## 

**Master Dissertation**

**EE5500**

Markus Just

Student number: 1644609

Dr. Alizrea Mousavi

Electronic and Computer Engineering

School of Engineering and Design 

Sunday, March 25, 2018

Inhalt

[List of Figures 4](#_Toc509775410)

[List of Tables 6](#_Toc509775411)

[List of Abbreviations 7](#_Toc509775412)

[1. Introduction 8](#_Toc509775413)

[1.1 Background to the project 10](#_Toc509775414)

[1.2 Aims and Objectives 12](#_Toc509775415)

[1.3 Structure of this dissertation 17](#_Toc509775416)

[1.4 Project plan 18](#_Toc509775417)

[2. Literature Research 22](#_Toc509775418)

[2.1 The WWTP example 22](#_Toc509775419)

[2.2 Theoretical concepts 30](#_Toc509775420)

[3. Design and Implementation 36](#_Toc509775421)

[3.1 System requirements 36](#_Toc509775422)

[3.2 Context view 41](#_Toc509775423)

[3.3 Use cases 43](#_Toc509775424)

[3.4 Mock-Ups 46](#_Toc509775425)

[3.5 Component diagram 50](#_Toc509775426)

[3.6 Technologies 54](#_Toc509775427)

[4. Results and Analysis 59](#_Toc509775428)

[4.1 UI interaction 59](#_Toc509775429)

[4.2 Integration of the Harmonization Service 66](#_Toc509775430)

[4.3 UI Testing 71](#_Toc509775431)

[5. Conclusions and further work 74](#_Toc509775432)

[5.1 Summary 74](#_Toc509775433)

[5.2 Further work 75](#_Toc509775434)

[References 78](#_Toc509775435)

# List of Figures

[Figure 1 - water treatment [1] 10](#_Toc509746652)

[Figure 2 – Population Growth [2] 12](#_Toc509746653)

[Figure 3 – WWTP system overview [3] 14](#_Toc509746654)

[Figure 4 – purpose of the system 16](#_Toc509746655)

[Figure 5 - Screenshot gantt chart 20](#_Toc509746656)

[Figure 6 - how the price is set [10] 29](#_Toc509746657)

[Figure 7 - system participants in the WWTP example 30](#_Toc509746658)

[Figure 8 – basic context view 42](#_Toc509746659)

[Figure 9 - context view in WWTP example context 43](#_Toc509746660)

[Figure 10 – Use Case Diagram 45](#_Toc509746661)

[Figure 11 - example view 48](#_Toc509746662)

[Figure 12 - example view 2 49](#_Toc509746663)

[Figure 13 - user management 50](#_Toc509746664)

[Figure 14 - screenshot user login 61](#_Toc509746665)

[Figure 15 - screenshot user view configuration 61](#_Toc509746666)

[Figure 16 - Draggable components 62](#_Toc509746667)

[Figure 17 - Empty chart 63](#_Toc509746668)

[Figure 18 - User view configuration 65](#_Toc509746669)

[Figure 19 - Login EngineerA 65](#_Toc509746670)

[Figure 20 - User view 66](#_Toc509746671)

[Figure 21 - Screenshot data sources 67](#_Toc509746672)

[Figure 22 - data source select 1 67](#_Toc509746673)

[Figure 23 - data source select 2 68](#_Toc509746674)

[Figure 24 - screenshot data field selection 69](#_Toc509746675)

# List of Tables

[Table 2 - Dissertation structure 19](#_Toc509567692)

[Table 3 - User Role 61](#_Toc509567693)

[Table 4 – User 61](#_Toc509567694)

[Table 5 – VisualizationComponent 62](#_Toc509567695)

[Table 6 - DataField 63](#_Toc509567696)

[Table 7 – DataOperation 63](#_Toc509567697)

[Table 8 - Data Source 64](#_Toc509567698)

[Table 9 - OperationType 64](#_Toc509567699)

# 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.

## 1.1 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]

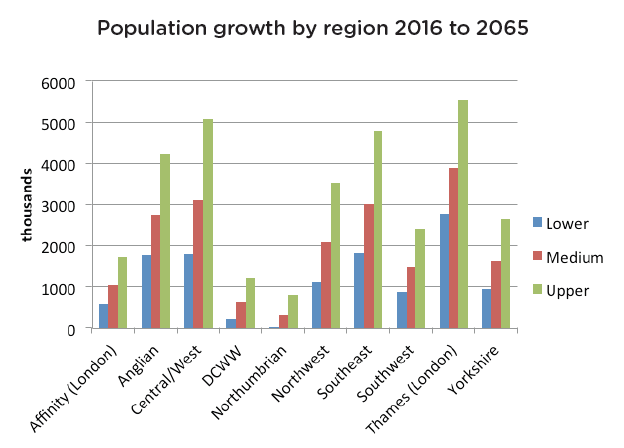


Figure 2 – Population Growth [2]

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.

# 1.2 Aims and Objectives

The main objective of this project is to develop a generic cloud service provider which is able to integrate, 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 3 – WWTP system overview [3]

The illustration shows that the water and wastewater treatment plants of different water providers are equipped 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 3.1. 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.

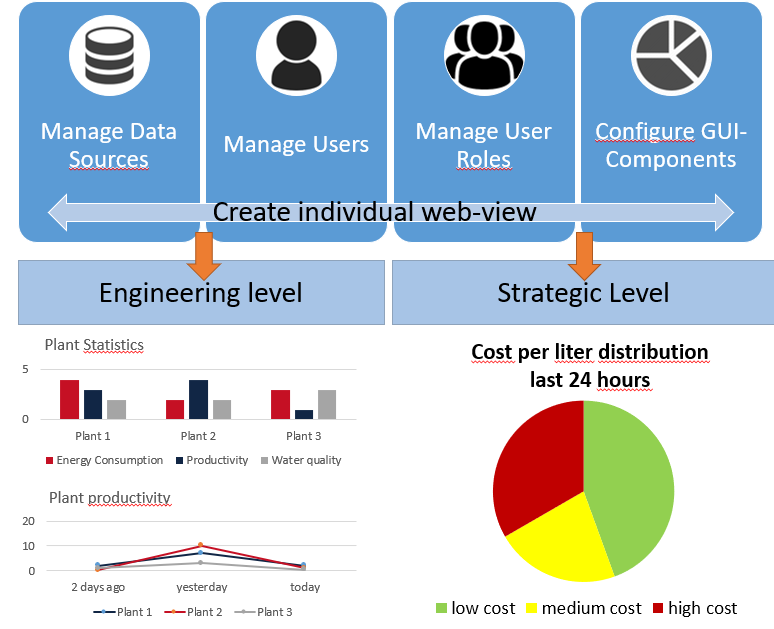


Figure 4 – purpose of the system

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

# 1.3 Structure of this dissertation

This master dissertation consists of five main chapters. The **Introduction** chapter gives an overview about the background, the general idea of this project and the project plan. The definition of the aim and objectives are also part of the first chapter. The second chapter is the **Literature Review** chapter and it explains the basic **theoretical concepts** behind this project and gives more insight into the WWTP example. 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

# 1.4 Project plan

The dissertation project should be finished until the end March 2018. To ensure that this project will be finished until then, a 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:

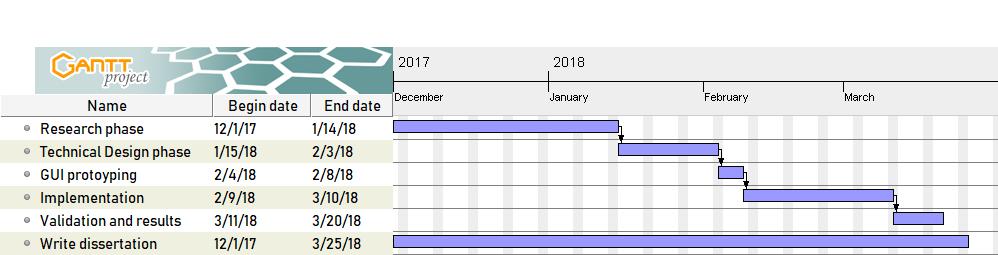


Figure 5 - Screenshot gantt chart

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 time plan gives a good orientation which will help during the work process. The following table describes the tasks briefly to give a clearer overview and understanding over the project:

|  |  |
| --- | --- |
| Task | Description |
| Research phase | The research phase marks the beginning of the project. In this phase information regarding the basic concepts and the WWTP example is gathered from different sources. |
| Technical Design Phase | 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. These choices have to be justified by showing the advantages and disadvantages over the available alternative choices. Another part of this project phase is the conception of the technical architecture, where the outcome will be the first (UML-) diagrams showing the system’s components. |
| GUI prototyping | After the technical research and design is finished, the conception of the user interface is started. In this phase the use cases for the application are discovered and the conception of the frontend is done by creating mock-ups with the assistance of available tools. |
| Implementation | The implementation phase is, apart from writing the dissertation, the most time-consuming phase during this project. The following goals should be achieved during this phase:  - setting up the development environment  - integrate frameworks  - implement the frontend  - implement the business logic  - implement persistence layer  - deploy to cloud environment  - intergate data harmonization service |
| Validation and results | In this phase the outcome of the implementation is analyzed and validated. The interaction with the user interface is tested and the result of the integration of the WWTP data harmonization project should be analyzed. |
| Write Dissertation | The reporting of the project work is always done in parallel to the other phases to enable the retracing of the project work and to document the conception as well as the final outcome. |

# Literature Research

The following chapter focuses on two main topics regarding the project. In the first part the WWTP example is taken to show which different factors have to be considered and how many data is required to enable the creation of a data visualization leading to a possibility of strategic decision making. The second part focuses on the thereotical concepts which serve as the basis for the implementation work, such as the concept of data integration or data visualization. The content of this chapter is mainly based on the literature research done in the early phase of the project.

## 2.1 The WWTP example

As it is important to understand why it is necessary to enable the integration of multiple different data sources, the following section gives an overview about different aspects, which has to be considered when a data analysis should be performed, aiming towards producing information for different kinds of consumers and stakeholders in WWTP process.

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 achieved by another project, which is called “data harmonization service”. 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”. [4]

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.” [5] 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:

[**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. [5]

[**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. [6]

[**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. [5]

[**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. [5]

[**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. [5]

### 2.1.1 Drinking water quality regulations

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). [7]

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. [7]

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. [7] 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. [8]

### 2.1.2 Further aspects

An external influence on the WWTP process is the current weather/climate 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 [9]: “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):

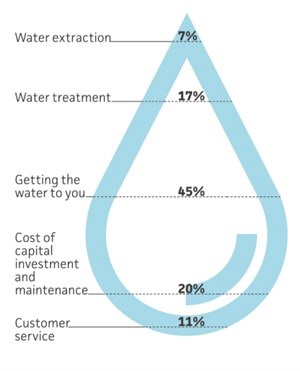


Figure 6 - how the price is set [10]

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]

In summary it can be said, that all the different kind of stakeholders (consumers, engineers, political decision makers) have different kind of interests in the data, which has to be considered when performing a data analysis with a visualized result and this indicates how crucial it is to provide a possibility for working with different data sources at the same time while producing a visual result based on the provided data.

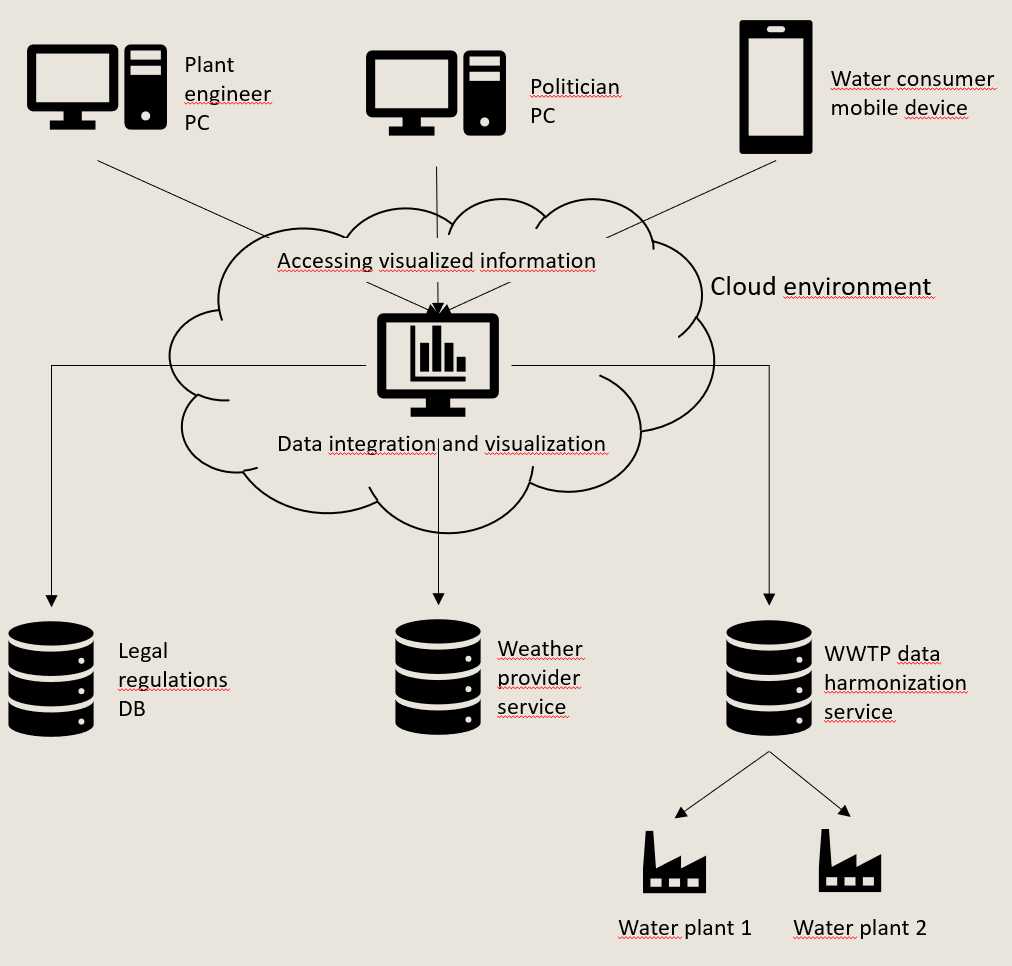


Figure 7 - system participants in the WWTP example

## 2.2 Theoretical concepts

The expected outcome of this project is based on several theoretical concepts which are mainly related to the topic “data”. This chapter aims towards giving an understanding of these concepts and clarifies how they are related to this project.

### 2.2.1 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]

### 2.2.2 Data integration

Data integration can be described as bringing together information from different data pools (data sources), which often differs in its structure (or it isn’t structured at all). Converting this data to a common structure is part of the data integration process which aims towards creating the possibility to use this data effectively. [12] Data integration is necessary when it is demanded that multiple, already grown, systems should be interconnected to solve an existing problem or to reach a specified goal. An example where data integration could be necessary could be bringing together different companies, work processes or different software applications. An implementation of data integration is for example a company’s data warehouse. A data warehouse is central database which stores data collected from differen kind of data sources for the purpose of performing data analysis based on this data. This makes sure that all the required data is present at a central place and it is ensured that all corresponding data is also present.

As already mentioned, one of the challenges in the data integration process is that it is often required to work with different structured data or with completely unstructured data. Unstructured data could be a text or a picture. Structured data means data which follows specified rules, so that a consumer knows what to expect when requesting this data. Structered data might be data sets from a relational database or the response of web service. When it comes to web services, there are several data formats which are commonly used to exchange data, for example XML, JSON or YAML. For the protoype developed during this dissertation work, the focus will be on data sources which provide the data as JSON, which became a very commonly used data type over the last years. Compared to XML, which was the standard data exchange format over many years, it is easier to read and easier to handle for application developers. However in the long-term the application should be able to deal with XML, YAML and databases as well.

### 2.2.3 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.

### 2.2.3.1 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.

### 2.2.3.2 Quantitive analysis

Quantitive analysis is about describing behaviour with models, correlations and numeric manifestations and enabling forecasting based on the results. [13] 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.

### 2.2.4 Data Visualization

Data visualization can be considered as the central topic of this project. It can be defined as “the use of images to represent information.” [14] Using it correctly may result in many potential benefits for business sectors or for science, however if there is lack of knowledge about how to use this “tool” correctly, it may produce the opposite of the intended effect, confusing rather than understanding. [14] Data visualization is basically achieved by the use of tables, graphs and diagrams. Historically, tables were the first manifestation of data visualization in human history. The earlies table discovered was created in the 2nd century in Egypt. [14] Although a table is only a textual representation of data, it is able to achieve the mediation of further information, simply by arranging the data in columns and rows. [14] Today there are many different techniques to visualize data, mostly in the shape of two-dimensional graphs. Given the fact that there is a great variety of graphs and charts today, it is very crucial to know which one to use to produce the information necessary to answer a given question. To visualize simple sales figures a classic bar chart, which is segmented into the 12 months of the year, may be sufficient. More unusual chart types, like a heatmap is able to suit the representation of percentual changes, for example. [15] It is important to know what should be achieved and which kind of information should be extracted from available data, when choosing the visualization technique. If the correct tools (i.e. graphs and/or tables) are used, it is possible to discover formely unknown relations between different aspects, which could lead to an answer of a scientific issue or to a new strategy leading to economic success. In the end, the concept and the utilisation of data visualization has increased rapidly over the last years and the importance is still growing, especially in combination with big data and cloud technologies.

# Design and Implementation

The purpose of this chapter is to picture the final concept of the cloud service provider application 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.

## 3.1 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.

## 3.2 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.

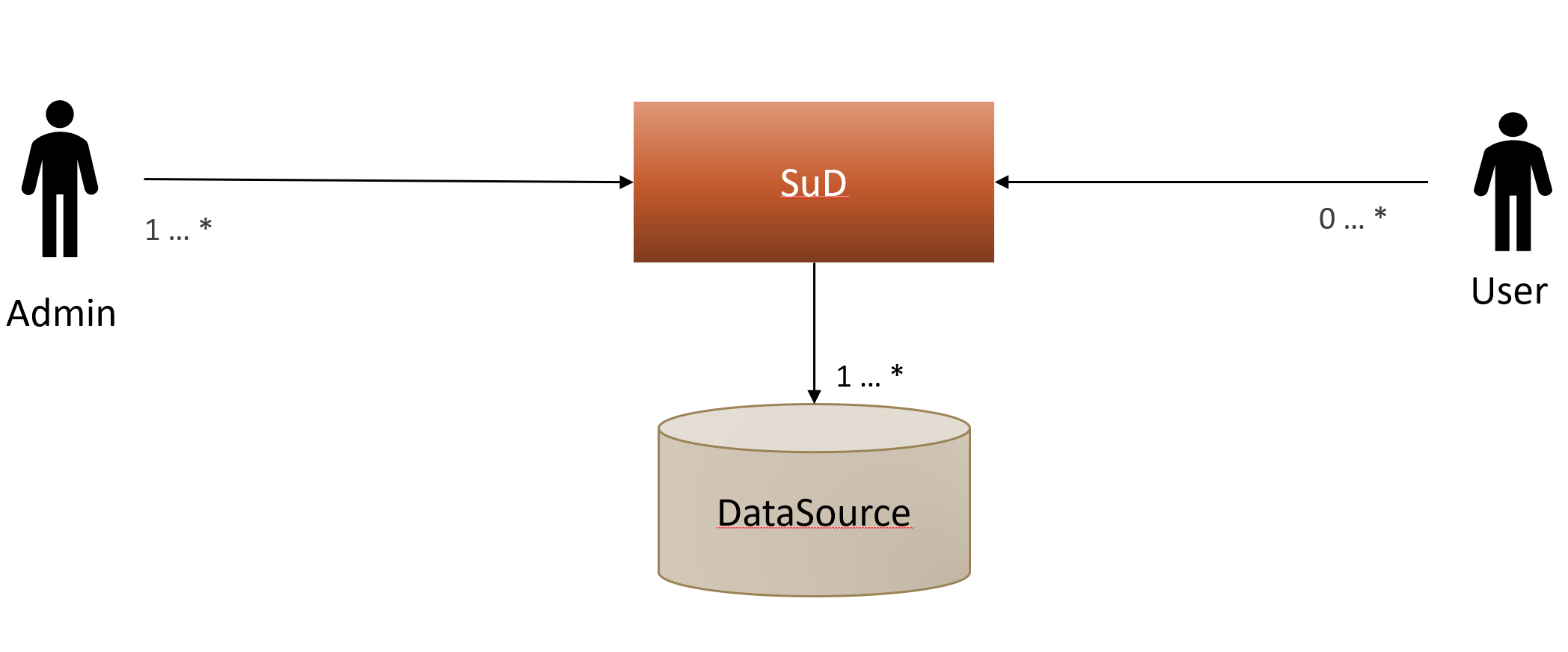


Figure 8 – basic context view

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:

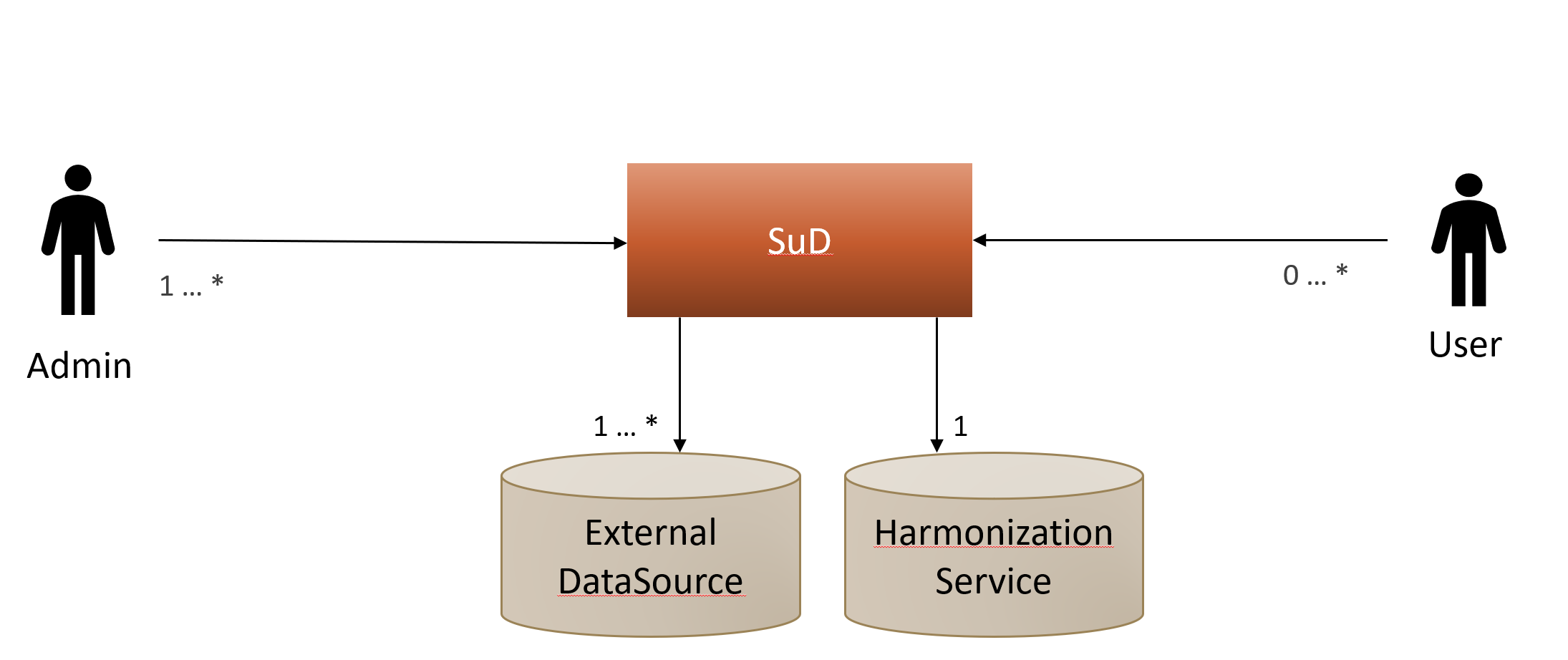


Figure 9 - context view in WWTP example context

### 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.

## 3.3 Use cases

The aim of each software development process is to develop a software which fulfills a defined set of requirements. In general, these requirements are determined at the beginning of the development process and they are the aims which should be achieved by the final product. However, the goals aren’t always defined cleary and completively. It is often the case that not all functions a software product should provide are known at the beginning of the development. A fitting example of this problem could be the development on an online shop. It is not unlikely that it is ascertained, during development, that an online shop could have almost an indefinite amount of different fuctionalities and it isn’t really clear what functionalities the online shop should really provide in the end.

A technique to prevent the occurrence of such scenarios, or to at least minimize the risk, is the use case diagram. The purpose of the use case diagram is to create an order for the many, sometimes very detailed, requirements, by giving an overview of the main features and functionalities the final software should provide. Technical and implementation details should not be considered when creating a use case diagram, because it should help to understand what the final product should achieve and not how it should do that. The use case diagram of the cloud based service provider looks as follows:



Master User

Service User

<<include>>

Figure 10 – 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 should be able to manage the user roles and the specific users as well as data sources. Managing these subjects means to create, edit and delete user roles, users and data sources.

### 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.

## 3.4 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.

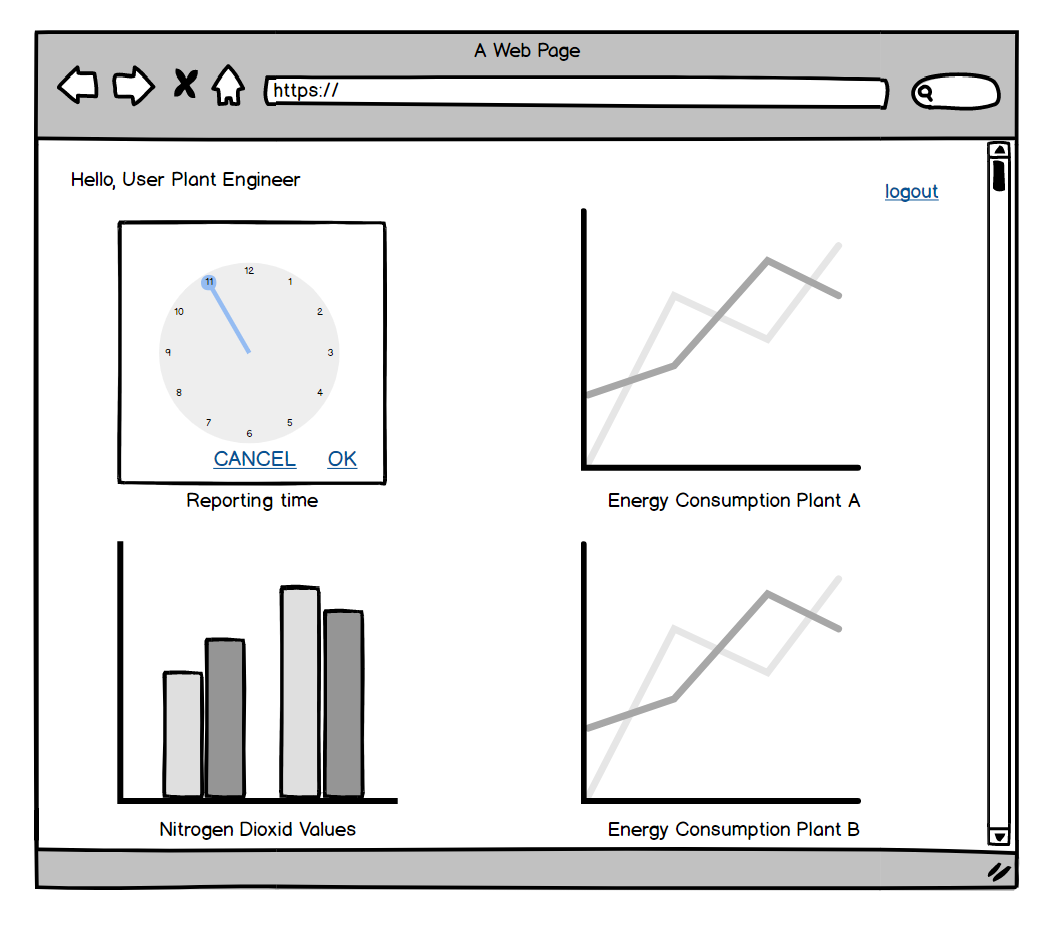


Figure 11 - example view

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.

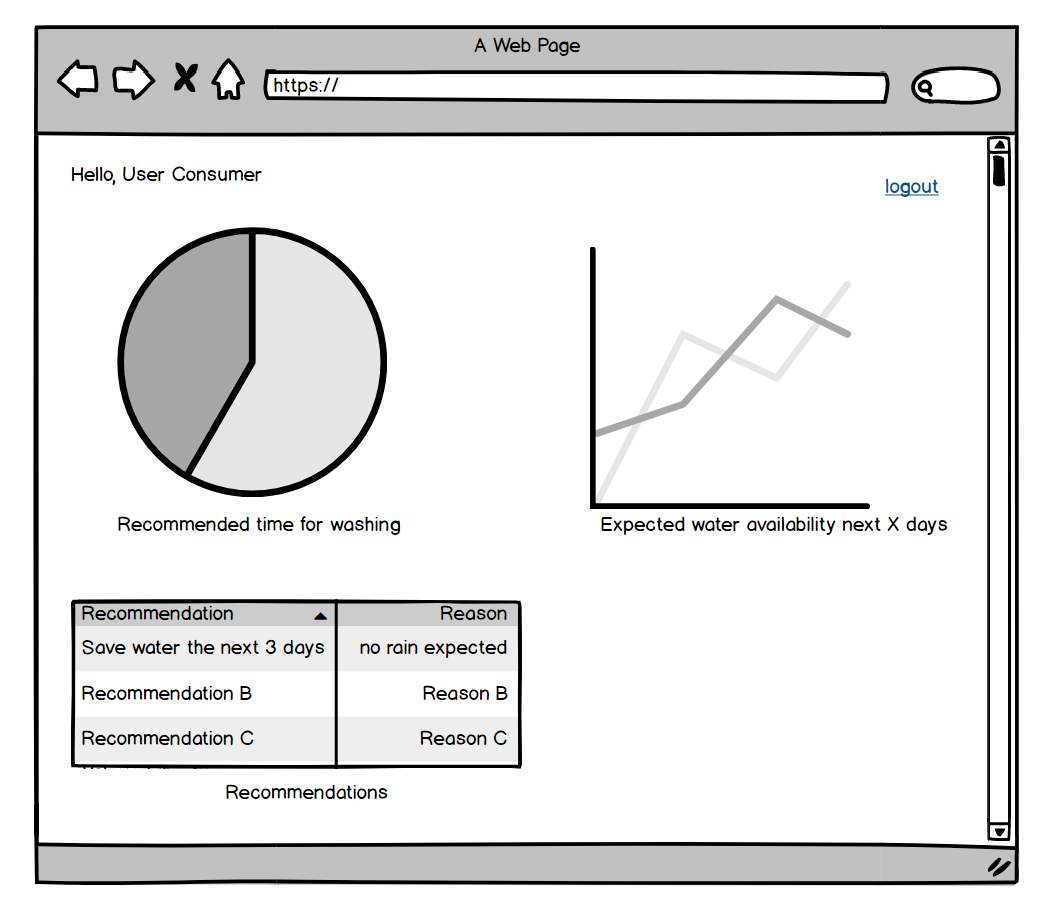


Figure 12 - example view 2

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:

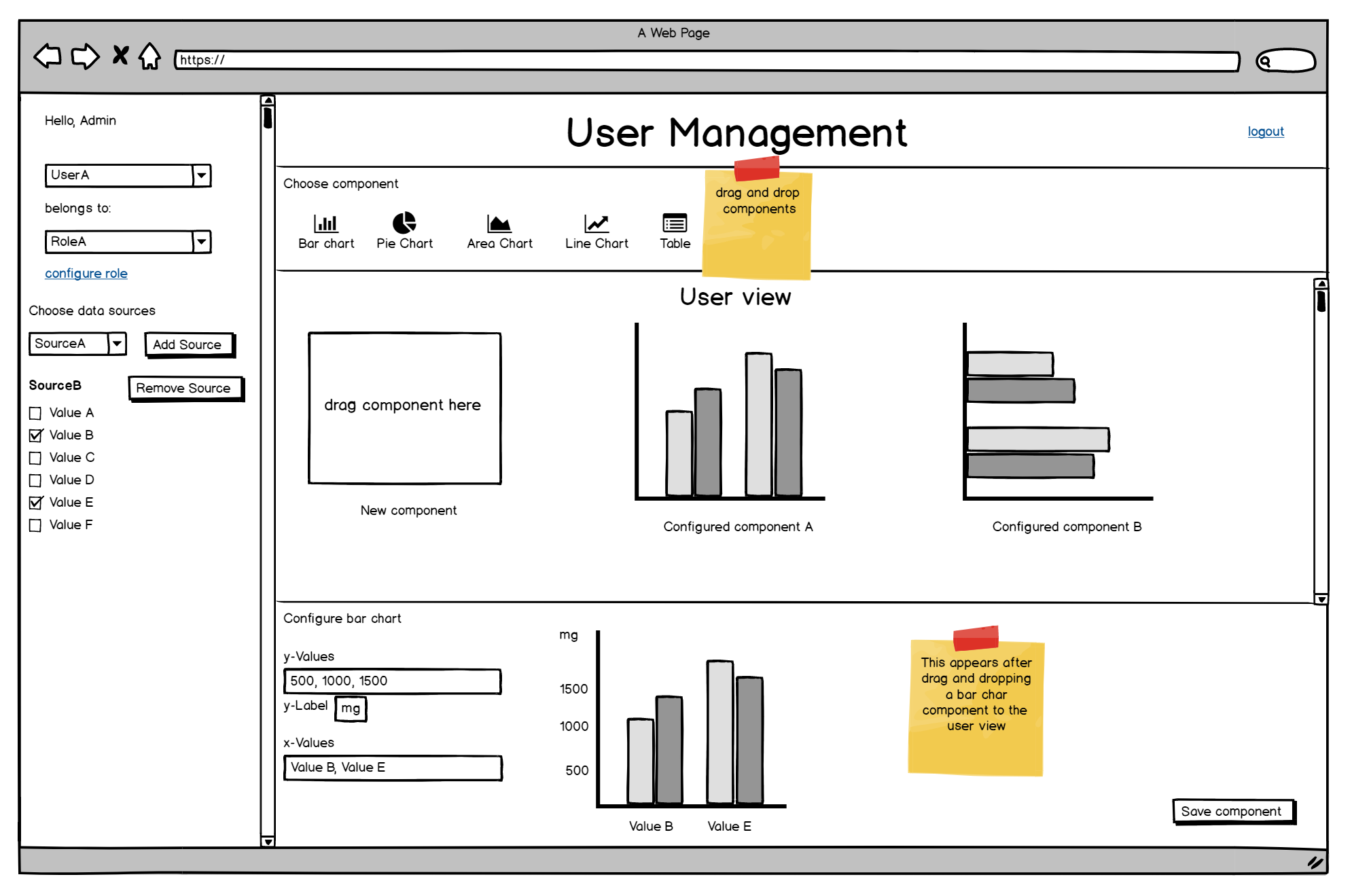


Figure 13 - user management

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).

## 3.5 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”* [16]

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 extendable, 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 entity relationship model isn’t created beforehand, instead the approach “code first” is used within the utilization of the Java frameworks JPA and Hibernate, which will be described in chapter 3.6

## 3.6 Technologies

This chapter provides a short overview about the utilised technologies chosen to realize the desired system. The main advantages of these technologies are briefly enumerated. However, not every programming framework or tool will be mentioned in this chapter as the goal is to show and explain the choices which are really relevant for the outcome.

### Amazon Web Services

As the cloud environment, Amazon Web Services is chosen. The main advantages are the cost efficiency, the simplicity of usage and the large amount of available services which support the main advantages of cloud environments like scalability, flexibility and cost efficiency. The “Amazon Relational Database Service” provides a large amount of relational database engines which can be used for the ConfigDB of the developed web app. The “AWS Elastic Beanstalk”-service is an easy way to host a web application while having the possibility to scale it by using further instances and it provides more features like automatic cpu scaling and monitoring. These services are also free to use as long as several limits aren’t exceeded which makes AWS a well fitting choice for this project.

### 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. [17] 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.”* [18]

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.

# 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.

## 4.1 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:

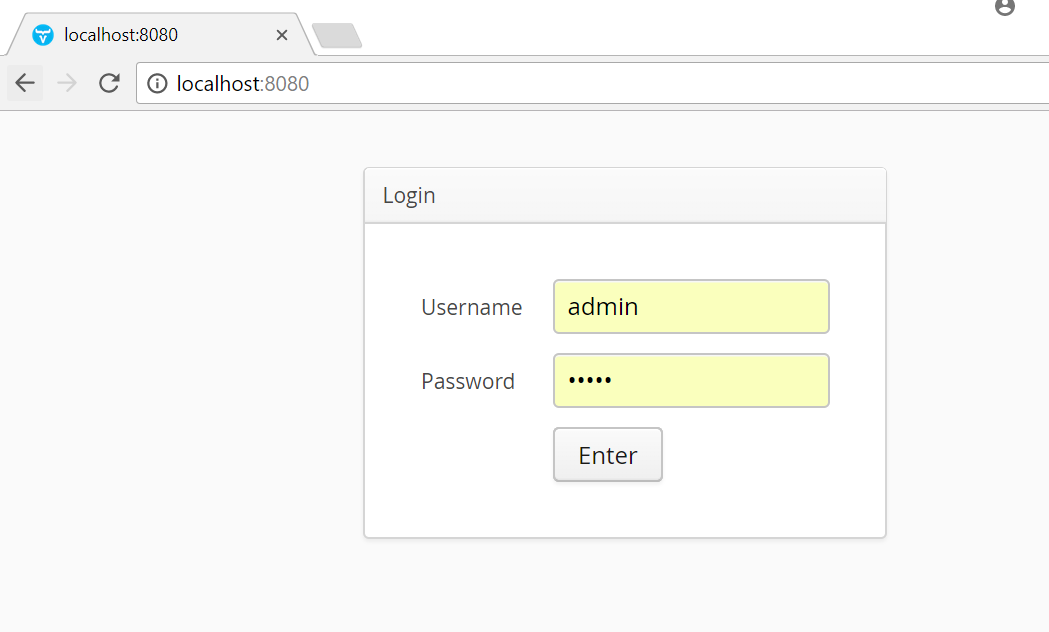
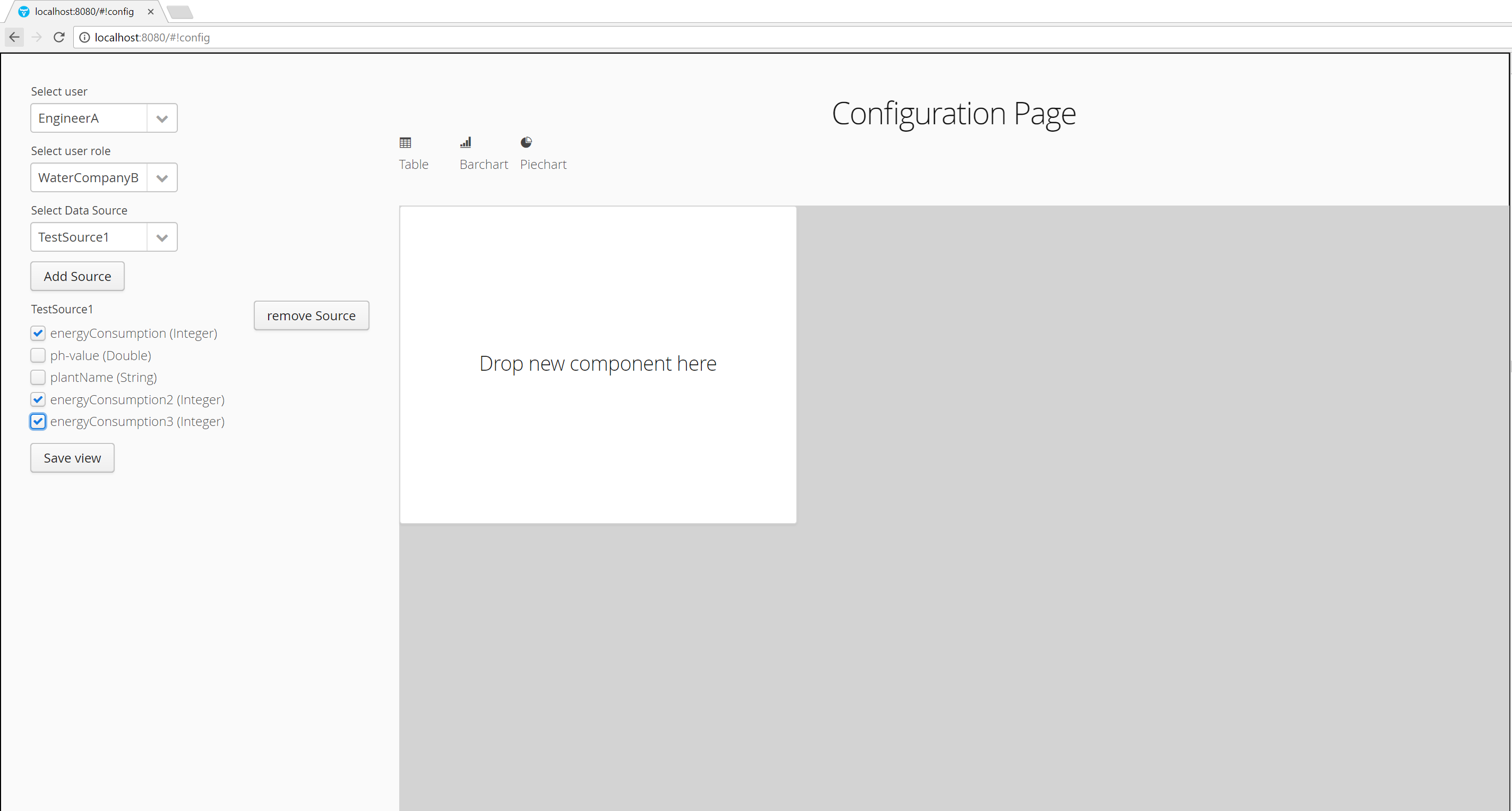


Figure 14 - screenshot user login

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 15 - screenshot user view configuration

### 4.1.1 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:

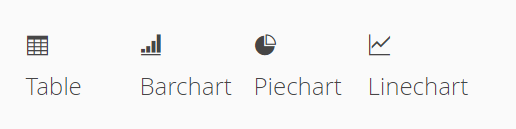


Figure 16 - Draggable components

If a component is dragged and dropped to the highlighted area 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 or when the component is modified in the component settings view at the bottom of the page. The following screenshot shows an empty linechart component, because no data source and data field was selected:

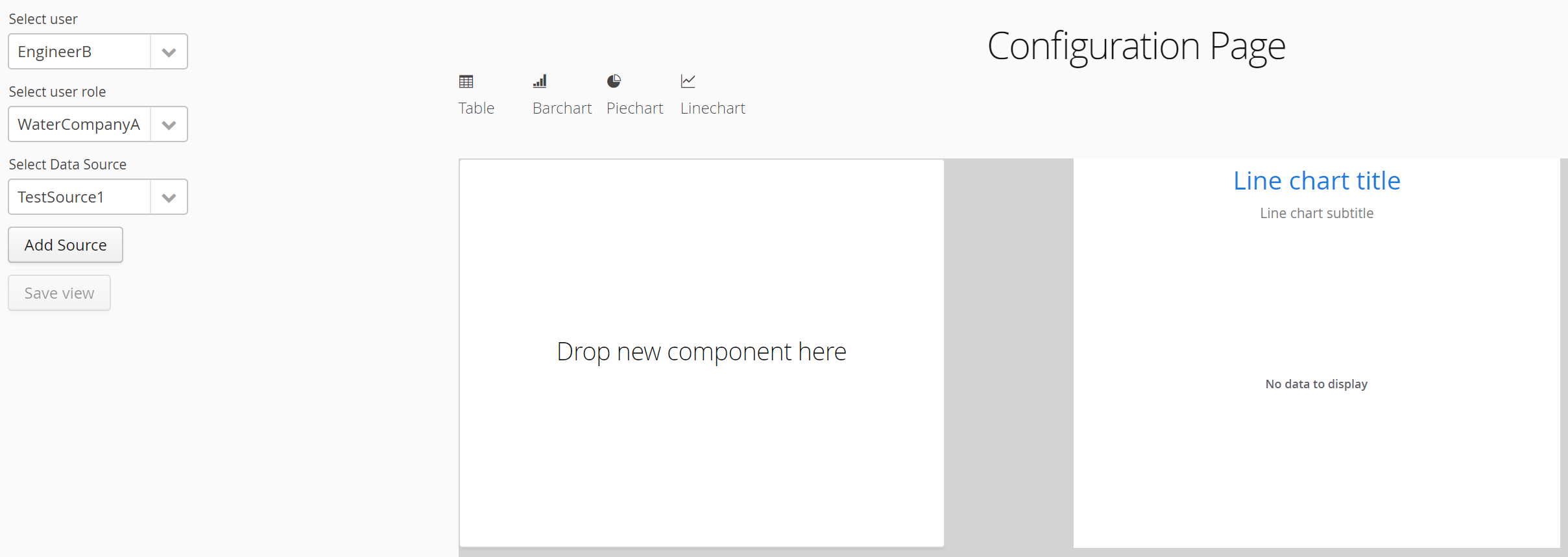


Figure 17 - Empty chart

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.

### 4.1.2 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:

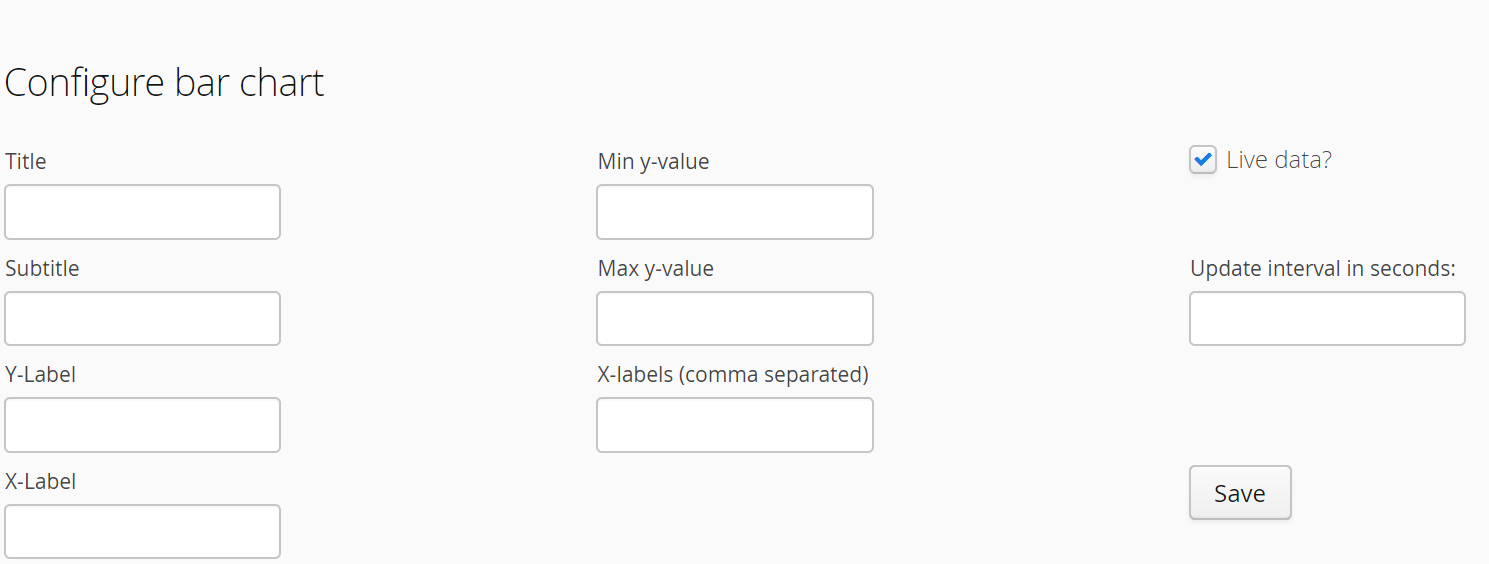


Figure 18 - Bar chart settings mask

In opposite to the data field selection, the view component isn’t updated until the “Save”-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.

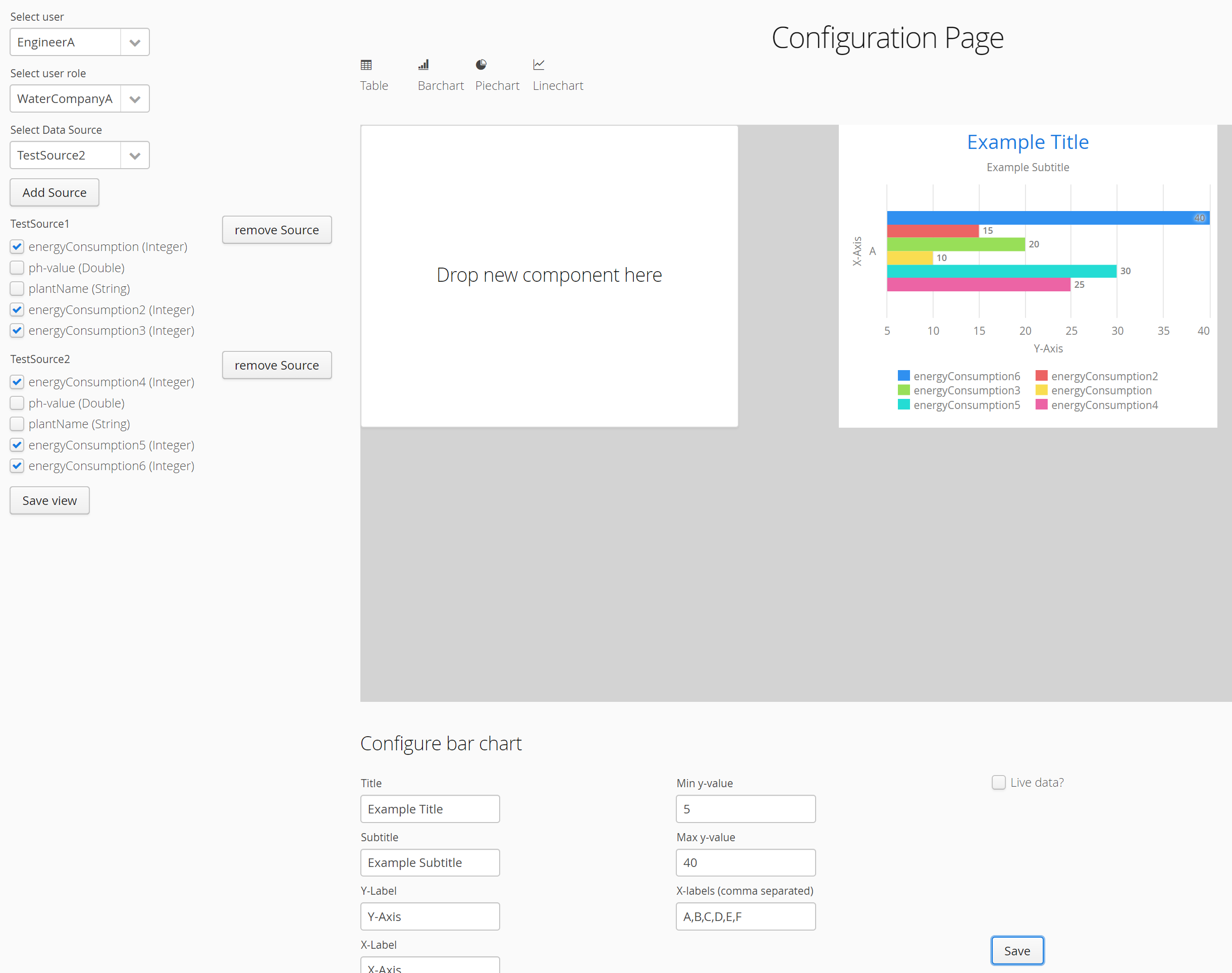


Figure 19 - Bar chart settings applied

### 4.1.3 User Views

As mentioned before, the admin user is able to configure the user views which means defining the data sources, drag and drop the components and configure the settings for each component by using the component configuration mask as described before. To verify this functionality, a view is configured for user “EngineerA”. The two mock data sources are added and the fields “energyConsumption1” to “energyConsumption6”, which return a static integer value are checked. After that, the bar chart and pie chart labels are dragged and dropped to the target are and the chart settings (title, subtitle, x-label, y-label) are applied as showed in the following screenshot:

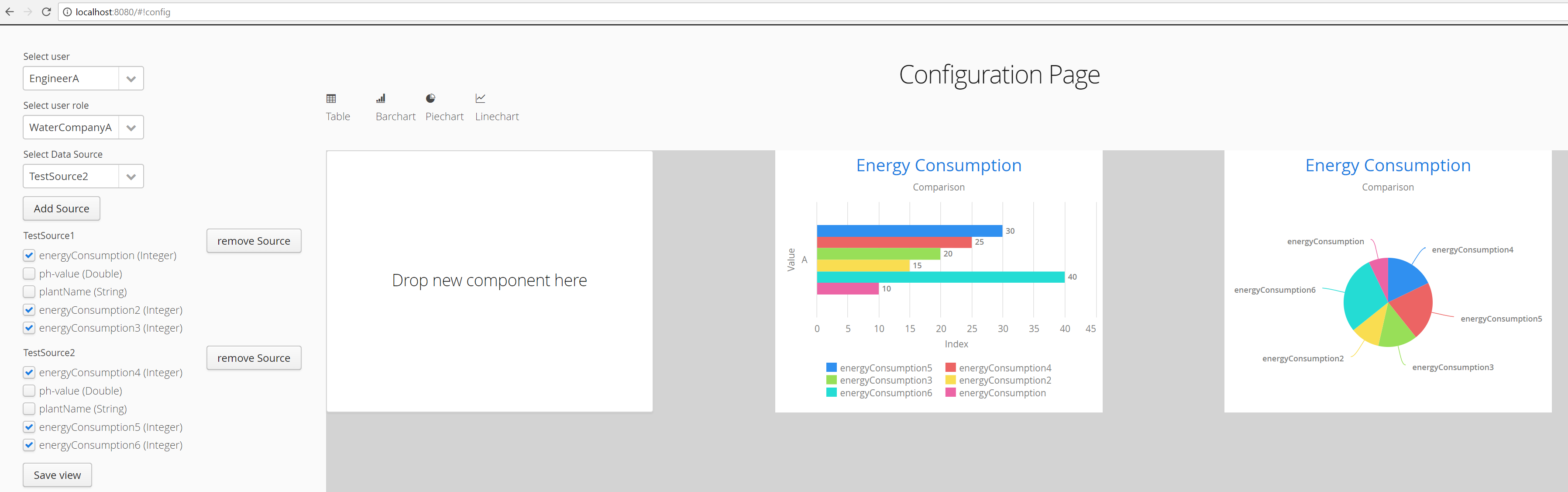


Figure 20 - User view configuration

As soon as the “Save View”-button is clicked this user view configuration is set for the current chosen user. After the user “EngineerA” logs in to the application, he is directed to his individual view.

Login:

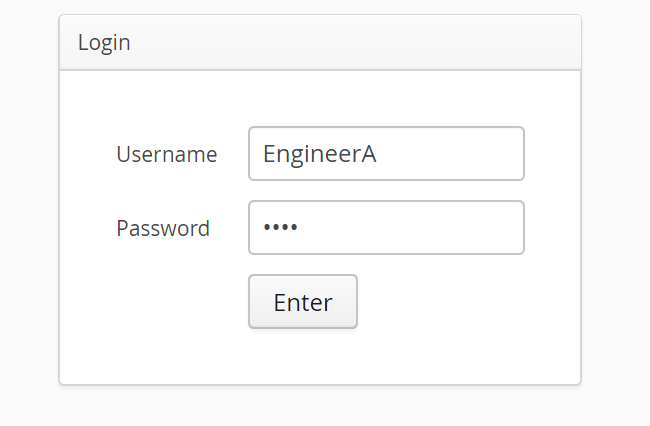


Figure 21 - Login EngineerA

After login:

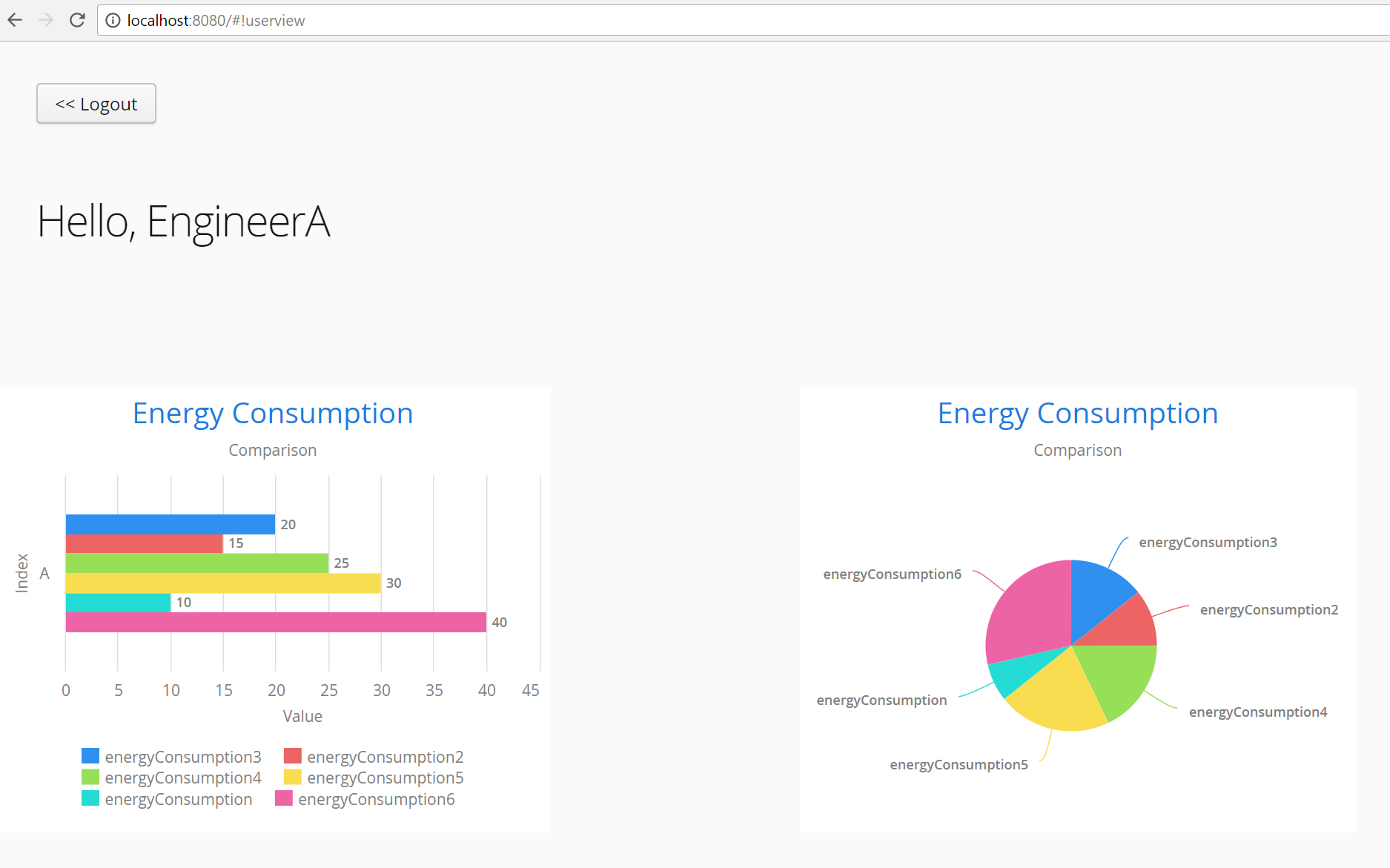


Figure 22 - User view

## 4.2 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. In this example, the goal is to compare the nitrogen dioxide (NO2) values arising from the sludge treatment and the water temperature of two water plants at the same water treatment step using a line chart.

It is necessary that an admin 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 stored:



Figure 23 - 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:

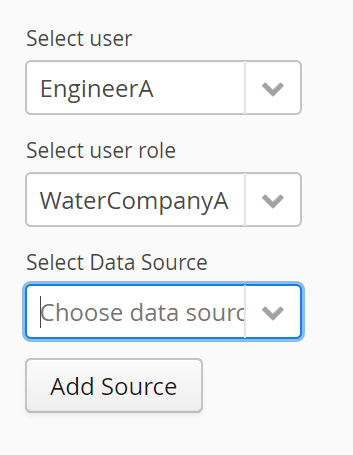


Figure 24 - data source select 1

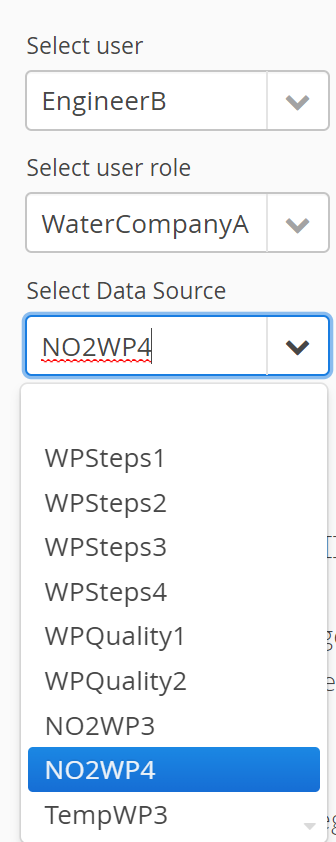


Figure 25 - data source select 2

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:

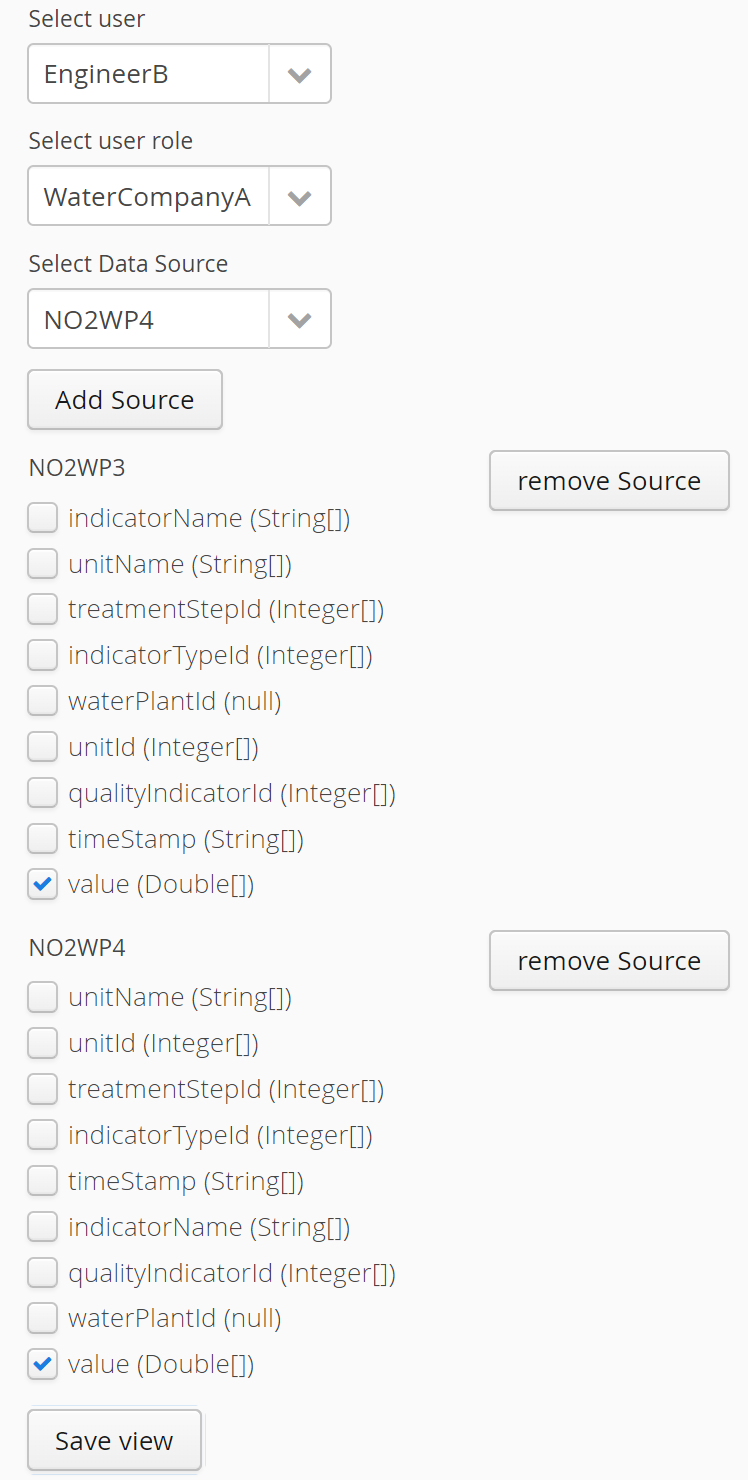


Figure 26 - Selected fields

As it is known from the harmonization service’s API documentation, the relevant field for comparison is the “value”-field. Due to the fact, that the fields chosen here, will be represented in the visual component the value field stays the only data field selected at that point. The other data fields may also be interesting for the visualization (for example as a label for an axis or a chart legend), but this isn’t implemented in the scope of this dissertation. However, it would be an important task for the further project work.

The following screenshot shows the user configuration overview after selecting the charts, sources and fields:

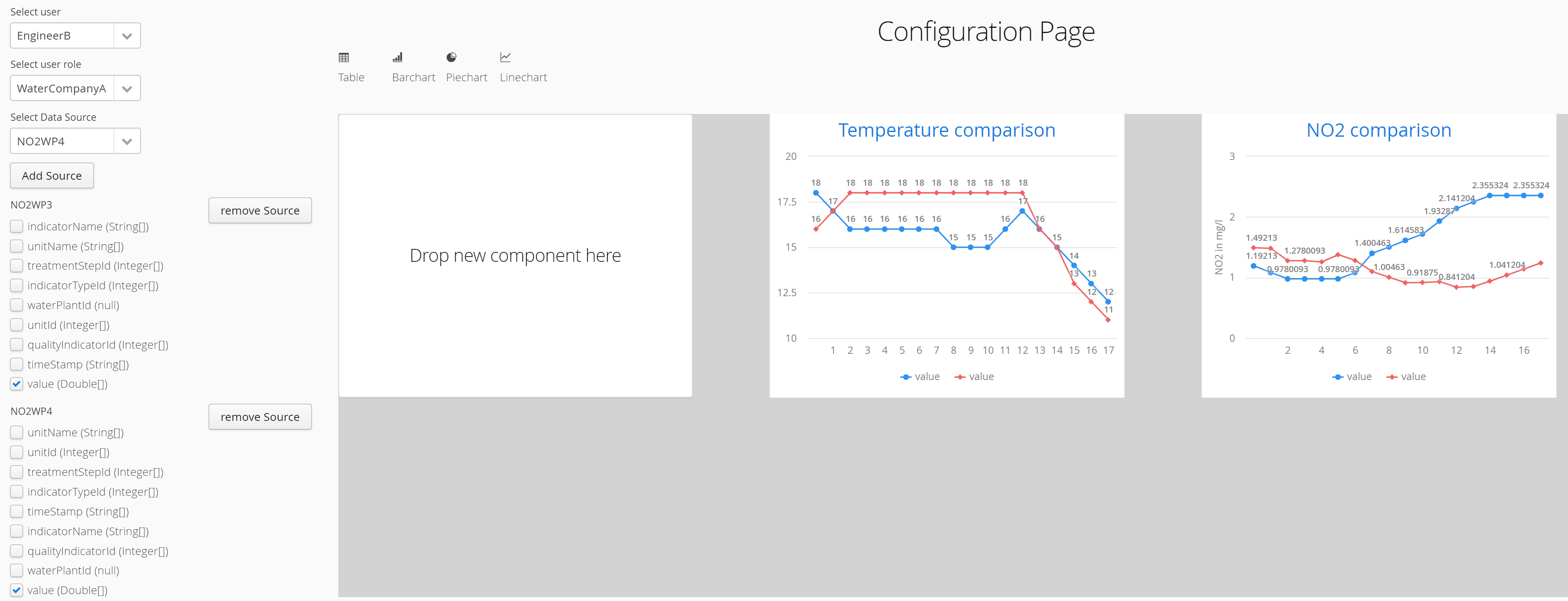


Figure 27 – Configure temperature and NO2 comparison

After saving, logging out the admin user and logging in the user “EngineerB” the two line chartsappear with data from the harmonization service:

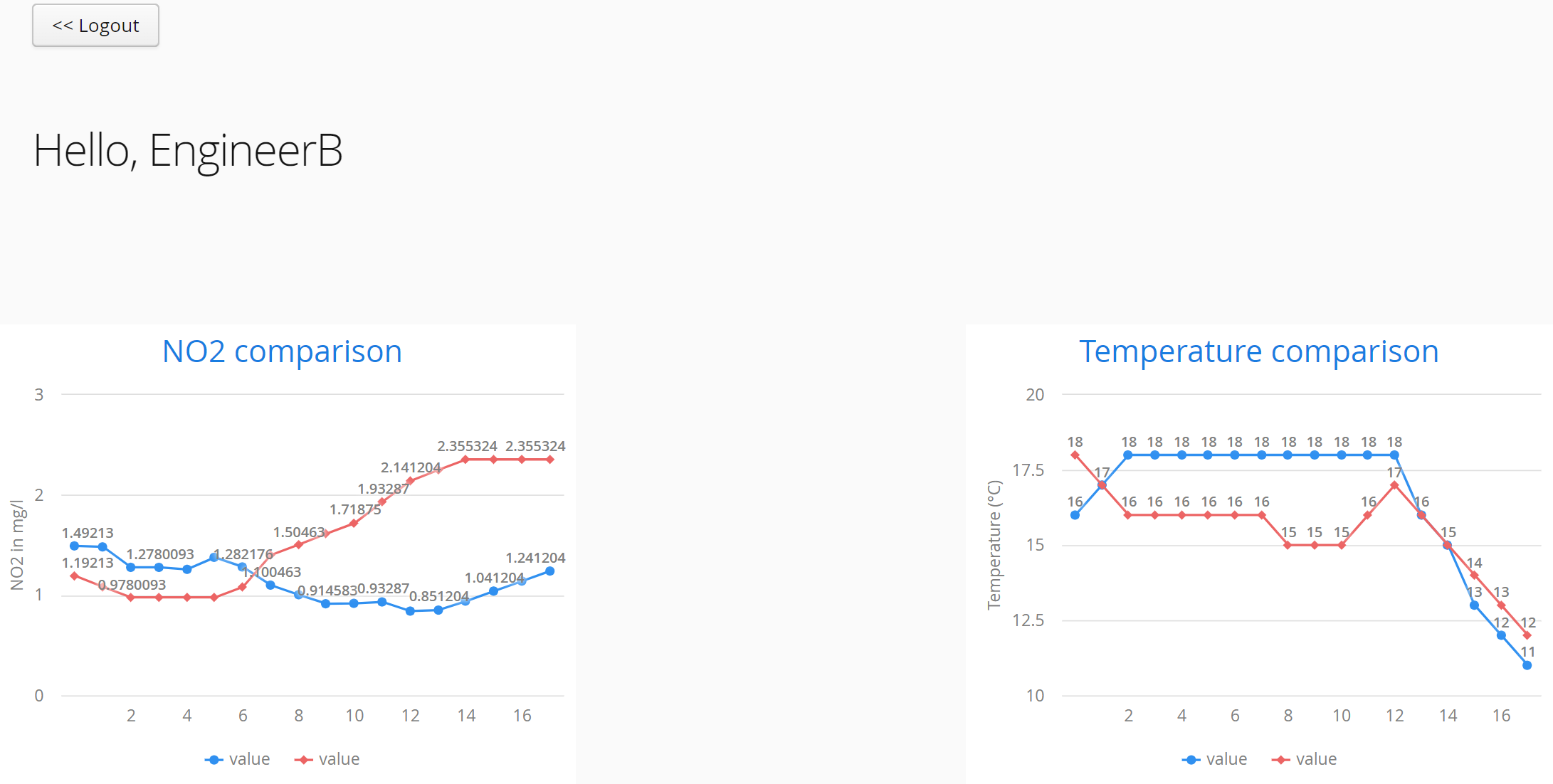


Figure 28 - User view temperature and NO2 comparison

## 4.3 UI Testing

Almost every software project today is demanded high quality standards to be successful in the competitive struggle in the software development business. While ordinary unit tests are well known and commonly used to make sure that present functionality doesn’t break when implementing new features, UI tests are more complicated and there are many different approaches. One approach is, not to test the UI at all, which is acceptable when most of the code can is tested via unit tests. Another approach is to test the UI via end-to-end tests. End-to-end tests have the advantage that they can simulate and test user interaction and business logic as well. Their disadvantage is often that they can be, in some environments, unstable and time-consuming and they are not easy to integrate into CI environments. Vaadin TestBench is part of the Vaadin framework and provides support for the following types of tests:

* Automated acceptance tests
* Unit tests
* End-to-end integration tests
* Regression tests

The documentation states:

*“Vaadin TestBench is a tool for creating and running browser based integration tests for your Vaadin application. TestBench simulates a user of your application, performs the tasks specified using Java code and verifies that the expected actions take place in the application.*

*TestBench can also visually inspect your application and detect unintentionally introduced changes, and verify that the application visually looks OK in all the browsers you are testing with. TestBench also includes special support for other Vaadin products, making testing easy and robust compared to generic web testing solutions.”* [20]

The Vaadin TestBench framework provides powerful possibilities to test Vaadin applications in many ways, however it is not easy to understand the principles of the so-called “Page Object Pattern” which introduces a test abstraction layer hiding the implementation details from the test methods. Although it would helps to improve the quality and maintainability of the application, it would take a big effort to test the whole code. The following code snippet shows an exemplary test for the login screen:

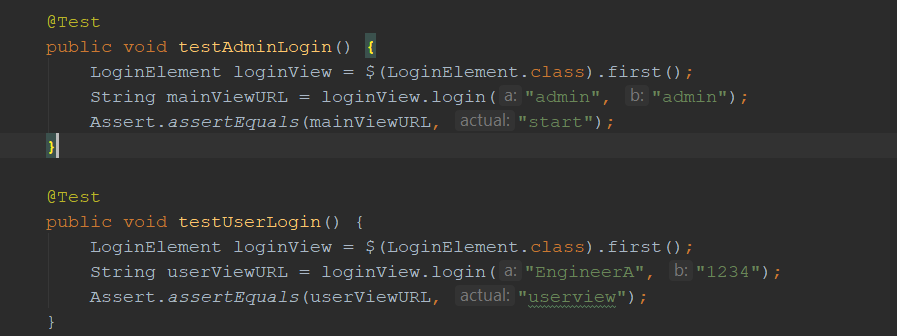


Figure 29 - Login test

This test checks if the login mask directs the user to the correct URL after entering his credentials. The admin user should see another view as a non-privileged user. Even if the test looks simple, the definition of the abstraction (LoginElement) is not trivial and the number of the lines of code is quite high for this simple test case.

# Conclusions and further work

## 5.1 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, which is strongly related to the complexity of the UI logic. Even if it seems that the outcome is only a small prototype, the event handling, view changes, interaction logic and styling management was a complex and time consuming task. 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, which was created in the early phase of the development, but then removed from this dissertation. It got clear 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. As a result, the ER-model was replaced by integrating JPA and Hibernate to use a code first approach for persistence, however it wasn’t possible to integrate a real database in the end because the remaining time was too little.

Comparing the final dissertation work with the interim report shows that the aims and objectives changed noteably. While in the beginning of the dissertation the priority topics were focused on the WWTP project, it changed to creating a service provider, which allows a user friendly creation of individual views consisting of data visualization components. The WWTP case should only deal as a proof of concept.

## 5.2 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.

# References

|  |  |
| --- | --- |
| [1] | “Water supply,” [Online]. Available: https://www.water.org.uk/about-water-uk/water-supply. [Accessed 24 09 2017]. |
| [2] | Water UK, “Water resources long-term planning framework,” Water UK, 2016. |
| [3] | A. Mousavi and E. Katsou, “WWTP-Global-300617,” 2017. |
| [4] | “Water UK,” [Online]. Available: https://www.water.org.uk/about-water-uk. [Accessed 23 09 2017]. |
| [5] | “Regulation,” [Online]. Available: https://www.water.org.uk/about-water-uk/regulation. [Accessed 24 09 2017]. |
| [6] | “DWI,” [Online]. Available: http://www.dwi.gov.uk/about/index.htm. [Accessed 23 09 2017]. |
| [7] | “Watrer UK: Drinking water quality,” [Online]. Available: https://www.water.org.uk/policy/drinking-water-quality/water-quality-standards. [Accessed 03 01 2018]. |
| [8] | “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. [Accessed 4th January 2018]. |
| [9] | “Water,” [Online]. Available: https://www.metoffice.gov.uk/services/industry/water. [Accessed 26 09 2017]. |
| [10] | “South East Water,” [Online]. Available: http://www.southeastwater.co.uk/your-account/about-your-bill/water-charges/how-much-does-your-water-cost. [Accessed 29 09 2017]. |
| [11] | “What is Data Dissemination,” [Online]. Available: http://www.learn.geekinterview.com/data-warehouse/dw-basics/what-is-data-dissemination.html. [Accessed 03 02 2018]. |
| [12] | F. N. Ulf Leser, Informationsintegration: Architekturen und Methoden zur Integration verteilter und heteorgener Datenquellen, dpunkt.verlag, 2007. |
| [13] | 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. [Accessed 27 02 2018]. |
| [14] | S. Few, “Data visualization: past, present, and future,” 10 01 2007. [Online]. Available: https://www.perceptualedge.com/articles/Whitepapers/Data\_Visualization.pdf. [Accessed 21 03 2018]. |
| [15] | W. Kobek, “Datenvisualisierung: Mehr als Balken und Torten,” 24 02 2015. [Online]. Available: https://jaxenter.de/datenvisualisierung-bi-projekte-15797. [Accessed 21 03 2018]. |
| [16] | D. Bell, “UML basics: The component diagram,” IBM, 15 12 2004. [Online]. Available: https://www.ibm.com/developerworks/rational/library/dec04/bell/index.html. [Accessed 02 03 2018]. |
| [17] | “Vaadin FAQ,” Vaadin Ltd., [Online]. Available: https://vaadin.com/faq. [Accessed 26 02 2018]. |
| [18] | “Java Community Process,” Oracle, [Online]. Available: https://jcp.org/en/jsr/detail?id=338. [Accessed 18 03 2018]. |
| [19] | B. Padmanabhan, “Unified Modeling Language (UML) Overview,” [Online]. Available: https://people.eecs.ku.edu/~hossein/Teaching/Fa13/810/Readings/UML-diagrams.pdf. [Accessed 2018 02 28]. |
| [20] | Vaadin Ltd., “Vaadin Testbench: Overview,” Vaadin Ltd., [Online]. Available: https://vaadin.com/docs/testbench/testbench-overview.html. [Accessed 25 03 2018]. |
| [21] | „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]. |
| [22] | “Data Integration,” [Online]. Available: https://www.ibm.com/analytics/data-integration. [Accessed 03 02 2018]. |
| [23] | J. Dong, Network Dictionary, Javvin Technologies Inc., 2007. |

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. [23] [↑](#footnote-ref-4)