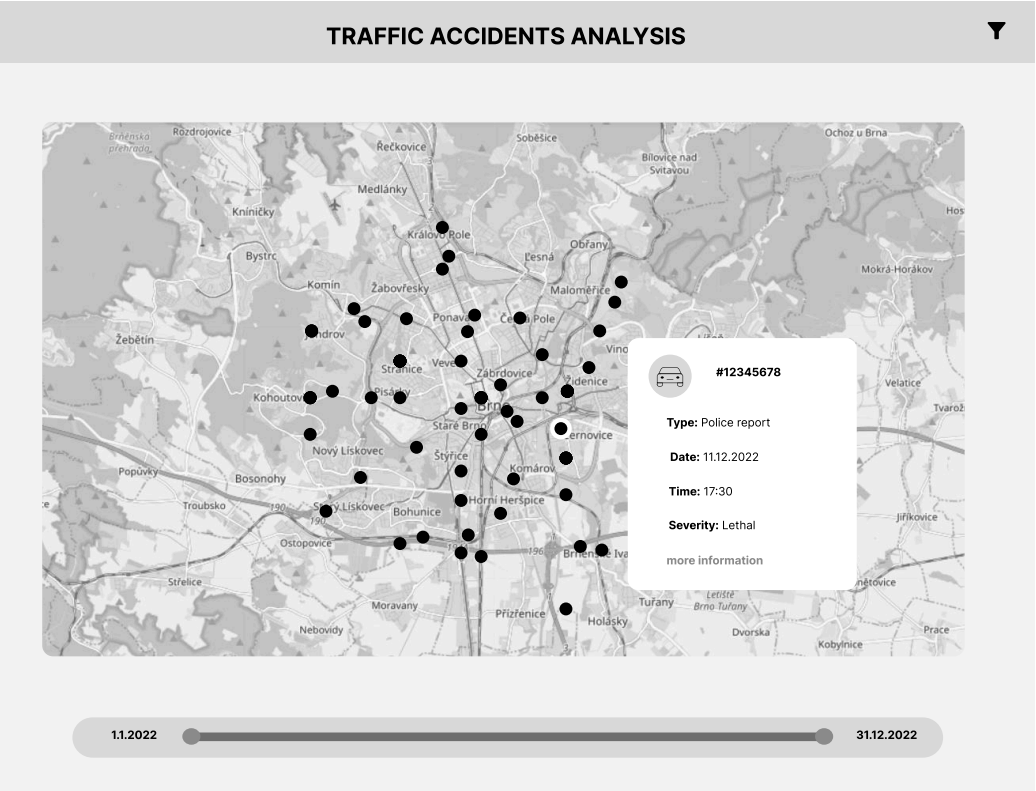

Figure 5.5: Early design of the map view.

Accident Detail View

For each police-reported accident, a dedicated detail view will be available, accessible by clicking on the markers on the map. When a user selects an accident, a panel will display key information such as the coordinates, time, date, severity, main cause, vehicle type, and whether alcohol was involved.

The detail view will also include a summary of any matched Waze alerts. If Waze reports have been linked to the accident, the total number will be shown as a clickable

element that navigates the user to a dedicated view. This secondary view will present a list of all individual Waze reports associated with the accident, including their timestamps and a match score based on spatial and temporal proximity. When the accident detail panel is open, the corresponding Waze report markers are also dynamically displayed on the map, allowing users to visually examine their locations in relation to the selected police-reported accident.

Each Waze-reported accident shown on the map will also have its own dedicated detail view. This view will present recalculated information derived from the clustered reports, including coordinates, time, date, road type, and a reliability score that reflects the confidence in the event's accuracy. In addition, the detail view will include a list of all original Waze reports that contributed to the cluster, allowing users to examine each individual submission.

Both police and Waze accident detail views will follow a consistent design where only a subset of the most relevant attributes is shown initially to avoid overcrowding the interface. Users can expand the view using the "see more" option to reveal the complete set of attributes. Additionally, a button will be available in both detail views to copy the coordinates for external use.

5.3.2 Dashboard

The dashboard will serve as the central point for exploring overall statistics and key trends within the traffic accident dataset. It will present a combination of summary metrics, time-based visualisations, and categorical charts to help users interpret the data from multiple perspectives. An early design of the dashboard view can be seen in Figure 5.6.

Summary Indicators

At the top of the dashboard, a set of four key metrics will provide a quick overview of the dataset's overall scale and severity:

- Total number of accidents
- Total number of fatalities
- Total number of seriously injured persons
- Total estimated property damage

These indicators will be positioned for immediate visibility to help users orient themselves before exploring the data in more detail.

Time Series Visualizations

Below the summary section, several time series charts will offer insight into how accident data changes over time. These visualizations will make it easier to detect long-term trends, identify seasonal patterns, and observe sudden spikes or drops in incident counts. This format is consistent with commonly used approaches in traffic analysis applications, where time-based insights are often central to planning and decision-making.

- The first chart will display the total number of reported accidents per day, combining data from both police and Waze sources into a unified view.

- A second chart will track the severity of accidents by plotting daily counts of fatalities, serious injuries, and light injuries in a single graph for comparison.

Categorical Charts

In addition to the time-based visualizations, the dashboard will include several bar and pie charts that break down categorical variables. These charts will provide insights into the distribution of accident types, causes and consequences, as well as the characteristics of the road infrastructure, including road classification, road geometry (such as curves, intersections, and straight sections), and accident position on the road (such as left or right lane, or the road shoulder).

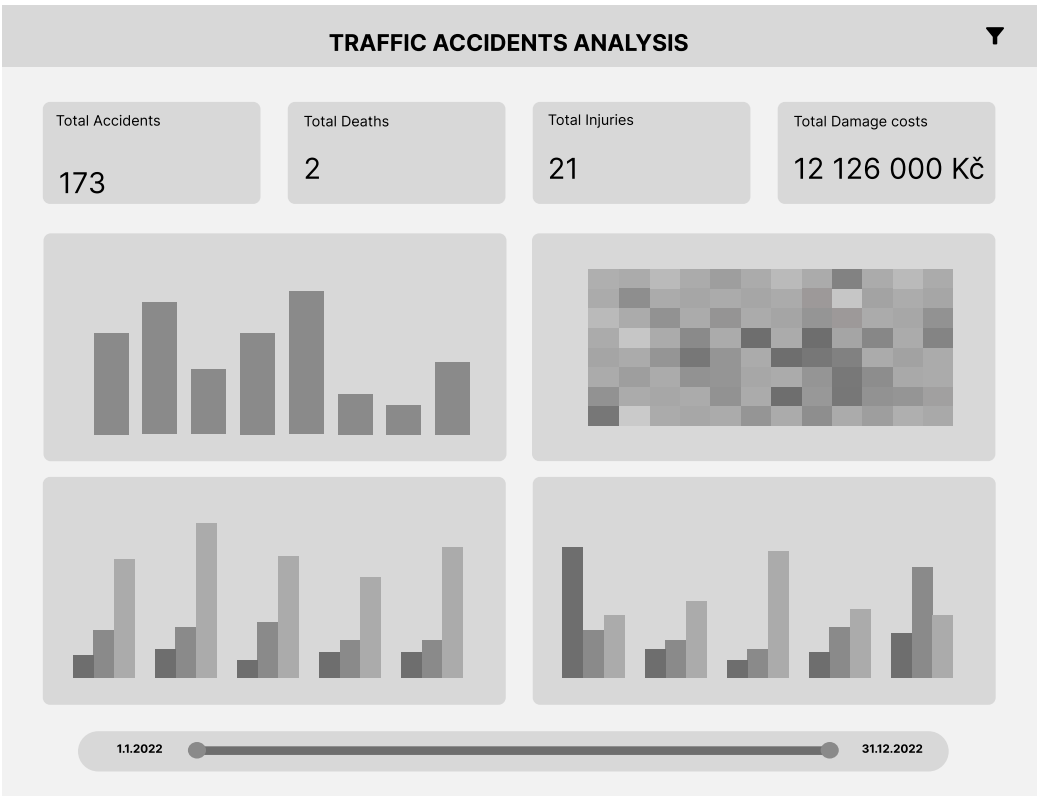

Figure 5.6: Early design of the dashboard view.

Heatmap

A heatmap showing the number of accidents by hour and day will be added to highlight typical accident patterns during the week. This visualization makes it easier to spot peak times and recurring risky periods.

5.3.3 Filtering

Filtering plays a key role in enabling users to interact with and explore the data more efficiently. Both the interactive map and the dashboard will support filtering to help users focus on specific subsets of the data and the selected filter settings will remain active across

both screens, ensuring a consistent view of the data. Two types of filters will be available: a simple time-based filter and a more advanced attribute-based filter.

Time Filter

The time filter will be a persistent control element placed at the bottom of the screen and will always be visible to the user. It will allow for quick temporal restriction of the displayed data by selecting a specific time interval. This filter can be used to explore accidents and Waze alerts that occurred within a given date range, which is especially useful when analyzing daily or seasonal patterns.

Attribute Filter

The attribute filter will enable advanced filtering of police accident records based on any field present in the dataset. Users will be able to filter by various attributes such as accident type, main cause, road surface condition, weather, visibility, and more. The full list of attributes is available in Appendix A.1. For each attribute, users will be able to select specific values using either the equality or inequality operator, allowing both inclusion and exclusion of categories.

Multiple attributes can be combined to construct more complex filter conditions. For example, users will be able to simultaneously filter for accidents that occurred during rain, were caused by speeding, and did not involve alcohol. This will allow users to build highly specific queries and supports deeper, user-driven exploration of the dataset.

Chapter 6

Implementation

The goal of this chapter is to describe the implementation of the web application extension based on the design decisions mentioned in the previous chapter. It describes algorithms used to create relations between multiple datasets, granting uniqueness and consistency of the presented data, as well as necessary operations, which are performed on the data, so they can be adopted for the project's use case. The application is based on a typical client-server architecture, using REST API endpoints for communication with the server, which exposes data in a predefined format, performing expensive calculations, and filtering on the server. The chapter also describes the implementation of the client-side part, using modern web development tools. Details of the functionality will be described later in the chapter.

6.1 Backend

The task of the backend application in the system is to provide processed data for the client application. As mentioned in Chapter 5, the backend is implemented in the Python programming language, using the `FastAPI` library for creating a REST API, which is then used by the web application to retrieve data for visualisation. The `main.py` file is an entry point to the application; it initialises routes and middleware, as well as performs the initial loading and processing of the data. The project is initialised using the `Poetry` dependency management and packaging tool, which addresses issues like version conflicts and ensures reproducible builds through the use of the `poetry.lock` file.

6.1.1 Data loading and preprocessing

As a start of the application, the `DataLoader` object is initialised. This is a class that uses a singleton design pattern with methods for loading data from various sources, such as static files or a PostgreSQL. In the final implementation of the project, the PostgreSQL database serves as the primary data source. This modular approach allows for flexibility in switching or extending data sources in the future without major changes to the application logic. The format of data is verified using the `Pydantic` library and the data is later saved to `Pandas Dataframe` for simple database-like manipulation and filtering. To make the data from the police dataset more understandable to humans, it is transformed using a predefined mapping dictionary located in the `models/data_map.py` file using the `_transform_accident(accident)` function. The original dataset records values using coded keys—for example, `{„p11": 1}` indicates an accident involving alcohol. The

dictionary contains key-value pairs for each attribute and its possible values. All subsequent operations on the data, such as filtering, are performed using the already transformed attributes. This approach simplifies the debugging and working with the data. The police dataset uses Krovak East North coordinate system (EPSG:5514), which is transformed to EPSG:4326 during data loading using the PostGIS function `ST_X(ST_Transform(geom, 4326))`. The Waze reports dataset uses the EPSG:4326 coordinate system, so it is beneficial to maintain consistency across datasets.

Due to the relatively high computational cost of preprocessing, it is not performed on every request, as this would significantly increase the loading time in the client application and negatively impact the user experience. Instead, preprocessing is executed at the start of the application and cached. To ensure the data remains up to date, a background job using `apscheduler` is utilised to periodically refresh the data.

6.1.2 Matching Accidents Based on Temporal and Spatial Proximity

Matching accidents based on spatial and temporal proximity is performed in two key steps within this project. The first step is to process duplicate Waze reports and cluster into unique events for visualization purposes, while the second focuses on matching individual Waze alerts to police-reported accidents, as seeing all matched alerts provides more contextual information than just one unique event.

Clustering Waze Reports

Due to the crowdsourced nature of Waze data, multiple users may report the same traffic incident, resulting in duplicate alerts. To consolidate these, Waze reports are clustered based on both spatial and temporal proximity. Specifically, the following thresholds are applied:

- Spatial proximity: 1 km radius,
- Temporal proximity: 90-minute time window.

Efficient spatial searches are performed using a KD-Tree data structure, which allows for fast retrieval of nearby reports. Within each cluster, the earliest published report is selected as the primary alert, serving as the representative of the entire event. Furthermore, the position of the primary report is recalculated to better represent the cluster. This recalculated location is computed as a reliability-weighted average of all reports in the cluster.

Given n reports in a cluster, the representative coordinates (x, y) of the event are calculated as:

$$x = \frac{\sum_{i=1}^{n} x_i r_i}{\sum_{i=1}^{n} r_i}, \quad y = \frac{\sum_{i=1}^{n} y_i r_i}{\sum_{i=1}^{n} r_i} \tag{6.1}$$

where x_i and y_i represent longitude and latitude of the i-th report, and r_i represents its reliability score found in the `reliability` attribute.

The following pseudocode describes the core algorithm used to cluster individual Waze alerts into one event.

Algorithm 1 Clustering Waze Reports to Remove Duplicates

```
for each Waze report do
    if report is already clustered then
        continue
    end if
    Find nearby reports within 1 km using KD-Tree.
    for each nearby report do
        Compute time difference Δt.
        if Δt ≤ 90 minutes then
            Add report to current cluster.
        end if
    end for
    Mark the earliest report in the cluster as primary.
    Recalculate primary's position using reliability-weighted average.
end for
```

On a representative data sample 5,434 unique traffic events were identified from 9,595 individual Waze reports after removing duplicates and consolidating overlapping reports, resulting in a **43.37%** reduction of redundant data.

Matching Waze Alerts to Police Accidents

Individual Waze alerts, not clustered events are linked to the police reported accidents. This approach allows analysis of how many Waze reports correspond to a single police accident and the time distribution of these reports. For each police accident, the system searches for candidate Waze reports within a spatial threshold of **1 kilometer** using a KD-Tree for efficient spatial queries. Temporal constraints are applied as follows:

- If the police report has a valid timestamp, Waze reports are considered if they fall within a time window of **-20** to **+120 minutes** relative to the police accident time.
- If the police report lacks a valid timestamp (approximately 30% of police reports do), Waze reports from the same day are considered.

Threshold selection was based on prior research [17] and typical driver behavior when reporting accidents driver-behavioural studies [18].

For every candidate Waze report, the geodesic distance d between the Waze report and the police accident is calculated. A match score is then computed to evaluate the relevance of the match:

$$\text{match\_score} = \begin{cases} \frac{1}{1+\frac{d}{100}+\frac{|\Delta t|}{30}} & \text{if police time is valid} \\ \frac{1}{1+\frac{d}{100}} & \text{if police time is missing} \end{cases} \tag{6.2}$$

where d is the distance in meters and Δt is the time difference in minutes between the police report and the Waze alert.

If a Waze report matches multiple police accidents, the system assigns it to the accident with the *best match score*. This process allows overwriting previous matches if a better

match is found, ensuring each Waze report is linked to the most relevant police report. If a police accident lacks a valid timestamp, the accident time can be estimated using the earliest matched Waze report.

The following pseudocode describes the core algorithm used to match individual Waze alerts to police-reported accidents.

Algorithm 2 Matching Police Reports with Waze Reports

```
for each police-reported accident do
    Find Waze reports within 1 km radius.
    for each candidate Waze report do
        Compute time difference Δt.
        if time constraints are satisfied then
            Calculate geodesic distance d.
            Compute match score based on d and Δt.
            if match score is better than previous match then
                Update Waze report with matched police accident ID.
            end if
        end if
    end for
end for
```

On a representative data sample where both official police reports and Waze alerts were available, 306 police-reported accidents (**20.07%**) were successfully matched with at least one corresponding Waze report, out of a total of 1,525 processed accidents.

6.1.3 REST API interface

The API is organised using Fastapi's router system, with each part of the application defined by its router. Routes for accident data, Waze reports, and charts are grouped under `/api/v1/accidents`, `/api/v1/waze`, and `/api/v1/charts`, respectively. The structure improves the maintainability and potential extensibility of the API in the future, while maintaining backward compatibility using API versioning. The individual controllers are located in the `routers` folder. Both `accidents` and `waze` controllers have `GET` endpoints for getting a filtered list of records, and an individual record specified by its `uuid` attribute.

Records can be filtered using query parameters in the URL of the query, for example `api/v1/waze?startDate=2024-12-01&endDate=2024-12-31` to get records in December 2024. Accidents from the police datasets can be filtered by various attributes, such as whether alcohol was involved or the consequences of accidents. A simple querying language was developed for filtering. The `api/v1/accidents?alkohol:eq=ano&pocet_vozidel:eq=3` would get accidents, where alcohol was involved, and the number of involved vehicles was equal to 3. The filtering is performed by the `get_filterd_df()` function located in `utils/filter.py` file, which parses the incoming query and returns a dataframe corresponding to the query.

The `/api/v1/charts` route provides endpoints for retrieving data used to generate various types of charts in the dashboard, such as accident frequency over time or distributions of accident attributes. The data is preprocessed on the server, simplifying the frontend's task by delivering it in a format ready for visualisation. The chart data generation either

uses raw SQL queries with aggregate functions, such as `GROUP BY` or `COUNT`, or Pandas utilities directly in Python for more complex operations.

6.1.4 Deployment

A **Dockerfile** with **Docker Compose** are used to simplify the deployment and running the application locally. The repository includes a database dump and a shell script to automatically initialise the database. Running `docker compose up -d -build` will start the application at port 8000. The **PgAdmin** tool has been integrated into the Docker Compose configuration to simplify direct operations with the database.

6.2 Frontend

The frontend of the web application is developed in **TypeScript** using the **React.js** framework, consistent with the original work, which allows for seamless integration of the existing functionality. Its primary role is to display the aggregated and processed data retrieved from the backend server in a clear and comprehensible manner, utilising maps and charts. The application structure is organised into three main parts: **layouts**, **pages**, and **components**. The layout defines the common user interface elements visible on every page, which belong to the layout, such as the top navigation bar and the time window selector. It also manages the routing logic and determines which page renders for the given URL.

For styling, the application uses a combination of `Tailwind CSS` and `SCSS`. `Tailwind CSS` is used in the *accidents* section of the application, while `SCSS` is applied in the *traffic jams* section. To avoid conflicts between the two styling systems, each layout imports its own specific stylesheets, ensuring style encapsulation and consistency across the application. **ShadCN UI** library is used for generic UI components, such as dropdown menus.

Other libraries are **Recharts** for various types of charts, **Axios** for making network requests, **Tanstack React Query** and **Zustand** for state management, or **React Leaflet** for interactive map visualisation.

6.2.1 Data fetching and state management

The frontend retrieves data from the backend server using asynchronous network requests. To handle these operations efficiently, the application uses the **Axios** library for making HTTP requests and **TanStack React Query** for managing server state. The The **TanStack React Query** library provides advanced features such as automatic handling of loading and error states, as well as efficient client-side caching. Each data record is associated with a unique query key, which allows for the instant retrieval of previously fetched data when the same key is requested again. This is particularly beneficial in scenarios such as comparing the effects of different filters, where switching between options can reuse cached results without requiring additional network requests. The **Zustand** library is used to manage client-side state within the application. It ensures the persistence of user selections, such as filter configurations, across different pages. To maintain state after a page refresh, the application synchronises the Zustand store with browser's `localStorage`.

6.2.2 User interface

Similarly to the original work, the layout of the application utilises a top navigation bar and a slider for selecting a time window, allowing users to filter events by time. Options

in top bar allow navigation between different pages, and opening the filter menu, used for more advanced filtering, allowing users to explore specific subsets of the data efficiently. The application supports localisation using the **i18n** library, allowing the user interface to be displayed in three languages: English, Czech, and Slovak.

Interactive Map

The application integrates an interactive map, seen in Figure 6.1, using the **React Leaflet** library. This map is used to display spatial data, such as the locations of accidents and Waze reports. To improve both performance and readability when handling large datasets, the map supports multiple visualisation modes. When the user is zoomed out, events are grouped into clusters to reduce visual clutter and ensure smooth performance. In addition, a heatmap mode is available, which highlights areas with a high concentration of events. Users can interact with the map by selecting individual events to view additional details, including the date, time, and other relevant information. The event detail panel allows users to easily copy the geographic coordinates of an event for use in external tools. In cases where an accident is matched with Waze reports, the panel also displays a list of the corresponding Waze records, providing additional context and verification. The user has the ability to export accident details in `JSON` format.

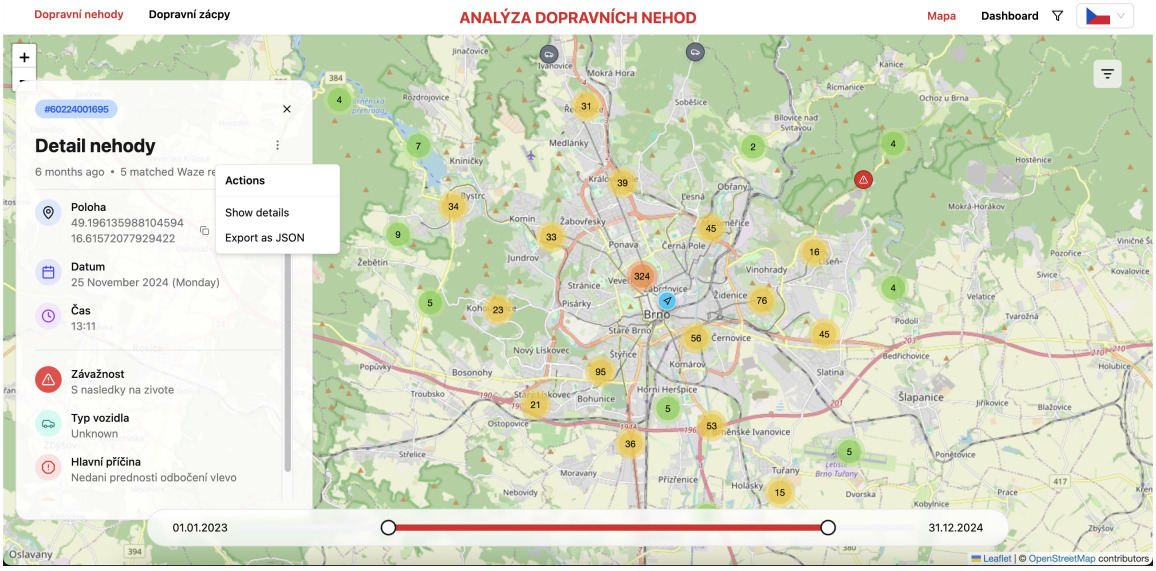

Figure 6.1: Screenshot of the interactive map in the final application.

Dashboard

The dashboard page, seen in Figure 6.2, consists of various charts that display insights derived from the filtered data. At the top, a summary statistics chart provides an overview of the selected period. The dashboard uses a responsive two-column grid layout, wherein individual charts occupy a single column by default, while more detailed visualisations span both columns to improve readability.

The `Recharts` library is used, providing a wide range of chart types, including line, area, and pie charts to render the dashboard visualizations. Each chart is implemented

as an independent React component responsible for fetching its own data based on the active filters. User-defined filter parameters are stored in a shared `Zustand` store, ensuring consistent access to filters across all chart components. Whenever a filter configuration changes, the affected chart component requests updated data from the backend. To optimise performance and prevent excessive requests, while users adjust filters, such as dragging handles on a temporal slider, a debounce mechanism using `useDebounce` hook is applied to delay data fetching until user input stabilises for a defined period of time. The use of `Tanstack React Query` ensures caching of fetched data for the given filter configuration. Data from the backend are retrieved in a format, which is ready for visualisation, therefore no expensive data processing, which may cause freezing of the UI, is needed to be performed on the client side. While the data are being fetched by a chart, a skeleton component is shown to represent the loading state.

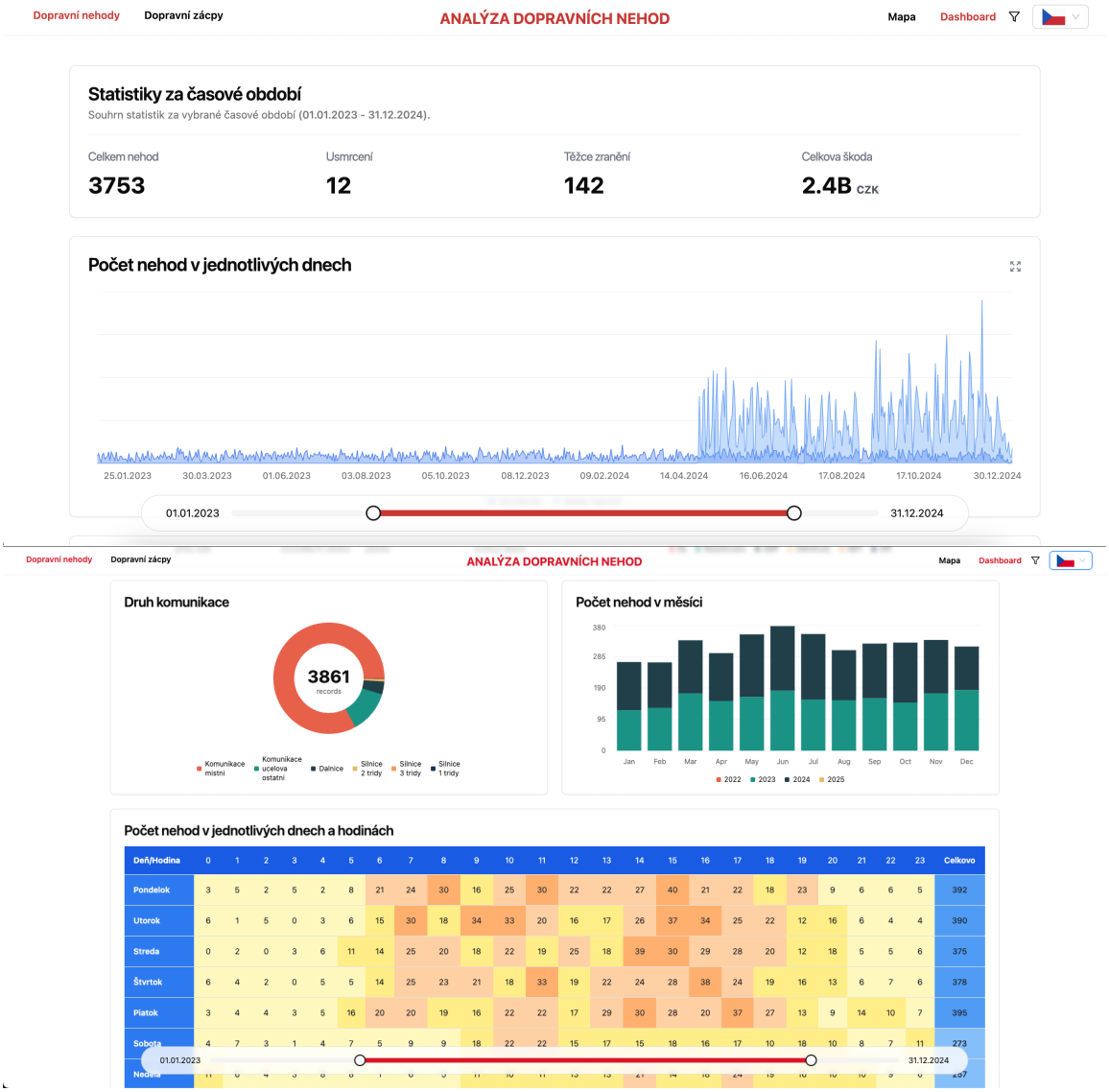

Figure 6.2: Screenshots of the dashboard in the final application.

Filtering

Advanced filtering functionality provides users with precise control over which accidents are displayed. The filtering system operates in two modes, depending on the selected attribute: a predefined list of values for categorical attributes—such as involvement of drugs—or a numeric input for quantitative attributes, such as the number of injured people. Once an attribute is selected, the user can choose a comparison operator (e.g., equals, not equals, greater than, or less than) and specify a value to filter the data accordingly.

The available filtering options are dynamically retrieved from the backend via the `/filter-schema` endpoint. This schema defines all supported attributes and their possible values or value types. Upon retrieval, the schema is stored in a global state management object called `ConfigStore`, making it accessible to all components during data fetching and rendering.

To enable consistent filtering across different parts of the application, such as individual charts on the dashboard, the current filter state can be serialised using the `asQueryParams()` function. This function converts the selected filters into a query string format suitable for use in backend API calls. Users can specify multiple filters, which are combined using a logical **AND** operation, meaning that only events matching all selected criteria are displayed.

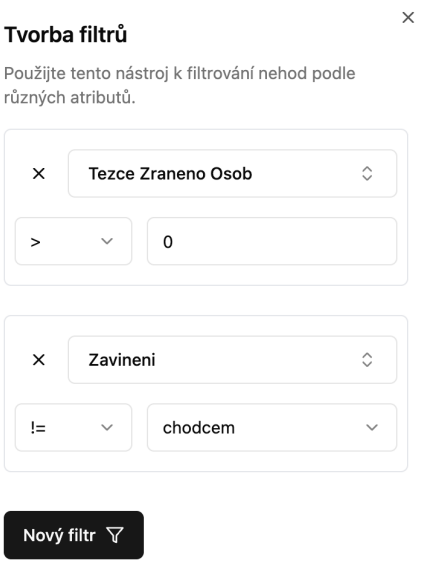

Figure 6.3: Screenshot of the filter builder.

6.2.3 Responsiveness and Mobile Optimization

The application is primarily designed for desktop use, where a larger screen allows for an overview of maps, dashboards, and detailed filters. However, basic mobile optimization has been implemented by adjusting the layout of key components. On smaller screens, elements that are displayed side-by-side on desktop are automatically arranged vertically, ensuring usability even on mobile devices.

Chapter 7

User Testing

Two target user groups were involved in the testing process. The first group, representing members of the general public, included primarily younger users with average technical proficiency but no deep background in traffic, data analysis or city planning. The second group consisted of master's degree students in the Urban Engineering program at the Faculty of Civil Engineering at Brno University of Technology, representing professional users with domain-specific knowledge.

7.1 Testing Methodology

These participants were asked to perform task-based interactions designed to evaluate the core functionality and usability of the application. All participants had to attempt a set of predefined tasks.

- Locate an accident that has at least one matched Waze report.
- Determine how many fatalities resulted from accidents involving alcohol and no pedestrians.
- Compare weekday and weekend accident frequency using dashboard charts.

After completing the tasks, users were encouraged to freely explore the application and share their impressions.

7.1.1 Feedback from General Users

The testing group of general-public users highlighted several usability issues during their exploration that needed to be addressed.

- Users found that when a large time range was selected, the map became cluttered with overlapping data points, making interpretation difficult. To address this issue, clustering for accident markers when zoomed out was implemented to reduce visual clutter.
- Some users also found it challenging to adjust the time range slider accurately for selecting specific days. A slider element was originally chosen for consistency with the base application. The original application only handled data dating back one year, but since the extension includes a broader period of data, sliding becomes less

precise. A decision was made to retain the slider element to maintain consistency, but once users move the slider close to the desired date, they can fine-tune the selection using the arrow keys on the keyboard.

- Lastly, there was feedback regarding the use of colours and placement of UI elements, which some found unintuitive or visually misleading. In response, these interface elements were revised to improve clarity and usability.

7.1.2 Feedback from Professional Users

The professional users, on the other hand, provided more domain-specific feedback based on their need for contextual awareness in analyzing incidents.

- Users expressed a need for more ways to understand the context of specific accident locations within the application. Specifically, they proposed integrating Google Street View to provide more visual information directly on the map interface. While this specific feature was not implemented, their feedback led to the addition of a button to easily copy geographic coordinates for use in external tools.

- Users also suggested using different icons to represent various types of incidents. While Waze and police-reported accidents were already displayed with distinct markers, they recommended further differentiation of police accidents based on severity. This would allow users to visually distinguish between minor and severe incidents at a glance, without needing to apply filters or click on individual markers for more details.

7.2 Evaluation Results

All tasks were completed successfully by all users within a short time frame and with only minor missteps, which were quickly recognized and corrected. This suggests that the core functionality of the application was intuitive and easy to use.

The results of the usability tasks and subsequent exploration highlighted aspects of the application that needed immediate refinement because they compromised the planned functionality of the app, but they also provided ideas that, while out of scope for this project, offer valuable direction for future development to better accommodate the needs of both user groups.

Chapter 8

Conclusion

This thesis successfully extends an existing traffic data analysis system by integrating traffic accident data from official police reports and crowdsourced Waze alerts. The objective was to provide an enhanced view of urban traffic safety conditions, focusing on the city of Brno, while ensuring the solution remains scalable to other regions within the Czech Republic.

Two complementary datasets were utilized: detailed, structured police accident records and user-reported Waze alerts. Combining these sources allowed the system to address data gaps, enhance situational awareness, and provide richer analytical insights. The developed extension enables users to visualize accidents on an interactive map, explore aggregated statistics through dashboards, and apply flexible filtering options for customized analysis.

The system was designed to meet the needs of two primary user groups: the general public and traffic professionals. During the design process it was important to balance the requirements of both groups, offering accessible visualisation tools while maintaining the depth of analysis needed by professionals. Feedback from user testing confirmed the system's usability and relevance for both target audiences.

Although developed and tested using data from Brno, the solution was designed to be easily extendable to other cities. In principle, once additional locations are added to the underlying database in a compatible format, the system should function without further modifications. However, this has not been practically verified due to the unavailability of suitable data for other regions.

While the current implementation achieves its primary goals, several opportunities for future work remain. These include integrating additional data sources, optimizing performance for larger datasets as the system expands to cover more regions, and enhancing contextual analysis through features like Street View integration. Such improvements would further increase the system's value for urban planners, policymakers, and the general public.

Bibliography

[1] Alvarez, F. M.; López Rubio, E.; Munoz Perez, J. and Campo, J. del. Event-Based Time Series Data Preprocessing: Application to Traffic Flow Analysis. In: *International Conference on Artificial Intelligence*. UPM, 2014, p. 533–543. Available at: https://oa.upm.es/36830/1/INVE_MEM_2014_195277.pdf.

[2] California, I. L. University of. *Research Guides: History 15A: Native American History: Home*. 2024. Available at: https://guides.lib.uci.edu/c.php?g=333028&p=8281490. Accessed: 2024-11-20.

[3] Campbell, J. and Shin, M. *Essentials of Geographic Information Systems*. 1st ed. FlatWorld Knowledge, 2011.

[4] Carpentry, D. *Introduction to Raster Data*. 2025. Available at: https://datacarpentry.github.io/organization-geospatial/01-intro-raster-data.html. Accessed: January 30, 2025.

[5] Consortium, O. G. *GeoTIFF Revision 4*. 2019. Available at: https://docs.ogc.org/is/19-008r4/19-008r4.html. Accessed: 2024-11-20.

[6] Dempsey, C. *Types of GIS Data Explored: Vector and Raster*. Geography Realm, 2024. Available at: https://www.geographyrealm.com/geodatabases-explored-vector-and-raster-data/. Accessed: 2024-11-01.

[7] Department of Geography, Penn State University. *Vector Versus Raster*. John A. Dutton e-Education Institute, College of Earth and Mineral Sciences, The Pennsylvania State University, 2012. Available at: https://www.e-education.psu.edu/geog160/node/1935. Accessed: 2024-11-01.

[8] Dunning, T. and Friedman, E. *Time Series Databases: New Ways to Store and Access Data*. Sebastopol, CA: O'Reilly Media, Inc., 2015. ISBN 978-1-491-91702-2. Available at: https://d1wqtxts1xzle7.cloudfront.net/37040996/Time_Series_Databases-libre.pdf. Accessed 2024-12-14.

[9] Earth, C. *Spatial Data Foundations for Nature*. 2024. Available at: https://newsletter.cecil.earth/p/spatial-data-foundations-for-nature. Accessed: 2024-12-21.

[10] Few, S. *Information Dashboard Design: The Effective Visual Communication of Data*. Sebastopol, CA: O'Reilly Media, 2006.

[11] Frank, J. and Wing, M. G. Balancing horizontal accuracy and data collection efficiency with mapping-grade GPS receivers. *Forestry: An International Journal of Forest Research*. Oxford University Press, 2014, vol. 87. Available at: https://academic.oup.com/forestry/article/87/3/389/2756007. Accessed on 24 November 2024.

[12] Geography, G. *Spatial Resolution vs Spectral Resolution*. GIS Geography, 2023. Available at: https://gisgeography.com/spatial-resolution-vs-spectral-resolution/. Accessed: 2024-11-02.

[13] Giulia Del Serronea, P. P. Speed data collection methods: a review. *Transportation Research Procedia*. Elsevier, 2023, vol. 69, p. 512–519.

[14] Haining, R. *Spatial Data Analysis: Theory and Practice*. Cambridge, UK: Cambridge University Press, 2003. ISBN 978-0-521-77437-1.

[15] Halachkin, A. Vehicle Classification Using Inductive Loops Sensors. In:. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, p. 302–304. ISBN 978-80-214-5496-5.

[16] Informatica. *What is ETL?* 2025. Available at: https://www.informatica.com/resources/articles/what-is-etl.html. Accessed: January 30, 2025.

[17] Li, X.; Dadashova, B.; Yu, S. and Zhang, Z. Rethinking Highway Safety Analysis by Leveraging Crowdsourced Waze Data. *Sustainability*. MDPI, 2020, vol. 12, no. 23, p. 10127. Available at: https://www.mdpi.com/2071-1050/12/23/10127.

[18] Neto, V. R.; Medeiros, D. S. V. and Campista, M. E. M. *Analysis of Mobile User Behavior in Vehicular Social Networks*. RMC16. Universidade Federal do Rio de Janeiro, 2016. Available at: https://www.gta.ufrj.br/ftp/gta/TechReports/RMC16.pdf.

[19] Ondrušková, M. *Analýza a vizualizace dopravních dat města Brna*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

[20] Organization, W. H. *Road Safety*. 2025. Available at: https://www.who.int/health-topics/road-safety. Accessed: 2025-01-28.

[21] Santos, S. Ribeiro dos; Davis Jr., C. A. and Smarzaro, R. Integration of data sources on traffic accidents. In: Federal University of Minas Gerais. *Proceedings XVII GEOINFO, November 27-30, 2016, Campos do Jordão, Brazil*. 2016. Available at: https://www.researchgate.net/publication/311457168.

[22] Topics, S. *Time Series Data*. 2024. Available at: https://www.sciencedirect.com/topics/mathematics/time-series-data. Accessed: 2024-12-15.

[23] Tufte, E. R. *The Visual Display of Quantitative Information*. 2ndth ed. Cheshire, CT: Graphics Press, 2001. ISBN 9780961392147.

[24] Wilke, C. O. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. Sebastopol, CA: O'Reilly Media, 2019. ISBN 9781492031086. Available at: https://clauswilke.com/dataviz/.

[25] Wolf, C.; Christmann, P.; Gummer, T.; Schnaudt, C. and Verhoeven, S. Conducting General Social Surveys as Self-Administered Mixed-Mode Surveys. *Public Opinion Quarterly*. Oxford University Press, 2021, vol. 85, no. 2, p. 623–648. Available at: https://academic.oup.com/poq/article/85/2/623/6374807.

[26] Zendulka, J.; Bartík, V.; Lukáš, R. and Rudolfová, I. *Získávání znalostí z databází*. Brno: FIT, october 2009.

[27] zVUT.cz. V centru AdMaS mají speciální vozidlo. Díky laseru dokáže rychle a efektivně zmapovat velké území. *ZVUT.cz*, 2017. Available at: https://www.zvut.cz/tema/tema-f38144/v-centru-admas-maji-specialni-vozidlo-diky-laseru-dokaze-rychle-a-efektivne-zmapovat-velke-uzemi-d136405.

Appendix A

Datasets

A.1 Police Accidents

Attribute	Description	Data Type
p1	Accident ID	Integer
p2a	Date of accident	Date
p2b	Time of accident	String (optional)
p4a	Region	Integer
p4b	District	Integer
p5a	Locality (urban/rural)	Integer
p6	Type of accident	Integer
p7	Type of collision	Integer
p8	Obstacle type	Integer
p8a	Type of animal	Integer
p9	Accident consequences	Integer
p10	Cause of accident	Integer
p11	Presence of alcohol	Integer
p11a	Presence of drugs	Integer
p12	Main cause code	Integer
p13a	Number of fatalities	Integer
p13b	Number of severely injured persons	Integer
p13c	Number of lightly injured persons	Integer
p14	Total property damage	Integer
p16	Surface condition	Integer
p17	Road condition	Integer
p18	Weather conditions	Integer
p19	Visibility conditions	Integer
p20	Line of sight	Integer
p21	Road division type	Integer
p22	Accident location on road	Integer
p23	Traffic control during accident	Integer
p24	Local priority control	Integer
p27	Special places/objects	Integer
p28	Directional characteristics	Integer
p29	Pedestrian category	Integer

Attribute	Description	Data Type
p29a	Reflective elements on pedestrian	Integer
p29b	Pedestrian on personal transporter	Integer
p30	Pedestrian's condition	Integer
p30a	Alcohol presence in pedestrian	Integer
p30b	Drug presence in pedestrian	Integer
p34	Number of vehicles involved	Integer
p35	Location of accident (type)	Integer
p36	Type of road	Integer
p37	Road ID number	String or Integer
p39	Type of crossing road	Integer
p44	Type of vehicle	Integer
p45a	Vehicle manufacturer brand	Integer
p47	Vehicle manufacturing year	String or Integer
p48a	Vehicle type (ownership/use)	Integer
p49	Skid occurrence	Integer
p50a	Vehicle condition after accident	Integer
p50b	Fluid leak presence	Integer
p51	Rescue method for persons	Integer
p52	Vehicle movement/position	Integer
p55a	Driver's license category	Integer
p57	Driver's physical condition	Integer
p58	Driver's external influences	Integer
p59g	Injury outcome	Integer
x	Longitude (geographical)	Float
y	Latitude (geographical)	Float

A.2 Waze Alerts

Attribute	Description	Data Type
uuid	Unique report ID	String
country	Country code	String
city	City name	String
type	Report type	String
subtype	Detailed report type	String
street	Street name	String (nullable)
report_rating	User rating of report	Integer
confidence	Confidence in report	Integer
reliability	Reliability score (0–10)	Integer
road_type	Road type ID	Integer
magvar	Magnetic variation	Integer
report_by_municipality_user	Reported by official user	Boolean
report_description	Optional text description	String (nullable)
location	Geographical location	String
published_at	Report publication time	DateTime
last_updated	Last update timestamp	DateTime
active	Report still active	Boolean