Freie Universität Berlin

Data Analysis Tool for the EU H2020 City.Risks Operation Center

Erik Zocher (4797383)
erik.zocher@fu-berlin.de

A thesis presented for the degree of
Bachelor of Science

Advisors:
Prof. Dr. Agnès Voisard
Prof. Dr. Katinka Wolter

Supervisor:
Paras Mehta

Freie Universität Berlin
Department of Computer Science & Mathematics
Institute for Computer Science
Research Group: Databases and Information Systems

2nd August 2017
Berlin, Germany

Abstract

The bachelor thesis is about the implementation of a Data Analysis Tool for the City.Risks Project funded by the Horizon 2020 (H2020) European Research and Innovation program. The aim is to develop an interface that is easy to interact with but in the same time provides all the necessary tools to conveniently gain knowledge out of a large dataset. This is achieved by using a search engine for the implementation and various user-centered design methods to design a user-friendly graphical user-interface that provides visualizations of the dataset as well as the possibility to search through the dataset by keywords. The underlying functionality is achieved by implementing a search engine with Lucene and Elasticsearch. The design of the user interface is realized by iteratively applying user-centered design methods like heuristic evaluation, action analysis and think-aloud.

Keywords user-centered design, search engine, observer design pattern, angular2+, elasticsearch, lucene

Die vorliegende Bachelorarbeit behandelt die Entwicklung eines Datenanalysetools für das von der europäischen Union im Rahmen des Forschungs- und Innovationsprogramms Horizon 2020 geförderten Projekts City.Risks. Ziel ist, eine Benutzeroberfläche zu entwickeln mit der sich leicht interagieren lässt, welche gleichzeitig alle nötigen Werkzeuge zur Verfügung stellt, um auf einfache Art und Weise neue Einsichten aus einem großen Datensatz zu erhalten sowie die Möglichkeit in dem Datensatz nach Schlagwörtern zu suchen. Die zugrunde liegende Funktionalität wird durch die Implementierung einer Suchmaschine mit Lucene und Elasticsearch erreicht. Der Entwurf der Benutzerschnittstelle wird durch die wiederholte Anwendung von nutzerzentrierten Designmethoden wie heuristische Auswertung, Aktionsanalyse und Think-Aloud realisiert.

Schlagwörter Nutzerorientierte Gestaltung, Suchmaschine, Beobachter (Entwurfsmuster), Angular2+, Elasticsearch, Lucene

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

API	Application Programming Interface
DAT	Data Analysis Tool
DSL	Domain-Specific Language
H2020	Horizon 2020, the European Research and Innovation Action
IDF	Inverse Document Frequency
ID	Identifier
IoT	Internet of Things
JSON	JavaScript Object Notation
NoSQL	Not only SQL
NPM	Node Package Manager
REST	Representational State Transfer
RxJS	Reactive Extensions for JavaScript
TF	Term Frequency
URL	Uniform Resource Locator

Statutory declaration

I declare that I have developed and written the enclosed Bachelor Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Bachelor Thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Berlin (Germany), 02nd August 2017

Erik Zocher

Chapter 1

Introduction

The first chapter provides the motivation for developing the software that is used to generate insights into spatial and temporal data, the goals of this thesis as well as the structure of the work.

1.1 Motivation

> "Data by itself is of no value unless it is processed to obtain information [..]"[1]

More and more data is produced every day. IBM stated in 2013 that "90% of the data in the world today has been created in the last two years alone."[2] Yet according to the MIT Technology Review Author Antonio Regalado "only about 0.5 percent of that data is ever analyzed."[3] Gaining valuable insights out of data has become an huge market in the last few years and increases a lot in the future. According to the International Data Corporation the Data and Business Analytics market is predicted to grow its revenue nearly 50% in five years from $122 Billion in 2015 to more than $187 billion in 2019. Nearly $55 billion of this revenue in 2019 is made from Software, like End-User Query, Reporting, Analysis and Data Warehouse Management Tools.[4] The crucial parts for those software solutions, according to Rita Sallam, research vice president at Gartner, Inc. are "design, implementation and support."[5]
As part of the City.Risks project funded by the European Union, an operation center is being developed in which information from citizens, authorities and devices is collected and further processed. This operation center lacks an

effective way to efficiently analyze and visualize the collected data over a long period of time. Such a tool is essential to all decision-makers and can form the basis for meaningful future decisions. Questions that can be clarified are, for example: *What time and in which neighborhood occur most thefts? Which objects are stolen most frequently?* Furthermore, by analyzing the user feedback, it is possible to find the most important topics for inhabitants of a district and to adapt the political and strategic decisions accordingly.

1.2 Outline of Contribution

The goal of the bachelor thesis is the creation of an effective tool for the analysis and visualization of the data collected within the framework of the City.Risks project. The application is supposed to be able to enrich and index data with metadata, to examine the data with common analysis methods and to find suitable ways to visualize them and help users gain new insights. The application is designed to integrate seamlessly into the existing structures of the City.Risks project, be easy to use, and have an attractive user interface.
Most applications for indexing and navigating large data sets already have all the necessary tools to store, index, and search the edited records in a user-friendly way. The two largest open source search engines are Lucene and Sphinx. Lucene lays the foundation for two open source software packages that add additional functionality to the aforementioned search engine and are named Elasticsearch and Apache Solr.[6] Elasticsearch, Apache Solr and Sphinx are presented in more detail during the course of the work and the appropriate search engine is selected for implementing the backend of our Data Analysis Tool. The main part of this thesis deals with the development of a user-friendly interface. A user-centered design approach is chosen. It starts with the think-aloud method of a prototype and is then further improved with a heuristic evaluation combined with the action analysis for every iteration stage.

1.3 Structure of the Thesis

The first chapter gives the motivation for the bachelor thesis, the outline of contribution and the structure of the thesis. Chapter 2 deals with the Background and the context of the thesis. It gives a brief overview over the EU H2020 City.Risks project in which the thesis is integrated, gives definitions for some of the underlying elements and explains the related work that has been done in similar fields. Chapter 3 deals with the design and evaluation of the Data Analysis Tool (DAT). It looks at the requirements for the DAT and examines the provided data. This chapter also introduces the available software to realize the backend and selects the right one for the application. Furthermore Chapter 3 deals with the design and evaluation process of a user interface for the DAT. Therefore target groups, Personas derived from that target groups and user stories are created. Furthermore two interface iterations are created and subsequently evaluated with heuristic evaluation, action analysis and think-aloud evaluation. The final iteration is implemented. The important steps of that implementation as well as interesting code snippets are described in Chapter 4. Chapter 5 provides the summary. Chapter 6 provides the conclusions, the limitations as well as the outlook of the bachelor thesis.

Chapter 2

Background

This chapter provides an overview over the framework for the bachelor thesis as well as important definitions and related work.

2.1 Context of Work

This section gives a brief overview over the City.Risks project and introduces the operation center that contains the Data Analysis Tool.

2.1.1 EU H2020 City.Risks

The framework for this bachelor thesis is the City.Risks research and innovation action project funded by the European Union within the Horizon 2020 program. One of the goals of City.Risks is to investigate whether there is a relationship between perceived and actual anxiety as well as the perception of security in large cities. It is intended to use the applications of the City.Risks project to provide a secure space for residents and authorities to organize in groups and to communicate directly. Examples could be that an authority informs all residents in a particular area about an unforeseen event in the neighborhood or individual citizens communicate to the authorities where problems occur in their surrounding area. Furthermore, a sensor is developed, that can be attached to objects and then localized with the help of smartphones and other receivers.[7]

2.1.2 City.Risks Operation Center

As part of the City.Risks project an operation center is developed where the user generated and the sensor data is managed in a central application. Users send information about incidents in their surrounding area via a smartphone application to the operation center. The data from the sensors also arrives at the operation center and gets further processed. The operator of the operation center can issue alerts that are sent to citizens who use the City.Risks smartphone application and warn them about incidents in their area. The module which is developed within the bachelor thesis is intended to be used in this operation center.

2.2 Definitions

Design Pattern

According to the book *Design Patterns – Elements of Reusable Object- Oriented Software*[8] "[a] design pattern names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design." or in general "[a design pattern] describes the core of the solution to that problem."

Elasticsearch

"Elasticsearch is a distributed, scalable, real-time search and analytics engine."[9] In contrast to a regular database, the process of inserting new rows into a table is called *indexing* in Elasticsearch and what would be a new table row in a relational database is called a *document*.

Inverted Index

An inverted index saves the locations of occurrence for a term in a text collection similar to an index in a book. It is used in most of information retrieval systems like search machines.[10]

Open Source

In software development open source means that the source code of the software is freely available for everyone to look at and in most cases the software can be improved or new features can be added by other developers.[11]

REST

Representational State Transfer (REST) is "an architectural style consisting of [a] set of constraints applied to elements within the architecture".[12] These constraints are client-server architecture, stateless communication, ability to cache the response of a request for later use, uniform or standardized interface that simplifies the architecture, a layered system for security as well as a code-on-demand style that allows downloading and executing of code.[12]

User-Centered Design

The ISO 9241 part 210: Ergonomics of human-system interaction - Human-centered design for interactive systems defines user-centered as

> "approach to systems design and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques."

2.3 Related Work

The main focus of this work is how to efficiently visualize and search through large datasets to get new insights about the available data. Therefor this section gives an overview over the state of the art of search and visualization of data and an overview over a special form of visualizing data, geospatial and temporal visualizations.

2.3.1 Data Visualization

The biggest paradigm shifts in data analysis software according to Gartner, Inc. are from IT-centric complete solutions to smaller modular data-centric software solutions and the shift from simple system of record reports to visual-based exploration to gain insights.[5] The large integrated software

solutions from old market leading manufacturers such as SAP and IBM had to adapt themselves and their range of functions to this paradigm shift in recent years in order not to lose the connection to new emerging companies such as the current market leader in this segment Tableau or QLik. The biggest advantage of these data-driven software solutions is that they can be implemented much quicker, therefore achieve results significantly faster and can be adapted to requirement changes quickly. The next major paradigm shift into the world of data analysis software is the arrival of machine learning and the automation of knowledge extraction.[5]

2.3.2 Search

A perfect example of why it is necessary to search through large datasets is the Internet. Without search engines like Google Search, Microsoft Bing or DuckDuckGo the Internet would not be as accessible as it is today. One of the biggest trends in search at the moment is contextual search.[13] That means the user provides additional information, like a location, his favorite food, etc. and the search engine delivers results depending on the external criteria. Companies like Google that provide a variety of services for users then use the provided data (e.g.: via Google Mail) to suggest results based on information from e-mails like appointment dates or gives information about traffic jams on a regular way home based on tracked smartphone GPS signal. The second trend deals with the integration of search into IoT devices. Amazon's Alexa, Apple's Siri or Microsoft's Cortana allow users to search for answers without typing them into an input element, simply by talking to the device.[14] The third general trend is to go away from universal search, like Google Search, that delivers results to every topic to more specialized search engines, like amazon.com for buying goods or skyscanner.de for cheap flights.[15]

2.3.3 Geospatial and Temporal Visualization

The form of visualizing complex geospatial and temporal data is in almost all cases tailored to highly individual projects. Requirements, like the necessity to be free and open source or the integration into existing projects, are usually too complex to be solved by standard software like the data visualization tool Kibana that is part of the Elasticsearch technology stack. The mentioned

application provides the basic functionality to visualize aggregated datasets also on maps using its built-in functionalities, but in practice fails to incorporate it into existing projects in a satisfying way. It provides two ways to use visualizations, first to create a Kibana branded dashboard with all wanted visualizations or to use single visualizations and incorporate them into an application via an iFrame. None of both ways meet the requirements. Other data visualization software like tableau provides the same basic, very well made functionality but fails to adapt to highly individual and special use cases. A very interesting project in the field of geospatial and temporal data visualization is the GeoVISTA CrimeViz project. It is a "web-based mapping application supporting visual analytics of criminal activity in space and time". It was developed between 2013 and 2015 for the Harrisburg Bureau of Police.[16] The application operates on a special dataset for the American city of Columbia. Adapting the software to the City.Risks use case wouldn't be an easy task to solve. But the core functionalities of CrimeViz, like the map panel for visualizing crime scenes, a data layer panel to adjust the outcome and the temporal panel to visualize crime over time lays out a good starting point for developing the application for the City.Risks project.

2.3.4 Research Results

The biggest advantages of the software developed in this bachelor thesis are the seamless integration into the City.Risks operation center, because of its modular data-centric approach as well as the development of the user interface according to user requirements using user-centered design methods. Furthermore, no software license fees are charged and the code is going to be completely available for inspection so it can be adapted to future change requests.

Chapter 3

Design and Evaluation of the Data Analysis Tool

This chapter starts with the requirements for our software. The main parts of this chapter are the design and evaluation of the frontend user interface using the user-centered design method and the design of the backend. As part of the backend design process this chapter examines the available data used in the City.Risks project and what available software is on the market that supports the development of our application backend in a convenient way.

3.1 Requirement Analysis

This section deals with the question of what functional and non-functional requirements the software should have.

3.1.1 Functional Requirements

Multiuser Many users must be able to use the software at the same time.

Data storage The data must be collected and stored for a long period of time.

Searchable The software has to enable users to search through the dataset of incidents via keywords.

Visualization The software visualizes the available data in a way that a user can gain knowledge from it.

3.1.2 Non-functional Requirements

Performance The search and visualization tools should deliver the user desired goals in a quick way.

Usability The user needs to reach his goal in an effective way. It must be easily understandable for the user what is possible with the software, the functionality should be self explanatory and easy to learn.

3.2 Design of the Frontend

This section describes the user-centered design method and the different processes that are executed in order to evaluate design of the frontend.

3.2.1 Method: User-Centered Design

The design for the frontend is done by using an iterative user-centered design approach. The first part of this section deals with the research about the target group, the creation of user personas and user tasks that subsequently become the basis for the design and evaluation process. For the design and evaluation process the methods think-aloud, heuristic evaluation and action analysis were used. These methods are used two times to generate findings in every iteration and lead into the creation of a new and improved prototype for every iteration.

Think-Aloud

The think-aloud method is one qualitative user-centered design method that helps finding inaccuracies and errors of a user interface by solving user stories and communicating every step and decision the user made.[17]

Heuristic Evaluation

Heuristic evaluation can be "quantitative, which uses statistical functions and countable techniques, and qualitative, using techniques such as those

that involves the experience and knowledge of experts or those belonging to knowledge discovery data (KDD) discipline (by example)."[18] For the evaluation of the user interface we focus on the qualitative approach to evaluate the user interface. For this approach the user feedback is not considered and the improvements are based on different design heuristics and principles. We used the list of heuristics from Jakob Nielsen[19] as the main source for deriving improvements for the next iteration.

Action Analysis

The action analysis is a quantitative approach "[b]ased on categorized user-program interactions"[20] like time until a task is completed, the number of clicks necessary to solve certain tasks or the number of errors that happen while fulfilling a task.

3.2.2 Target Group

In order to efficiently work with the user-centered design method it is necessary to define a target group for the application. This helps to better understand the user needs and to have a clear idea of who is the audience that uses the software. The City.Risks operation center is used initially by the city administration of three major European cities: London, Rome and Sofia. The operation center is operated by "Agents" that are supervised by a "Supervisor". In the European Union it is more likely that the Agent or assistant in civil service is female (65%) and the supervisor or in general an administrator is male (62%).[21] From this we derive two personas, Sofia Rossi (Figure 3.1) a female "Agent" and Robert Lewis (Figure 3.2), a male "Supervisor".

3.2.3 Personas

The personas are based on two of the 12 archetypes from Carol S. Pearson's book *Awakening the Heroes Within: Twelve Archetypes to Help Us Find Ourselves and Transform Our World*. [22] While Sofia Rossi represents *the Orphan*, Robert Lewis represents *the Ruler*.

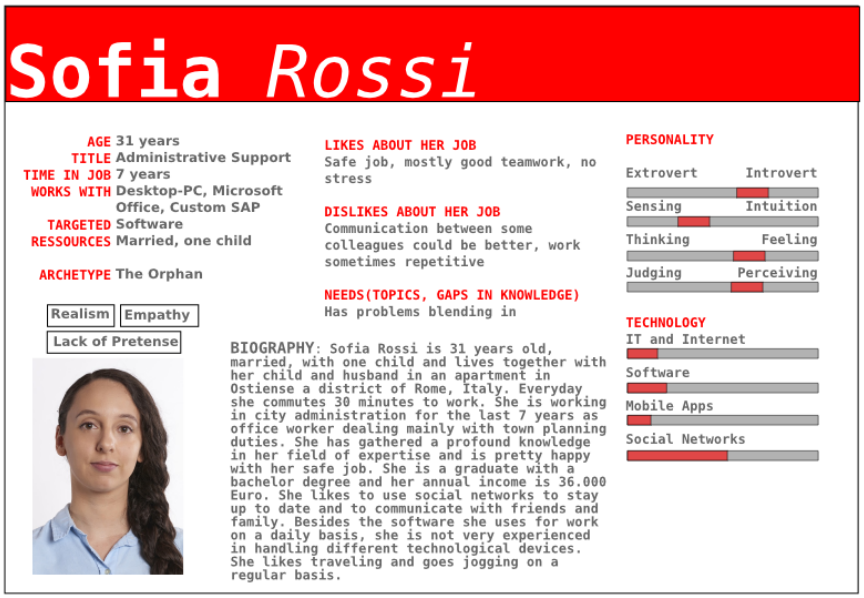


Figure 3.1: Persona Sofia Rossi[1]

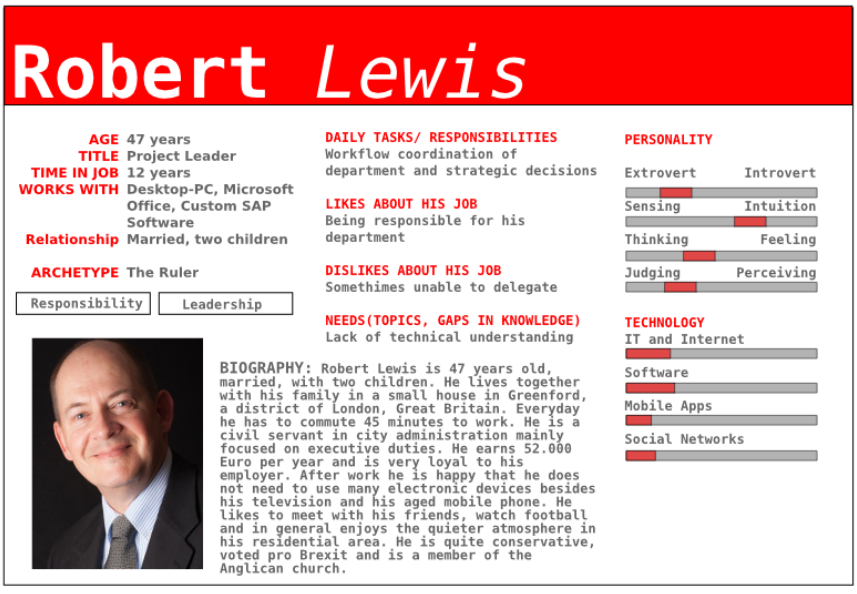


Figure 3.2: Persona Robert Lewis[2]

"Agent" Sofia Rossi Sofia Rossi is 31 years old, married, with one child and lives together with her child and husband in an apartment in Ostiense a district of Rome, Italy. Everyday she commutes 30 minutes to work. She is working in city administration for the last 7 years as office worker dealing mainly with town planning duties. She has gathered a profound knowledge in her field of expertise and is pretty happy with her safe job. She is a graduate with a bachelor degree and her annual income is 36.000 Euro. She likes to use social networks to stay up to date and to communicate with friends and family. Besides the software she uses for work on a daily basis, she is not very experienced in handling different technological devices. She likes traveling and goes jogging on a regular basis.

"Supervisor" Robert Lewis Robert Lewis is 47 years old, married, with two children. He lives together with his family in a small house in Greenford, a district of London, Great Britain. Everyday he has to commute 45 minutes to work. He is a civil servant in city administration mainly focused on executive duties. He earns 52.000 Euro per year and is very loyal to his employer. After work he is happy that he does not need to use many electronic devices besides his television and his aged mobile phone. He likes to meet with his friends, watch football and in general enjoys the quieter atmosphere in his residential area. He is quite conservative, voted pro Brexit and is a member of the Anglican church.

3.2.4 User Stories

User stories wrap the requirements for a product in an easy to understand and clear way that can be communicated with other stakeholders.[23]

"As an an official in charge for town planning, Sofia Rossi wants to find new insights fast and easy, like in which town district security needs to be improved, so that she don't has to waste a lot of time on researching."

"As a civil servant in city administration, to gain new insights, Robert Lewis

[1]Image Credit: https://pixabay.com/en/woman-serious-bored-satisfied-2385789/, accessed 31.07.2017

[2]Image Credit: https://pixabay.com/en/business-man-suit-office-1032839/, accessed 31.07.2017

doesn't want to use complicated software or work through enormous amount of un- or semistructured data, like excel sheets."

3.2.5 First Iteration

We used the mockup tool Balsamiq[3] for both iterations. Figures 3.3 and 3.4 show two separate views for visualization and search as well as errors that were found during the evaluation of both views. Higher quality images of the two views can be found in appendix A.1 and A.2.

Think-Aloud Method

For the first iteration four test subjects were asked to solve following tasks using the provided user interface:

1) Look at the page and describe what you see
2) Show all thefts in April without datepicker
3) Find the day with most riots
4) Search for all incidents, then find the theft entry with the description "my blue car got stolen" in the table and click it (describe what you see)
5) Find all days with thefts
6) Find all thefts that happened after 15.05.2017
7) Find all incidents with keyword "stolen"
8) Find all incidents between 15.05.2017 and 20.05.2017

8b) Reset search

The transcribed interviews can be found in appendix C. The key findings for the first round of evaluation via think-aloud method were: Users 1, 2 and 4 stated that both views should be combined (Figures 3.3 and 3.4, No. 1). The users made clear that the input area is too similar in both views and it is of high interest to search for a keyword and visualize the results.

[3]https://balsamiq.com/, accessed 19.07.2017

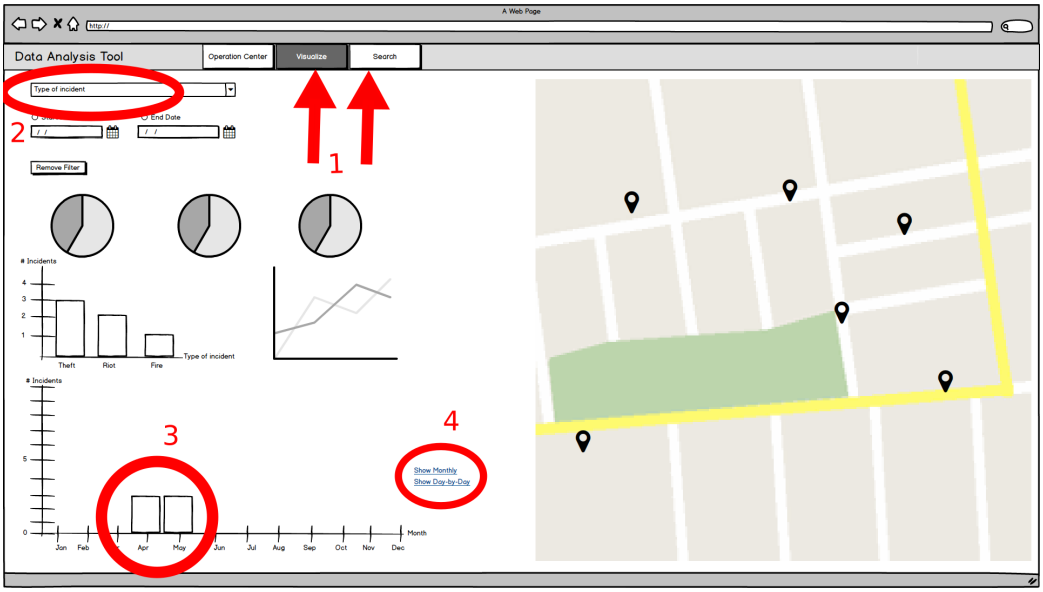


Figure 3.3: Mockup for first Iteration of Visualize Tab

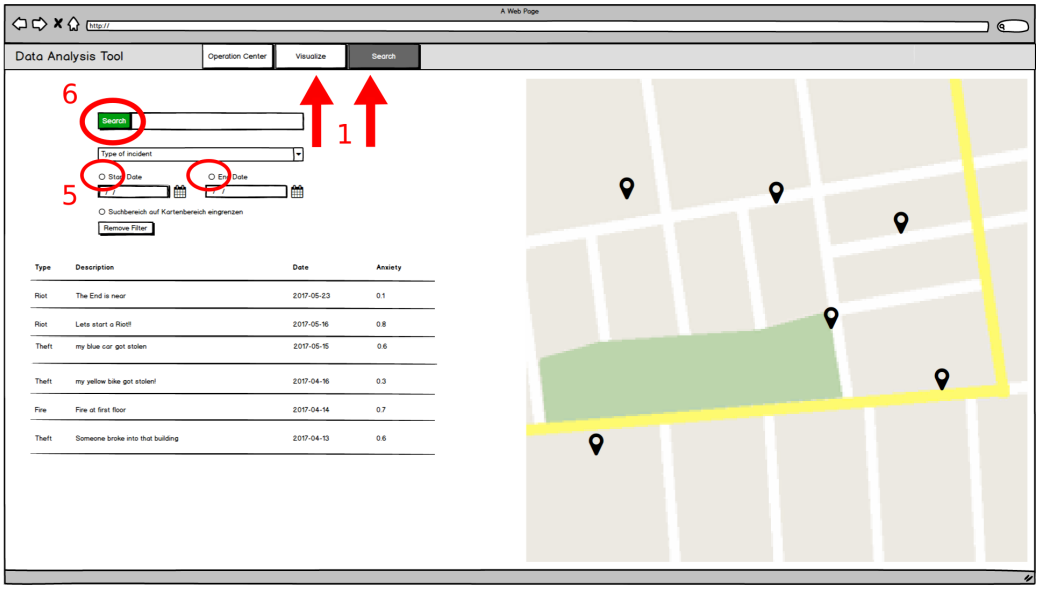


Figure 3.4: Mockup for first Iteration of Search Tab

User 1 and 4 stated that it would be less confusing if the interface allowed to prepare a statement and click a button that then fires the query to receive a result and not fire a query every time the user changes an element.
Users 1 and 2 pressed a button to see all incidents for a month ("show monthly") but it changed nothing in the view.
For user 1 it was not plausible that the charts were interactive and could be used to narrow down results (Figure 3.3, No. 3).
Users 1 and 3 had problems finding the buttons to switch between monthly and day-by-day visualization of the incidents (Figure 3.3, No. 4).
The need to activate or deactivate the date picker was a problem for users 2, 3 and 4 (Figure 3.4, No. 5).

Heuristic Evaluation

The heuristics used for evaluating the user interface according to Nielsen [19] are:

- **Visibility of system status**: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **Match between system and the real world**: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- **User control and freedom**: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **Consistency and standards**: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- **Error prevention**: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

- **Recognition rather than recall**: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

- **Flexibility and efficiency of use**: Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

- **Aesthetic and minimalist design**: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

- **Help users recognize, diagnose, and recover from errors**: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

- **Help and documentation**: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

After working through the usability heuristics for evaluating user interface design the following errors of the first iteration have been found:

Search Tab The input field "Type of incident" in both views is missing an "All incidents" feature (Figure 3.3, No. 2).
After the user has chosen a type of incident it is not clear anymore that the input field is about selecting the type of incident. At this point the user interface is not providing enough feedback to the user.
The distance of the "show monthly" and "show day-by-day" buttons and the chart they belong to is too wide (Figure 3.3, No. 4). Therefore the elements don't relate to each other. There is no visual feedback which one of the two buttons is active.
It is not obvious that the center left and bottom left charts are interactive

and different bars can be clicked to narrow down results. The elements that are used to narrow down results, like the type of incident dropdown or the date picker don't provide enough feedback about their status to the user. The "Remove Filter" button is not clearly marked as an escape. The function of the radio button next to the start and end date picker is not explained and inconsistent to the rest of the input fields.

Visualize Tab The search button is the only element that needs to be clicked in order to receive results (Figure 3.4, No. 6). All other elements that are used to narrow down results change something in the user interface immediately. This is clearly inconsistent to the other elements.
The "Remove Filter" button is not clearly marked as an escape 3.4, No. 7). It is not possible for the user to search for keywords and than see the results visualized in the same view.

Action Analysis

Table 3.1 shows the tasks in the first column and the minimum amount of clicks for solving each task is displayed in seconds. The third column provides the amount of time a test subject spent who had to solve the task and saw the user interface for the first time. The fourth column shows the time necessary for a test subject that had a significant amount of time to get comfortable with the user interface.

Key Findings

To avoid confusion and prevent users from getting lost in the two views, the two different tabs for visualization and search are combined into one view. This further promotes the heuristic of a minimalistic design because duplicate elements are avoided.
The interface elements are grouped on one side into interactive areas where the user can modify different search and visualize criteria and on the other side into areas that are only used for visualizing results. They are separated visually and named accordingly to prevent confusion.
According to Nielsens 10th heuristic[19] it is a good practice to provide help and documentation. When reviewing the user interface some elements don't provide enough information and there functionality does not speak for themselves. Therefore more help functionality needs to be implemented to guide

Tasks	#Clicks (minimum)	Time [sec] (untrained)	Time [sec] (trained)
Show all thefts in april without datepicker	3	24,94	9,67
Find the day with most riots	3	24,36	10,08
Find all incidents between 15.05.2017 and 20.05.2017	5	29,79	12,26
Search for all incidents, then find the theft entry with the description “my blue car got stolen” in the table and click it	2	22,52	8,69
Find all days with thefts	3	27,71	9,29
Find all thefts that happened after 15.05.2017	5	26,31	14,33
Find all incidents with keyword “stolen”	3 +keyword	30,08	12,85

Table 3.1: Action Analysis for first Iteration

and help users when they are in doubt. The visual feedback of what is a button was insufficient. And it needs to be consistent whether certain tasks have to be triggered by a button or are automatically performed.

3.2.6 Second Iteration

The starting point for the second iteration is the mockup shown in figure 3.5. A high resolution version can be found in appendix A.3. It shows the combined visualize and search tab and and now also includes some help functionality, like sub lines for components or buttons with question marks that provide additional information to support the user. It also shows the marked errors that were found in the second iteration.

Think-Aloud Method

For the second iteration five test subjects (users 5 - 9) were asked to perform the same tasks as in the first iteration. User 5 stated that she feels distracted by the empty bottom left chart that shows nothing on initial startup (3.5,

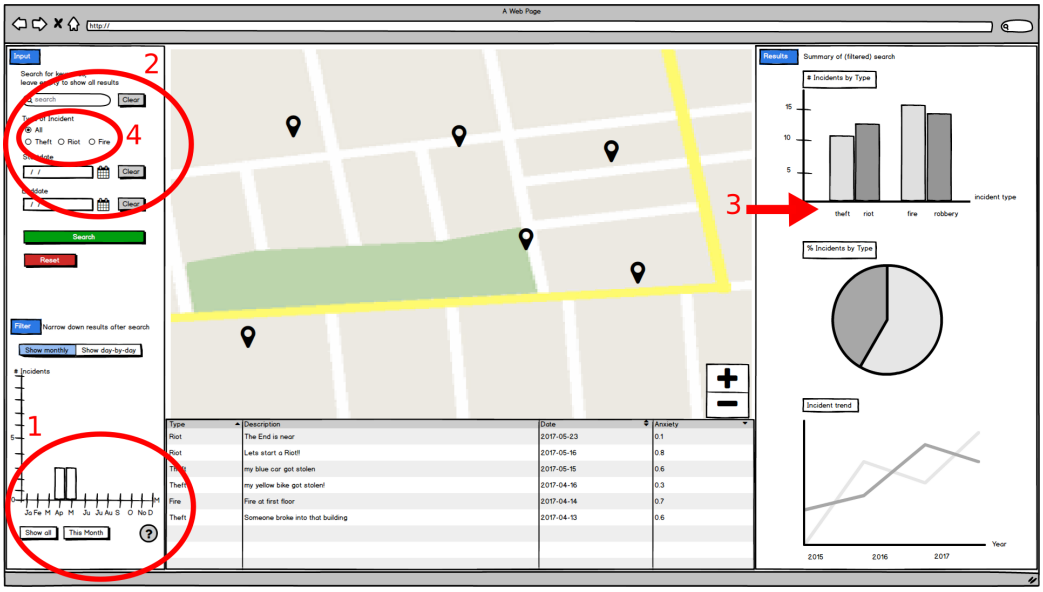


Figure 3.5: Second Iteration Mockup

No. 1). She was so much distracted by empty components that she did not see the search button. User 5 also stated that the order of the input fields is misleading (3.5, No. 2). If you can specify the type of incident before a search and also get results without using the keyword search, it may be good to put the types of incident input field at the first position of the input component. This way you get an overview of all results for a certain type of incidents and can narrow down the results with the search for keywords and date range later. User 6 agreed to user 5's observation and stated that it is uncommon to apply a type of incident "filter" without searching for keywords. User 7 stated that he expects to see only information regarding his search query, especially when he used the filter by "type of incident". Information that is not relevant to the search should be omitted (3.5, No. 3). A general user trend was that they were missing extra information on what exactly they are seeing and what happened.

Heuristic Evaluation

Due to the fact that the types of incidents are a free form field in the application that produces the incident data, it is cumbersome to use radio buttons to chose the type of incident (3.5, No. 4). The amount of types of

incidents could exceed 30, which would make it look cluttered. Therefore a drop-down menu is chosen to provide the incident types. The user interface provides empty charts before the user issues a search. This clearly violates the principle of aesthetic and minimalistic design.

Action Analysis

Tasks	#Clicks (minimum)	Time [sec] (untrained)	Time [sec] (trained)
Show all thefts in april without datepicker	3	15,62	4,88
Find the day with most riots	3	14,38	5,87
Find all incidents between 15.05.2017 and 20.05.2017	5	18,68	9,75
Search for all incidents, then find the theft entry with the description “my blue car got stolen” in the table and click it	2	12,69	4,82
Find all days with thefts	3	11,43	5,02
Find all thefts that happened after 15.05.2107	5	19,02	5,60
Find all incidents with keyword “stolen”	3 +keyword	14,82	7,53

Table 3.2: Action Analysis of second Iteration

The table 3.2 shows a significant improvement of the amount of time that was necessary to fulfill the tasks for trained and untrained test subjects compared to the action analysis of the first iteration.

Key Findings

To skip the two different views and combine them into one was a major improvement for the test subjects that had also tested the first iteration of the interface.
The most significant speedup from the first iteration to the second was that

users did not had to switch between the two separate views. Also the structure of the search input fields on the left from top to bottom provided a clearer structure to achieve the user intended goals. The major key finding of the second iteration was that users expected to have hidden components if those were not relevant for the moment or didn't provide any additional content. User guidance and design principles remain a high priority for the user interface.

3.3 Design of the Backend

The backend consists of a search engine that deals with indexing and storing the data from the operation center. It is accessible through an API and delivers user specified search results in a fast and convenient way. The following section deals with picking the appropriate search engine for this task.

3.3.1 Incident Data

The data that is indexed and visualized is derived from the operation center of the City.Risks project and consists in the form of incidents. The significant characteristics of an incident are a unique ID, a single location, different types, reports, alerts and sightings. It is not necessary for an incident to have a location but it can also have an area provided via a polygon instead of a single location. reports are issued by citizens who want to report anything regarding a specific incident and alerts are issued by the operators of the City.Risks operation center to warn users in a certain area. Sightings are automatically recorded with the help of the bluetooth broadcast device that is developed during the City.Risks project and receivers like smartphones or telephone provider antennas. Every incident can have many reports, alerts and sightings. Most important parts of every report and sighting are a timestamp and a location. This can help to identify routes or the current position of a lost or stolen object. An example incident can be seen in listing 3.1.

Listing 3.1: Sample Incident (JSON)

```
{   "@class": "Incident",
    "id": "0ecee0c5-1df4-440a-bfd6-53645f17f557",
    "state": "open",
    "location": {
      "type": "Point",
      "coordinates": [13.41, 52.52, 0]
    },
    "types": [{
        "type": "theft",
        "confidence": 1
      }],
    "reports": [{
        "@class": "Report",
        "id": "0955e576-54df-4304-833f-199629b9e6bd",
        "src": {
          "@class": "TheftReport",
          "id": "0955e576-54df-4304-833f-199629b9e6bd",
          "description": "black/green Centurion Cyclo Cross 4000",
          "time": 0,
          "incident": "0ecee0c5-1df4-440a-bfd6-53645f17f557",
          "location": {
            "type": "Point",
            "coordinates": [13.41, 52.52, 0]
          },
          "address": "7E:23:B1:56:2F:B9",
          "lastTimeSeen": 1483618580,
          "state": "active",
          "created": 1483618580
        },
        "incident": "0ecee0c5-1df4-440a-bfd6-53645f17f557",
        "state": "restricted"
      }],
    "sightings": [{
        "@class": "NormalizedSighting",
        "id": "7E:23:B1:56:2F:B9",
        "time": 1464103770,
        "signal": 64,
        "location": {
          "type": "Feature",
          "properties": {
            "precision": 64,
            "timestamp": 1464103770
          },
          "geometry": {
            "type": "Point",
            "coordinates": [13.42, 52.52]
          }
        }
      }],
    "token": "7E:23:B1:56:2F:B9",
    "flags": 2,
    "theft": true }
```

3.3.2 Technology Analysis

Available Technology

Lucene Lucene is an open source search engine library developed in Java.[24] It indexes documents, allowing it to search them effectively for keywords. Extensions like Elastichsearch or Solr provide additional functionality such as a server, a database or a REST interface for easy integration into other projects.

Elasticsearch and Kibana Elasticsearch, provides a server with REST interface, works distributed, stores the incoming data in a NoSQL database, and uses Apache Lucene to index the data [25]. Elasticsearch also has a whole host of other products that simplify the creation of clear dashboards with visualizations or help storing the data in the database. The so-called *Elastic Stack* consists at least of Elasticsearch, Logstash and Kibana. Logstash is a tool which is used to receive different data formats from different sources, to convert them into an Elasticsearch-readable format (JSON) and send them to Elasticsearch. Kibana provides a browser-based graphical user interface for Elasticsearch that can be used to easily create and publish dashboards with different visualizations.

Apache Solr Apache Solr is quite similar to Elasticsearch. It is an open source search server on top of the Apache Lucene search library and provides storage for the data as well as a REST-like API for easy searching through the documents.[26]

Sphinx Sphinx is also an open source search engine with REST-ful API, data storage and server like Elasticearch or Solr but relies on its own indexing library and not on Lucene.

Results of Evaluation

Criteria	Sphinx	Apache Solr	Elasticsearch
Open Source	yes	yes	yes
Github contributors	9	65	857
Github branches	3	67	101
Github last commit	6 days ago	an hour ago	14 minutes ago
Github stars	1186	975	23,942
Stackoverflow Q&A	12,165	39,156	59,101
angular2 compatible	yes	yes	yes
license	GPLv2	Apache License 2.0	Apache License 2.0

Table 3.3: Comparison of different Search Engines

The table 3.3 provides a comparison of the three most used open source search engines. All numbers for GitHub and Stackoverflow have been accessed on 23. May 2017. As a result of the comparison Elasticsearch is used for the project. It has the most questions and answers on stackoverflow.com, an important source for developers, and the most stars on github.com, where the code of the project is shared. You can derive from the numbers that it has the biggest or at least most active userbase and it is therefore more likely that questions that occur during development have already been answered by others or are answered faster compared to the other search engines. All of the above mentioned examples are important facts for maintainability and being able to improve the code base of the project.

Chapter 4

Implementation of the Data Analysis Tool

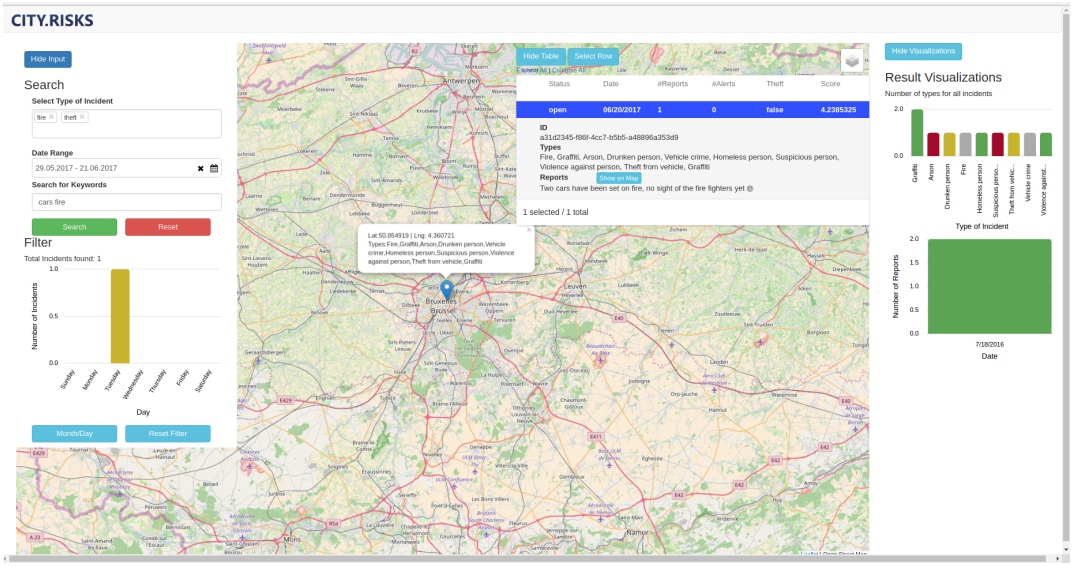


Figure 4.1: Implemented Data Analysis Tool

This chapter starts with describing the underlying design pattern that was chosen for implementing the Data Analysis Tool. The rest of this chapter is separated into two main parts, the implementation of the frontend in the first half and the implementation of the backend in the second half.

The code for this implementation can be found on the attached DVD or under https://github.com/frometor/oc-dat.
Throughout this chapter examples are provided to better understand the application. In contrast to the user-centered design process of the user interface in chapter 3, this section describes the general software architecture and the applied design pattern used for implementing the Data Analysis Tool.

4.1 Architectural Pattern

To deal with the asynchronous nature of a user input, the need of individual components to communicate with each other as well as reacting to data changes, the publish-subscribe architectural paradigm is used. In a publish/ subscribe architecture "[s]ubscribers have the ability to express their interest in an event, or a pattern of events, and are subsequently notified of any event, generated by a publisher, which matches their registered interest".[27]

4.2 Design Pattern

The design pattern used for implementing the frontend is the observer pattern. It is a behavioral model "that defines and maintains a dependency between objects"[8]. It is widely used in frontend development for applications with different components exchanging data. The basic idea behind the observer pattern is that components, called observers, depending on the same data, subscribe to the same source called subject or observable and get notified whenever the data changes. None of the components knows how many components have subscribed to the same source. Only the observable, that handles the data, knows that it has to communicate changes to an array of observers.

4.3 Implementation of the Frontend

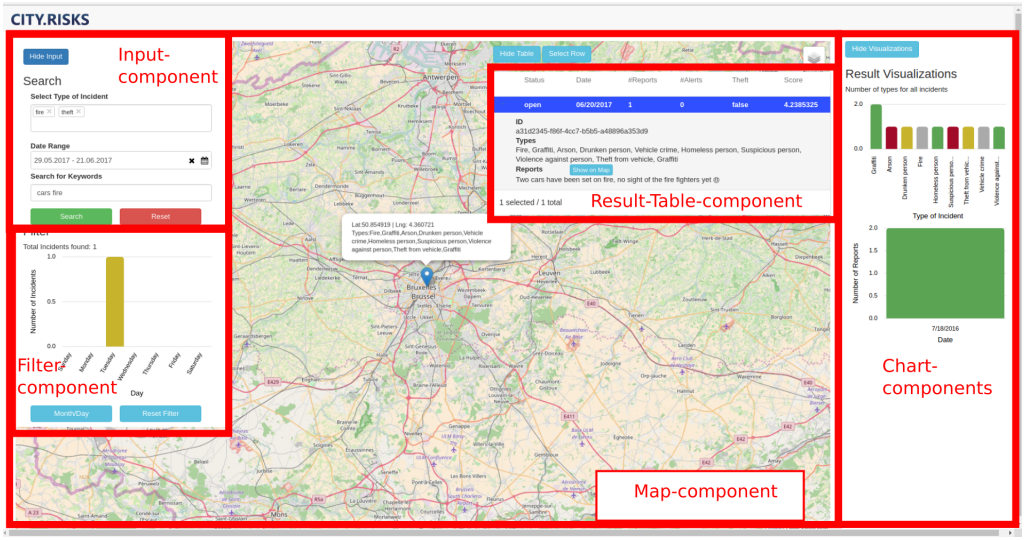


Figure 4.2: Implemented Data Analysis Tool with Components

The following example is given to show the different steps from inserting the data until receiving a result.

Example 4.1 *"A user wants to know about incidents of type "fire" and "theft" with the keywords "cars" and "fire" that happened between 29.05.2017 and 21.06.2017."*

Figure 4.2 shows the implemented version of the Data Analysis Tool with the input values from example 4.1 inserted into the input component. A high resolution version can be found in appendix A.4. The search query leads to a single result incident that is visualized as a marker on the map component, can be found on the filter component as well as in the result-table component. If the filter component has more than one incident, it is possible to filter the incidents by a single day or by a single month simply by clicking the corresponding chart. The frontend consists of many components that are responsible for different tasks. The biggest components are the input component, the map component, the data-table component and different chart components and are marked with red rectangles in figure 4.2. Each of

these components consist of smaller components that provide a functionality inside their respective main component. Figure 4.3 provides a top-level overview over the communication of the different components and consists of five steps:

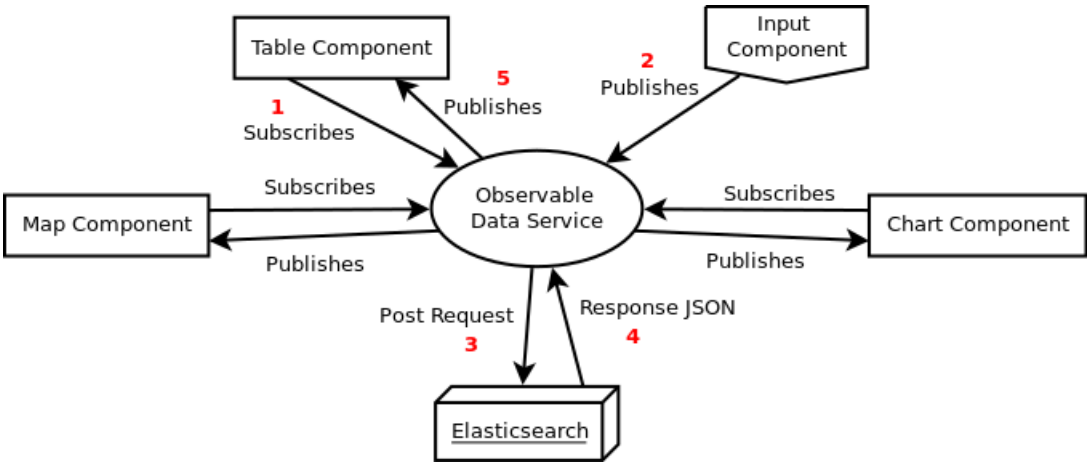


Figure 4.3: Observable Data Service

First, all components, that need data to work with, subscribe to the observable data service.
Second, the input-component publishes the search criteria from the user to the observable data service. The input component is where the user can type keywords, specify a date range or choose incidents by type. For our example the user specifies the keywords "cars" and "fire" in the keyword input field, uses the date range picker to specify the date range and also chooses the types of incident "fire" and "theft" from the type of incident field.
Third, all the provided data is used to form an Elasticsearch query and sent to the Elasticsearch backend. The backend part is discussed in detail in Section 4.4.
Fourth, the result of this post request is sent back to the observable data service via a response JSON object that contains all matching incidents.
Fifths, the data is published to all the components that subscribed to the observable.

Listing 4.1: Result-table Component Initialization

```
1  this.incidentService.incidents$.subscribe(
2      newIncidents => {
3        this.allIncidents = newIncidents;
4        this.fillColums(incidents,{"name":"all"});
5      }
6    );
```

Listing 4.1 shows a code snippet for one component, the result-table component. On initialization the component subscribes to the data service observable and whenever the observable emits a new value, the result-table component sets the local variable allIncidents (line 3) and calls the function fillColumns (line 4) to redraw the table with the new values.

Listing 4.2: Observable Data Service

```
1 // private subject = new BehaviorSubject(this.EMPTY_SEARCH);
2 // incidents$: Observable<any> = this.subject.asObservable();
3
4 getIncidents(payload: any): Observable<any> {
5 [..]
6 return this.http.post(this.url, postDataType, headers)
7       .map(res => res.json())
8       .do(newIncidents => this.subject.next(newIncidents));
9 }
```

In listing 4.2 is shown what happens inside the observable data service. Whenever the function getIncidents() gets called it receives a payload that is used to form the postDataType variable. In our example the postDataType variable contains information about the keywords, the date range and the types of incident the user specified. This data gets sent to an URL, in this case the Elasticsearch endpoint (line 6). If the post request was successful, the returning values get mapped to JSON (line 7) and the observable emits a new value to all subscribers including the result-table component (line 8).

Example 4.2 "*The user clicks on one of the markers on the map component and expects the associated row in the table to be highlighted.*"

The observable data service provides yet another feature which is the communication between two elements. This helps to communicate user input in one component to other components that need to react to this input and

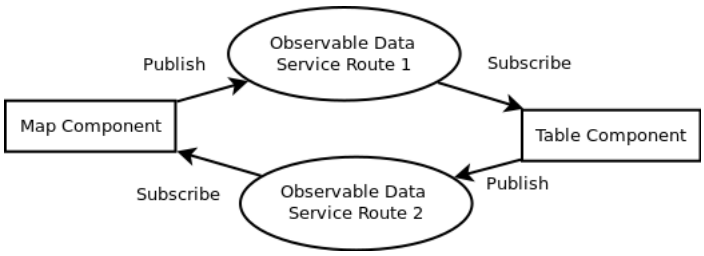


Figure 4.4: Communication between two Components

therefore providing a coherent user experience. We use example 4.3.2 to clarify this form of communication. For example the map and the result-table component communicate with each other via defined routes of the observable data service, shown in Figure 4.4. This form of communication simply delivers the information, which of the incidents is selected, to the other element.

Listing 4.3: Communication between two Components

```
1  //map component
2   this.markerLayerGroup.on("click", function (event) {
3   let clickedMarker = event.layer;
4   self.incidentService.sendClickedMarkerFromMap2Table(clickedMarker);
5  }
6
7  //result table component
8   this.incidentService.getClickedMarkerFromMap2Table().subscribe(
        clickedMarker => {
9         this.clickedMarker = clickedMarker;
10        highlightTableRow(this.clickedMarker)
11      });
12
13 //observable data service
14 sendClickedMarkerFromMap2Table(message: any): void {
15      this.subjectfromMap2Table.next(message);
16    }
17
18 getClickedMarkerFromMap2Table(): Observable<any> {
19      return this.subjectfromTable2Map.asObservable();
20    }
```

The code for the communication between two objects can be found in listing 4.3. It shows the important methods for the map and the result-table component as well as the corresponding methods from the observable data service. For the example we provide only one direction of communication from map to table component.

For this component-to-component communication we decided to implement routes for every direction. In theory, a component could use the observable to emit a new value and at the same time subscribe to this observable to receive changes from other components. This requires additional handling of every received message in every component and is more prone to errors. This whole process could be avoided by using separate routes.
In the provided code snippet shown in listing 4.3 the map component registers an event handler on the markerLayerGroup array which consists of all markers representing the different incidents (line 2). The map component calls the data service observable function sendClickedMarkerFromMap2Table() which takes the necessary information from the marker, if it has been clicked by the user (line 4). Inside the data service observable the called function emits a new value namely the clickedMarker that came from the map component (line 15). If the result-table component subscribed to the observable (line 8), it receives a new message from the observable, sets the local variable this.clickedMarker (line 9) and calls the function this.highlightTableRow() to highlight the corresponding row in the table (line 10).

4.3.1 Frontend Framework

For the implementation of the frontend the newest version of angular (4+) was used. To create new angular components and services we used the command line interface (angular-cli) tool.[1] The main benefits of using the angular-cli are the speed up of creating template structures for new components and services, providing a development server for testing the application and automatically compiling the component based code structure into one coherent code base whenever code changes happen. The main reason for using angular was that it uses the observable pattern extensively and the use of observables is supported out of the box with the RxJS library, that is part of the core libraries of angular. RxJS is a support library used to achieve the reactive programming paradigm, a way of implementing methods that makes it easier to deal with asynchronous data streams.

[1] https://github.com/angular/angular-cli, accessed 27.07.2017

As already mentioned angular has a component based architecture and every component consists of a pseudo Model-View-Controller Architecture. A single component has a view representation, a component class that can be considered a controller and services and directives that are used for communicating with a model.[28]

4.3.2 Plugins, Libraries, Frameworks

For speeding up the implementation several plugins, libraries and smaller frameworks for common use cases were used. These support applications were chosen by the quality of their documentation, the availability of working examples and an active community represented by the amount of github stars and questions on stack overflow. To install all components we used the node package manager (NPM).[2] NPM allows it to install new plugins simply by adding them to the list of dependencies in the package.json file and running *npm install* to install them. For the input component, we used the mydaterangepicker[3] and the ng-select2[4] plugins. For the map component the leaflet[5] library and the angular 2+ packages from asymmetrik[6] were used. For the datatable component we used the ngx-datatable[7] framework and for the different charts the ngx-charts[8] framework was used. To make the whole application responsive, we used the bootstrap[9] framework.

[2]https://www.npmjs.com/, accessed 28.07.2017
[3]https://github.com/kekeh/mydaterangepicker, accessed: 15.07.2017
[4]https://github.com/valor-software/ng2-select, accessed: 15.07.2017
[5]http://leafletjs.com, accessed: 15.07.2017
[6]https://github.com/Asymmetrik/angular2-leaflet, accessed: 15.07.2017
[7]https://github.com/swimlane/ngx-datatable, accessed: 15.07.2017
[8]https://github.com/swimlane/ngx-charts, accessed: 15.07.2017
[9]http://getbootstrap.com, accessed: 15.07.2017

4.4 Implementation of the Backend

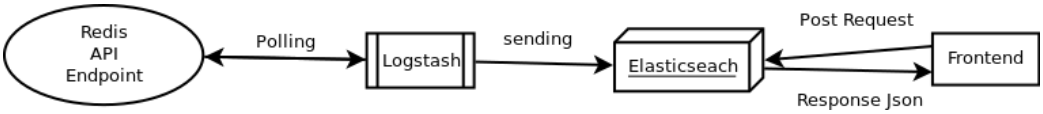


Figure 4.5: Backend Architecture

Figure 4.5 gives an overview over the backend architecture. The starting point for the application is an API endpoint that provides an array of incidents from the City.Risks operation center in JSON notation. The array consists of incidents that haven't been resolved, indicated by the *"status":"open"* value or have been resolved in the last 24 hours. An instance of Logtash[10] is used to pull the array from the API endpoint. This polling is achieved via the http-poller[11] input plugin for Logstash. The configuration file for Logstash can be found in listing 4.4. The main benefits from using Logstash is that it takes the data from any source, transforms it so Elasticsearch can understand it and feeds it to the Elasticsearch instance. If Elasticsearch is inaccessible due to heavy load from other sources or outtakes, Logstash automatically retries to connect to the Elasticsearch instance after a customizable amount of time. The "output" part (line 21 -28) of listing 4.4 deals with updating incidents that are already indexed and the creation of new documents if the incident is not yet indexed. The pipeline takes care of indexing each Elasticsearch document (document ID) with the primary "id"-key from each incident document.

[10]https://www.elastic.co/de/products/logstash, accessed: 15.07.2017
[11]https://github.com/logstash-plugins/logstash-input-http_poller, accessed: 15.07.2017

Listing 4.4: Logstash Configuration

```
input {
  http_poller {
    urls => {
      ocendpoint => {
        method => get
        url => "http://url.to/api/enpoint"
        headers => {
          Accept => "application/json"
          auth=> "<password>"
        }
     }
    }
    request_timeout => 60
     schedule => { cron => "* * * * *"}
     codec => "json"
  }
}

[..]

output {
  elasticsearch {
    index => "incidents"
    document_type => "incident"
    pipeline => "rename_id"
    document_id => "%{id}"
    doc_as_upsert => true
    action => "update"
  }
}
```

The Elasticsearch document structure needs to be manually mapped before indexing new documents. The domain-specific language (DSL) query for this purpose can be found in appendix B. It is considered best practice to map the structure before new documents get indexed to avoid inconsistencies or unwanted behavior when using the build in mapping functionality. The main reason for not using the automatic mapping for indexing the incident documents in our case is that a single incident can contain arrays of reports or alerts. These arrays are automatically flattened by Elasticsearch to simple key value pairs to speed up the search process later on.

Listing 4.5: Flatten Objects in Nested Array in Elasticsearch

```
//This:
{
    "followers": [
        { "age": 35, "name": "Mary White"},
        { "age": 26, "name": "Alex Jones"},
        { "age": 19, "name": "Lisa Smith"}
    ]
}

//Will be flattened to:
{
    "followers.age":    [19, 26, 35],
    "followers.name":   [alex, jones, lisa, smith, mary, white]
}
```

The Elasticsearch documentation provides an example for this flattening process and can be found in listing 4.5[12]. In the provided example the correlation between the follower names and their ages is lost due to this flattening. The same would happen to reports and alerts of our incidents. If we would only search for incidents, it would be bearable to flatten the reports and alerts. But in order to present each of the alerts and reports separately, we need to preserve the correlated information between single reports and alerts. This can be achieved by providing the `"type":"nested"` option for reports and alerts while manually mapping the incidents.

The Elasticsearch search engine communicates with its clients via an API post request. The body of the request sent to the Elasticsearch instance contains the DSL query. Listing 4.6 shows the query with keywords, type of incidents and date range from our example. The array of returned documents receive a score how well they match the given query expression. This is mainly achieved via the *practical scoring function* that consists of a boolean model, a calculated term frequency/ inverse document frequency (TF/ IDF) and a vector space model, explained online in more detail on *Elasticsearch: The Definitive Guide*, section *Theory Behind Relevance Scoring*.[29]

[12] https://www.elastic.co/guide/en/elasticsearch/guide/current/complex-core-fields.html#object-arrays, accessed: 31.07.2017

Listing 4.6: Elasticsearch DSL Query Example

```
{
  "query": {
    "bool": {
      "filter": {
        "bool": {
          "must": [
            {
              "nested": {
                "path": "reports",
                "query": {
                  "bool": {
                    "must": [
                      {
                        "range": {
                          "reports.src.created": {
                            "gt": 1496008800,
                            "lte": 1497996000
                          }}}]}}},
            {
              "nested": {
                "path": "types",
                "query": {
                  "bool": {
                    "must": [
                      {
                        "match": {
                          "types.type": "fire"
                        }}]}}},
            {
              "nested": {
                "path": "types",
                "query": {
                  "bool": {
                    "must": [
                      {
                        "match": {
                          "types.type": "theft"
                        }}]}}}]}},
      "must": {
        "query_string": {
          "query": "cars~ fire~"
        }}}}
}
```

The query operator "bool" (line 3) in listing 4.6 indicates that the query is evaluated as a boolean expression. It further recalls this boolean model with the use of keywords "must" and "should" that act as boolean $\vee$ (OR) and $\wedge$ (AND) operators in the query. The "query_string" operator evaluates the two keywords "cars~" and "fire~" as a boolean OR clause. The "~"(tilde) at the end of each keyword adds fuzziness to the keyword search via the Levenshtein edit distance. This way the DSL query in listing 4.6 who gets sent to Elasticsearch evaluates to this boolean expression:

$$searchresults = (keyword1 \vee .. \vee keywordX) \wedge types \wedge daterange$$

The *term frequency/inverse document frequency* consists of three factors that are taken into account for scoring a keyword:

- **term frequency**: how many times the keyword appears in the specified field of the document
- **inverse document frequency**: how many times the keyword appears in all documents that have been found
- **field-length norm**: If the keyword has been found deep inside the document, the document receives a lower score than a document where the keyword was found in a first level key value pair.

The *vector space model* transforms documents and queries into vectors. The keywords receive a weight according to their score in the TF/IDF. Next the documents and the search query are plotted as vectors in an n-dimensional graph, where n equals the amount of keywords. It is easy to calculate the distance from each document to the given query and rank them according to distance. The array of documents sorted by relevance gets sent back to the user and can be used to visualize the results in the frontend.[29]

Chapter 5

Results

This section first gives a summary about the results that have been found during the bachelor thesis and provide the limitations as well as an outlook to future work.

5.1 Summary

We started this bachelor thesis explaining the motivation for this work, subsequently explaining the outline of contribution and the structure of the thesis. In the second chapter the background information in which the thesis is embedded was given, as well as some definitions and the related work touched by our thesis. The third chapter explained the design process of the frontend via the user-centered design approach including target groups, personas and user stories. These fundamentals provided the basis for the think-aloud, heuristic evaluation and action analysis methods, used for improving the frontend in an iterative process. The same chapter provided detailed information for the design of the backend as well as the requirement analysis for the whole application. Chapter 4 described the whole implementation part for the front- and the backend. Used design patterns, chosen frameworks as well as important code snippets are treated in this chapter.
The major improvement between the first and the second iteration was the combination of the two separate views for visualizing and searching the incident data. The key finding for the second iteration was that components that are empty or don't provide any additional information, should be hidden to not confuse users. The second think-aloud method for the second

iteration received a much better feedback from the test candidates. User 6 and 8 stated that the interface was clearly understandable and the tasks were easily solvable.
An important factor for realizing the frontend for the Data Analysis Tool was to implement familiar structures that are similar to other applications in order to reduce the learning curve and make the application more accessible. User 6 confirmed this assumption by saying "I expected this button there, I know this from other applications, it looks familiar". User 9 stated "I think it has come out well, i think it looks good."
The results from the Action Analysis provided in table 5.1 show a signifi-

Tasks	1. Iteration untrained [sec]	trained [sec]	2. Iteration untrained [sec]	trained [sec]
Task 1	24,94	9,67	15,62	4,88
Task 2	24,36	10,08	14,38	5,87
Task 3	29,79	12,26	18,68	9,75
Task 4	22,52	8,69	12,69	4,82
Task 5	27,71	9,29	11,43	5,02
Task 6	26,31	14,33	19,02	5,60
Task 7	30,08	12,85	14,82	7,53

Table 5.1: Action Analysis Comparison

cant reduction in time users had to spent on solving the tasks. Users were sometimes able to decrease the time necessary to solve the task by half using the second iteration interface compared to the first iteration.
During the first iteration half of the test subjects stated that they are confused by the application performing actions automatically whenever they change the input fields. We improved this by making it clear that a user has to click a search button in order to receive results. This is an interesting fact, because it means some user prefer to need more clicks to achieve a result.

Chapter 6

Discussion

This Chapter provides the conclusions, the limitations as well as the outlook of this thesis.

6.1 Conclusions

The goals of this thesis were to build an application that is able to visualize and search through large datasets and designing a user-friendly interface in an iterative user-centered design process.
As seen in the summary section the user-centered design approach for designing user interfaces helped increasing the productivity and speeding up the users workload with every iteration. The observable data service pattern used for implementing the frontend is a well tested, well documented design pattern for frontend user interfaces. Angular, the main framework chosen for the frontend implementation, follows best practices and coding standards, is maintained by a large company, has a large userbase, is well documented and popular for frontend applications. All additional plugins, frameworks and libraries have been selected according to their userbase size, their quality of documentation and their provided examples. This provides a good maintainability for our application and the chance to add new features in the future as well as providing a template as starting point for other applications that want to visualize data.
The implementation of the backend with the help of the Elasticsearch technology stack provides a useful setup that can be used in future works to build effective applications that handle all different kinds of searchable datasets.

6.2 Limitations

Although the user interface has been successful implemented and the user interface has received positive feedback, there are still limitations. The back-end handling and frontend visualization of a huge dataset with more than 10.000 documents has not been tested. The initial testing phase for the whole collection of City.Risks applications starts after the end date of this bachelor thesis. So all testing was done using test data that was emitted from the participating members of the City.Risks project via the provided API endpoint. The application still provides space for further new visualizations. These could be easily integrated as separate components after new user feedback, when used in production or when the amount of data increases.

6.3 Outlook

The successfully implemented Data Analysis Tool can further be used to incorporate new features, thanks to its component based structure. Ideas for future improvements could be additional layers for the map like heatmaps, visualizing different districts of a city or provide additional information on nearby police or fire departments. Also it might be of interest to choose separate characteristics of the incident and provide a visualization for the chosen characteristics.

References

Cited references are ordered by appearance.

[1] V. Rajaraman, "Big data analytics," *Resonance*, vol. 21, no. 8, pp. 695–716, 2016. [Online]. Available: http://dx.doi.org/10.1007/s12045-016-0376-7

[2] I. Corporation, "What is big data?" 2013, [accessed April 12, 2017]. [Online]. Available: https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html

[3] A. Regalado, "The data made me do it," 2013, [accessed April 12, 2017]. [Online]. Available: https://www.technologyreview.com/s/514346/the-data-made-me-do-it/

[4] I. IDC Research, "Worldwide big data and business analytics revenues forecast to reach $187 billion in 2019, according to idc," 2016, [accessed April 12, 2017]. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS41306516

[5] I. Gartner, "Gartner says worldwide business intelligence and analytics market to reach $18.3 billion in 2017," 2017, [accessed April 12, 2017]. [Online]. Available: http://www.gartner.com/newsroom/id/3612617

[6] solidIt, "Db-engines ranking von suchmaschinen," 2017, [accessed February 14, 2017]. [Online]. Available: http://db-engines.com/de/ranking/suchmaschine

[7] C. Research and I. Action, "Project overview," 2017, [accessed 29.07.2017]. [Online]. Available: http://project.cityrisks.eu/project-overview/

[8] *Design patterns : elements of reusable object-oriented software / Erich Gamma*, 37th ed., ser. Addison-Wesley professional computing series. Boston, Mass. ; Munich [u.a.]: Addison-Wesley, 2009.

[9] Elasticsearch, "Preface," 2014, [accessed 19.07.2017]. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/guide/current/_preface.html

[10] S. Büttcher, C. L. A. Clarke, and G. V. Cormack, *Information Retrieval: Implementing and Evaluating Search Engines*. Cambridge: MIT, 2010.

[11] D. M. Berry, *Copy, Rip, Burn : The Politics of Copyleft and Open Source*. London: Pluto, 2008.

[12] R. T. Fielding, "Architectural styles and the design of network-based software architectures," dissertation, University of California, Irvine, 2000. [Online]. Available: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[13] A. Friedman, "The future of search engines is context," 2015, [accessed April 16, 2017]. [Online]. Available: http://searchengineland.com/future-search-engines-context-217550

[14] S. Sirich, "Smarter digital assistants and the future of search," 2016, [accessed April 16, 2017]. [Online]. Available: https://searchenginewatch.com/2016/04/25/understanding-intent-through-voice-search/

[15] J. Gans, "Google, yelp, and the future of search," 2015, [accessed April 16, 2017]. [Online]. Available: https://hbr.org/2015/07/google-yelp-and-the-future-of-search

[16] R. E. Roth, K. S. Ross, and A. M. Maceachren, "User-centered design for interactive maps: A case study in crime analysis," *ISPRS International Journal of Geo-Information*, vol. 4, no. 1, pp. 262–301, February 2015. [Online]. Available: https://doaj.org/article/81d999342d11462ba5a94e0733dee471

[17] E. Charters, "The use of think-aloud methods in qualitative research an introduction to think-aloud methods," *Brock Education Journal*, vol. 12, no. 2, pp. 68–82, May 2003. [Online]. Available: http://dx.doi.org/10.26522/brocked.v12i2.38

[18] M. González, L. Masip, A. Granollers, and M. Oliva, "Quantitative analysis in a heuristic evaluation experiment," *Advances in Engineering Software*, vol. 40, no. 12, pp. 1271 – 1278, 2009, designing, modelling and implementing interactive systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997809000477

[19] J. Nielsen, "10 usability heuristics for user interface design," 1995, [accessed April 18, 2017]. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/

[20] C. Spannagel, M. Gläser-Zikuda, and U. Schroeder, "Application of qualitative content analysis in user-program interaction research," *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, vol. 6, no. 2, 2005. [Online]. Available: http://www.qualitative-research.net/index.php/fqs/article/view/469

[21] E. Commission, "More women in senior positions key to economic stability and growth," Publications Office of the European Union, Luxembourg, Ms.

[22] C. Pearson, *Awakening the Heroes Within: Twelve Archetypes to Help Us Find Ourselves and Transform Our World*. Cambridge: Pluto, 1991.

[23] K. Breitman and J. Leite, "Managing user stories," in *International Workshop on Time-Constrained Requirements Engineering*, 2002, p. 168.

[24] T. A. S. Foundation, "Apache lucene core," 2016, [accessed February 06, 2017]. [Online]. Available: http://lucene.apache.org/core/

[25] E. BV, "Elasticsearch basic concepts," 2015, [accessed July 22, 2017]. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html

[26] T. A. S. Foundation, "Solr features," 2017, [accessed February 16, 2017]. [Online]. Available: http://lucene.apache.org/solr/features.html

[27] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, Jun. 2003. [Online]. Available: http://doi.acm.org/10.1145/857076.857078

[28] R. Shrestha, "Angular 2: A component-based mvc framework," 2016, [accessed 30.07.2017]. [Online]. Available: https://dzone.com/articles/angular-2-a-component-based-mvc-framework

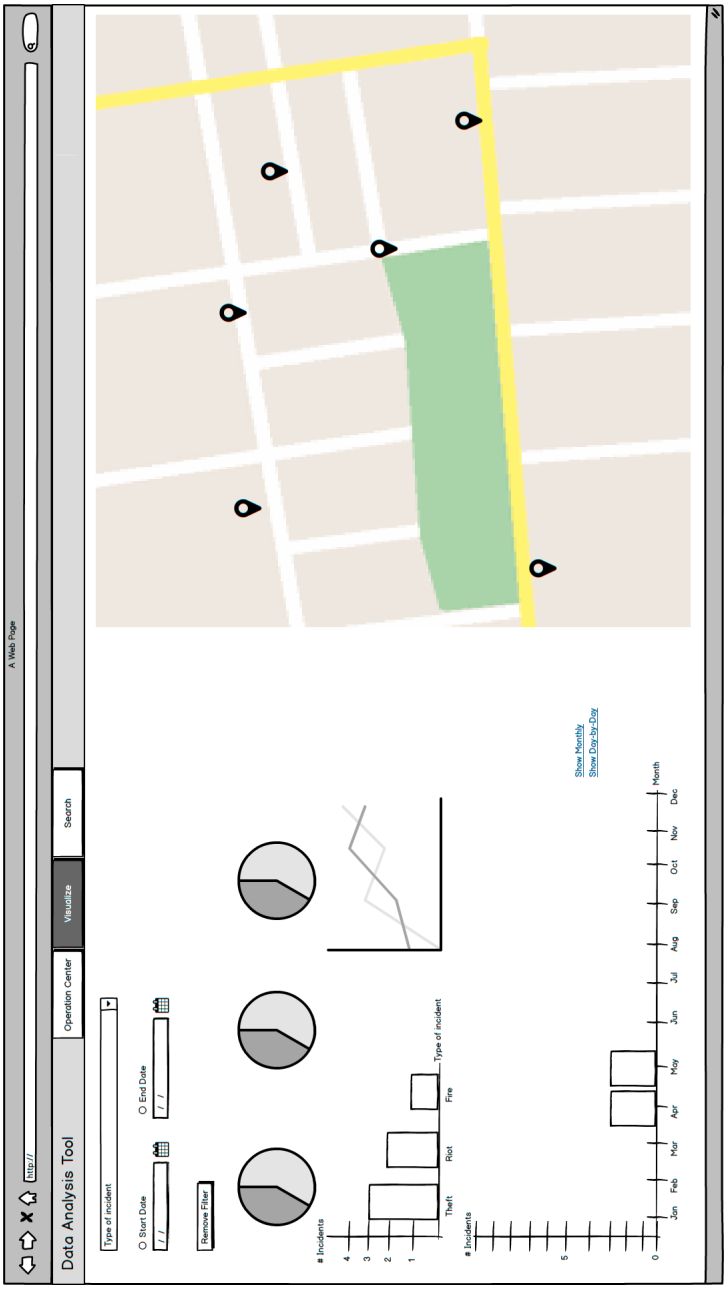
[29] Elasticsearch, "Theory behind relevance scoring," 2014, [accessed 18.07.2017]. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html

Appendices

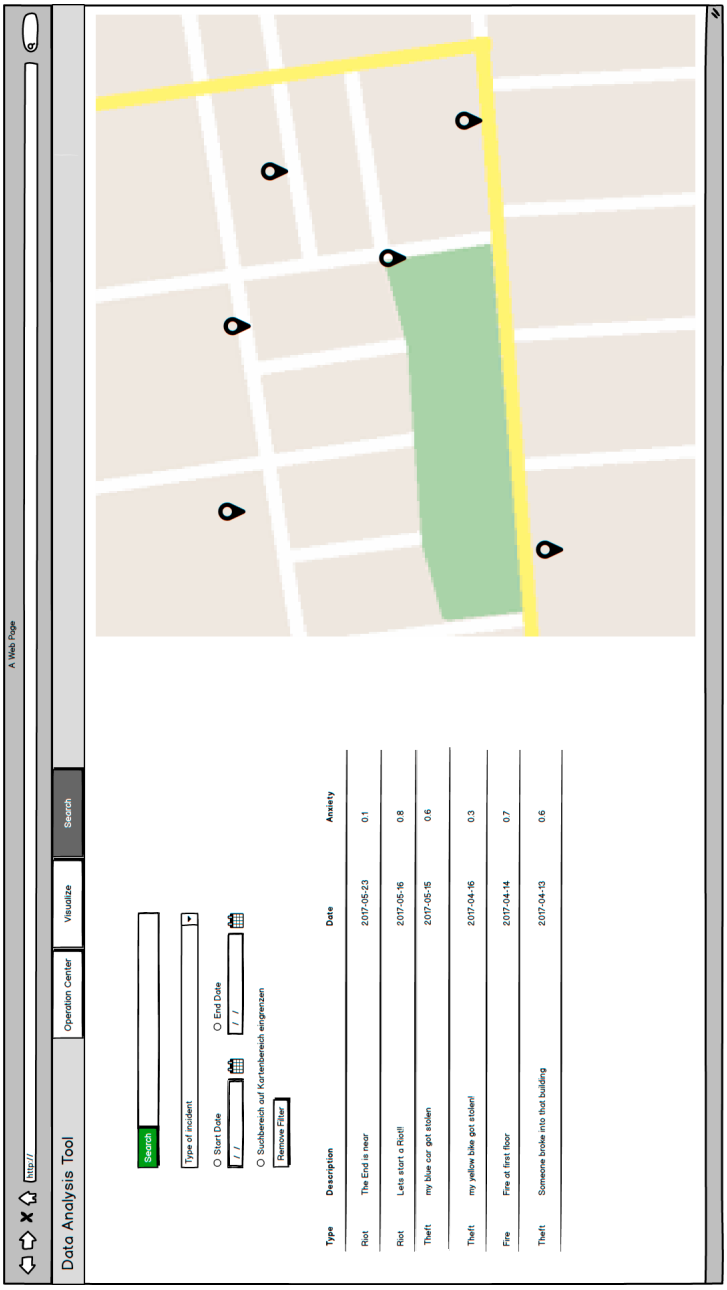
Appendix A

Images of Mockups and final Implementation

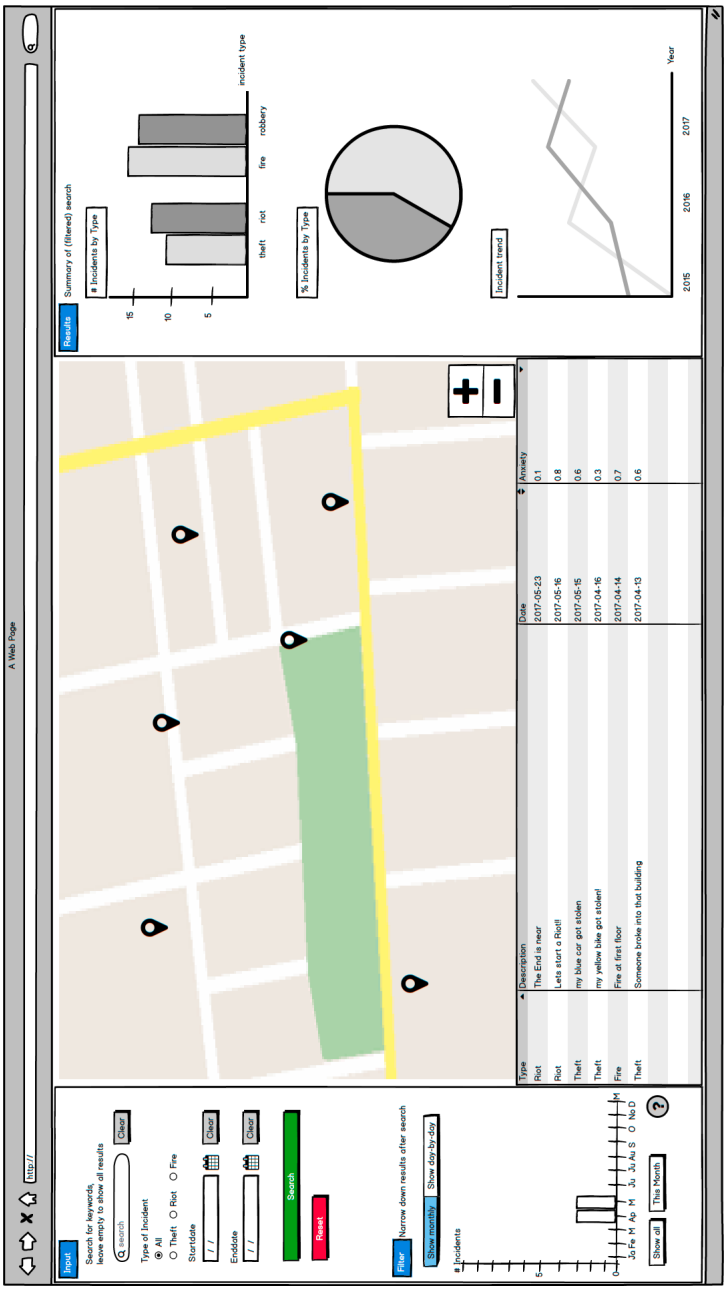
A.1 First Iteration Visualize View



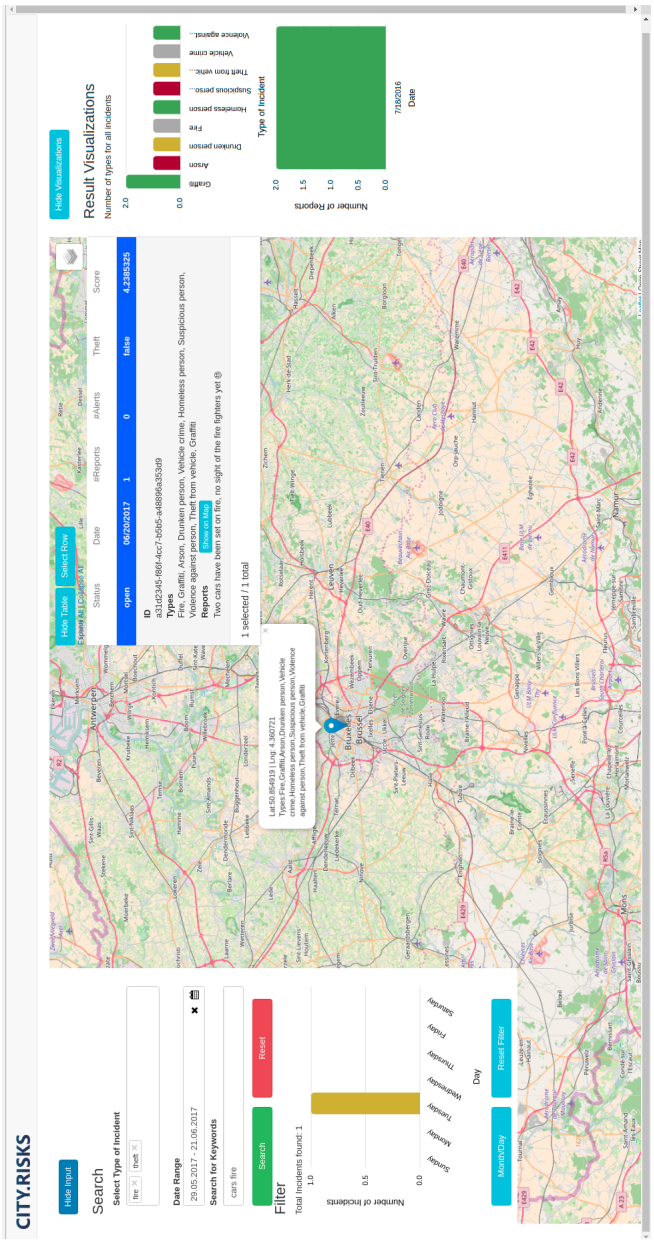
A.2 First Iteration Search View



A.3 Second Iteration Mockup



A.4 Implementation of Data Analysis Tool



Appendix B

Mapping of Elasticsearch Index

Listing B.1: Mappings for Elasticsearch Docment Index

```
{"mappings":{"incident":{"properties":{"@class":{"type":"text","
    fields":{"keyword":{"type":"keyword","ignore_above":256}}},
  "alerts":{"type":"nested","properties":{"@class":{"type":"text","
      fields":{"keyword":{"type":"keyword","ignore_above":256}}},
    "default_language":{"type":"text","fields":{"keyword":{"type":"
       keyword","ignore_above":256}}},
    "event_type":{"type":"text","fields":{"keyword":{"type":"keyword
       ","ignore_above":256}}},
    "geometry":{"properties":{"coordinates":{"type":"float"},
      "type":{"type":"text","fields":{"keyword":{"type":"keyword","
         ignore_above":256}}}}},
    "id":{"type":"text","fields":{"keyword":{"type":"keyword","
       ignore_above":256}}},
    "incident":{"type":"text","fields":{"keyword":{"type":"keyword",
       "ignore_above":256}}},
    "languages":{"properties":{"en":{"properties":{"description":{"
       type":"text","analyzer":"standard","fields":{"keyword":{"type
       ":"keyword","ignore_above":256}}},
      "headline":{"type":"text","fields":{"keyword":{"type":"keyword
         ","ignore_above":256}}},
      "instruction":{"type":"text","fields":{"keyword":{"type":"
         keyword","ignore_above":256}}}}}}},
    "msg_type":{"type":"text","fields":{"keyword":{"type":"keyword",
       "ignore_above":256}}},
    "note":{"type":"text","fields":{"keyword":{"type":"keyword","
       ignore_above":256}}},
    "provider_id":{"type":"text","fields":{"keyword":{"type":"
       keyword","ignore_above":256}}},
    "scope":{"type":"text","fields":{"keyword":{"type":"keyword","
       ignore_above":256}}},"sent":{"type":"long"},
```

```
16      "severity":{"type":"text","fields":{"keyword":{"type":"keyword",
            "ignore_above":256}}},
17      "status":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}}}},
18    "flags":{"type":"long"},"id":{"type":"text","fields":{"keyword":{"
        type":"keyword","ignore_above":256}}},
19    "location":{"properties":{"coordinates":{"type":"float"},"
        geometries":{"properties":{"coordinates":{"type":"float"},
20      "type":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}}}},
21      "type":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}}}},
22    "reports":{"type":"nested","properties":{"@class":{"type":"text","
        fields":{"keyword":{"type":"keyword","ignore_above":256}}},
23      "id":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}},
24      "incident":{"type":"text","fields":{"keyword":{"type":"keyword",
            "ignore_above":256}}},
25      "relevance":{"properties":{"@class":{"type":"text","fields":{"
            keyword":{"type":"keyword","ignore_above":256}}},
26        "code":{"type":"long"},"confidence":{"type":"float"},"
              report_id":{"type":"text","fields":{"keyword":{"type":"
              keyword","ignore_above":256}}},
27        "status":{"type":"text","fields":{"keyword":{"type":"keyword",
              "ignore_above":256}}},"value":{"type":"float"}}},
28      "src":{"properties":{"@class":{"type":"text","fields":{"keyword"
            :{"type":"keyword","ignore_above":256}}},
29        "address":{"type":"text","fields":{"keyword":{"type":"keyword"
              ,"ignore_above":256}}},"anxiety":{"type":"float"},
30        "created":{"type":"long"},"description":{"type":"text","
              analyzer":"standard",
31          "fields":{"keyword":{"type":"keyword","ignore_above":256}}},
                "flagging":{"type":"boolean"},
32        "id":{"type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}},
33        "incident":{"type":"text","fields":{"keyword":{"type":"keyword
              ","ignore_above":256}}},"lastTimeSeen":{"type":"long"},
34        "location":{"properties":{"coordinates":{"type":"float"},"type
              ":{"type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}}}},
35        "media":{"properties":{"heading":{"type":"float"},"type":{"
              type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}},
36          "url":{"type":"text","fields":{"keyword":{"type":"keyword","
                ignore_above":256}}}}},
37        "state":{"type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}},
38        "time":{"type":"long"},"type":{"type":"text","fields":{"
              keyword":{"type":"keyword","ignore_above":256}}}},
```

```
39      "state":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}},
40      "translation":{"properties":{"@class":{"type":"text","fields":{"
            keyword":{"type":"keyword","ignore_above":256}}},
41        "code":{"type":"long"},"description":{"properties":{"value":{"
              type":"text","analyzer":"standard",
42          "fields":{"keyword":{"type":"keyword","ignore_above":256}}}}
                },"media":{"properties":{"probabilities":{"type":"float"}
                ,
43          "terms":{"type":"text","fields":{"keyword":{"type":"keyword"
                ,"ignore_above":256}}}}}},
44        "report_id":{"type":"text","fields":{"keyword":{"type":"
              keyword","ignore_above":256}}},
45        "status":{"type":"text","fields":{"keyword":{"type":"keyword",
              "ignore_above":256}}}}}}}},
46    "sightings":{"type":"nested","properties":{"@class":{"type":"text"
          ,"fields":{"keyword":{"type":"keyword","ignore_above":256}}},
47      "id":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}},
48      "location":{"properties":{"geometry":{"properties":{"coordinates
            ":{"type":"float"},
49        "type":{"type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}}}},
50        "properties":{"properties":{"precision":{"type":"long"},"
              timestamp":{"type":"long"}}},
51        "type":{"type":"text","fields":{"keyword":{"type":"keyword","
              ignore_above":256}}}}},
52      "signal":{"type":"long"},"time":{"type":"long"}}},"state":{"type
            ":"text","fields":{"keyword":{"type":"keyword","ignore_above"
            :256}}},
53    "theft":{"type":"boolean"},"token":{"type":"text","fields":{"
          keyword":{"type":"keyword","ignore_above":256}}},
54    "types":{"type":"nested","properties":{"confidence":{"type":"float
          "},
55      "type":{"type":"text","fields":{"keyword":{"type":"keyword","
            ignore_above":256}}}}}}},
56    "settings":{"number_of_shards":2,"number_of_replicas":1}}
```

Appendix C

Transcription of Interviews

The transcribed interviews along with the audio and video files can be found on the attached DVD.