## 

**Master Dissertation**

**EE5500**

Markus Just

Student number: 1644609

Dr. Alizrea Mousavi

Electronic and Computer Engineering

School of Engineering and Design 

Monday, March 19, 2018

Table of contents

[List of Figures 5](#_Toc509148491)

[List of Tables 6](#_Toc509148492)

[List of Abbreviations 7](#_Toc509148493)

[Introduction 8](#_Toc509148494)

[Background to the project 10](#_Toc509148495)

[Regulations 15](#_Toc509148496)

[Drinking water quality 15](#_Toc509148497)

[External factors 16](#_Toc509148498)

[Aims and Objectives 19](#_Toc509148499)

[Structure of this dissertation 24](#_Toc509148500)

[Mock-Ups 25](#_Toc509148501)

[Experimental/investigative methods to be adopted 30](#_Toc509148502)

[Theoretical concepts 31](#_Toc509148503)

[Data dissemination 31](#_Toc509148504)

[Data integration 32](#_Toc509148505)

[Data analysis 33](#_Toc509148506)

[Qualitative analysis 34](#_Toc509148507)

[Quantitive analysis 34](#_Toc509148508)

[Generic cloud-based service provider 36](#_Toc509148509)

[System requirements 36](#_Toc509148510)

[Context view 41](#_Toc509148511)

[Administrator 42](#_Toc509148512)

[User 42](#_Toc509148513)

[HarmonizationService 42](#_Toc509148514)

[External data source 43](#_Toc509148515)

[Use cases 43](#_Toc509148516)

[Short-descriptions of the use cases 45](#_Toc509148517)

[Use-case “Manage Users/Roles” 46](#_Toc509148518)

[Use-case “Configure Views” 46](#_Toc509148519)

[Use-case “Viewing information” 46](#_Toc509148520)

[Domain Model 47](#_Toc509148521)

[Component diagram 47](#_Toc509148522)

[GUI: 49](#_Toc509148523)

[Configuration Service: 49](#_Toc509148524)

[DataProvider: 49](#_Toc509148525)

[CommunicationService: 49](#_Toc509148526)

[ConfigDB: 50](#_Toc509148527)

[ER-model 50](#_Toc509148528)

[Entitites 52](#_Toc509148529)

[Design and Implementation 56](#_Toc509148530)

[Amazon Web Services 56](#_Toc509148531)

[Programming Language and Frameworks 56](#_Toc509148532)

[Java 56](#_Toc509148533)

[Vaadin 57](#_Toc509148534)

[JPA and Hibernate 58](#_Toc509148535)

[Software Architecture 59](#_Toc509148536)

[Class diagram 59](#_Toc509148537)

[Results and Analysis 60](#_Toc509148538)

[Integration of the Harmonization Service 60](#_Toc509148539)

[Conclusions and further work 64](#_Toc509148540)

[Summary 64](#_Toc509148541)

[Further work 65](#_Toc509148542)

[Project plan 67](#_Toc509148543)

[Time-plan 67](#_Toc509148544)

[References 71](#_Toc509148545)

# List of Figures

[Figure 1 - water treatment [1] 9](#_Toc508287231)

[Figure 2 – Population Growth [2] 11](#_Toc508287232)

[Figure 3 - how the price is set [9] 18](#_Toc508287233)

[Figure 4 – WWTP system overview [10] 20](#_Toc508287234)

[Figure 5 – purpose of the system 22](#_Toc508287235)

[Figure 6 - example view 27](#_Toc508287236)

[Figure 7 - example view 2 27](#_Toc508287237)

[Figure 8 - user management 28](#_Toc508287238)

[Figure 9 – basic context view 41](#_Toc508287239)

[Figure 10 - context view in WWTP example context 42](#_Toc508287240)

[Figure 11 – Use Case Diagram 45](#_Toc508287241)

[Figure 12 - initial data model 49](#_Toc508287242)

[Figure 13 - ER-model 53](#_Toc508287243)

[Screenshot 14 - Time Plan 65](#_Toc508287244)

# List of Tables

[Table 2 - dissertations tructure 24](#_Toc508287169)

[Table 3 - User Role 53](#_Toc508287170)

[Table 4 – User 53](#_Toc508287171)

[Table 5 – VisualizationComponent 54](#_Toc508287172)

[Table 6 - DataField 55](#_Toc508287173)

[Table 7 – DataOperation 55](#_Toc508287174)

[Table 8 - Data Source 56](#_Toc508287175)

[Table 9 - OperationType 56](#_Toc508287176)

# List of Abbreviations

# Introduction

Today it is natural that clean water is available whenever it is needed. Water is not only used for drinking, it is also used for cleaning, cooking, washing or many other things in daily life. Besides for individuals it is important for whole industries like agriculture industry, catering industry, food industry and many more. It is one of the most important basics in today’s life that clean water is accessible for everyone with a certain quality and quantity. To make sure that clean water is present all the time, there are industrial processes, which use water and wastewater treatment plants to make use of natural water resources and sewage to produce clean water which can be used by consumers. The following picture shows the process for natural water resources:



Figure 1 - water treatment [1]

The purpose of this report is to give an initial overview about the master dissertation project, which topic is water resource planing in the UK with the help of modern web technologies.

This project is strongly realted to another project, so this report will briefly describe the general idea and will then focus on the part which is relevant for this dissertation. After the information about the backgound of the project, there will be a part which describes the initial survey done. Furthermore, this report gives information about the aims and objectives of the dissertation and the expected outcomes as well as the time plan for the project.

## Background to the project

“Customers’ top priority for water services is a safe, reliable supply of water at a price they can afford” [2]. Because of different factors the risk of droughts in the UK increases steadily and there have also been some droughts during the last 40 years which were worse than those that had been used as the basis of planning. [2] One factor which is responsible for the increased risk of a drought is the climate change as well as the resulting increased evaporation during the warm months of the year. Another factor is the population growth, which is estimated to be between 6,6 million and 16 million by 2040 for England and Wales. Especially regions with the least resilience regarding water resources are subject to most population growth and climate change. [2]

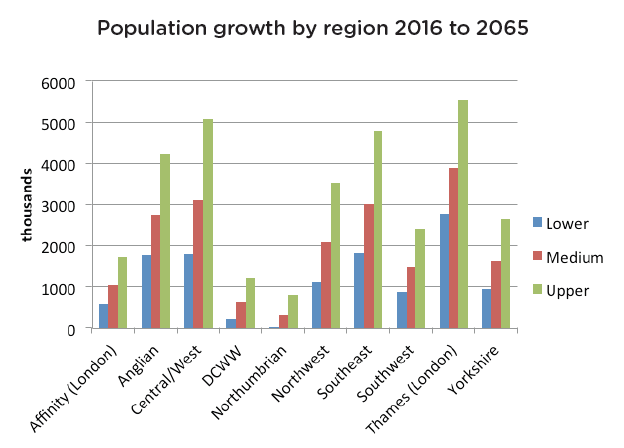


Figure 2 – Population Growth [2]

To protect customers as well as the industry from the consequences of serious droughts and to enable an economical and environmental efficient use of water resources, politics and the water industry have to make strategical long-term and short-term decisions regarding water infrastructure to be able to gain resilience.

The problem is that in water industry there is no technology which provides information about the overall state of the current water resources, neither for consumers nor for providers or political decision-makers. Because of this missing knowledge, there is a lack of efficiency regarding different aspects, e.g. energy consumption or cost. It is not possible to achieve an efficient water resource planning without information about important key indicators and predictions.

The basis of this project is the data acquired by modern sensors of water and wastewater treatment plants. This data is categorized in the industry’s key performance indicators. Those indicators are Carbon Footprint, Energy Consumption & Generation, the yield of the By-Products and the Productivity, which means the performance of the Equipment and Reliability. It has to be dealt with the real-time data as well as with historical data. As already mentioned, the acquisition, harmonization and provision of the data from different water providers is done in another project. This dissertation focuses on the data analysis and data representation. That’s why it has to be analyzed which stakeholders there are to create a concept how the data analysis has to be done and which factors has to be considered. A source for this kind of knowledge is Water UK:

“Water UK is a membership organisation which represents and works with the major water and wastewater service providers in England, Scotland, Wales and Northern Ireland”. [3]

Water UK provides all kind of information about water supply, wastewater and regulations regarding those processes. Focusing on the regulators, Water UK makes the following statement:

“A wide range of organisations work with water companies to ensure customers get the best services for the best possible price, and that the environment is protected.” [4] The mentioned organisations are categorized in governmental organisations, regulators, consumer watchdogs and water companies. These categories will be relevant for the concept of the data visualization:

*Governmental organisations:*

[**Defra**](https://www.gov.uk/government/organisations/department-for-environment-food-rural-affairs) **(**[Department for Environment, Food & Rural Affairs](https://www.gov.uk/government/organisations/department-for-environment-food-rural-affairs), [Environment Agency](https://www.gov.uk/government/organisations/environment-agency)):

UK government department responsible for looking after natural environment. Defra sets the overall rules for water services in England and is responsible for water quality and orders regarding droughts.

[**Welsh Government**](http://gov.wales/?lang=en): devolved Government for Wales which works to help improve the lives of people in Wales and make the nation a better place in which to live and work. The Welsh Government sets the overall rules for water services in Wales. [4]

*Regulators:*

[**Drinking Water Inspectorate**](http://www.dwi.gov.uk/): the DWI was formed in 1990 and provides independent reassurance that water supplies in England and Wales are safe and drinking water quality is acceptable to consumers. [5]

[**Environment Agency**](https://www.gov.uk/government/organisations/environment-agency): non-departmental public body which is responsible for regulating industry waste, as well as water quality and resources in England. They are also responsible for managing the risk of flooding from rivers, reservoirs, estuaries and the sea. [4]

[**Natural England**](https://www.gov.uk/government/organisations/natural-england): the government's adviser for the natural environment in England, helping to protect England's nature and landscapes for people to enjoy and for the services they provide.

[**Natural Resources Wales**](https://naturalresources.wales/?lang=en): works to ensure that the environment and natural resources of Wales are sustainably maintained, enhanced and used, both now and into the future.

[**Ofwat**](https://www.ofwat.gov.uk/)**:** the economic regulator for the water and sewerage sectors in England and Wales. It works in the interest of customers by setting price limits, ensuring companies run efficiently and encouraging resilience. [4]

*Consumer Watchdog:*

[**CCWater**](http://www.ccwater.org.uk/): The Consumer Council for Water promotes consumers' interests to governments, regulators and water companies. They also provide a free advice and complaint handling service for consumers, research their views on key topics, and keep them informed on the issues that affect their services. [4]

All these stakeholders have different kind of interests in the data, which has to be considered when doing a data analysis with a visualized result. What exactly these interests are must be researched beforehand which is one of the first steps during the project.

## Regulations

### Drinking water quality

In the UK, drinking water quality has the highest priority for water suppliers. Independent drinking water inspectorates regularly[[1]](#footnote-1) check and ensure the quality of the water provided to customers. The national regulations specify strict standards derived from the EU Drinking Water Directive, which is based on advice from the World Health Organization (WHO). [6]

It is the requirement of each member state of the EU to translate requirements of the EU Drinking Water Directive to local laws. The UK follows this requirement and adds also additional rules, which leads to very high standards in regards of drinking water quality. As mentioned before, the EU and the UK regulations are based on the advice of the WHO, which are regularly updated mainly because of new gained knowledge. To make sure that the regulations of the EU and the UK are updated accordingly, the European Commission review the current stadards at least every five years and update them if it is necessary. [6]

Besides the drinking water quality the EU directive lays down strict requirements for monitoring, analysing and reporting of measured data as well as requirements about actions which has to be taken if standards are exceeded. [6] The DWI states that water companies have to ensure that samples for e.coli, coliform bacteria, colony counts, residual disinfectant, turbidity and nitrite are taken at a specific frequency from the point the water leaves a treatment works. [7]

## External factors

The data analysis which has to be done is based on two data sources. On the one hand it is the data which is collected on the plant level. This means real-time and historical data about the productivity, the energy consumption or the water quality measured by the sensors of the water treatment plants. On the other hand there are external data sources which will be required to perform fitting analyses. So it is important to identify which external factors are relevant for the water resource planning and what’s the impact of these factors.

Maybe the most important external factor is the weather situation. It has impact on the total availablity of water, on the demand for water by the people and on the water quality. The Met Office[[2]](#footnote-2) states [8]: “With climate change predicted to increase, the likelihood of significant weather events from extreme rainfall to heatwaves, will have a significant impact on the water industry.“ Extreme weather conditions may have a serious impact on water quality e.g. too much rain can cause sewers to flood and overflow into water course. Heatwaves with few rain may lead to droughts which could result in a reduction of availability of water as well as a higher demand of the consumer

s. If less water is available then the overall cost of delivering water to the end consumer may increase for the industry. With the knowledge of upcoming extreme weather conditions, actions could be taken early by regulators or consumers.

As mentioned before the cost of the whole process is also interesting for the stakeholders. Ofwat sets price limits for customers, so that the providers have to work efficiently and look after resilience to avoid sudden increase of cost. The following image shows how the price for water is set by a specific water provider (south east water):

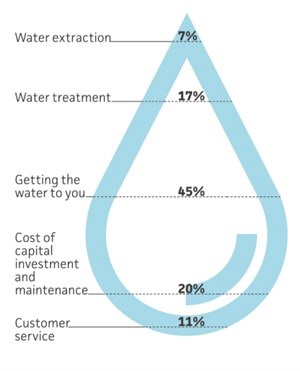


Figure 3 - how the price is set [9]

Almost 20% of the price is related to the water treatment process. So the absolute cost of the treatment process will influence the overall cost for whole process. This information could be very useful for water companies and may lead to strategic decisions depending on the current situation. From 2016 to 2017 the average household water and sewage bill in England and Wales was about £389. [9] The prices increased by £2, which is part of a 5-year-plan confirmed by Ofwat. This plan enables companies to invest £44 billion over 5 years in better services, greater resilience and environmental improvements. [9]

# Aims and Objectives

The main objective of this project is to develop a generic cloud service which is able to collect, analyze and visualize data independent of its origin and meaning. The previously described problem will be taken as an example use case and should show the validity of the concept as well as possible problems.

Taking the WWTP example it should be possible to integrate data from different sources, analyse them and create different results and representations, which can be categorized in different levels:

- strategic level: level which enables users to create strategies (e.g. price shouldn’t go over x)

- operational level: engineering level combined with external factors (e.g. weather)

- engineering level: lowest level/plant level

For example an end consumer wants to see different kind of data than a representative of a water provider company or a politican which aims towards developing a strategy for water resource management. This should be achieved by analyzing real-time data and/or historical data combined with different external factors like weather conditions. It is necessary to have a fitting security concept to guarantee that every user role can only see information which it should be allowed to see. For example a member of a water provider company shouldn’t be able to see sensitive data from another provider company (of course a consumer shouldn’t also be able to see sensitive copmpany data).



Figure 4 – WWTP system overview [10]

The illustration shows that the water and wastewater treatment plants of different water providers are equiped with different kinds of modern sensors, measuring data regarding several key indicators like Carbon Footprint, Energy Consumption & Generation, the yield of the By-Products of the processes, and the Overall Performance of the Equipment and Reliability (Productivity). This sensor data from different plants is collected and will be migrated to a common data model.

Each service provider categorizes the data from its plants in different indicators. The Large Network Performance Collider collects this data and converts it into a common model. This collected data is the basic information for the data analysis which has to be done to create the representation for stakeholders to enable strategic decision making.

The KPIs which are acquired at plant level and communicated over the service providers’ internal networks will be translated to a provider-specific evaluation of three aspects:

*Sustainablity:* Efficiency, water networks, waste

*Economic:* Economic factors (cost)

*Societal:* Well-being of public

Those results will be collected from different service providers to create an overall view which enables strategic decision-making. Real-Time data should be combined with historical data to create a systematic knowledge and data engineering capability. Additionally external factors have to be considered as well at this point as described in chapter (REFERENZ).

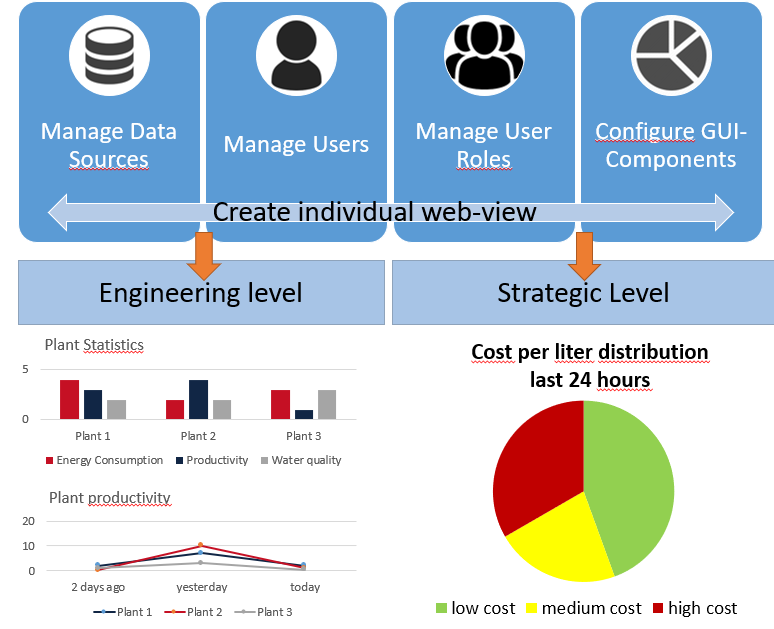


Figure 5 – purpose of the system

Long-term aims and objectives:

* Development of a generic cloud based data analysis service which enables data processing and data visualization independent of the specific problem
* Raising the resilience in the UK’s water and wastewater treatment process by using this cloud service together with the data harmonization project

Aims and objectives for this dissertation work:

* Creating a concept for a generic data analysis and data visualization service platform
* Developing a prototype with the key features required for the long-time aim
* Integrating the WWTP data harmonization project’s API and visualize the provided data

# Structure of this dissertation

This master dissertation consists of 5 main chapters. The **introduction** chapter gives an overview about the background and the general idea of this project. The definition of the aim and objectives are also part of the first chapter. The second chapter explains the basic **theoretical concepts** behind this project. This part is mainly based on the literature research done in the early phase of the dissertation work. The third chapter, which is the largest one, is the **Design and Implementation** chapter and it consists of all technical and structural aspects regarding the development of the system, such as design approach, used technologies and software architecture. The penultimate chapter **Results and Analysis** describes the outcome of the implementation, names possible problems and analyses if the approach was successful. In the final chapter **Conclusions and Further Work** the key results are summarized and it will be listed which would be required to approach the long-term goal of the project.

Table 2 - dissertations tructure

## Mock-Ups

One of the first steps when developing a web application or a desktop application with an user interface is often the creation of mock-ups. Mock-ups are graphical models and their purpose is to give an idea of the look and feel of the application and the possible user interactions. They are also useful for the developer and the customer to get a common understanding of the application’s purpose and they enable the customer to provide feedback at an early time point in the development process.

In this project they are used to discover the functionality of the system and to help developing the fuctional requirements and data models. The following Mock-ups are created with the web based tool Balsamiq, which provides an user interface to support the mock-up creation and they show an exemplary view for users and admins, which are the two main external actors.

It is important to mention that mock-ups don’t necessarily represent the final state of the application. Instead they give an overview about the ideas and possible features, which may still be subject to change during the development phase.

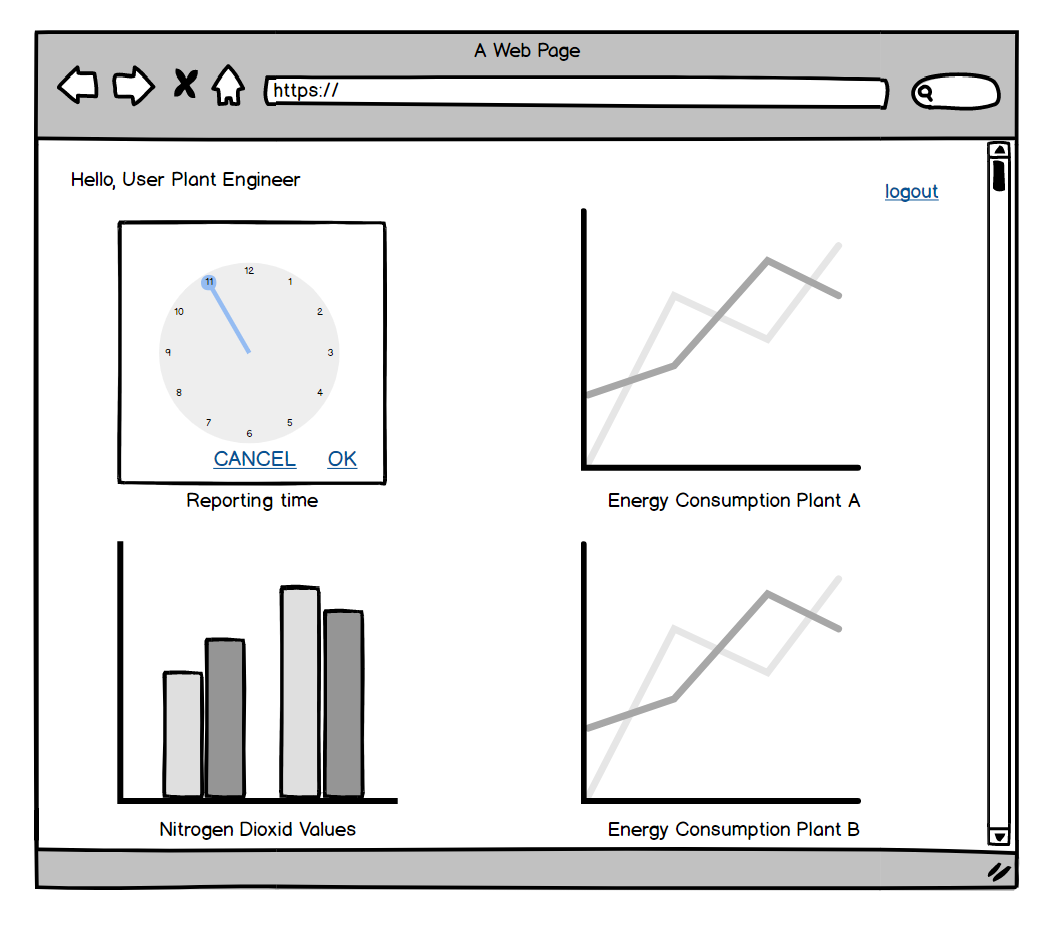


Figure 6 - example view

In this mock-up an example view is shown for a water plant engineer from a water provider company in the WWTP example context. A user view consists of one or many view components which represent the values of data source, in this case the sensor values of different water plants. The user may have the possibility to set the time period of the report by using a timepicker or datepicker component.

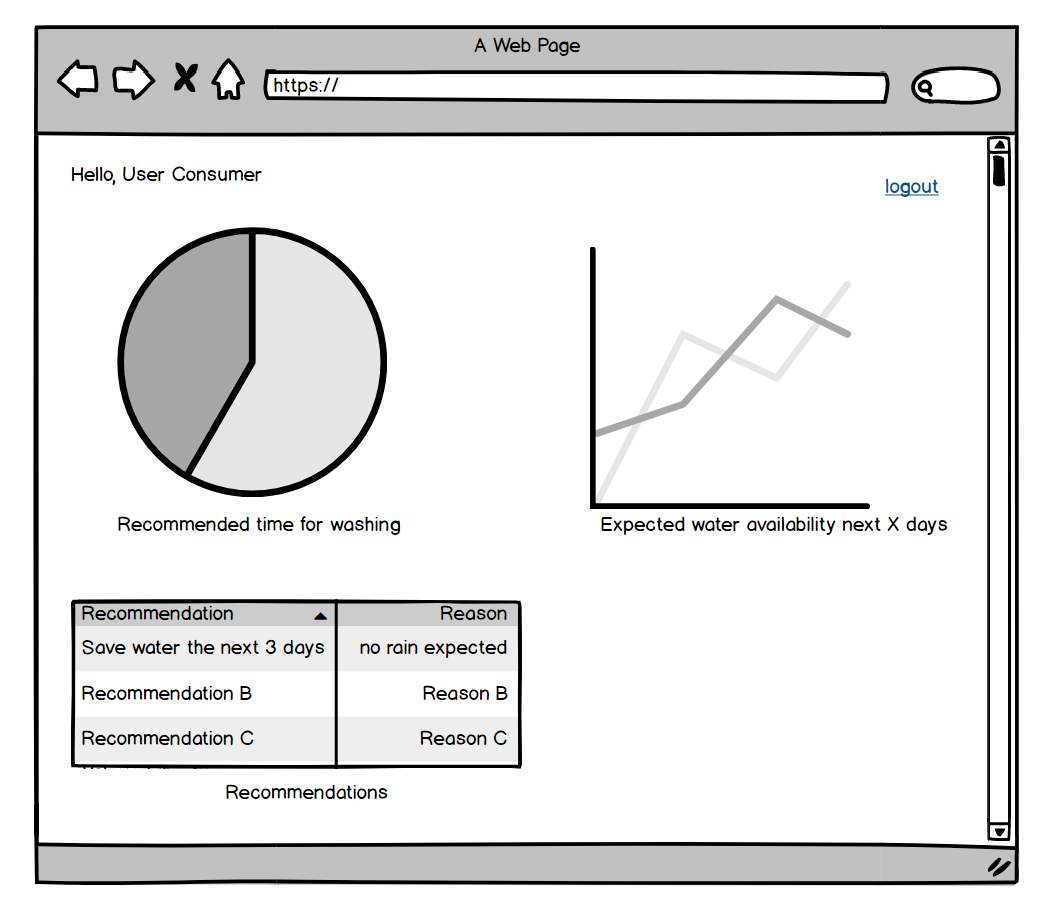


Figure 7 - example view 2

Similar to the view of the engineering user in the last mock-up, a consumer in the WWTP example sees the pre-configured components representing values from data sources. In opposite to the first view, the data comes from different data sources (e.g. water plants and weather providers) and they are analysed and evaluated using special rules, previously configured by an administrator. This leads to recommendations for the user visualized by the pie chart and the table.

The following mock-up shows a more complex interface with many possibilities of user interaction. This is the example of the user management view, which is an admin’s view:

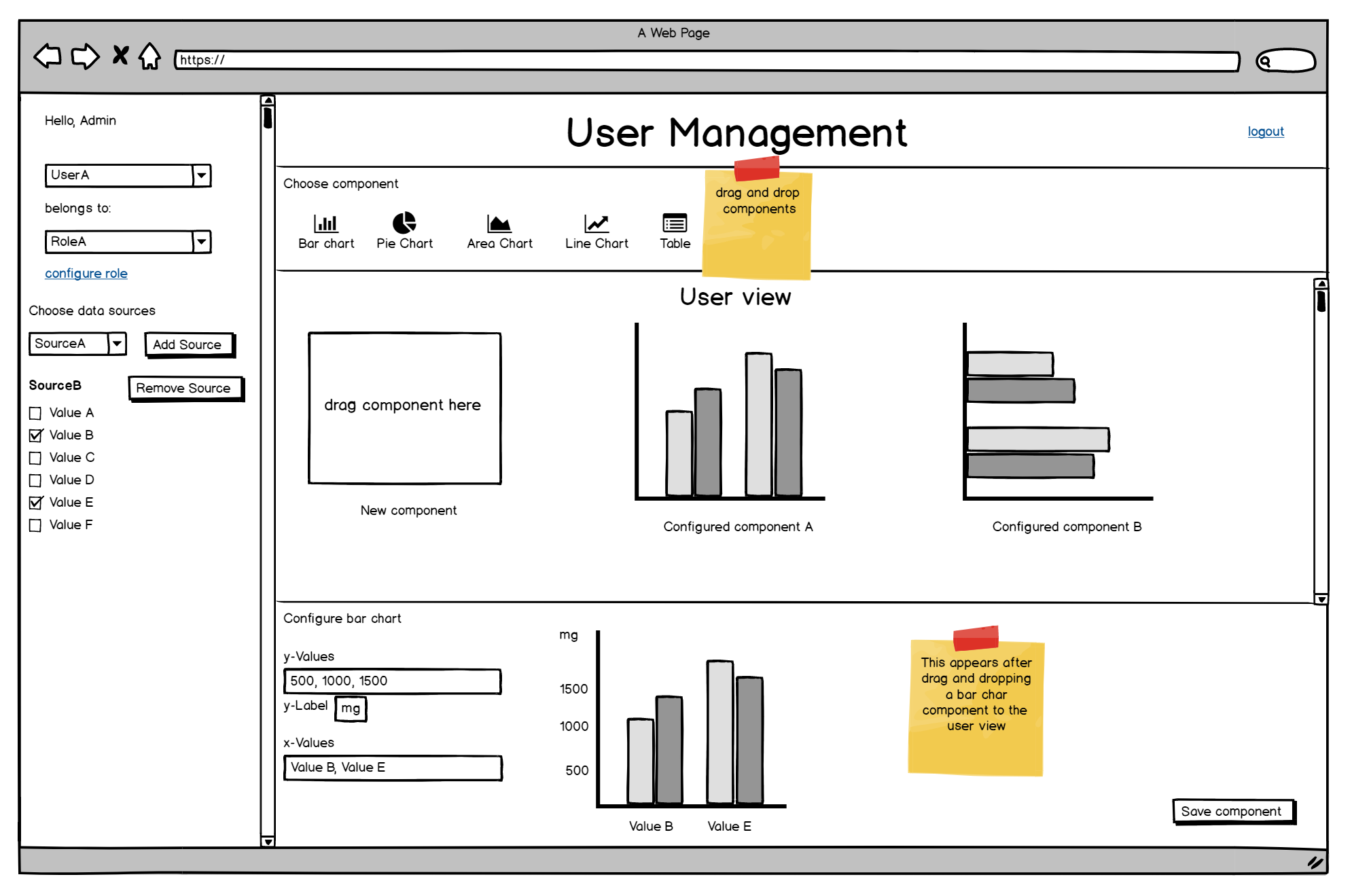


Figure 8 - user management

The purpose of this view is to configure an individual user view like shown in the first two mock-ups. The left part of the screen is the configuration of the user, which role the user belongs to and which data sources are used for the visualization components. After the choice of the user and the user role, the admin chooses from the pre-configured data sources (the configuration of the data sources is not part of this screen) and checks the data values which should be visualized by a view component. The available view components are in the upper part of the screen (below the headline) and they can be chosen by drag-and-dropping them to the placeholder in the center view of the page. The center view also shows the current view of the user and by clicking on one of the already present components, they can be modified. The bottom part of the page gives a preview of the component and provides the possibility to configure the new or an already existing component. This bottom view must be individual for every kind of component and should also provide the possibility to apply different kind of operations to the data values (e.g. multiplication with 1000 to convert miligrams to grams).

# Experimental/investigative methods to be adopted

It has to be analyzed which cloud technologies will be used to create a fitting representation of the current situation considering external factors and/or historical data. Furthermore a strategy has to be developed how to analyze the harmonized data from the different plants and create meaningful results, depending on the users’ request. It must also be evaluated which external data sources must be used to produce useful information. Another important task will be the role concept for the application. A fitting security strategy has to be developed to ensure that every user is only allowed to see what he/she should see. Critical data has to be secured from unauthorized access to avoid the misuse of information.

Due to the fact, another project is working on the problem of the data aqcuisition and data harmonization of different data sources (different providers), there is a need for “simulated” data which can easily replaced by the real data when it is available.

For the visualization of the results, a fitting user interface must be developed. It has to be researched which user will be interesed in which kind of information and after that the user interface has to be designed by using Mock-ups.

# Theoretical concepts

## Data dissemination

The concept of data dissemination is present all over the Internet. Every second millions of internet users receive data from different servers all over the world. This is made possible through communication protocols such as HTTP (Hyper Text Transfer Protocol) in combination with Internet Browsers and HTML (Hyper Text Markup Language). Using this technologies a user is able to navigate through the world wide web and view websites presenting data to the user. Another widely spread mechanism for data dissemination thorugh the internet is E-Mail communication. Using the SMTP protocol (Simple Mail Transfer Protocol) E-Mail is used to disseminate data through the internet or through intranet systems which are used by companies to enable their staff to communicate. [11]

Despite the efficiency of electronic means of data dissemination, there are still drawbacks which may take a long time to overcome, if at all. Privacy is one of the most common problems with electronic data dissemination. The internet has thousands of loop holes where people can peep into the private lives of other people. Security is also a related problem with electronic data dissemination. Every year, millions of dollars are lost to electronic theft and fraud. Every time a solution is found for a security problem, another malicious programs spring up somewhere in the globe.

Many companies set up precautionary measures against security invasion in their information systems. Some set up user accounts with varying privileges to data access. Many set up internet firewalls and anti virus software on their computers to prevent intrusions.

Data dissemination is a very substantial aspect of business operation. Most of today’s businesses are data driven. It is a common scenario where business organizations invest millions for data warehouses including hardware, software and manpower costs, to make data dissemination fast, accurate and timely. Information gathered from disseminated data form as basis for spotting industry trends and patterns and decision making in companies. [11]

## Data integration

Unter **Informationsintegration** versteht man das Zusammenführen von Informationen aus verschiedenen Datenbeständen (Datenquellen) mit in der Regel unterschiedlichen Datenstrukturen in eine gemeinsame einheitliche Datenstruktur.

Dabei sollen vor allem [heterogene](https://de.wikipedia.org/wiki/Heterogenit%C3%A4t_(Informationstechnik)) Quellen möglichst vollständig und effizient zu einer strukturierten Einheit zusammengeführt werden, die sich effektiver nutzen lässt, als dies bei direktem Zugriff auf die einzelnen Quellen möglich wäre. Informationsintegration ist vor allem dort notwendig, wo mehrere gewachsene Systeme miteinander verbunden werden sollen, also beispielsweise bei der Zusammenführung von Unternehmen, [Arbeitsabläufen](https://de.wikipedia.org/wiki/Arbeitsablauf) und Anwendungen oder bei der Informationssuche im [Internet](https://de.wikipedia.org/wiki/Internet).

Die Integration komplexerer Systeme ist erst in den 1990er Jahren in den Blickpunkt der [informatischen](https://de.wikipedia.org/wiki/Informatik) Forschung gerückt und somit in der Entwicklung begriffen.

## Data analysis

Data analysis can be defined as a numeric and statistic process to discover structures in large data sets, for example grouping data by specific aspects or finding dependencies between different factors. A data analysis is about getting to know existing data, processing this data and visualizing it to develop hypotheses or issues. The primary aim is to discover the information which is hold by the data and to be able to describe and present this information. Data analysis can be categorized into quantitative analysis and qualitative analysis.

### Qualitative analysis

Qualitative research is often used in social sciences and aims towards understanding things like human behaviour and social phenomenons for example. The analysis techniques are quite dynamic and flexible, some examples are interviews, case by case analysis or observations.

### Quantitive analysis

Quantitive analysis is about describing behaviour with models, correlations and numeric manifestations and enabling forecasting based on the results. [12] It aims towards gaining statistical evidences by isolating cause and effect and by measuring and quantification of phenomenons. In opposite to the qualitive analysis, the approaches are often [standardised](https://www.linguee.de/englisch-deutsch/uebersetzung/standardised.html) and follow strict rules to guarantee a statistical relevant data collection. There are three important criterias when it comes to quantitive analysis, which are the following:

|  |  |
| --- | --- |
| objectivity | Relates to the observer’s independent description of the facts. External factors aren’t considered if this it’s possible. |
| reliability | Relates to the measuring instrument itself and demands accuracy of measurements. Two measurements under the same conditions must have the same results. A measuring instrument might be a sensor for example. |
| validity | Validity describes the formal correctness of the measurements. This means that a sensor, for example, should really measure the values it should measure (and nothing else) |

In this project the focus is on the quantitive data analysis, because the data comes mainly from sensors and the analysis models and visualizations are based on numeric values and statistics.

# Generic cloud-based service provider

## System requirements

A system requirement is defined as a condition or a capability which is necessary to enable a user to solve a specific problem or reach a specific goal. Defining system requirements is usually one of the first steps in a software development process and they aim towards getting a common understanding between the customer and the developer about what problems the system should solve.

In general, requirements are categorized as functional and non-functional requirements, where the functional requirements specifiy the functionality of the system, i.e. what a user should be able to achieve by using the system while the non-functional requirements are demands regarding quality aspects, such as scalability, security or performance.

In the following section the requirements for the cloud-based service provider will be listed. Each requirement has an individual identifier (number) and the functional and non-functional requirements will be grouped as:

* 100: purpose of the system
* 1xx: functional requirements
* 2xx: non-functional requirements

**Requirement 100:**

The system should provide the possibility to create an individual data analysis and data representation per user by configuring the data sources and view components individually. Each user should see its own, previously configured, user view when he logs in to the application.

**Requirement 101:**

An admin user should be able to create and delete users.

**Requirement 102:**

An admin user should be able to create and delete user roles.

**Requirement 103:**

An admin user should be able to add and remove data sources for a specific user or user role.

**Requirement 104:**

An admin user should be able to add and remove data sources for a specific user or user role.

**Requirement 105:**

An admin user should be able to select and unselect data fields from a specified data source.

**Requirement 106:**

An admin user should be able to select a view component by drag-and-drop it from the view component selection.

**Requirement 107:**

An admin user should be able to define rules for the representation of the data fields within the view components.

**Requirement 108:**

An admin user should be able to save the current configuration persistently in a database.

**Requirement 109:**

A user should be able to login to the web application by providing its credentials.

**Requirement 110:**

A user should see his individual user view with the preconfigured data representation after the login.

**Requirement 111:**

The system should be able to update the data from the data sources after a previously configured interval.

**Requirement 112:**

The view components should be able to update itself when the source data has changed without reloading the web page.

**Requirement 201:**

The application should be deployable to any cloud environment.

**Requirement 202:**

The application should be accessible with any modern internet browser by accessing the URL.

**Requirement 203:**

The user interface should be easy to handle by any user without the need to read further documentation.

**Requirement 204:**

A user should only be able to see the data which was individually configured for him and shouldn’t have access to other data.

**Requirement 205:**

All interactive user interface elements should be clearly identifiable by its appearance.

**Requirement 206:**

The system should communicate via HTTP with external data sources.

## Context view

The context view of a system describes the system under development (SuD) as a blackbox and how it is embedded in its environment. This view is used early in the development process to define the scope of a project and it can be considered as a connections between the written requirements and the system architecture. This view aims towards clarifying the system boundaries and the interfaces to the outside world.

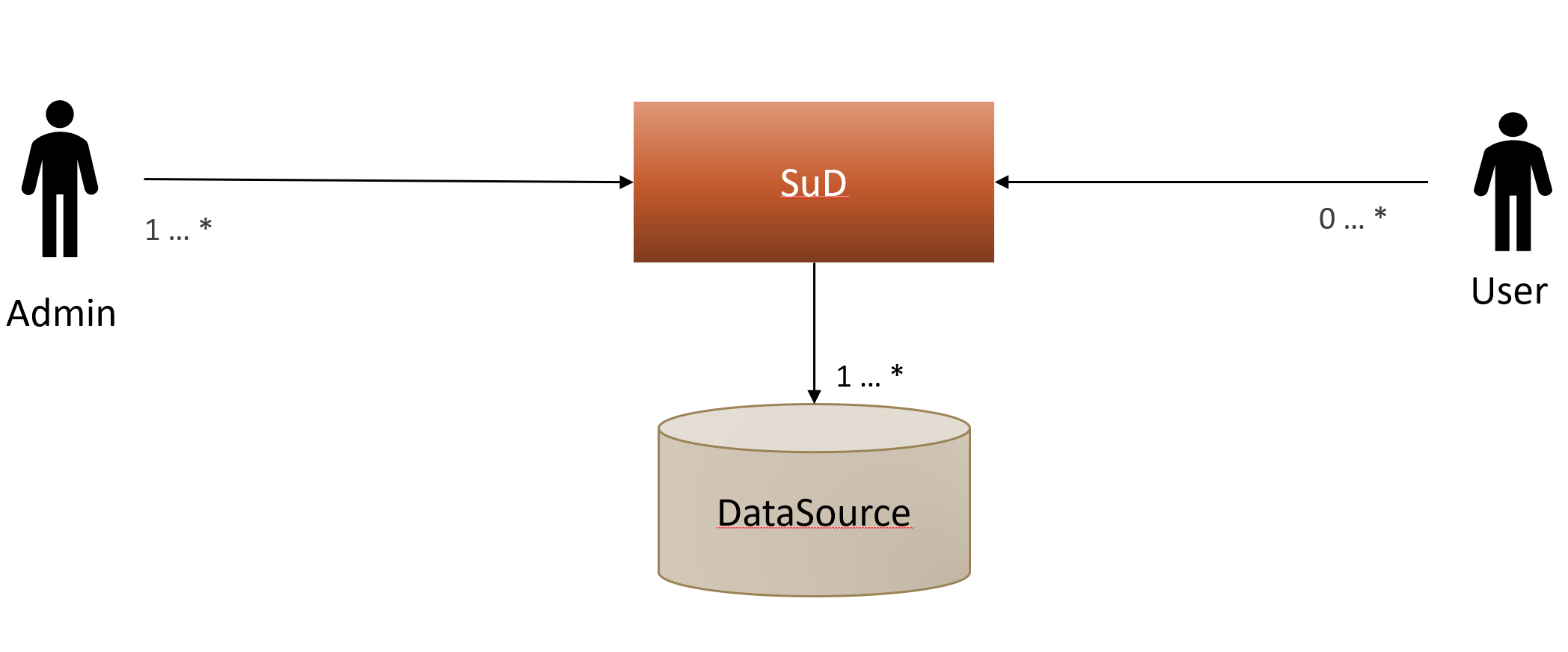


Figure 9 – basic context view

The basic context view of the system shows that there aren’t many external actors/systems in the basic context. As the WWTP project is taken as a proof of concept, the context view gets a little more concrete knowing one of the data sources which will be used:

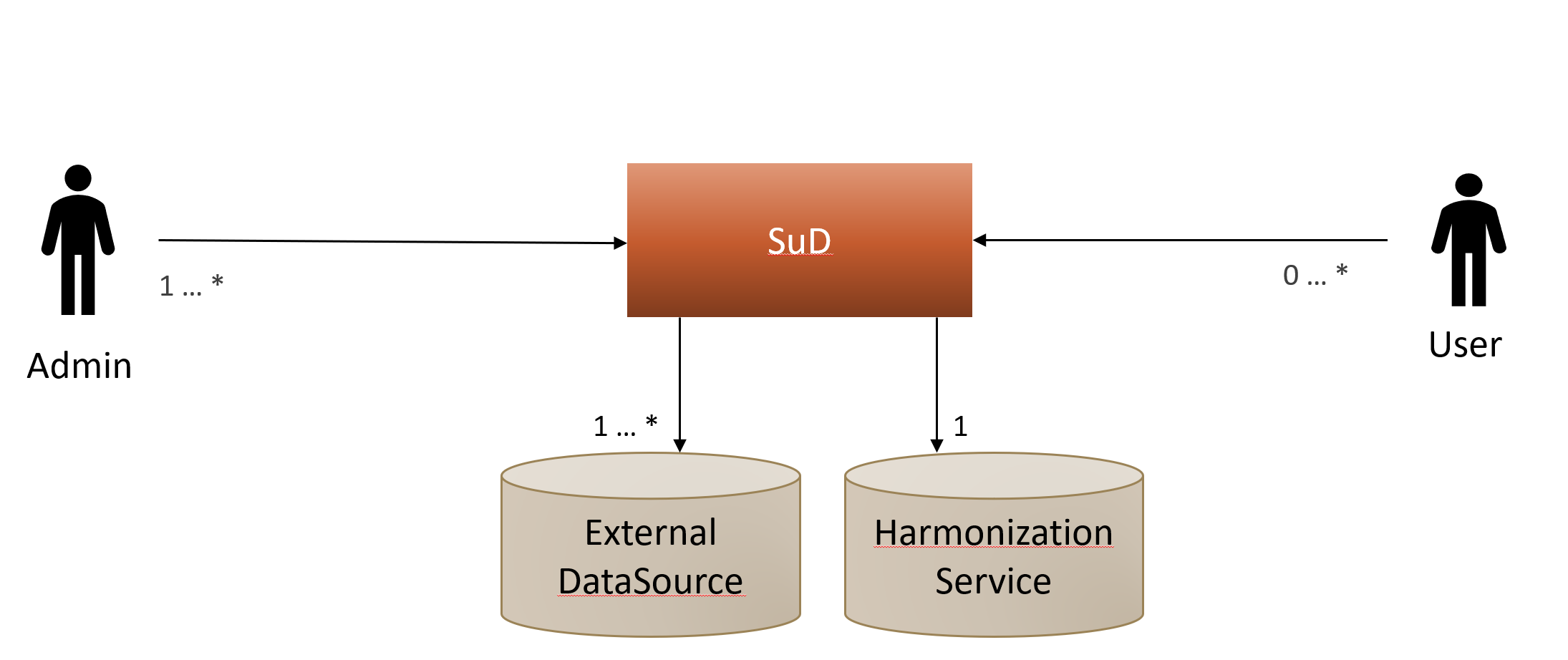


Figure 10 - context view in WWTP example context

### Administrator

The Admin user is member of the company which provides the this cloud service and he uses the system to configure user roles, users, data sources and individual views.

### User

The user is the customer who logs into the web application to see the his individual view (tables, diagrams) which was configured beforehand by an administrator.

### HarmonizationService

The HarmonizationService is the primary data source for the data analysis and data representation. Actually, this is just an external data source, but in the concrete example of the WWTP project it is treated specially.

### External data source

External data sources serve as the basis of the data integration and data analysis which is performed by the system. There is no limitation regarding the amount of external sources. As mentioned before, the DataAggregator is an external source as well, but treated specially here. In the WWTP example an external data source might be a weather provider or a database providing legal regulations for the industry.

## Use cases

Das Ziel jedes Softwareentwicklungsprozesses ist es, eine Software zu entwickeln, die ganz bestimmte Anforderungen erfüllt. Die Entwicklung einer Software fängt mit der Zielsetzung an: Die Software soll, wenn fertiggestellt, die zu Beginn des Entwicklungsprozesses festgelegten Anforderungen erfüllen. Niemand entwickelt Software grundlos und wartet darauf, dass zufällig eine für irgendetwas brauchbare Software vor seinen Augen entsteht. Es gibt immer Ziele, die erreicht werden sollen.

Leider sind diese Ziele nicht immer klar definiert. Man nimmt sich zum Beispiel vor, einen Online-Shop zu entwickeln, und stellt dann während des Entwicklungsprozesses fest, dass es unendlich viele unterschiedliche Funktionen in einem Online-Shop geben kann und man sich eigentlich nie klar gemacht hat, was man denn nun für Funktionen im Detail braucht. Man muss den Entwicklungsprozess daher wiederholt unterbrechen und inne halten, um sich zu überlegen, welche Funktionen, die einem bei der Entwicklung gerade eingefallen sind, notwendig sind und welche nicht.

Besonders schwierig wird die Situation, wenn der Auftraggeber des Entwicklungsprozesses nicht gleichzeitig der Entwickler ist. In diesem Fall kann der Entwickler nicht entscheiden, welche Funktionen notwendig sind - dies weiß nur der Auftraggeber. Dies führt zu einem ständigen Frage-Antwort-Spiel zwischen Entwickler und Auftraggeber, wenn der Entwickler nicht - noch schlimmer - die Entscheidungen selbst trifft und hofft, dies jeweils im Sinne des Auftraggebers zu tun.

Wenn Anforderungen an die zu entwickelnde Software nicht zu Beginn des Entwicklungsprozesses klipp und klar sind, wird der Entwicklungsprozess an sich unnötig erschwert. Denn das, was Sie entwickeln, richtet sich nach den bekannten Anforderungen. Jede Anforderung, die Ihnen oder Ihrem Auftraggeber später einfällt, führt dazu, dass Sie das, was Sie bisher entwickelt haben, ändern müssen. Denn die neue Anforderung hatten Sie logischerweise in Ihrer bisherigen Entwicklung nicht berücksichtigt. Grundsätzlich gilt, dass je später Anforderungen in einem Entwicklungsprozess bekannt werden, umso aufwändiger und daher teurer der Entwicklungsprozess wird. Anders gesagt: Wenn alle Anforderungen von Anfang an bekannt sind, bevor der Entwicklungsprozess gestartet wird, wäre das ideal.



Master User

Service User

<<include>>

Figure 11 – Use Case Diagram

### Short-descriptions of the use cases

The three use cases shown in the use case diagram are described briefly in the following section to get an understanding of the target system’s mechanisms and to help identifying the required components and possible technical issues.

### Use-case “Manage Users/Roles”

The master user or administrator, who belongs to the service providing company is able to manage the user roles and the specific users. Managing these subjects means to create, edit and delete user roles and specific users.

### Use-case “Configure Views”

The master user is able to configure the individual view for a specific user or user role. A view consists of one or multiple GUI-components representing a data set. An example for a view is a table with real-time sensor data along with a diagram which visualizes the development of sensor data over the last 24 hours. The master user is also able to define rules for the data representation, e.g. define limits for specific values while crossing these limits results in a change of the view.

### Use-case “Viewing information”

The service user (the customer) is able to see the pre-configured views after he logs in to the application. A view consists of one or more visualization components which may be updated if the source data changes. These updates should take place without reloading the web page.

## Domain Model

The creation of data models is often one the first steps in software development to build the foundation of the software architecture. Data models are independent of the used technology and can be used for different target systems, e.g a desktop application, web service or a database. Data models are useful for developers to get an idea how to build the system, however it is also useful for customers to clarify the purpose of system together with the developers. In general, data models show objects of the real world and the static relations between these objects. In this project it is a bit different, because the system should be able to deal with data which is not known during development. So the data model must be build with generic data objects.

## Component diagram

In opposite to the context diagram of a system, the component diagram shows a white-box view. This means that the internal components of a system (or a subsystem) are visible. In the series UML basics from IBM the component diagram is described as follows: *“The component diagram's main purpose is to show the structural relationships between the components of a system”* [13]

The UML specification states that a component is a replaceable, modular entity of a software-system with defined interfaces. A component might provide interfaces or might require interfaces, which is visualized in the component diagram. While the system is shown as a white-box in this diagram, the components within the system are visualized as black boxes, which means that their internal structure is hidden.

ConfigDB

GUI

Data Provider

Communication Service

Configuration Service

### GUI:

The GUI-component contains the logic for the user interface presentation. It calls the configuration service to retrieve information about which components need to be rendered for the current user. It provides the interface to the outside world (i.e. to the internet browser).

### Configuration Service:

The Configuration Service component contains the main business logic of the system. It accesses the ConfigDB to retrieve and store information about users, user roles and their respective user views. It calls the Data Provider to retrieve the data from the external data sources to update or initialize the GUI-components for the user views.

### DataProvider:

The Data Provider component handles the data (e.g. values from water plants) from the data sources. It calls the Communication Service to retrieve the current data, processes this data and provides the processed data to the Configuration Service.

### CommunicationService:

The Communication Service contains the logic for the actual communication with the data sources. When called by the Data Provider, it requests the configured data sources via HTTP requests (in general with HTTP method GET). This component should be built expendable, so that it would be possible to use alternative communication ways in a later step (e.g. message queues, direct DB access).

### ConfigDB:

The ConfigDB persists the configurations about users, user roles and data sources. The user and user role configuration must contain information about configured visual elements and their source data as well as general information about data sources, such as URL, request parameters or HTTP method. The data model for this database will be described in one of the following chapters.

### ER-model

As most of the configuration set by the administrator have to be persistent, the web application requires a database which is able to persistently store user and data source information. An important task preceding the implementation and integration of a database is to develop an ER-model (Entity-relationship model). The entity-relationship model serves the purpose of describing the structure and relations of the persistent data model and sets the basis for the design and implementation of the database.

UserRole

User

VisualizationComponent

DataSource

DataField

DataOperation

visualizes

is modified by

has

belongs to to

comes from

1

N

N

1

1

N

1

N

M

N

has

N

M

Figure 12 - ER-model

### Entitites

In the following section, the purpose of the database tables and their fields are described briefly. As this is the initial concept, it is not guaranteed that this model is the best fitting one for the requirements and use cases. If problem with this models are identified during the development, it will be addressed in a later chapter.

|  |  |
| --- | --- |
| *UserRole* |  |
| id (number) | The primary key |
| name (string) | The name of the user role |

Table 3 - User Role

The entity UserRole defines a groups of users and their default configuration.

|  |  |
| --- | --- |
| *User* |  |
| id (number) | The primary key |
| roleId (number) | The foreign key for the user role, this user belongs to |
| name (string) | The name of the user |
| Password (string) | The user password (base64 encoded) |

Table 4 – User

The entity user defines a specific user and stores the user’s credentials along with the associated user role.

|  |  |  |
| --- | --- | --- |
| *VisualizationComponent* |  | |
| id (number) | The primary key |
| type (number) | The type of the visualization component, such as “table”, “bar chart” or “pieChart” | |
| roleId (number) | The foreign key for the user role | |
| userId (number) | The foreign key for user | |

Table 5 – VisualizationComponent

The entity VisualizationComponent defines a visible, data presenting component, such as a table or a diagram. It also references the user and the user role, the component belongs to.

|  |  |
| --- | --- |
| *DataField* |  |
| id (number) | The primary key |
| dataSourceId (number) | The foreign key for the data source |
| vComponentId (number) | The Id of the visualization component where the data field is used |
| Key (string) | The name of the attribute |
| updateInterval (number) | The interval of time between data updates (polling) |

Table 6 - DataField

The entity DataField defines an attribute froma data source by referencing the data source and the attribute name.

|  |  |
| --- | --- |
| *DataOperation* |  |
| dataFieldId (number) | The foreign key of the data field |
| dataSourceId (number) | The foreign key of the data source |
| vComponentId (number) | The foreign key of the visualization component. |
| additionalValue (string) | Additional value for the operation, if required. |
| operationType (string) | The type of the operation, e.g. “sum, multiplication etc.” |

Table 7 – DataOperation

The entity DataOperation defines an mathematical operation (e.g a sum) which is applied to one or more data fields. It also provdes the possibility of applying an operation with an additional value, e.g. if miligrams should be converted to grams. In that case the field index would point to the milligram value in the VisualizationComponent entity (dataFields), the operationType would be a multiplication and the additionalValue would be 1000.

|  |  |
| --- | --- |
| *DataSource* |  |
| id (number) | The primary key |
| uri (string) | The uri under which the data source is located |
| queryParams (string) | Optional query parameters (key=value, serperated by “;”) |

Table 8 - Data Source

The entity DataSource defines an HTTP endpoint where data can be fetched. It is identified by the endpoint-URI and query parameters.

|  |  |
| --- | --- |
| *OperationType* |  |
| id (number) | The primary key |
| type (string) | The operation type which is applied for two values. Examples are “+”, “-“, “<”, “>=” etc. |

Table 9 - OperationType

OperationType is a list of applicable operations on the data values.

# Design and Implementation

The purpose of this chapter is to picture the final software archtitecture as well as the used cloud and programming technologies. The main advantages of the chosen technologies are shown and they are compared to alternative choices in some cases.

### Amazon Web Services

As the cloud environment, Amazon Web Services was chosen. The main advantages are the cost efficiency and the simplicity of usage.

### Programming Language and Frameworks

In this chapter the programming language and frameworks which were used during the implementation are described and their advantages will be highlighted to justify their use. There are several alternatives which could have been used for the implementation too, however their analysis won’t be a part of this chapter.

## Java

As the main programming language, Java is chosen, which is the most commonly used object-oriented programming language. There are plenty of Third-Party librarys able to solve specific problems or to simplify the development. Especially in this project where it’s necessary to use concepts like data serializing, database abstraction or HTTP requests, it is efficient to have access to proven librarys. Another advantage of Java is its portability and platform support. As a java application runs in a Java-virtual-machine (JVM), it is almost independent of the underlying operating system that means that the development process can be started even if the hosting environment is not known in the earyl phase.

## Vaadin

Vaadin is a free of charge Java-framework which is used to build Rich-Internet-Applications (RIA)[[3]](#footnote-3). In opposite to most frontend development librarys and plugins, Vaadin has a server-driven architecture, which means that most of the application logic runs on a server instead of in the browser. In this project, Vaadin has some advantages over traditional frontend development technologies (HTML/Javascript/CSS). A major advantage is that Vaadin’s purpose is to build web applications instead of just websites, so it comes with a programming model which is similar to the programming model of desktop applications, which simplifies the implementation of user interaction and the handling of user input events. [14] Vaadin comes with a built-in set of basic user interface components, like tables, combo-boxes and radio button groups and it also has an extension library which provides differen kinds of charts. Together with its drag-and-drop features these components and charts are very useful to implement the required use cases (e.g configuring a user view) efficiently and this makes Vaadin a well-fitting choice for this project.

## JPA and Hibernate

The JSR[[4]](#footnote-4) 338 states that *“the Java Persistence API is the Java API for the management of persistence and object/relational mapping in Java EE and Java SE environments. It provides an object/relational mapping facility for the Java application developer using a Java domain model to manage a relational database.”* [15]

JPA is an interface for Java applications, which simplifies the mapping of Java objects to database entries. In general, the state of a Java object is stored in memory as long as the current session is active. JPA provides the possibility to store Java runtime objects persistently, so that they can be used session independent. Even if relational databases aren’t made for object-oriented data structures, they can be used with JPA to persist Java runtime data.

JPA is generally used together with Hibernate. Hibernate is responsible for object-relational mapping (ORM). ORM is a technique to map runtime objects with their attributes and methods to database entries and to map these database entries back to their respective runtime objects. Dependencies between runtime objects are mapped to related database entries accordingly. As almost every database has its own SQL dialect, Hibernate needs to know the underlying database type, so that it can generate the correct SQL instructions.

In this project JPA and Hibernate are used to simplify the managing and the implmenetation of the configuration database which holds information about users, their roles and their configurations.

### Software Architecture

### Class diagram

Class diagrams are part of the UML (Unified Modeling Language) structure diagrams. These diagrams emphasize the things that must be present in the system being modeled. Since they represent the structure, they are used extensively in documenting the software architecture of software systems. [16]

The purpose of a class diagram is the abstraction of different objects and their relationship and interaction between each other. This enables the modelling of a clearly delineated system in object-oriented analysis and design.

KLASSENDIAGRAMM

# Results and Analysis

This chapter shows the outcome of the implementation and evaluates if the initial goals could be reached. The main focus will be on the UI interaction and the integration of the data harmonization service.

## UI interaction

One of the aims in this project was to build a platform with an easy to understand user interface which provides the possibility to create user views by drag and dropping the components of which this user view consists. This configuration can only be done by an admin user who has the privileges to create user views. For the identification of the current user the start page of the application shows a login screen. After entering the user name and the user password the application backend checks the input and chooses the next view depending of the user’s privileges:

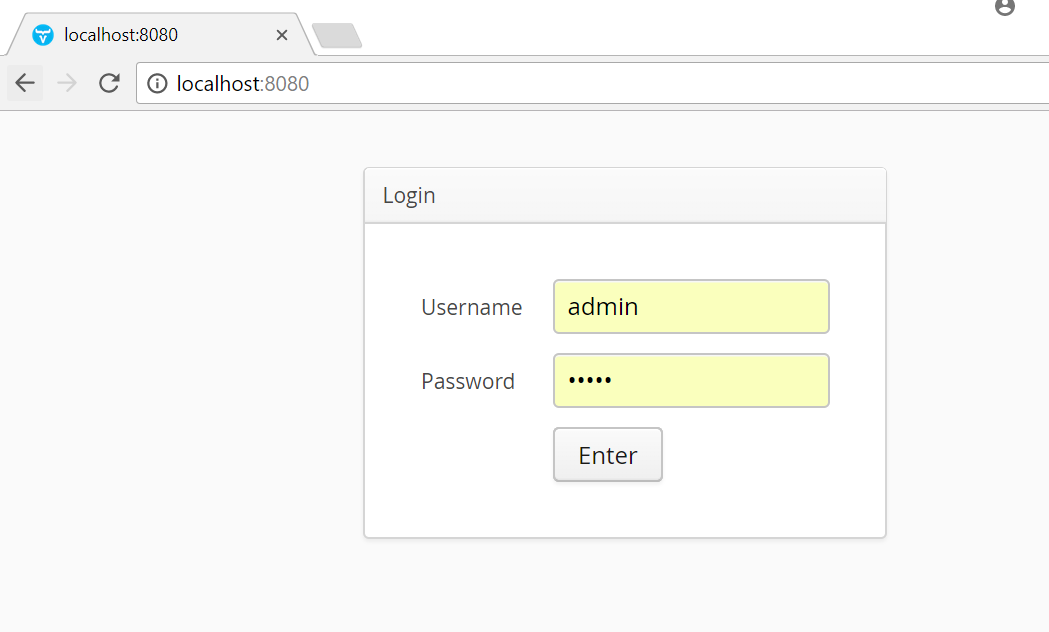
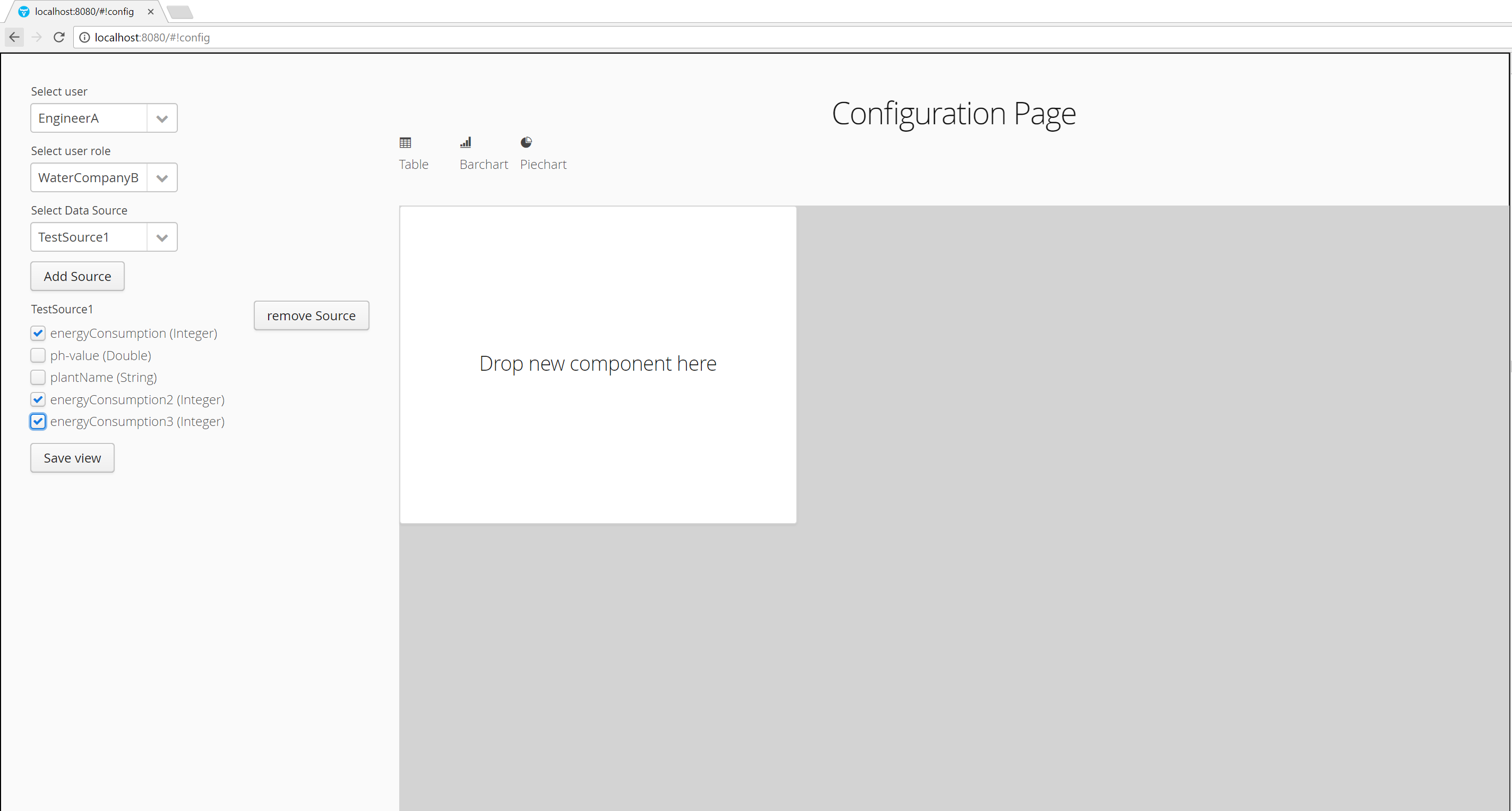


Figure 13 - screenshot user login

If an admin user logs in to the application he has the choice to configure data sources and user views. The data source configuration is shown in the sub chapter which is about the integration of the harmonization service. This chapter focuses on the user view configuration. The following screenshot shows the configuration page with an empty user view:



Draggable components

Figure 14 - screenshot user view configuration

### Draggable components

An important feature of the configuration UI is the building of an user view via drag and drop. The following screenshot shows the selection of the draggable components, indicated by a label and its corresponding icon:

Screenie

If a component is dragged and dropped to the highlighted are it is added to the view. If a data field of a data source was selected beforehand, then the view component will visualize the selected fields instantly. If no data field was selected, an empty component is added to the view. In both cases, the component is now in configuration mode and updates itself instantly when a data field is selected or unselected.

Screenie

Dragging and dropping a view component to the highlighted are will always add a new component to the view. The last added component is always in configuration mode until the “Save View” button on the left side is clicked. Other components can be edited by clicking on them.

### Component edit section

As soon as a component is in edit mode, the component configuration section appears at the bottom of the page. This section is indivudal for each view component and the reason for that is that each view component has individual configuration possibilities. In case of a bar chart, the label and the scale of the x-axis and the y-axis can be configured, while this isn’t the case regarding tables and pie charts. The following screenshot shows the configuration setting for a bar chart:

Screenie

In opposite to the data field selection, the view component isn’t updated until the “Save” (Wie heißt der?) button is clicked. If one of the inputs is invalid, a notification appears as soon as the “Save” button is clicked. The changes won’t be applied to the component in this case.

## Integration of the Harmonization Service

The data harmonization service is a web application which aggregates the sensor data of different water plants from different water companies and it converts the data from the provider and plant specific model to a common model. This harmonized data serves as the basis for evaluating the concept of this project.

It is necessary that a user is able to add and edit the data sources which are needed to perform data analysis and data visualization. For this purpose, the following data source overview was implemented and the URIs of the data harmonization API are added and persisted:

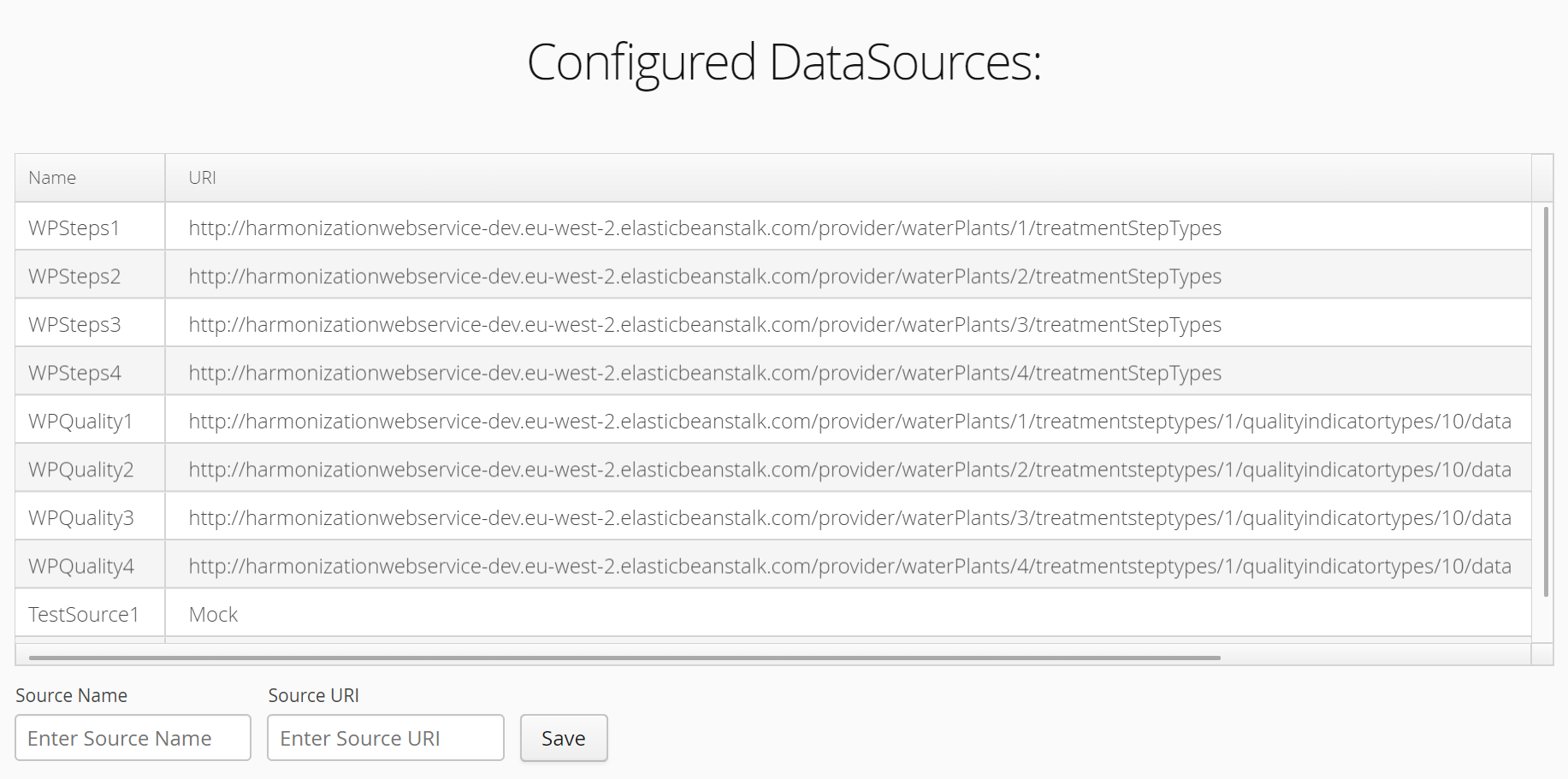


Figure 15 - Screenshot data sources

As these data sources are configured, they can be used to build user views by visualizing the data with view components. In the user view configuration the data sources can be chosen and added by selecting them in the combo box:

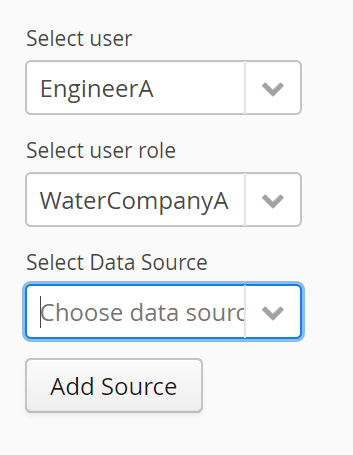


Figure 16 - data source select 1

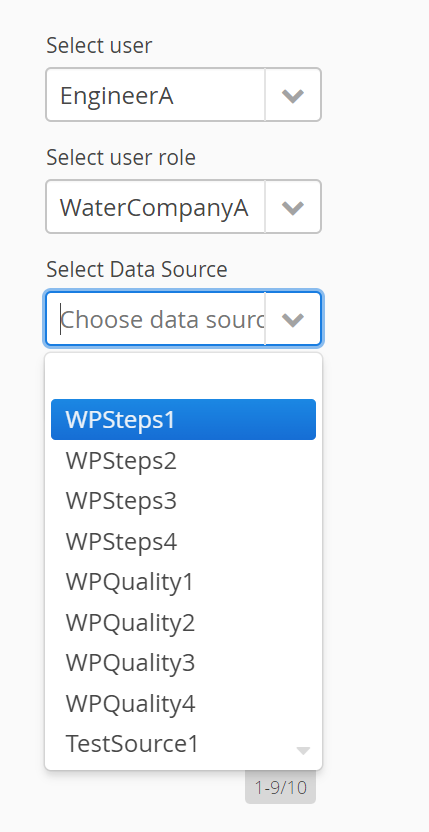


Figure 17 - data source select 2

After selecting the data source, a click on the “Add Source”-button leads to an HTTP request to the chosen endpoint. If the request is successful and data is returned, then this data will be parsed by the data provider component. This results in a list of data fields with their respective data type. These fields can be chosen for presentation by selecting the corresponding checkboxes:

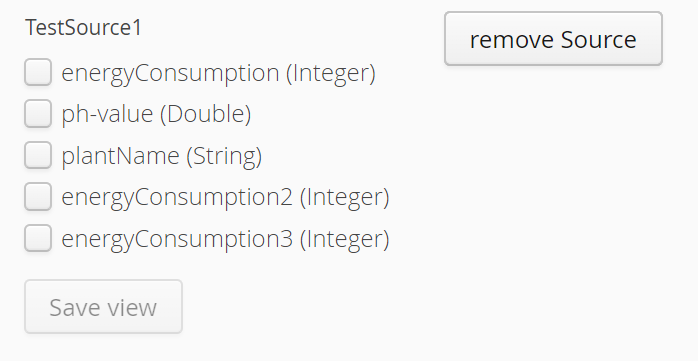


Figure 18 - screenshot data field selection

# Conclusions and further work

## Summary

At the end of the project, the work can be summarized as successful and the goals for this dissertation were reached. However the initial time calculation for the implementation task was a bit underestimated. The most time consuming task was the implementation of the UI logic and the styling. The vaadin framework provides an easy understandable UI component system which enables developers to create views in very short time. The disadvantage is however that when it comes to edit the style and the behaviour of the UI, the possibilities are limited with vaadin.So at that point, there is no way around using own CSS and integrate it with the application.

Another issue that didn’t work as planned was the database model described in a previous chapter. It got clear in the early development phase that the ER-model isn’t sufficient to persist the information about the visualization components, especially with the application of data operations on the data fields and the individual configurations of components like bar charts for example.

## Further work

Although the aims and objectives for this dissertation work could be reached, there are further improvements along with challenges which have to be worked on to reach the long-time goals.

At the current state the communication with external sources only work with HTTP GET requests. Besides creating the possibility to execute POST or other HTTP methods as well, there might be the wish to gather data with a completely different method, like reading directly from a database’s table or connecting to a message queue and receive updates regularly. Besides that, only endpoints without a security mechanism can be requested at the current state, so it would be necessary to develop a possibility to work with credentials and/or certificates to get the required data.

Another task would be to implement a fitting security system. At the current state users and their respective passwords are stored in the underlying database while the passwords are base64 encoded. This can be regarded as susfficient for a prototype application, but in a production environment with sensitive data, this might be too much of a risk and a better fitting security system should be integrated.

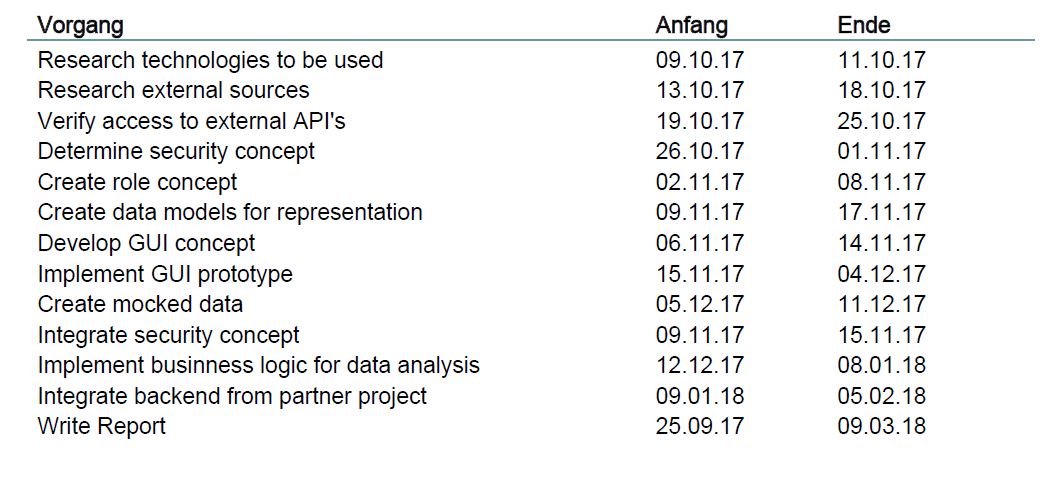
For the data representation, more view components can be integrated in the future. The Vaadin Charts plugin provides a large amount of different chart components where each component has specific advantages over the other ones in a specific case. This would lead to more flexibility in building a user view and to an improved user experience as the user view is individually build for showing the desired specific information.

The most challenging part in the further development is probably the building of the possibility to perform a more complex data analysis. With the prototypical implementation it is possible to build a presentation for values received from one or more data sources by drag and dropping the view components and selecting the data fields. However it is not yet possible to perform complex mathematical operations with multiple data values to get an visual output based on the results, which might be necessary to build an user view which contains recommendations about the user’s water consumption for example. Although there is much work to do in the future to achieve more flexibility in the data analysis, the foundation has been set successfully at the end of this dissertation project.

# Project plan

# Time-plan

The dissertation project should be finished until the end March 2018. To ensure that this project will be finished until then, an initial time plan is created which contains the key activities that have to be done with a time estimation. This time plan can be used as a tracking mechanism to detect possible delays early and take actions against it early (TODO: reformat time table):



Screenshot 19 - Time Plan

It is not guaranteed that all the tasks are found on creating the initial plan. It is likely that further tasks emerge during the research and the development. Nevertheless the initial time plan gives a good orientation which will help during the work process. The following section describes the tasks briefly to give a clearer overview and understanding over the project:

|  |  |
| --- | --- |
| Task | Description |
| Technology research | The initial research about the software technologies used to achieve the project aims. It is necessary to make decisions about several technologies like databases, cloud providers, frameworks etc.  Those choices have to be justified by showing the adventages over available alternatives. |
| External source research | There will be external sources which have to be accessed by the web app to enable analysis of different aspects. During this task the required external sources have to be identified. |
| API verification of external sources | After the identification, the access to the external sources has to be verified. It must be determined if these sources offer an external API and under which requirements they can be accessed. |
| Determine security concept | It is important that not every information is accessible to everybody, so a security concept is required to protect the web application. This concept has to be developed during this task, also considering provided services by the hosting provider. |
| Create role concept | There will be a role concept to distinct different users (e.g. admin user). The required roles have to be identified and the possibilities for the technical implementation must be verfied. |
| Create data models | A domain model has to be developed in this task. It should show the relations between the different entities and should be the basis for the presenetation of the data. |
| Develop GUI concept | Outcome of this task will be a first mockup of the web application GUI. Static views as well as possible user interaction should be part of this concept. |
| Implement GUI prototype | The outcome of the previous task should be implemented as a protoype with placeholders. |
| Create mock data | It is likely that it won’t be possible to access real data during the development, so it will be necessary to create mock data, which should be easy replaceable as soon as the integration of data providing systems is possible. |
| Integrate security concept | The defined security concept should be implemented in this task. |
| Implement business logic for data analysis | During this task the core of the web application should be implemented. The data from the different sources must be accessed and a data analysis will be performed which results in a meaningful visualization (HIER NOCH ETWAS DAZU SAGEN) |
| Integrate backend from other project | If possible, the data from the data aggregation and harmonization project should be accessed and used by the web application to perform the data analysis. |
| Write report | All the steps should be documented in this dissertation as well as the initial research. |

# References

|  |  |
| --- | --- |
| [1] | „Water supply,“ [Online]. Available: https://www.water.org.uk/about-water-uk/water-supply. [Zugriff am 24 09 2017]. |
| [2] | Water UK, „Water resources long-term planning framework,“ Water UK, 2016. |
| [3] | „Water UK,“ [Online]. Available: https://www.water.org.uk/about-water-uk. [Zugriff am 23 09 2017]. |
| [4] | „Regulation,“ [Online]. Available: https://www.water.org.uk/about-water-uk/regulation. [Zugriff am 24 09 2017]. |
| [5] | „DWI,“ [Online]. Available: http://www.dwi.gov.uk/about/index.htm. [Zugriff am 23 09 2017]. |
| [6] | „Watrer UK: Drinking water quality,“ [Online]. Available: https://www.water.org.uk/policy/drinking-water-quality/water-quality-standards. [Zugriff am 03 01 2018]. |
| [7] | „GUIDANCE ON THE IMPLEMENTATION OF THE WATER SUPPLY (WATER QUALITY) REGULATIONS 2000 (as amended) IN ENGLAND,“ 20th March 2012. [Online]. Available: http://dwi.defra.gov.uk/stakeholders/guidance-and-codes-of-practice/WS(WQ)-regs-england2010.pdf. [Zugriff am 4th January 2018]. |
| [8] | „Water,“ [Online]. Available: https://www.metoffice.gov.uk/services/industry/water. [Zugriff am 26 09 2017]. |
| [9] | „South East Water,“ [Online]. Available: http://www.southeastwater.co.uk/your-account/about-your-bill/water-charges/how-much-does-your-water-cost. [Zugriff am 29 09 2017]. |
| [10] | A. Mousavi und E. Katsou, „WWTP-Global-300617,“ 2017. |
| [11] | „What is Data Dissemination,“ [Online]. Available: http://www.learn.geekinterview.com/data-warehouse/dw-basics/what-is-data-dissemination.html. [Zugriff am 03 02 2018]. |
| [12] | S. Winter, „Quantitative vs. Qualitative Methoden,“ 15 05 2000. [Online]. Available: http://nosnos.synology.me/MethodenlisteUniKarlsruhe/imihome.imi.uni-karlsruhe.de/nquantitative\_vs\_qualitative\_methoden\_b.html. [Zugriff am 27 02 2018]. |
| [13] | D. Bell, „UML basics: The component diagram,“ IBM, 15 12 2004. [Online]. Available: https://www.ibm.com/developerworks/rational/library/dec04/bell/index.html. [Zugriff am 02 03 2018]. |
| [14] | „Vaadin FAQ,“ Vaadin Ltd., [Online]. Available: https://vaadin.com/faq. [Zugriff am 26 02 2018]. |
| [15] | „Java Community Process,“ Oracle, [Online]. Available: https://jcp.org/en/jsr/detail?id=338. [Zugriff am 18 03 2018]. |
| [16] | B. Padmanabhan, „Unified Modeling Language (UML) Overview,“ [Online]. Available: https://people.eecs.ku.edu/~hossein/Teaching/Fa13/810/Readings/UML-diagrams.pdf. [Zugriff am 2018 02 28]. |
| [17] | „Water UK,“ [Online]. Available: https://www.water.org.uk/news-water-uk/latest-news/household-water-and-sewerage-bills-2016-17. [Zugriff am 28 09 2017]. |
| [18] | „Data Integration,“ [Online]. Available: https://www.ibm.com/analytics/data-integration. [Zugriff am 03 02 2018]. |

1. The checks take place each year [↑](#footnote-ref-1)
2. National meterologic service of the UK [↑](#footnote-ref-2)
3. A Rich-Internet-Application is an interactice web application with many possible user interactions, comparable to desktop applications [↑](#footnote-ref-3)
4. JSR means „Java Specification Request”, which are formal documents that describe proposed specifications and technologies for adding to the Java platform. [19] [↑](#footnote-ref-4)