Digital Transformation

Implementation of Business Intelligence Solution for the Pharmaceutical Sector

Akshay Whig

Internship Report presented as the partial requirement for obtaining a Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

DIGITAL TRANSFORMATION

Implementation of Business Intelligence Solution for the Pharmaceutical Sector

by

Akshay Whig

Internship Report presented as the partial requirement for obtaining a Master's degree in Information Management, Specialization in Information Systems and Technologies Management

Advisor / Co-supervisor: Vitor Manuel Pereira Duarte dos Santos

February 2020

ACKNOWLEDGMENTS

First of all, I must have to express my sincere gratitude to my family and friends who supported and encouraged me during the course of this master's degree.

I would like to thank SDG Group, for giving me the opportunity to carry out my internship based on their company's project with Janssen, as well as to thank the employees and my co-workers for their support.

Finally, I would like to thank my thesis advisors Vitor Manuel Pereira Duarte dos Santos and Ana Carina Fernandes Rodrigues for their continuous support, guidance and constructive feedback over the last year. It would not have been possible without them.

ABSTRACT

Mertens describes the term Business Intelligence as the integration of all activities evolving around the company, starting from IT procedures up to management methods and analytical processes, with the aim of providing information and new insights as a decision-making basis for the management.

Already in the 1960s, information systems attempted to support decision making for the management. Over time, the umbrella term Management Support Systems (MSS) evolved, re-emphasizing the importance of information and communication technology.

In the present project report, the aim is to establish and enhance these decision-making processes for the company Janssen Pharmaceutical. The objectives, therefore, are to establish a single source of truth with all data sources feeding one data lake and on top of that data lake building various reports, dashboards and visualizations in both a web-based solution and standalone app solution.

KEYWORDS

Business intelligence, Pharmacy, Health insurance, Dashboard, Qlikview, ETL, QlikSense, Mashup, Data warehouse

TABLE OF FIGURES

LIST OF TABLES

INDEX

1. INTRODUCTION

The aim of this internship report is the development of a web-based business intelligence solution which analyses the data of Janssen Pharmaceutical in various contexts and for different focus groups.

The objective for this report is to present and review the technologies used and the approach which led to developing the application. In order to build a solid foundation, first, the general concepts regarding the subject Business Intelligence and its best practices must be elaborated. After a solid foundation of BI and its components has been established, the technologies, tools and methodologies used will be presented. Subsequently, the development activities beginning from data preparation, transformation, processing and all the way to data visualization will be demonstrated. At last, a conclusion of the internship and also of all the activities within the internship will be given, in a way to reflect on experiences made and lessons learned.

1.1. ACADEMIC CONTEXT

This internship report is part of the second year of the Master´s in *Information Management with specialization in Technology Management and Information Systems* from University of Nova IMS in Lisbon. The final requirement to receive the master's degree is to do either a thesis, a working project or an internship with a written report. My internship started on the middle of November 2018 and finished on June 2019. I had the opportunity to do this internship in Lisbon, for a mid-size BI consulting company, as by the rules it is necessary that the internship has to be related to the masters or its courses.

The master's was structed with the following courses:

Table 1 - Course plan

First Semester	Second Semester
Data Mining I	Data Mining II
Information Project Management	Business Process Management
Knowledge Management	Information Technology Architectures
Information Management Systems	Customer Relationship Management

All the above-mentioned courses have a focus on a theoretical as well as a practical part. The best way to combine both was done by having theoretical classes daily and meanwhile having a project to be hands-on and utilize what was learned earlier. Most projects were to be done in teams consisting of 3 to 5 students every team had their own approach on solving the provided task, which in the end enriched teamwork, commitment and communication.

The internship report here, focuses on a similar approach. The difference is only that you are tackled with a real-life problem, which needs to be solved. In the internship you are asked to work with a team as well as to work on a project and deliver instead to a professor a department or company. At the end of this internship, students need to deliver a written report and present it to a jury consisting of professors and supervisors. After this, students are graduating with a master title from Nova IMS.

1.2. Business context

The project is being conducted at SDG Group Lisbon, Portugal. SDG Group is a global management consulting firm, having a leading vision in the practices of Business Analytics, Corporate Performance Management and Data-driven Services. Currently more than 750 professionals are employed at SDG Group in over 20 offices, providing Management Intelligence services to more than 400 worldwide customers in multiple industries.

The BI solution is to be developed for Janssen Pharmaceuticals and SDG was contracted to be responsible for the BI solution, Data Lake and also managing the project using the SCRUM methodology.

Janssen Pharmaceuticals is one of the world's leading research-based pharmaceutical companies. It was founded in Belgium in the year 1953 by Dr Paul Janssen. In the year 1961, Janssen Pharmaceutical was acquired by the New Jersey-based American company Johnson & Johnson (J&J) and became part of Johnson & Johnson Pharmaceutical Research and Development (J&J PRD), which was renamed Janssen Research and Development (JRD). The Company conducts research and development activities related to a variety of human medical conditions. The project organization setup is structured the following.

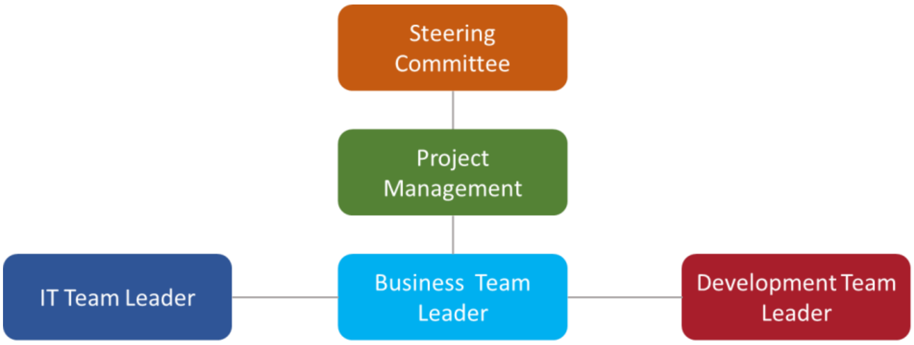


Figure 1: Project organization structure

The highest level, the steering committee in project management, refers to the high-level decision-making authority for an individual project or group of projects, in general the tasks of a steering committee are to support, monitor and supervise the progress. The task of a project is to fulfil the project goal through its implementation. The project management, which is on the second level, commits itself towards the project owner (business owner) to fulfil all tasks resulting from the project within the agreed scope. Finally, the last level consisting of IT, Business and Development all fulfil different roles and objectives in the project and are also each interdependent of each other as for instance the IT needs to establish a pipeline and framework of tools and technologies for the Business whereas the Business needs to provide requirements for the Development team and describe their vision of the expected result and at last the Development team develops the features the business requested for their project. However, this explanation is rather abstract, a conclusive description will be given thoroughly in the chapters three and four.

1.3. Objectives of Internship

In this section, I will briefly summarize the goals of the internship and at the same time also the objectives of the project itself.

As mentioned, the timeframe of this internship was set for 6 months, which can be structured in the following phases:

1. **Onboarding**: During the onboarding the idea is to get to know the company, their values, way of working and of course the technologies. The onboarding phase approximately was around 4 weeks in which, I needed to understand the basics of the technology QlikSense and web development. Since SDG Group Portugal mostly deals with the pharma industry, I had to get a basic understanding of the processes within the pharma industry and their data.

2. **Project**: In this phase the business department was discussing their ideas with the development team as well as their IT team. As I am part of the development team, this phase was crucial as its up to my team to understand the requirements and needs of the business to develop the features as expected. The project itself has been split in three phases. In the first phase the goal was to establish a single source of truth, which in short means identifying all sources the business is acquiring its data from and store it in one central source, namely a data lake. In the second phase, build one or multiple dashboards to represent the data in an easy to understand way but also in a way the business can make use of it, for that the business defined how they want to represent their data in terms of what measures and what dimensions should be used. In the third phase it was asked to create a dynamic input

interface, where business is inputting information regarding clients and product metrics which is directly stored in the data lake and at the same time visualized in the dashboard.

In short during this project it is asked to create a central hub for their data, a data lake, then a dashboard visualizing their KPI’s and metrics as specified and at last an interface to input data.

For the internship I will be working full time and my team leader is directly responsible for me. The team of the project consists of two developers and one team leader, therefore the responsibilities on my side are really high as I have never handled such a big scope.

As we are working with sensible data in this report all sensitive information will be masked as I have to respect to the compliance rules of Johnson & Johnson. But more about the project and the data will be elaborated in the upcoming chapters.

2. LITERATURE REVIEW

This chapter is going to cover all the theoretical basis required to understand the project. First a clear definition and a categorization of the term Business Intelligence will be provided. Furthermore, the emerge of BI and its current state will be demonstrated, by elaborating best practices of the whole concept, as this ensures a clear understanding of the project.

The structure of this chapter will be the following, first a brief introduction of the emerge of BI will be given which eventually work towards a clear definition of the term BI. Afterwards the architecture will be elaborated, which includes components such as the ETL process, Data Warehousing and so on, this will cover the pillars of the theoretical basis.

Subsequently, the analytical part of the report will be covered, which has a strong emphasis on reporting and dashboarding. Here, the best practices will be elaborated as these are of fundamental importance for the project.

As BI technology keeps evolving two of its new features, self-service BI and Mobile BI will also be presented.

At the end of the chapter the development lifecycle, namely the agile methodology SCRUM, which was used to develop the solution will be presented.

2.1. BUSINESS INTELLIGENCE

The continuing growth of data, as well as the massive changes in the market environment and ever-increasing internal and external requirements for transparency of decision making must be incorporated into the calculations of successful corporate management. Traditional individual management systems can no longer meet those requirements, here Business Intelligence (BI) emerges as solution. In this chapter, the objective is first to define and classify the term Business Intelligence. Furthermore, the analytical concepts which support the decision-making process will be described.

2.1.1. Definition and classification

The term Business Intelligence describes the integration of all activities in the company, from IT procedures to management methods and analytical processes, with the aim of providing information and new insights as a decision-making basis for management (Mertens, 2002). Already in the early 60s, information systems were used to support decision making for the management. Over time, the collective term Management Support Systems (MSS) emerged, which re-emphasizes the importance of information- and communication technology. The intention at the time was not only to use individual computers or systems as a support tools, but to look at the entire environment as a support factor for the whole organization (Müller & Lenz, 2013). In the late 80s, the Gardner Group shaped the new terminology Business Intelligence, which has established itself to this day.

Initially, the term was used more as a collective term for front-end tools in the environment for data analysis, reporting and query mechanisms. This derivation sparked intense discussion and reorientation in supportive information and communication technologies. Thus, the term gained more and more interest and was often tried to rewrite. The author (Gluchowski, 2001) tried to develop a connection between data provision and evaluation:

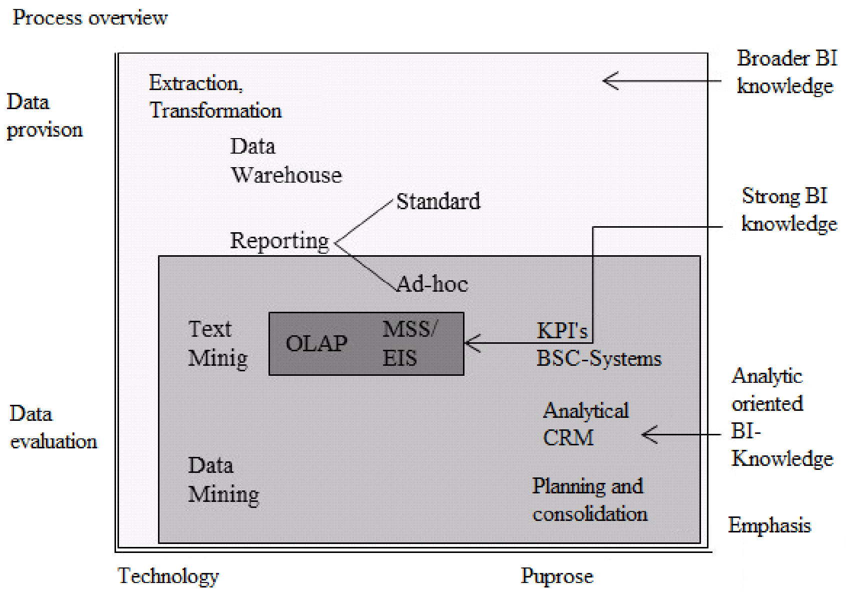


Figure 2: Business Intelligence factes adapted from (Gluchwoski & Chamoni, 2016)

The model provides possible views between data provision and evaluation as well as the focus on technology and purpose. Yet, three of those categories accumulated, play one major role. Business Intelligence in the broader sense deals with data preparation and storage, in short data provision. Terms such as data warehouse, ETL and reporting are relevant here. For strong BI knowledge, the primary factor is decision support. That includes Online Analytical Processing (OLAP) and Management Information Systems (MIS). Finally, the analytical oriented BI knowledge describes the use of the system by employees (Cleve & Lämmel, 2016). Examples for this case are planning and consolidation, as well as text and data mining. Nevertheless, there is still some uncertainty regarding a clear definition of business intelligence. The author (Mertens, 2002) tries a new interpretation and identifies seven different variants for BI:

1. BI as a continuation of data and information processing:
   Information processing for company management
2. BI as a filter in the flood of information: Information Logistics
3. BI = MIS (Management Information System), but especially fast / flexible evaluation
4. BI as a warning system
5. BI = *Data Warehouse*
6. BI as information and knowledge storage
7. BI as a Process: Symptom Elevation

However, these variants cannot be classified specifically and have different effects on the systems used. When we talk about BI nowadays, the data warehouse always comes into play, regardless of whether a central, a distributed- or the Hub&Spoke approach was chosen (Schön, 2016). In this context, we should point out Kemper's structure of the three-layer BI framework.
In the following graphic, Kempers structure of a three-layer BI framework is explained (Kemper, 2016):

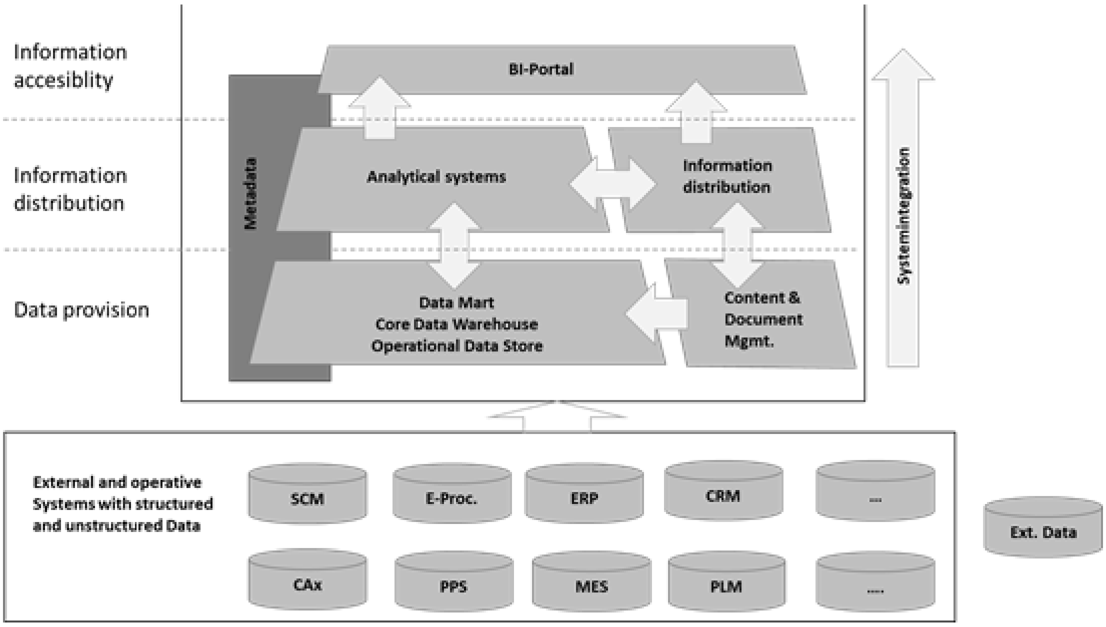


Figure 3: BI-framework adapted from (Gluchwoski & Chamoni, 2016)

The external and operational systems form the first level and describe the origin of the data. The data must be classified into operational data, i.e. internal data that is subject to a constant change process, and external data, e.g. from trade associations, market research institutes or external databases. The data is then extracted from the source systems, which can be distinguished as structured or unstructured data. Data is provided via a data warehouse as the first layer of the framework is indicating, however a more detailed explanation of Data Warehouse will be given in the following chapter.

The next step is the generation of information, in which various analysis systems are aligned to the individual characteristics of the company. The information gained serves as a basis for decisions on further strategies. Lastly, the BI portals are mapped, which are used to process the newly generated information across departments.

Nowadays, BI methods are mainly used in the fields of business administration, information technology, operations research and statistics. This historical approach has been further developed; today Data Mining, Predictive Analytics and Advanced Analytics play an increasingly important role.

Models such as Business Intelligence 2.0 or 3.0, which can be accessed via Web Services, also offer many new possibilities. The intention to convert company data into business-relevant information with BI or BI-technology has remained (Mayer & Quick, 2015).

2.1.2. Analytical methods

In order to generate business-relevant information from operational data, it is necessary to analyze the data accordingly. In following section, the fundamental analytical concepts used for Business Intelligence will be presented.

2.1.2.1. OLAP

OLAP is the abbreviation for On-Line Analytical Processing and represents the process of explorative, interactive analysis of archived and stored data in a data warehouse based on a multidimensional data model. This mainly involves the support of queries for analysis purposes or the preparation of business data for managers and decision makers in a company. The main focus of OLAP is the execution of complex analysis projects, which cause a very high data volume, but at the same time enable a flexible, intuitive evaluation. The data is summarized from the data sources in a multidimensional data cube and then presented in reports with tables and graphics. The user can select and combine the criteria that are of interest to him.

The OLAP concept originally goes back to the database specialist Edgar F. Codd. In 1993, Codd established a series of properties for OLAP systems, which were later reduced to five essential factors ("FASMI"). These are in detail:

- Fast: An OLAP system should answer regular queries in a maximum of five, complex queries in 20 seconds.
- Analysis: The system should perform an intuitive analysis for arbitrarily complex calculations.
- Shared: Multiple users can use the system at the same time
- Multidimensional: At its core is a multidimensional view of the data, independent of the database structure used.
- Information: The scalability of the application should also be given for larger amounts of data.

As a result of the five required features, OLAP databases are high-performance and easy to use.

This multidimensional cube, that was mentioned above is known as the *OLAP cube.* In that cube the data is, as the name already suggests, presented as a cube. In this way, the data can be viewed from multiple perspectives and in multiple stages of granularity. This allows you to analyze key business figures such as sales or costs on a multidimensional basis using dimensions such as products, regions or time. Especially this type of data presentation is quite easy to understand. The figure below illustrates such a multidimensional cube. The dimensions and "product", "region" and “measures” correspond to the axes and span the cube. Each cell of the cube contains exactly one value (highlighted in color), which for example indicates the turnover of product X in a certain time.

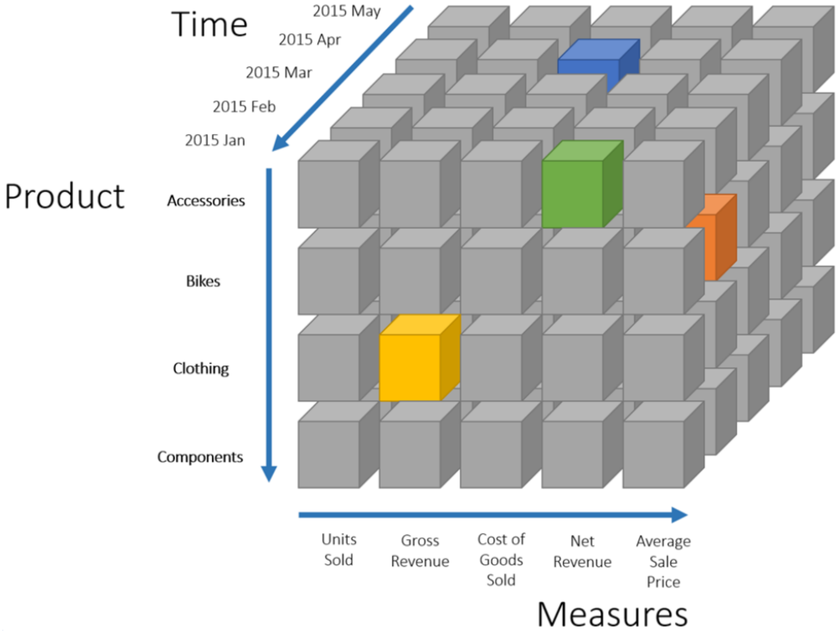


Figure 4: OLAP cube (Data Leader, 2017)

You can analyse the cube with the help of OLAP functions. The desired data sets can be generated by a number of methods, such as slicing or dicing the data cube. Certain views, such as viewing a specific product or a specific year, are obtained by cutting out layers, as shown in the figure below (Mertens, 2002).

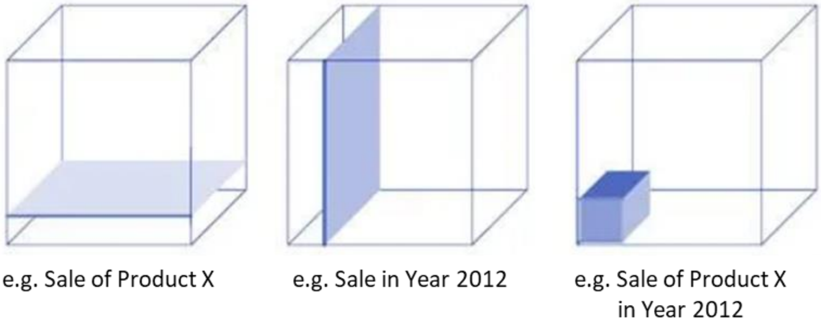


Figure 5: Cube operations (Data Leader, 2017)

A few techniques such as “Drill down” or “Roll-up” emerged, which enable analysing data from other perspectives. For example, during rollup, all single values are summarized or aggregated to a hierarchy attribute at a higher level. The drill-down function on the other hand provides a more detailed display of the data, which basically does the opposite of the roll up.

2.1.2.2. Reporting

In the business context, a report is a standardized overview of business topics that relate to a particular area of responsibility. The representation of those overviews generally takes place through visualizing the focus area in diagrams in order to improve the reception of the information by the recipient. As focus areas one can understand all areas of companies and at all hierarchical levels in order to gain a quick insight into relevant business contexts, which in the best case can empower business decisions (University of Agder, 2018).

Increasingly therefore, companies are using technology whose functionality enables the rapid and almost intuitive creation of a wide variety of reports. The most important reporting objects of a company can be differentiated into dimensions, hierarchies, attributes and key performance indicators (KPIs).

KPIs are used to check how successful certain activities are in companies. All processes in organizations can be monitored using these key performance indicators. By means of key performance indicators, management and controlling can analyse processes in the company and through consistent monitoring, processes can be adjusted and optimized accordingly (Fasel & Meier, 2016).

As the basis for reporting has been established, a closer look has shown that there are different types of report. Below a few reporting types with their functionality will be presented:

- **Standard reporting** takes place at fixed dates and cycles to precisely defined recipients, e.g the top management. The content and form of the standard reports are defined and can no longer be changed by the recipient. For this reason, the content and design of these reports must also be defined within the framework of determining information requirements to such an extent that as little or no information gaps as possible can arise for the recipient.

- **Exception reports** are generated when defined tolerance values, variance values or threshold values are reached. If defined limits are exceeded, they are usually automatically presented to the recipients for information analysis and help to draw attention to particularly control-relevant issues (Gluchowski, 2001).

2.1.2.3. Dashboarding

The BI Dashboard is a data visualization tool that provides a clear and easy-to-understand visual representation of the insights gained from business intelligence. The dashboard shows the current status of important KPIs and other relevant company figures, such as metrics, numbers and sometimes rating systems such as traffic lights or scale systems. As a metaphor to better understand what a dashboard represents, you can compare it with a car dashboard, it always shows the current status at any time. The aim of the dashboard is to bring together the various figures and metrics, condense them and visualize them in a way that is as easy to understand as possible.

Dashboards can be customized for specific roles and display metrics that target a particular focus or department. The core components of a BI dashboard are therefore a customizable user interface and the ability to process real-time data from different sources.

One of the key features of a BI dashboard is the clarity and simplicity of its representation of data. The most important core findings of the dashboard should be recognizable at a glance by the respective target group. The actual key figures and the derivation of the results are no longer considered in the visualization. Moreover, the contents presented are chosen in such a way that they are overarching and generally accepted indicators. For this reason, an individualisation of each individual company and company division is deliberately avoided (Rouse, 2015).

An important factor in order to deliver reliable and accepted statements to business executives is, that the information provided by the BI dashboard must be kept consistent with other company analyses, trends or plans.

Recognition of efficient and inefficient business processes and display of new trends and developments are just a few to mention advantages, when the dashboard is used correctly.

2.2. Data Warehousing

The objective of the data provisioning layer of a BI solution is to provide a suitable and consistent database for information and analysis purposes. In order to meet this requirement, both a decision-oriented database and a permanent filling structure of existing and newly added data must be defined and implemented. The required technological concepts and implementations examples will be subject of the following chapter. In the following sections, the conceptual principles of data warehousing will be presented, briefly, afterwards an architecture with its components of a data warehouse solution will be elaborated.

2.2.1. Extract – Transform – Load (ETL)

The ETL process plays a key role in Data Warehousing let alone Business Intelligence, and the reason for that is simple. ETL, as already mentioned in the header stands for "Extract", "Load" and "Transform", but what does that refer on to?

Basically here, the focus is on data. The ETL process can be defined as the data supply concept for the delivery of data into the data warehouse and it takes place in three steps. The following graphic visualizes how data is being extracted from one or more data sources and then transformed accordingly to the format of the target database and then finally loaded into the data warehouse.

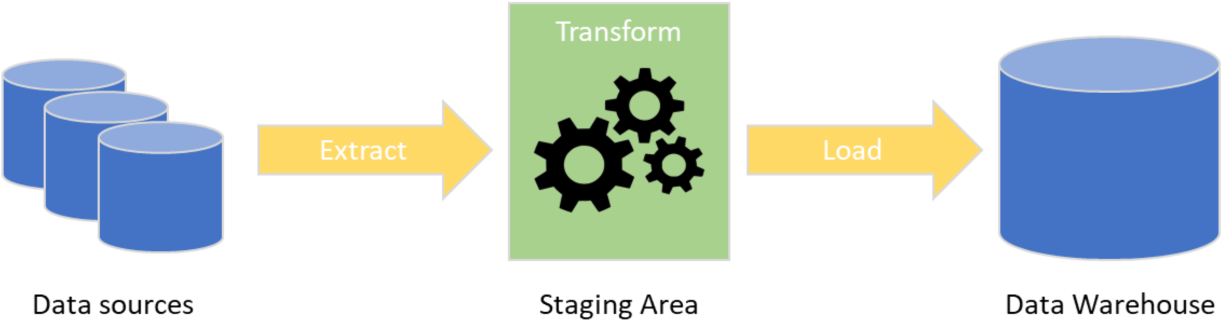


Figure 6: ETL process simplified

In the first ETL step, **extraction**, relevant data formats and structures of the various data sources are being identified and stored as direct extracts in the staging area of the data warehouse. The **staging area** is known as the temporary buffer for the data transformation, here the whole ETL process takes place. The various data sources that have been made available to the data warehouse, which came from internal and external data sources will have data that needs to be merged. This process of data integration is a task of the second step, **transform**. The data formats and structures in the staging area must be converted or transformed to the formats of the target database in the data warehouse system. The data formats in the target database have to be defined in a way that it can be interpreted from a business point of view. The reason for that is that most of the generated reports and dashboards finally, will need to have labels that are conclusive. Furthermore, the transformation of data formats and structures, therefore, can partially be done automatic and others manually as adjustments or corrections can only be identified and carried out by that (Manhart, 2008).

In general, the transformation process can be summarized in the following subtasks, as the table below shows:

Table 2 - Transformation tasks

Transformation process	Task
Filtering	In this step, only data that is relevant for our data warehouse is being extracted, irrelevant data will be discarded. Defective data, which can be understood as syntactic or contextual defects has to be corrected too.
Harmonization	Harmonization refers to the process of reconciling filtered data from a business point of view, which can be summarized in correcting inconsistency in data formats and lengths
Aggregation	Adjustments of units e.g. specifying in which unit a measurement should be for example kilo or gram. Also, semantic standardization, which can be a record that means the same as the other and therefore can be summarized.
Enrichment	Completing missing data, in order to establish a good data quality.

Finally, in the third ETL process step, the source data is transferred to the target database using the extraction and transformation rules as they had been defined in the load process. There are two different ways to execute the load process, the full load or the incremental load (Manhart, 2008).

The following table will highlight the differences between both:

Table 3 - Incremental vs. full load

	Full Load	Incremental Load
Data	Truncates all rows and loads data from the beginning.	Loads new and updated records only.
Time	Requires more time.	Requires less time.
Difficulty	Trustworthy, it can be guaranteed that load was successful.	Difficult, as new records have to be checked, if they were updated correctly.

There is no right or wrong option here, business has to specify the rules for themselves as this highly depends on how fast they want to have their data available. Often, the loading processes are triggered during non-working hours, therefore at nights or weekends. Transformations that serve to standardize the data structure are also known in the software industry as data migration. When migrating data, emphasis is placed on the aspect of data quality. Not infrequently there is a lack of sufficient data quality in the source systems, where redundant, obsolete or erroneous values can be stored. These can be cleaned up in the transformation process if they are identified in advance. Finally, it should be mentioned here that poor data quality in particular can not only come from the data sources, but can also occur during the ETL process if, for example, the transformation rules contain errors (Gerken, 2018).

2.2.2. Data Warehouse Concept

The provision of information is and remains an essential aspect of management support and business intelligence systems. The collection, aggregation and selection of decision-relevant information can only take place on the basis of consistent company-wide data management. The first steps towards standardizing information access for management levels were taken relatively early on by the IT industry. However, the way to make decision-relevant information directly available to end users quickly led to the information overload brought about by the first-generation MIS (Nirmalya Bagchi, Management Information Systems). In order to avoid this so called information overload, the data warehouse concept comes into play, which fundamental aspects focuses on only storing relevant content to enable decision-making processes. Ideally, such a database should be company-wide and cover the information needs of various departments. The ultimate objective of these data warehouse concepts is therefore to give decision-makers in organizations uniform access to all their data, regardless of where it was originally stored or what form it takes. From a technical point of view, it makes sense to isolate such a central data warehouse (DW) from the upstream systems that provide the data and to operate it on a dedicated platform. The collected data must be extracted from the various operational data sources and then stored in an independent database in such a way, that the user's information requirements are covered.

The term Data Warehouse (DWH) itself, has both a number of precise but also generic definitions. One definition by Bauer and Günzel is: "A data warehouse is a physical database that provides an integrated view of (arbitrary) data to enable analysis" (Gluchwoski & Chamoni, 2016). The definition is rather generic, and the focus lies on the analytical component. In order to get a more precise view one can say that a data warehouse describes the collection, transformation and preparation of decision-relevant data from different sources into one central data storage system, on which then one can perform support planning, analysis and decision-making tasks, on a management level basis (Müller & Lenz, 2013). The basic idea of a data warehouse can be traced back to the pioneers Ralph Kimbal and W. H. Bill Inmon. Inmon developed the top-down approach, while Kimbal followed a bottom-up approach. These approaches can be explained in the following. Inmon provides a definition via the data warehouse with the following statement: *"A data warehouse is a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management´s decision"* (Schön, 2016). Hence, it is clear that the central characteristics of a data warehouse are subject-oriented, integrated, nonvolatile and time-variant. This topic orientation about facts in a company is important. For the analysis, it is not operative processes that are decisive, but certain data objects according to which the data is subdivided, for example time, location or product. A further point is the integration of the data from the various previous data storage systems. In order to enable logical connections via consistent data storage and structures, it is necessary to integrate or standardize external and internal data from the various legacy systems. The next step is the permanent collection of data that can no longer be changed in order to map a large spectrum of available data. The last point describes the time reference as part of the data. A DWH focuses on more time-related data, which can periodically collect information, process it according to use and make it available as needed (Gluchowski, 2001).

2.2.3. The traditional Data Warehouse architecture

Businesses measure themselves by the availability of data and the knowledge that can be derived from it (Fasel & Meier, 2016). It turns out that a clean architecture is essential for development, operation and a fast time-to-market implementation in the company. The so-called Single Point of Truth was clearly defined for a long time. It was based on an Enterprise Data Warehouse, which was equipped with a Hub&Spoke approach. The hub stands for the DWH, which takes over the tasks of integration, storage and distribution of the data. The data marts serve as spokes that copy a specific partial dataset of the DWH for analysis purposes. The aim is to integrate different operative source systems with the data warehouse. To do this, data must be extracted from the source systems, transformed and loaded into the data warehouse (ETL). Subsequently, data extracts are incorporated into multidimensional data marts that are technically separated from each other and are then used by classic BI tools. BI platforms include OLAP, data mining, standard reporting, and so on. The advantage of a Hub&Spoke architecture is the integration, quality assurance, and data distribution to the data marts, which the organization arranges according to the analysis requirements. In practice, however, an exact Hub&Spoke architecture rarely exists, but a mix of different architectural approaches that are based on the Single Point of Truth. This concept is still valid for certain requirement contexts. Individual companies that divide their evaluation views by region, time, customer and product receive consistent evaluations. In reality, however, it is becoming increasingly difficult to maintain such an architecture. Problems are caused by continuous changes on the part of specialist departments or differently structured business models. In this case, a central data warehouse cannot offer an optimal solution, since constant changes and different weighting of the business models inhibit the synergy of a data warehouse. This results in a kind of information silos and heterogeneous architectures in dispositive landscapes, which often can no longer guarantee the transparency of data integration and preparation processes (Schön, 2016). The different business departments are competing for the fastest data deliveries. In the process, they try to achieve timely analysis in the form of spread marts or self-service solutions via the data copies of the data warehouse ad hoc with additional internal or external data. The resulting result reduces data quality and compromises traceability.

The following graphic illustrates the most common data warehouse architecture, the Hub&Spoke architecture:

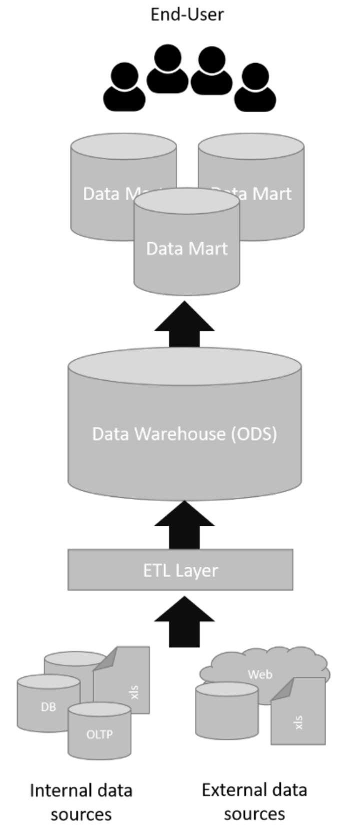


Figure 7: Hub & Spoke architecture

First, data will be provided from internal databases and flat files or external sources. The Extract Transformation Load (ETL) process is then initiated. The ETL process forms the core of data warehousing. It is divided into three steps: The first step is extraction, in which the data from the source systems is stored in the staging area. The second step is to merge the data. During the transformation, the data extracts are automatically or semi-automatically converted into the desired data formats and structures of the target database. The main tasks of the transformation process is to clean, integrate, homogenize and aggregate the extracted data in the staging area. The last step is to save or load the data into the architecture shown here, shows the Hub&Spoke architecture, which is the most commonly used in practice. The data marts shown here are dependent on the DWH and correspond to the top-down approach. the target database. After the ETL process, the data is permanently stored in the data warehouse. The DWH itself consists of a variety of components, such as archive, repository, metadata, core data warehouse and much more. Finally, data marts provide the data for the respective users. Data marts are nothing more than extracts from the data warehouse (Müller & Lenz, 2013). The data stock from the DWH is distributed over the data marts and made available to the departments or application areas. The advantages over direct access from the DWH are special data structures via Online Analytical Processing (OLAP), improved performance and use by authorized users only. They are relatively easy to derive and form consistent analyses of the data warehouse. Since you have to create these for each application area, the development time of a central DWH is the biggest challenge. The independent data marts offer a different approach by not using a central, companywide DWH, but data warehouses per department. The principle corresponds to the bottom-up approach. The main benefit is the much faster creation of the individual data warehouse systems. Disadvantages are data duplication between the separate DWHs, another ETL process, and the more difficult company-wide analysis possibility. Nowadays, different types of DWH exist. In addition to the classic DWH, which uses the ETL process to forward data to the DWH on a daily basis, there are also closed loop DWH, real-time DWH and active DWH. The closed-loop DWH offers the feature of backward integration of the data into operative systems to enable analyses already in the preceding systems (Schön, 2016). The real-time DWH however, forwards the data constantly in real time to the DWH. Furthermore, the Active DWH offers the possibility to directly integrate operative business processes into the DWH (Schön, 2016).

2.3. Mobile BI

Basically, mobile BI includes the same concepts for analyzing enterprise data as traditional BI. However, these concepts are being extended into the domain of mobility, which means that BI functionalities can be used independently of location and time with the aid of mobile devices, such as smartphones or tablet computers (Fuchß, 2009). However, the emerge of mobile BI was only possible due to the rapid development of smartphones and mobile devices, in the last years.

The popularity of mobile BI is out of question, numerous companies are reacting by increasingly developing these applications. Mobile BI increases the efficiency of organization's process due to a simple fact to have the data or information available and accessible from anywhere and anytime. Although mobile BI has somewhat the same functionalities and benefits as traditional BI systems, organizations still do not fully rely on them. This is because smartphones or other mobile devices are not completely secure or at least not as well secured as desktop systems. Information security is a big factor when working with data, which may be highly sensible (Patel, 2014).

Regardless, the use of Business Intelligence on mobile platforms is a major competitive advantage since it allows the interactivity between the user and the device to be combined with the creation of alerts and reports customized to their needs on any time (Patel, 2014). When developing mobile BI applications, a lot of factors have to be taken into account, as mobile devices might not be as accessible as traditional BI applications. Basically, the application which is being reflected on the desktop should look like what is being reflected on the mobile device. The author is highlighting a few key factors when developing mobile BI, however these will be focused later on when talking specifically about the developed application in the project.

2.4. Self-Service BI

According to Gartners IT Glossary, Self-Service BI can be defined like the following:" end users designing and deploying their own reports and analyses within an approved and supported architecture and tools portfolio" (Gartner, 2019).

Before self-service BI, BI applications were created by BI developers, and these applications were often created for customer-specific requirements. After development and deployment, these applications often became static. In case of new business requirements or user needs new applications or reports had to be created.

Therefore, organizations are largely considering BI applications which offer this kind of functionality. As their analysts can easily generate their own queries, reports, and direct their own analyses without the need for assistance from an information technology professional. It does not pose a problem to seek assistance from an IT professional, however with Self-service BI, analysts can make better decisions and more quickly since they do not have to wait for their request to be satisfied by an IT professional. On the other hand, the IT team can devote themselves to create new strategies in the area of information technology, as (Patel, 2014) describes.

However, for these tools to be efficient, they must be extremely intuitive and easy to use. Most users do not have enough technical knowledge to work with traditional, complex, and sophisticated BI tools.

3. TECHNOLOGIES & TOOLS

This chapter briefly describes the technologies and tools used during the internship. The emphasis is on presenting an overview of the individual technologies with their purpose and necessity and closing the circle of tools used to successfully develop the objectives of the internship.

The technologies and tools will be elaborated in a chronological order to establish a good understanding of the workflow and BI process.

3.1.1. Amazon RedShift

Johnson and Johnson uses various information technologies to generate and store data. However, due to the vast amount of data that is being processed it’s easy to lose track of what is happening. Therefore, a very important process has been initiated during the last two years. The IT of J&J has to handle a very serious task, which is to establish Master Data Management (MDM). MDM in general is the core process of managing, organizing, centralizing, localizing and enriching master data as defined by the business in order to comply marketing and operational strategies.

As BI itself keeps evolving, data storing technologies do too. Data storing technologies such as data lakes, empower Business intelligence operations and processes. Businesses require information in a never seen velocity and variety and the problem starts at the data storage architectures.

The data lake was chosen as the solution because J&J generates both structured and unstructured data. This makes the data lake a compatible system as it supports both data formats, is scalable and enables fast querying for analysis purposes.

Amazon Redshift as technology itself, is a framework of many technologies which enables data warehousing and analytic services made possible by Amazon Web Services (AWS). An overview of the scope of AWS can be taken from the following illustration.

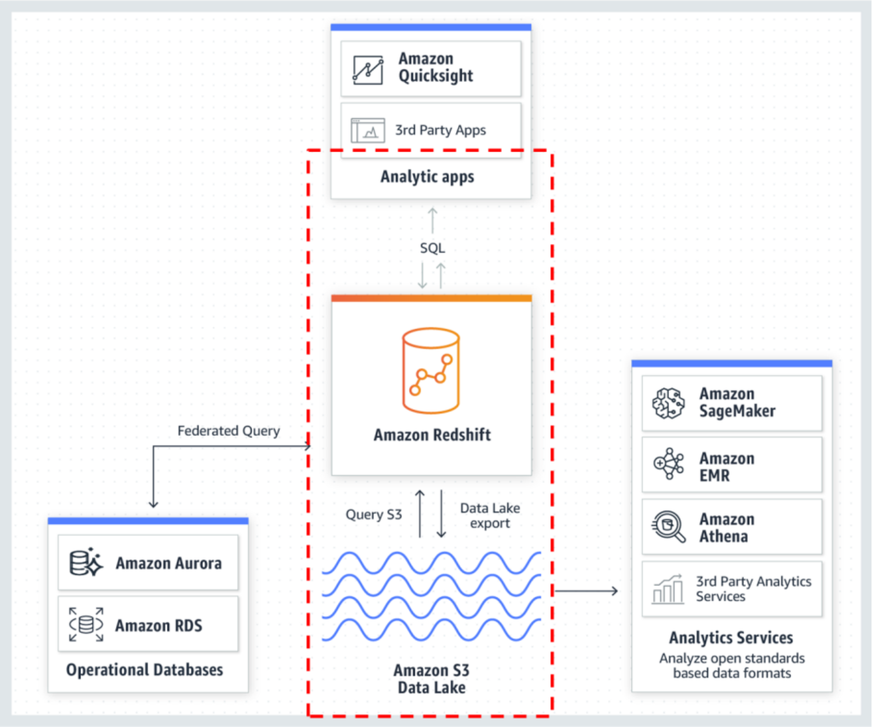


Figure 8: Amazon RedShift framework adapted from (Amazon, 2019)

According to the product specification, with Redshift it is possible to query petabytes of data using SQL statements. One of RedShift's feature is to store a query's result on to the data lake for further analysis with third party or Amazon services. Doing so results are much faster retrievable if one was to query it again.

The focus for the project's purpose is on the highlighted area.

Already mentioned above, a data lake is a suitable solution as a regular data warehouse would not fit to get information from unrestructured data. In short, a data lake is a centralized storage system designed to store all structured and unstructured data at any scale on the cloud. Two features of Amazon S3 as mentioned in the graphic are the D*ata Lake export and Federated Query S3*, both are significanttly important in order to improve managing the existing data warehouse and integrate with a data lake. A rough description of both functionality could be that on one hand the data lake export feature allows to unload data from a Redshift cluster in S3 in an efficient open column storage format, which is optimized for analysis purposes. On the other hand there is the federated query feature which enables the functionality to query data in the data lake and in Relational Database Services (RDS) using PostgreSQL. PostgreSQL (PSQL) is not much different than the regular SQL, the significant differences are that in PSQL more functionalities and data types are available (Amazon, 2019).

3.1.2. Aginity Workbench for Redshift

Aginity Workbench is a unified visual SQL database development tool designed for database architects and developers. It can be used for creating, managing and track both a whole database schema as well as SQL queries and that is due to the intuitive user interface. The main features of this workbench are that its relatively easy to import data from local files such as Excel files or CSV's or even from an external database into a table. The most used feature is the SQL-Editor as it allows to create DDL or SQL files which can be used to store important queries or even objects, hence the DDLs. The Query Analyzer allows the user to connect to multiple databases at once, which is quite helpful as due to the complexity of this project a lot of testing was required to compare against the source in order to verify if transformations have been applied correctly (Aginity, 2018).

3.1.3. QlikSense

QlikSense is the mainly the technology that is being used throughout the internship. Therefore, this part is going to be structured in three parts whereas first the technology will be introduced with its core features and functionalities. In the second part we will look more into the server capabilities. Finally, the methodology of Qlik will be elaborated to understand how the technology operates.

QlikSense is a business intelligence and visual analytics platform that supports a range of business cases which includes customized, standalone, analytic driven applications and dashboards as well as self-service capabilities all within a scalable and fully supported framework. It emerged from Qlik's first product QlikView. In QlikView you can manipulate the data in a lot of technical ways through its scripting language, which is somewhat of a hybrid of SQL and Qlik functionalities.

Nevertheless, the motive of Qlik Sense is to show and analyse data in the best possible graphical ways. There are numerous possibilities to visualize your data, it ranges from KPI's, tables, a variety of charts up to different type of maps (Rouse, TechTarget, 2016).

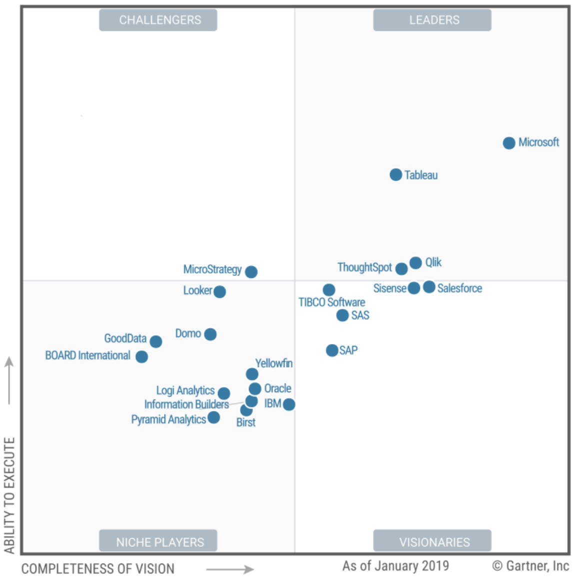
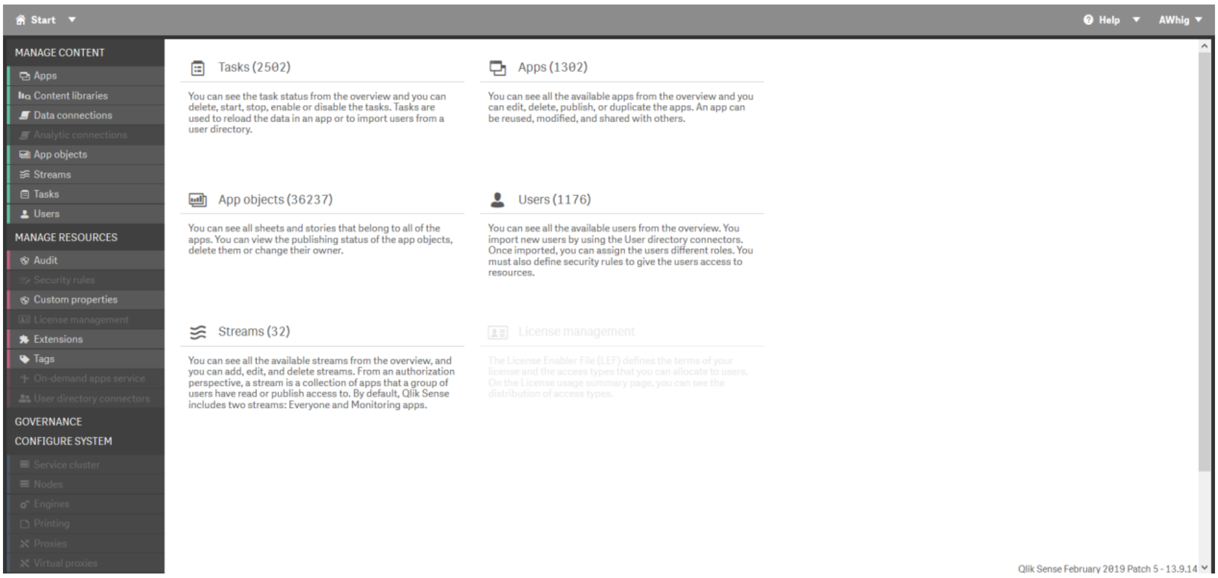


Figure 9: Gartners Magical Quadrant (Gartner, 2019)

According to GARTNER's Magical Quadrant for Analytics and Business Intelligence Platforms, as displayed in Figure 9, QlikSense, Tableau and PowerBI are the top leaders in terms of BI platforms. The Quadrant is structured in four types of technology providers, niche players, challengers, visionaries and leaders. As presented, it's clear that Qlik is amongst the big players here.

QlikSense offers both a desktop version, but also an enterprise version. As for the desktop version, it can be said that we have used this version locally on our desktops to create proof-of-concepts (PoC) for our customers in order to implement other solutions that can be used with BI. Aside from creating PoCs we also used the desktop version to create scalable solutions, try out new functionalities and test performance, as desktop computers have limited resources, in terms of computation power.

QlikSense for enterprises could be described as the enterprise ready client-server version of Qlik Sense Desktop. Every application is stored centrally on the server, which is responsible for security and administration. A powerful tool for QlikSense enterprise is the Qlik Management Console (QMC), it is used for all types of administration. The QMC basically, is a webpage which can be accessed from the organization's server. During the internship this was the most used tool to release new versions of the dashboard or refresh data. The picture below displays the QMC.



Tasks (2502)

You can see the task status from the overview and you can delete, start, stop, enable or disable the tasks. Tasks are used to reload the data in an app or to import users from a user directory.

Apps (1302)

You can see all the available apps from the overview and you can edit, delete, publish, or duplicate the apps. An app can be reused, modified, and shared with others.

App objects (36237)

You can see all sheets and stories that belong to all of the apps. You can view the publishing status of the app objects, delete them or change their owner.

Users (1176)

You can see all the available users from the overview. You import new users by using the User directory connectors. Once imported, you can assign the users different roles. You must also define security rules to give the users access to resources.

Streams (32)

You can see all the available streams from the overview, and you can add, edit, and delete streams. From an authorization perspective, a stream is a collection of apps that a group of users have read or publish access to. By default, Qlik Sense includes two streams: Everyone and Monitoring apps.

License management

The License Enabler File (LEF) defines the terms of your license and the access types that you can allocate to users. On the License usage summary page, you can see the distribution of access types.

Qlik Sense February 2019 Patch 5 - 13.9.14

Figure 10: Qliksense Management Console

The most commonly used administration tools in the QMC have been the options *data connection, apps, tasks and extensions*. The data connection as the name suggests are there to manage data connections for your apps. Here we define the connection to the data lake with user, password and server. Mostly here we defined multiple connections as for development and production data. In the apps section, we can create new applications to create dashboards on top of them. Most of the time one application can be assigned to a certain region or even department within a company. A way to handle this, is to create streams. In streams it is possible for each app to be assigned to a certain stream, which allows certain user groups to have access only to those apps which are defined in their stream. Basically, one can say this serves as a low form of security policy, however within an app we can still define security rules, configuring that certain user have only access to certain dashboards or limiting data so not every user sees the dashboard as the top management. In the tasks we define whether the data should be refreshed daily, monthly or as business wants it to be refreshed. As already described in the literature review section we differentiate between incremental or full reloads. Here, however we create tasks which we only incremental or full reload specific tables. Often due to data refreshes in the data lake it is necessary to refresh the data in the dashboard as well, it is also possible to manually trigger a reload in the QMC task section.

Extensions in the QMC can be various things. QlikSense itself comes with a bundle of built in graphs like charts, scatter plots etc. However, if its necessary to visualize data on a heatmap or even a geographical map, Qlik offers extensions which can manually extend this functionality.

Extensions also enable dashboards on the web, for that Qlik offers something they call *Mashup*. In the context of Web development, a mashup is a Web application that uses information from more than one Qlik application to create a single new service that is displayed on a single graphical interface, hence a website. There are many reasons to why a dashboard should be made available on the web, in this context it was to enable users to have their metrics available on the go, more on this topic was already described on the literature review.

3.1.4. Web technologies

QlikSense offers diverse functionalities as already mentioned above, due to the demand of the business it was critical to deliver a solution that works "on the go", which can be translated as Mobile BI.

The Mashup as Qlik describes it, needs to be built with various technologies. These technologies include JavaScript, HTML and CSS. Each technology plays a different part in building a Mashup, however each is equally important. On one hand JavaScript is needed to connect to the Qlik API, which provides the logic and decides how the webpage should behave and on the other hand HTML and CSS are necessary to align web objects in a more design appropriate way.

Mostly it is considered to use Qlik Mashups because the general functionalities are somewhat limited, and the idea here is to overcome those limitations. Some limitations can be solved by additionally loading more extensions or widgets to include minor functionalities in QlikSense. However, in order to build interactive and customized dashboards to visualize data in a more storytelling approach the Mashup here supports these demands. The following table will underline some of the capabilities and limitations the Mashups has over the desktop version.

QlikSense Desktop	QlikSense Mashup
Limited number of chart functionalities (in terms of dimensions and measures)	No limitations in including multiple dimensions and measures
Less room for customizing charts (in terms of design)	No limitations as a webpage is highly customizable
Charts have to be built multiple times	Convert existing charts to different types of charts
Limited number of charts in one sheet	Unlimited number of charts in one webpage

The focus here lies on the representation of data, obviously the data model, measures etc. must already been built in the app.

4. DEVELOPMENT CYCLE

The importance of this project is out of question, as SDG Group is a consulting firm, which provides IT solutions to its clients. Companies struggle with the huge amount of data they are storing, they don't know which insights they can generate out of their data. Building a customized solution which is capable of storing, processing and analysing such data can make significant change in their businesses. SDG Group has been providing various clients with such solutions, however each project in this matter is different. Not all companies have the same IT infrastructure, therefore new projects are always challenging.

The project here consists of two phases, one of them is, data has to be extracted from all different existing sources, transformed in the new warehouse system and load into it, in the second phase data has to be presented in a story telling way, summarizing all important findings to enable decision making.

4.1. PROJECT OBJECTIVES

The objective of this project in terms of a technological point of view is simple. First, the data which the client is providing has to be understood in terms of what fields or more particularly which data sources have to be aggregated. As for now the client was hosting their data in an outsourcing firm, which was also providing a BI solution. However, that solution did not deliver the expected result.

Therefore, understanding the data and its sources first is top priority. Furthermore, building the correct data model is essential for this solution as data modelling is all about creating conditions to make any analysis on the data possible. A good communication between the business is highly important here as we they have to make sure we understand what is required in order to meet the demands of their business.

So, in short, the objectives are the following:

1. Identify business requirements
2. Identify the data sources and load them into the central data lake
3. Design the data model with all its relations
4. Clean Data, if necessary
5. Build reports, dashboards and identify KPIs based on client's requirements
6. Deliver Documentation

4.2. METHODOLOGY

The described project will follow the existing methodologies of SDG Group. SDG Group has a unique framework, which is used to build BI solutions. The development of the application will first of all follow the SCRUM methodology and the technologies used will be QlikSense which is the BI software, namely the Backend of the app and the frontend will be done with HTML, Javascript, AngularJS and of course CSS.

Currently the data of Janssen Pharmaceutical is stored in a relational database but also other sources such as CRM, Excels and so on. The first task is to aggregate the data which is needed and Transform and Load it into QlikSense. In order to do that, a star schema has to be built, with one fact table and many dimension tables. The number of the dimensions is still yet unknown, therefore the usage of the SCRUM Methodology which leads to continuous improvement of the application, without interference of functionality will determine that at the end of the project.

The team of this project consists of two scrum teams, one internally in SDG Group and one from Janssen. The team in general, typically consist of a product owner and a data team and us, the developers. We are structured in one mashup developer (front end) and one application developer, however as I am new to this field my tasks concern both back- and front end. This was done as to lay a foundation for both areas.

As already mentioned, the development of this project was done with the SCRUM methodology. SCRUM is a project management method that focuses on an incremental approach to develop software, however it can be used in different scenarios and is not exclusively used for software development. Typically, there are three different roles in SCRUM, a product owner, a SCRUM master and a development team. A SCRUM project usually consists of a series of sprints, and each sprint usually takes between two to four weeks. The following project plan will deliver an overview of the project and how it was structured.

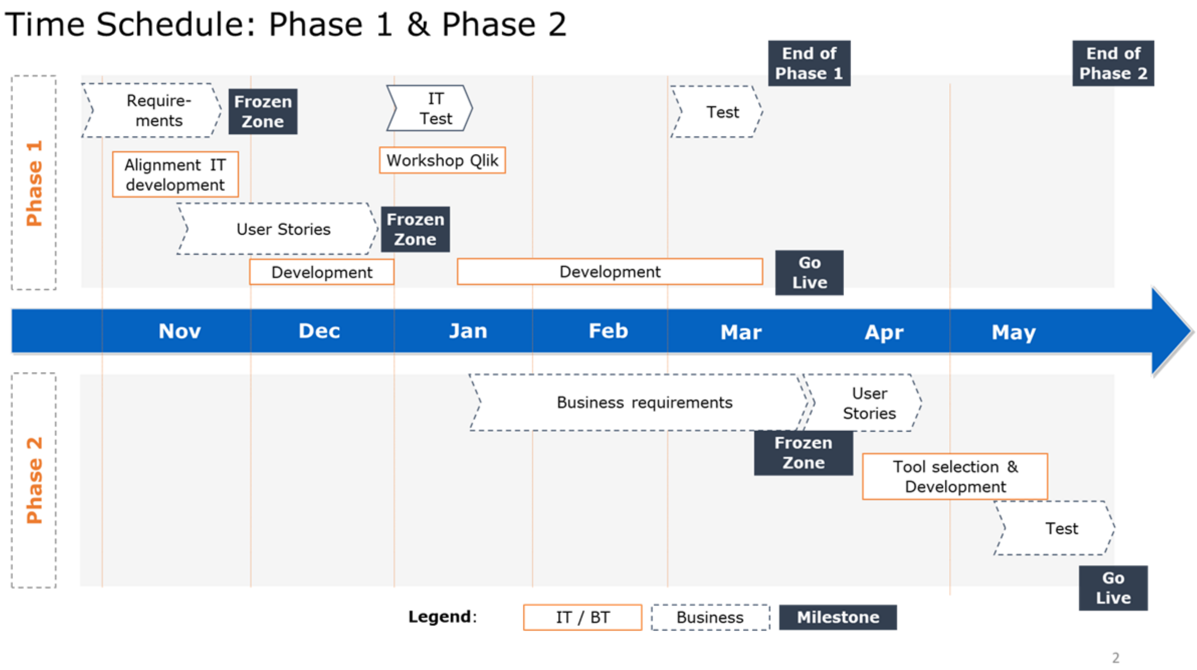


Figure 11: Project plan

In order to rollout a SCRUM project, the product owner must create and prioritize a backlog that consists of user stories. These user stories were groomed with the development team during a workshop where the business users discussed their requirements with us, the developers. For the first phase the scope was to move all data that is necessary to the data lake. For this purpose, the scrum master has defined sprints of 2 weeks and additionally User acceptance tests (UAT) for 1 week. During a UAT the business tests the developments to verify if what was developed is aligned to the requirements, which were specified in a user story. If requirements have been missed or developed wrong, bugs are being created which have to be solved during the UAT, as here the development team does not start working on new stories but provide support and solve issues that may have arisen. Both phase 1 and phase 2 are dependent of each as the second phase can only begin once phase 1 has finished.

4.3. Development activities

Talk about data in different sources … establish single source of truth blab la is best practice and shit

4.3.1. Phase 1 – Establish a single source of truth

Based on what we have learned in the classes in Nova IMS, a single source of truth needs to be established in order to manage data in the most efficient way. However, most enterprise rarely implement it in an ideal way, and this is due to having multiple information systems, which all often relate to the same entities and therefor can cause inconsistencies and also trust issues on the data.

In Janssen this was the reason to rethink the data management - they have started to establish this single source of truth by aggregating all data to a central data lake, which stores all information to feed dashboards and reports.

Currently, there are about four different data sources which all need to be centralized into the data lake. The current situation can be seen in the following graphic.

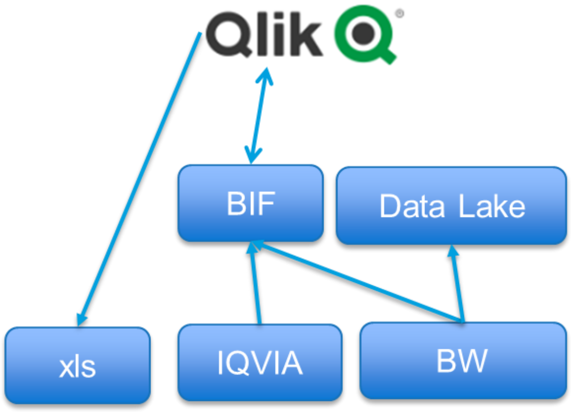


Figure 12: Current data architecture

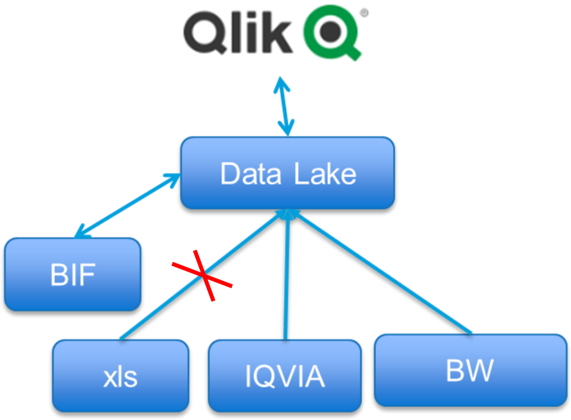


Figure 13: Ideal data architecture

Initially, the idea of the business department was to feed our dashboard, which will be developed in phase 2 using all these external sources without harmonizing the data sources. Clearly, we already pointed out issues that may arise. We will face data which can be duplicate, inconsistent and not trustworthy. The business is using a large number of Excel files to calculate their measures. IQVIA is an external data vendor, which provides data that is not available to the business itself, however its crucial as it can contain competitor product information, which can be used to empower decision making. BIF and BW are both internal data sources, which contain a lot of master data but also sales data. Our proposal of the ideal data architecture can be seen in Figure 13. The reason we proposed this idea, is because of the limitations the business had. In the end we will centralize all data of all Excel files, the data sources IQVIA and BW in the data lake. In order to do so, we wrote different transformation jobs to store the data in the data lake, which are running daily to guarantee consistency, however we will understand these jobs more in detail later.

In order to understand what data is even needed, we had to talk to the business and understand what they need in their dashboard, to do so we held multiple meetings and workshops to get everything right and fully understand the needs. In short, the business context of this project is to calculate bonuses for hospitals and pharmacies, namely accounts with the amount of medications they are going to buy. Because if these accounts buy a certain number of medications, they will receive a discount. However, not all accounts are also buying products, we can have information about potential accounts too, however in order to differentiate those, we need to somehow split the information. Obviously, all this data is associated with timestamps, because organizations keep track of this information. Apart from that, we can see that we have some kind of sales and quantity information, because we need to check how much has been made in terms of quantity of products sold and sales volume. So, by already having so little information we could identify different master tables and one fact table, which is important throughout the project. Bundling all this information I was able to build an abstract data model (as shown in Figure 14), which I had to confirm with the business to make sure everything was captured.

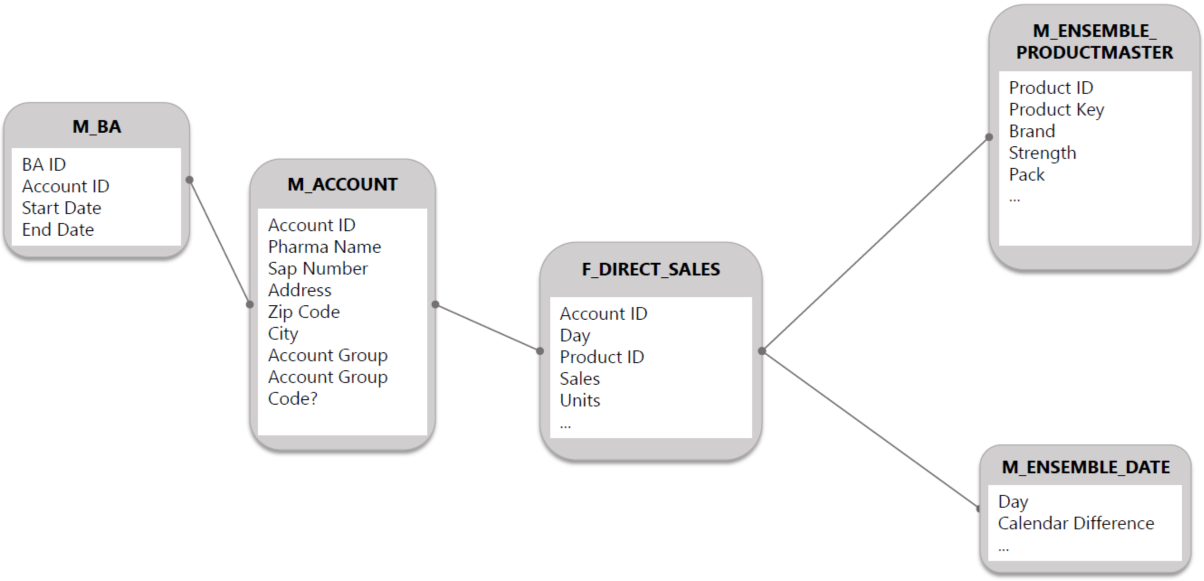


Figure 14: Abstract data model

The result of this data model made it easier for me to understand, which data will be needed to make available in the data lake. As during this process, we were able to eliminate certain data, that is not necessary. The fields which are shown in the data model are just a few, which were identified in the early stage. We did a field mapping with the business to identify, what information they need in their dashboards. However, in phase two we will look into what fields were identified.

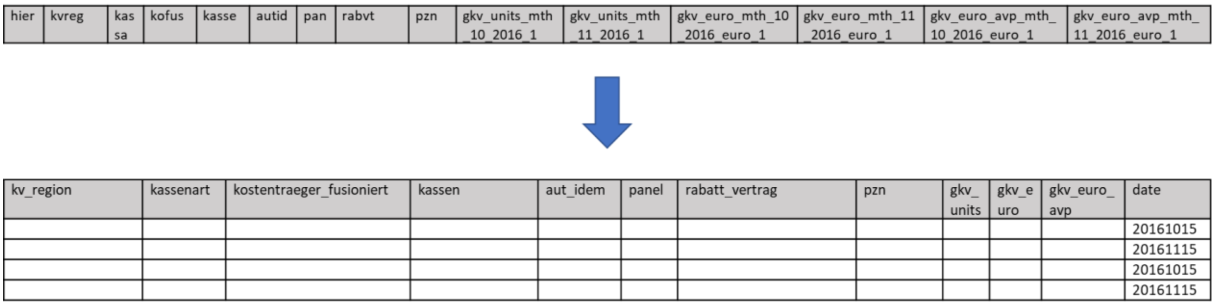
Business defined the multiple user stories, in order to keep track of the work and work with the SCRUM methodology as elaborated in the earlier section. The following tables are just a sample of the user stories, however those mentioned, will be elaborated in terms how these were developed.

Table 4 - User story #1 - IQVIA

User story	Integrate data delivered by IQVIA in the data lake
Description:	The data vendor IQVIA will deliver 102 Excel files each month with sales data. All 102 files have the same structure, except for the information. Data had to be split for each region and into 6 health insurances. • The flat file will be deliverd to the following incomming path: ***.../Layer1/DE/Sales/*** • The transformations will take in the refined layer: ***.../Layer2/DE/Sales/*** • The outgoing path of the processed file is: ***.../Outgoing/DE/Sales*** The table in which the data should be stored is ***DE_ContractSales*** Columns are comma seperated and files structure is fixed and similar in the 102 files. File contains always 36 months of sales. Columns of the files are the following: hier **(ignore)** kvreg as **kv_region** kassa as **kassenart** kofus as **kostentraeger_fusioniert** kasse as **kassen** autid as **aut_idem** pan as **panel** rabvt as **rabatt_vertrag** pzn as **pzn** gkv_units_mth_xxx as **gkv_units** gkv_euro_mth_xxx as **gkv_euro** gkv_euro_avp_mth_xxx as **gkv_euro_avp** **(xxx = mm/yyyy)** Older file(s) should be archived into a specific historical folder. **Transformations**: rename field names and transform into a row oriented way, by including a date field.

The scope of this user story is clear, what needs to be done is to integrate the 102 files into the data lake. There are no difficult transformations as can be seen in the description of the story. In Janssen, they use a set of specific software to carry out their work. For this purpose, I was to use a technology called *MBOX,* which is a tool created by Janssen itself. Initially, working with MBOX was difficult as I have never heard of it. Nonetheless, it is a really simple way to integrate Excel files into the data lake, all that needs to be done is to define the three paths as described in the user story and then apply the transformations. Unfortunately, I cannot disclose the technology in this paper as it belongs solely to Janssen.

In general, we had to offer the data vendor an FTP server in which they would drop the files, as defined in the description, which we would then pick up and aggregate into one Excel file. In the MBOX we would then begin with the transformation process, which is first to remove duplicates as duplicates have no use for the business and can in the end just falsify information. Additionally, I would rename the columns as specified in the user story. As the requirement was to transpose the information in a row-oriented way, as we have sales information separated in columns for all the 36 months. In the next figure we can see, how the transformation was intended.



hier	kvreg	kas sa	kofus	kasse	autid	pan	rabvt	pzn	gkv_units_mth _10_2016_1	gkv_units_mth _11_2016_1	gkv_euro_mth_10 _2016_euro_1	gkv_euro_mth_11 _2016_euro_1	gkv_euro_avp_mth_ 10_2016_euro_1	gkv_euro_avp_mth_ 11_2016_euro_1

kv_region	kassenart	kostentraeger_fusioniert	kassen	aut_idem	panel	rabatt_vertrag	pzn	gkv_ units	gkv_e uro	gkv_euro_ avp	date
											20161015
											20161115
											20161015
											20161115

Figure 15: Transposed IQVIA data

The reason to transform the data in a row-oriented way was not only due to the fact that it is easier to read but also because the data lake is designed in a traditional row oriented relational database.

Due to the fact, that for other countries IQVIA data was already being imported into the data lake, I didn’t additionally need to apply any other data cleaning processes. IQVIA, itself holds all the fact information, as shown in the draft data model.

Furthermore, after successfully importing the data into the data lake, I needed to test the data to verify, if everything was loaded successfully. For that I created test scripts, which would guarantee that the transition from Excel to the data lake was flawless. The test script for this user story looked like, as described in the following table.

Table 5 - SIT test script

Step	Data	Expected Result
Verify if all specified columns exist and if they were renamed correctly and the transformation to a row oriented format, including the date field was applied.		Transformations are applied correctly.
Verify if the outgoing file contains all data for the past 36 months.	Month of extraction -35 months.	Data is complete.
Verify if the outgoing file contains all 17 KV regions	Baden Wuerttemberg Bayern Berlin Brandenburg Bremen Hamburg Hessen Mecklenburg-Vorpommern Niedersachsen Nordrhein Rheinland-Pfalz Saarland Sachsen Sachsen-Anhalt Schleswig-Holstein Thueringen Westfalen-Lippe	Data is complete.
Compare records against Source file(s) and verify if the records are matching.	Filter for November 2018	Records are matching

For each step I had to provide a screenshot highlighting, that everything is as it is supposed to be. These steps for us developers are some kind of insurance to show, that at the time the story was successfully developed everything was working as defined in the user story. The steps mentioned above are making sure, that the requirements defined in the user stories were met, we call these tests System Integration Tests. The testing, itself has to be done by the developer himself. Once the test was passed and evidence was attached, the developments carried out will be moved from the development environment to testing environment, where the business users itself will test the developments on their own, to ensure everything was done as asked for.

Evidently, these are not all the Excels which had to be imported into the data lake, however the remaining ones have a similar set up and therefore won't be considered.

Furthermore, having all IQVIA data imported into the data lake, the next challenge was to move on and integrate the Excels and SAP BW data into the data lake. This data is mostly master data, however the data lake already holds master data that is shared across the whole Europe, the Middle East and Africa (EMEA) region, therefore I needed to make sure, that we don't integrate data that's already in the data lake, as the goal is to build a central hub for data without inconsistency.

The masters, as described in the abstract data model are:

- *Products*
- *Calendar*
- *Accounts*
- *Buying associations*

For all the underlying master data, what I needed to verify, was to check if the data already exists somewhere in the lake. For that I had to check against the SAP data source to make sure everything is exactly as it is. If there was a source that is not in the data lake, I had to integrate it. Fortunately, every master was there but, for that I had to create views that only hold data for Germany, as the other countries are not for our concern.

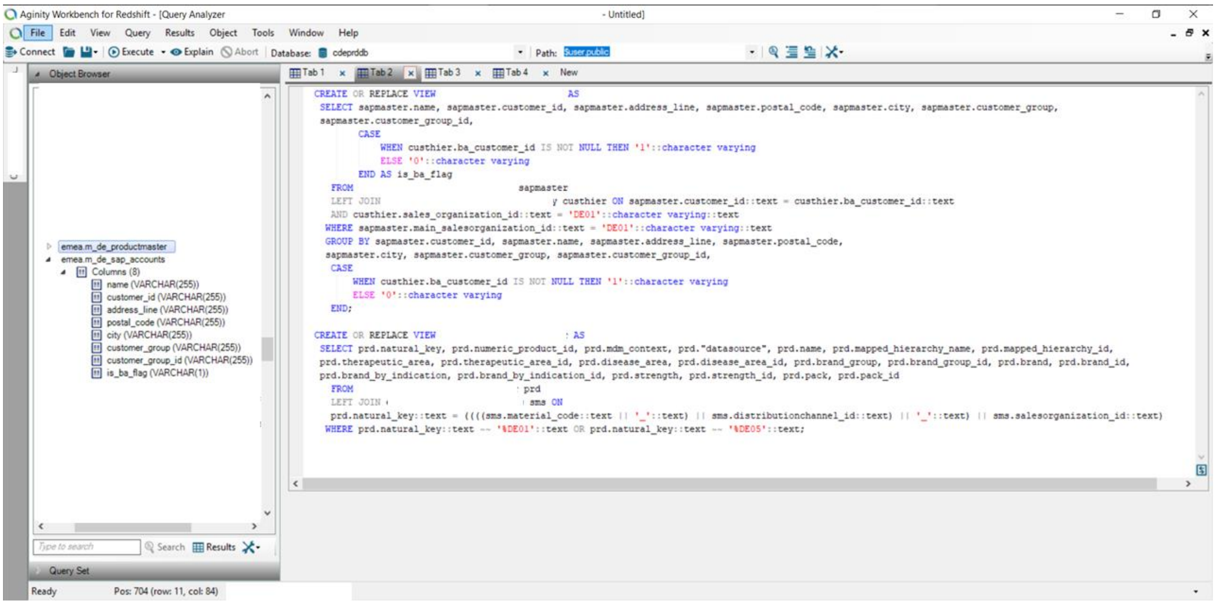
Therefore, the business created user stories for each master dimension to be created as a view in the data lake, under the premise that the data is available.

Table 6 - User story #2 - Create views

User story	Create views for Accounts and Products
Description:	▪ Create Redshift View **M_DE_Accounts** Customers Data (Source is SAP Master account table): **Fields:** ○ name ○ customer_id ○ address_line ○ postal_code ○ city ○ customer_group ○ customer group id ○ is_BA Flag (Customer is Buying association) 1= BA found, 0=BA not found **<u>Important</u>**: Filter on salesorganization_id = 'DE01' for Germany only. ▪ Create Redshift View **M_DE_Products** view Product Data (Source is SAP Productmaster table) **Fields**: ○ natural_key ○ numeric_product_id ○ mdm_context ○ datasource ○ name ○ mapped_hierarchy_name ○ mapped_hierarchy_id ○ therapeutic_area ○ therapeutic_area_id ○ disease_area ○ disease_area_id ○ brand_group ○ brand_group_id ○ brand ○ brand_id ○ brand_by_indication ○ brand_by_indication_id ○ strength ○ strength_id ○ pack ○ pack_id Important: Additionally filter the natural_key only for DE01 and DE05 products.

I have put the two user stories in the table above as they are developed similar. In order to check whether the needed data is already in the data lake, I had to run a query on the main product table in the data lake, afterwards I had to make sure if data for German products is there, as this concerns us in this context.

Furthermore, in Aginity Workbench I had to query on the already existing tables so check if we have data for Germany, as shown in the graphic below. As I have already explained before, we use Aginity Workbench to create views, tables and query on them. The programming language used here is PSQL.



```
CREATE OR REPLACE VIEW                          AS
 SELECT sapmaster.name, sapmaster.customer_id, sapmaster.address_line, sapmaster.postal_code, sapmaster.city, sapmaster.customer_group,
 sapmaster.customer_group_id,
        CASE
            WHEN custhier.ba_customer_id IS NOT NULL THEN '1'::character varying
            ELSE '0'::character varying
        END AS is_ba_flag
  FROM                              sapmaster
  LEFT JOIN                               y custhier ON sapmaster.customer_id::text = custhier.ba_customer_id::text
  AND custhier.sales_organization_id::text = 'DE01'::character varying::text
 WHERE sapmaster.main_salesorganization_id::text = 'DE01'::character varying::text
 GROUP BY sapmaster.customer_id, sapmaster.name, sapmaster.address_line, sapmaster.postal_code,
 sapmaster.city, sapmaster.customer_group, sapmaster.customer_group_id,
  CASE
      WHEN custhier.ba_customer_id IS NOT NULL THEN '1'::character varying
      ELSE '0'::character varying
  END;

CREATE OR REPLACE VIEW                          : AS
 SELECT prd.natural_key, prd.numeric_product_id, prd.mdm_context, prd."datasource", prd.name, prd.mapped_hierarchy_name, prd.mapped_hierarchy_id,
 prd.therapeutic_area, prd.therapeutic_area_id, prd.disease_area, prd.disease_area_id, prd.brand_group, prd.brand_group_id, prd.brand, prd.brand_id,
 prd.brand_by_indication, prd.brand_by_indication_id, prd.strength, prd.strength_id, prd.pack, prd.pack_id
  FROM                              prd
  LEFT JOIN                          sms ON
  prd.natural_key::text = ((((sms.material_code::text || '_'::text) || sms.distributionchannel_id::text) || '_'::text) || sms.salesorganization_id::text)
 WHERE prd.natural_key::text ~~ '%DE01'::text OR prd.natural_key::text ~~ '%DE05'::text;
```

Figure 16: Creation of account and product view

Gladly, I was able to find out that we already have the accounts stored in the data lake and they were matching with the information in our SAP BW. I was able to confirm that the data matched, due to a few data quality checks I had prepared. The data quality checks I did both checked, if the data is matching with the current SAP source and complete. The checks I preformed are like a test script, as shown in the table below.

Step	Data	Expected Result
Verify if we have the same number of customers in the view as in SAP.	`SELECT DISTINCT count(customer_id) FROM m_de_accounts;`	Number of customers matches SAP.
Verify if all fields are available in the view as specified in the user story.		Data is complete.
Compare records against Source file(s) and verify if the records are matching.	`SELECT DISTINCT customer_group FROM m_de_accounts;`	Records are matching

Obviously, it is impossible to check each record one by one, manually. Therefore, I was trying to check the completeness by samples.

Once I was fully sure, that the data is the same, I created a view in Aginity only based on German data. In the above-mentioned figure, I am applying a condition on the *sales_organization* field to only filter for *DE01*. Additionally, I created a flag, that indicates if a customer is a buying entity or just a regular customer. The difference here is that, basically, every customer is a potential buying entity, however he or she may not be buying and medications yet but can be contacted. The value *0* here indicates that he's just a regular customer and *1* indicates that he is already buying medications. A customer in this context is either a pharmacy or a hospital.

Similarly, to the customer view I created the product view with fields defined. The only significant difference was that we are filtering for *DE01* and *DE05.* The reason that we are doing this is because we also want to analyze competing products, which are under *DE05*. Now the product view has many fields, and this is due to the granularity they want to represent their data in the dashboard.

Now, if we again look at Figure 13, we now have eliminated the Excels and SAP data source as we successfully loaded everything into the data lake, with the development steps explained above. The only source we cannot switch completely to the data lake is BIF but it was identified that the BIF data was already in the data lake in other tables and we therefore had to create views on top of them so we can have ownership of those.

In Phase 2 I will elaborate how I built the dashboard. As this was a team effort, my team leader mostly took care of building the data model and my part was to build the front end in the Mashup, in the following chapter I will explain how this was done.

4.3.2. Phase 2 – Develop a dashboard

The business wanted an intuitive and easy to understand dashboard, they have defined what metrics they want to show as KPIs and how they want to show other data, whether in a pie chart, table or line chart. It was my responsibility to build them and apply them on the web front end.

The effort be carried out, has to be separated in two steps. On one side I have to create all the different visualizations on a QlikSense application and on the other side I have to create a webpage and align the visualizations in a story telling way. Furthermore, I already created the sheets in QlikSense as I would create the frontend in the webpage, as the idea from my side was to first demonstrate a frontend layout first and once it looks appealing to business, they could then say how they would like to have it changed for the final web front end look.

The total number of visualizations that I created was around 34, therefore I cannot explain each one of them, therefore I will provide one example below.

The first sheet I created was to deliver an executive summary of the most important metrics, which is of great importance to the management. Of course, there are much more things to show, however I was trying to keep it simple and clean and not overload the sheet with too much information.

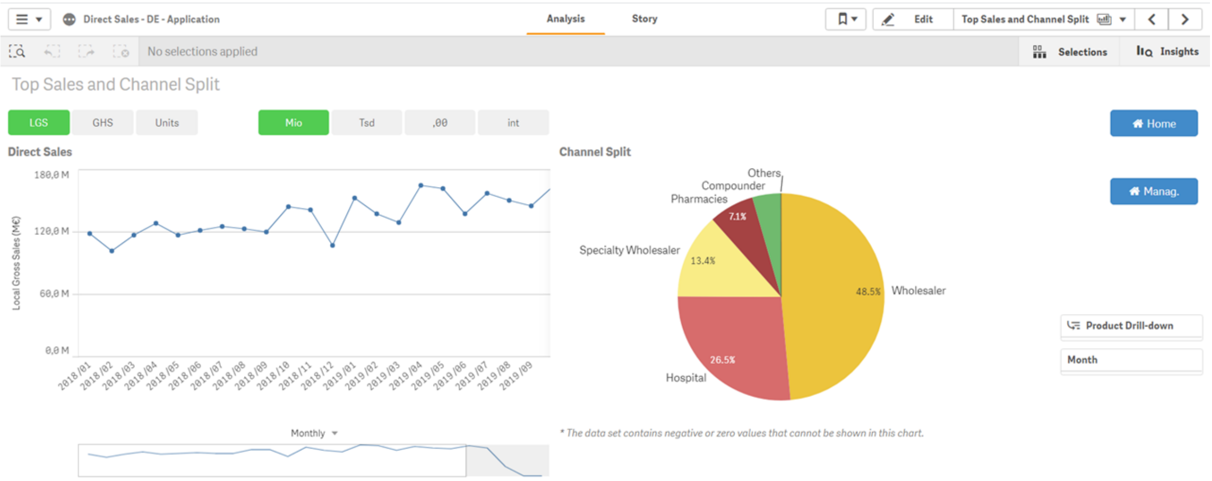


Figure 17: Mangement summary sheet

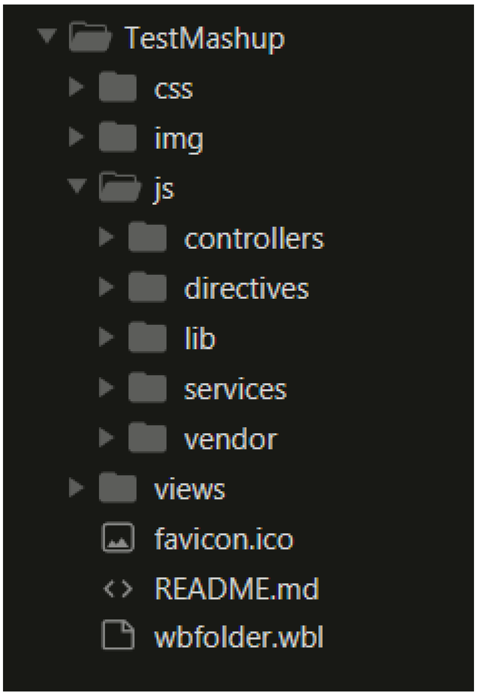
Looking at the first sheet, as shown in Figure 17, it's clear that I want to visualize the Gross Sales over a timely period in the line chart and in the pie chart I wanted to visualize the same but in terms of distribution by channel. On the top we have toggles, that make it possible to switch measurement unit. Almost on the bottom right corner, we have two filters, that allow the user to change for a specific product and also filter for a specific month.

In order to build these visualizations in QlikSense, you have to drag and drop the specific visualizations you want to build and specify the dimensions you want to use such as *Month, Product* etc., afterwards you have to define a measure that you want to use. In this case I summed the sales with additional filters to accept selections made by users.

Now, as mentioned above, I have created multiple visualization object, but we don't have to go through all of them, the example provided above can be applied to the remaining ones.

Furthermore, as the business accepted the suggested dashboard, the next step was to implement this on a web based front end.

For this, lets first have a look at the structure of the Mashup, as shown in Figure 18:

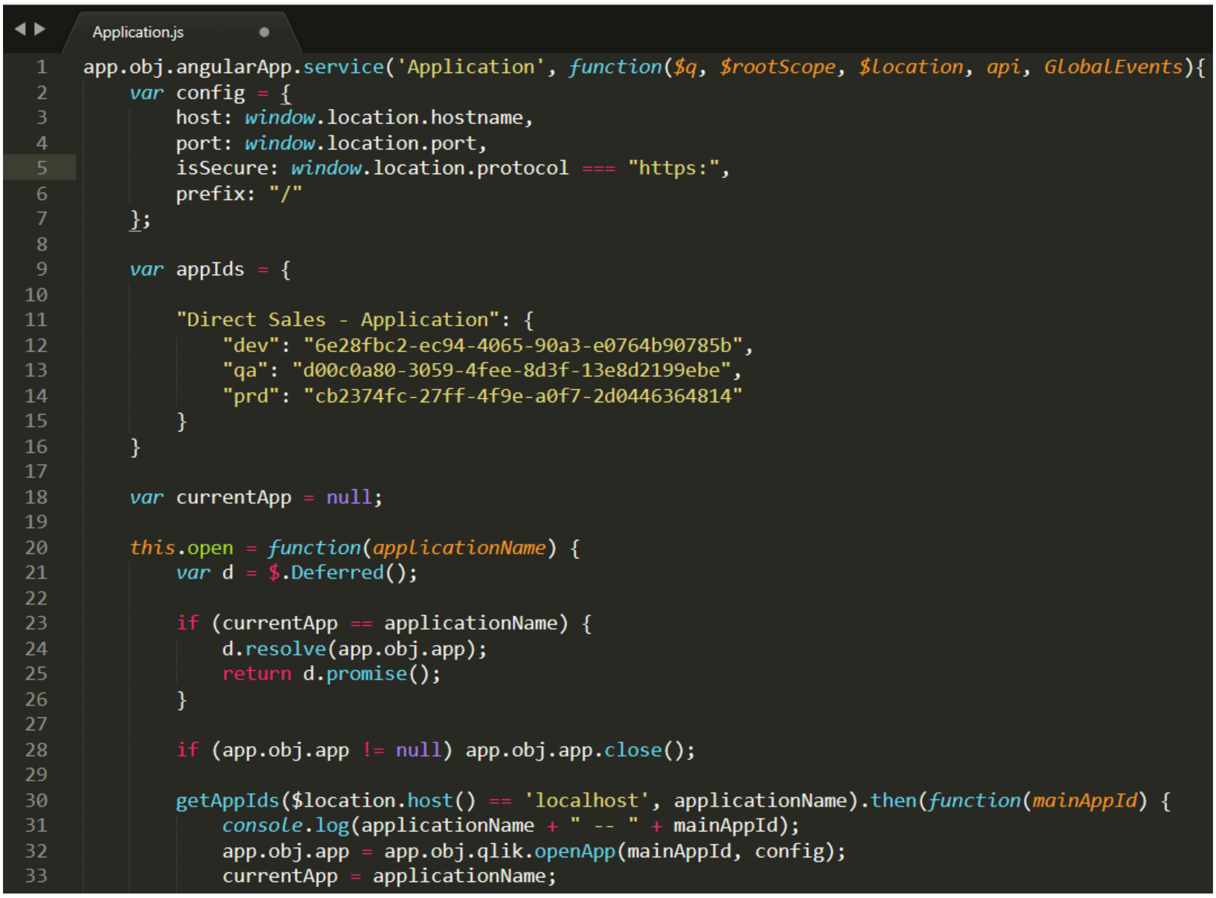


```
TestMashup
  css
  img
  js
    controllers
    directives
    lib
    services
    vendor
  views
  favicon.ico
  README.md
  wbfolder.wbl
```

Figure 18: Mashup structure

Clearly, as for any other web page, we have CSS files with which we can define the layout and the general design of the webpage, then we have an image folder, in which we store all images which are going to be used throughout the webpage. Furthermore, we have some JavaScript files that are structured in multiple folders such as: controllers, directives, libraries, services and vendor. The reason why we separate, these JavaScript files is because each of the named folders has a different necessity. Since we are working with AngularJS which uses Model-View-Controller (MVC) framework, we have to structure, it this way. Basically, MVC is the most common used software development framework for web applications.

Since I already created a QlikSense application with all the charts, the next step was to connect this application with the Mashup. In the JavaScript file Application.js I have to establish the connection for all environments, thanks to Qlik API its really easy to do so. In the below graphic, its visible how this was done



```
app.obj.angularApp.service('Application', function($q, $rootScope, $location, api, GlobalEvents){
    var config = {
        host: window.location.hostname,
        port: window.location.port,
        isSecure: window.location.protocol === "https:",
        prefix: "/"
    };

    var appIds = {

        "Direct Sales - Application": {
            "dev": "6e28fbc2-ec94-4065-90a3-e0764b90785b",
            "qa": "d00c0a80-3059-4fee-8d3f-13e8d2199ebe",
            "prd": "cb2374fc-27ff-4f9e-a0f7-2d0446364814"
        }
    }

    var currentApp = null;

    this.open = function(applicationName) {
        var d = $.Deferred();

        if (currentApp == applicationName) {
            d.resolve(app.obj.app);
            return d.promise();
        }

        if (app.obj.app != null) app.obj.app.close();

        getAppIds($location.host() == 'localhost', applicationName).then(function(mainAppId) {
            console.log(applicationName + " -- " + mainAppId);
            app.obj.app = app.obj.qlik.openApp(mainAppId, config);
            currentApp = applicationName;
```

Figure 19: Establish connection between Mashup and App

I am doing a few things here, first of all I am defining the applications across all environment namely, Development, UAT and Production. Moreover, in the end I am making use of the Qlik API, which can be seen when I use *qlik* in the functions. The method *openApp* basically expects an App Id and a configuration object, which can be empty, however for this purpose this configuration object is defined on the top and just specifies the URL.

After the connection has been established, I can finally implement the created visualizations in the webpage. For each web page I have to create one view and one controller. As I already proposed a frontend design in the app to the business, it was easy to implement this in the web application. The outcome was the following.

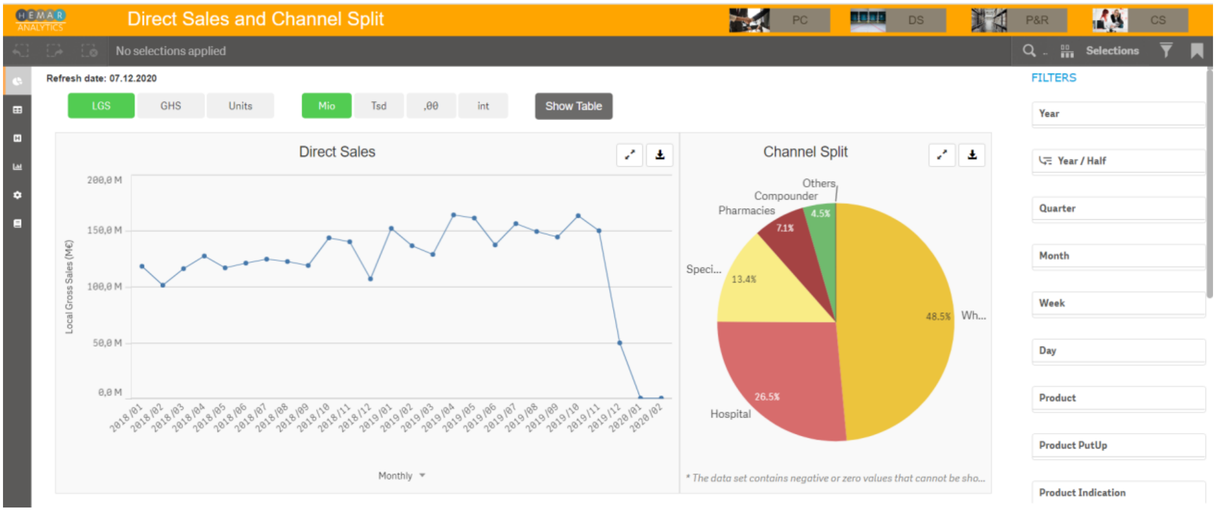


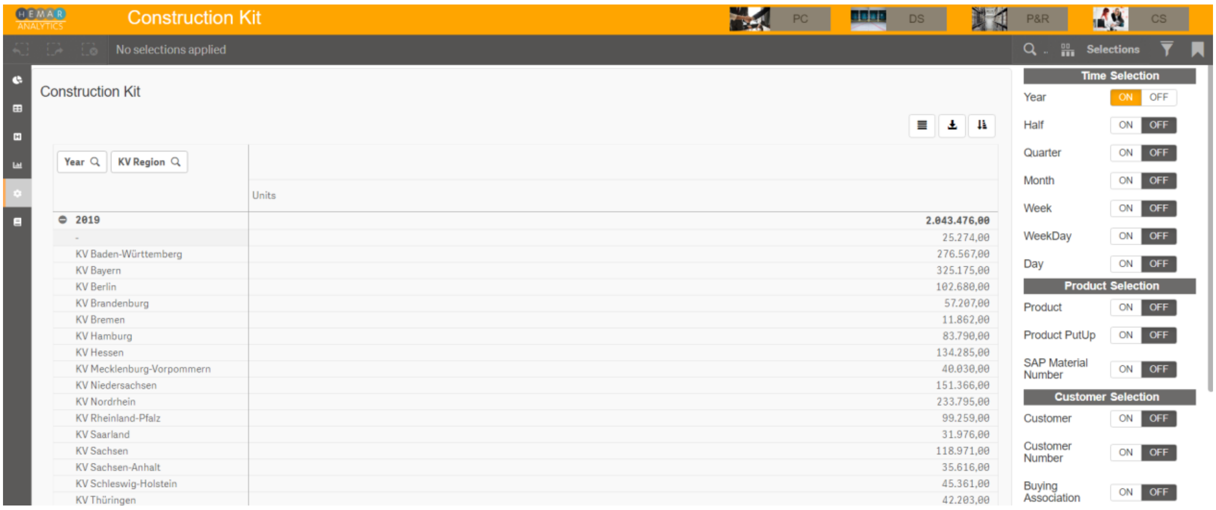
Figure 20: Direct Sales and Channel Split

This final result resembles, the mock-up I created in the application. There are only a few extras here. For once, I implemented the functionality to download a CSV file with the data in the graphs, which can be seen as the download icon and next to it there is a button to full screen the graph to have a better understanding. The *Show Table* button will transform the line chart into a table. In terms of how this was developed, basically, the Qlik API is powerful and easy to understand I created a few directives that make it easy to call the created visualizations in the web application. In Angular directives are used to extend the power of the HTML by giving it new syntax and in order to create a directive you both need to create the functionality in JavaScript and the syntax in HTML.

```
<get-object name="VisualizationName"></get-object>
```

All that needs to be done to get an object from the app into the Mashup is to create the charts in the app as *MasterItems* and those have unique names, which then can be called in the Mashup in the placeholder *“VisualizationName”.*

Apart from this view, the remaining one are pretty similar, except for one. The business wanted some kind of Self-Service functionality in which they can create tables on their own with their preferred dimensions and measures. This actually was a big challenge because the objects are created "on the fly" which is difficult to monitor, and it has to be done dynamic. For this purpose, I created a Hypercube, which allows me to create a Cube with predefined dimensions and measures, with the difference that this Cube is created in each session and is not therefore static. As this is really performance heavy, I had to specify a selection of dimensions and measures with the business. The result in the front end looks like the following image, on the right we have the available selections and, in the center, we can see the table, with the selected dimensions and measures.



Construction Kit

Year / KV Region	Units
2019	2.043.476,00
-	25.274,00
KV Baden-Württemberg	276.567,00
KV Bayern	325.175,00
KV Berlin	102.680,00
KV Brandenburg	57.207,00
KV Bremen	11.862,00
KV Hamburg	83.790,00
KV Hessen	134.285,00
KV Mecklenburg-Vorpommern	40.030,00
KV Niedersachsen	151.366,00
KV Nordrhein	233.795,00
KV Rheinland-Pfalz	99.259,00
KV Saarland	31.976,00
KV Sachsen	118.971,00
KV Sachsen-Anhalt	35.616,00
KV Schleswig-Holstein	45.361,00
KV Thüringen	42.203,00

Time Selection

- Year [✓ ON] OFF
- Half ON [✓ OFF]
- Quarter ON [✓ OFF]
- Month ON [✓ OFF]
- Week ON [✓ OFF]
- WeekDay ON [✓ OFF]
- Day ON [✓ OFF]

Product Selection

- Product ON [✓ OFF]
- Product PutUp ON [✓ OFF]
- SAP Material Number ON [✓ OFF]

Customer Selection

- Customer ON [✓ OFF]
- Customer Number ON [✓ OFF]
- Buying Association ON [✓ OFF]

Figure 21: Self Service table

The goal of this Self-Service functionality is that each user can create a table with their own input fields, by doing so we are avoiding to create multiple views with for each user specified table and avoid redundancy.

In summary, the whole purpose of this Business Intelligence project is to provide clear, accurate, automated and timely information to decision makers and business users. Our client is now able to identify possible outliers and anomalies in their previously untapped millions of records and present them in a user-friendly interface. Naturally, this dashboard will be further and continuously developed and monitored to ensure stability for the future.

5. CONCLUSION

The internship at SDG Group has been a wonderful experience both personally and professionally. In this section I would like to conclude this internship in three parts, whereas first I want to evaluate the experience, considering the impact on my professional experience and general motivation. Secondly, I want to critically asses the internship in all aspects and at last my future perspectives.

5.1. EVALUATION OF INTERNSHIP

My motivation to work in yet, another consultancy was a really easy choice. I have always preferred having a strong communication with other teams and work with project management methodologies. My bachelor's degree in Business Information Systems had already prepared me perfectly to work with several IT technologies and programming languages such as SQL, which has been quite handy throughout the internship. In the early stages, I was a bit nervous as this would have been my first professional experience in business intelligence, however I quickly noticed that I already know a few things as, during my masters in NOVA IMS I was already thought a few principles and best practices for data warehousing and business intelligence. Furthermore, working for consultancy companies is always a challenging experience as here we have deadlines to meet and we work in several fields all in once. For example, during the internship I had to learn about health insurances and pharmaceutical business, which for me was something completely new. I had no idea about these business fields and moreover I didn't know their day to day business. Nonetheless, I chose to apply here in SDG Group Portugal. I could have done my internship also in Germany; however, I always wanted to acquire international experience and see how different Germans work in comparison to Portuguese, gladly there was no difference at all. In the end it was just a matter of a language barrier that I had to overcome, however during work hours our communication language was English, which made the work easier. Still, the reason, why I did an internship first, was due to the fact that this was my first experience in BI and I quickly wanted to use the knowledge I acquired during my masters in all the courses from theory in a practical environment, too.

Looking at the project I was assigned to at SDG Group, I can say it was not an easy task that I was assigned to. The development I had to carry out in both SQL and QlikSense were somewhat challenging, as I had no practical programming experience. Still due to the help and guidance of my team and my colleagues, I quickly learned and could overcome those difficulties. I started off as a Junior Consultant and still I was treated equally, even the business was really complying as they knew I was a beginner. I could quickly gain their trust as I was delivering what was asked for in a short period, therefore I can say I was happy once I started receiving feedback, it really helped me building confidence and like the work I was doing. The team, I was working with, had strong discipline and self-control, by that we could avoid disputes. Even, if there were issues, we quickly could talk them out in order to improve our teamwork. Overall, thanks to this environment and ability to change or even talk it was extremely comfortable to work here.

Despite all this positivity during the internship, there were a few drawbacks that I must mention. Early on, when I joined it took the company about 2-4 weeks to assign a project to me and the fact that the team changed often, it was sometimes difficult to build relationships and trust with a colleague. Also, my aim was to work more in the backend of BI, however in the end I was doing frontend work, which honestly was not bad, but I was looking forward to deepening my knowledge about ETL and data modeling. Nonetheless, some tasks about data modeling were done. That being said, altogether I would asses this internship as a very rewarding and enriching experience.

5.2. CRITICAL APPRAISAL

In order to asses myself it makes sense to use the feedback I received from my teammate and team leader, namely the people I was working closely with. Both were there from the beginning except for a change in my teammate. All in all, both were quite satisfied with my performance. For a beginner, I was able to learn quickly and apply the knowledge acquired during trainings. Still my team leader noticed, due to the lack of technology skills I needed more time to adapt and understand. Nonetheless, I was glad that I was given the necessary time to learn and understand the technologies needed. In the begin the term Mashup was something I have never heard anything of and building a dashboard on the web was a difficult task. Once, I got used to the technologies I could quickly start developing dashboards on a web-based solution. My superiors saw my learning curve and I quickly got promoted to a Consultant. Giving me both more responsibilities and trust at the same time. I was joining more calls with the clients and providing my own ideas to deliver the best possible solutions. Moreover, following the promotion my manager noticed that I was getting too confident, which resulted in not following her advice as I was thinking my approaches were the better ones. I had to learn that that’s not always correct, my manager was forthcoming an understanding. When I was promoted, I also started working in different projects and sometimes the time management was an issue, because every dashboard is different even though the business is the same the contexts change. In summary, according to my manager and our client, they were satisfied with the work that was carried out and the reason for that was, that we managed to deliver the final solution on time and with all the requested features.

5.3. Future perspectives

Without a doubt I can say this internship has been a good experience. Quickly after I finished the internship I was offered to be hired as a Consultant, which means a promotion. Nonetheless, the business, namely Janssen offered me a job internally to be a Business Analyst for them as the job I performed as a BI Consultant was outstanding, according to them. Apart from that I also got an offer from Siemens, as a QlikSense in house Consultant. They have heard from me during an upcoming in German projects in Siemens and they need German speakers to communicate requirements and day to day ideas with the business.

Now, I didn't want to leave the company that taught me everything about BI and as I am not really confident enough to work in the pharmaceutical sector due to the fact I don't have the required business knowledge, I declined Janssens offer with the reasoning that I want to establish a solid BI knowledge. About Siemens, I told them that I cannot leave a company without being there for at least 1 year. So right after I finished a year at SDG and got promoted to Qualified Consultant it was time for me to move on and find a new challenge. The reason why I chose Siemens over SDG were two factors. First of all, I was not being challenged at SDG anymore despite having a team under me and teaching my knowledge to others. The development tasks started to get repetitive and I didn't feel like learning new things anymore. Siemens offered me new challenges as my job is to develop dashboards used by Siemens itself. Siemens is a large organization and their dashboards are being used by a large amount of people. Developing dashboards in that scalability will strengthen my knowledge about BI. Apart from that Siemens offers a lot of possibilities that can make benefit for my future career. That's why in the end I chose to work for Siemens and see where it takes me.

REFERENCES

Aginity. (2018, August). Retrieved from https://www.aginity.com/documentation/

Amazon. (2019). *Amazon Documentation*. Retrieved from https://docs.aws.amazon.com

Amazon. (2019). *AWS*. Retrieved from https://aws.amazon.com/de/redshift/

Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung. (2013, Januar). Informations- und Kommunikationstechnologien (IKT). Retrieved from https://www.bmz.de/de/mediathek/publikationen/archiv/reihen/strategiepapiere/Strategiepapier326_02_2013.pdf

Cleve, J., & Lämmel, U. (2016). *Data Mining.* Berlin: De Gruyter Oldenbourg Verlag.

Data Leader. (2017, 02). Retrieved from https://awskimschmidt.com/2017/02/28/traditional-data-warehouses-chapter-3-5-in-all-aws-data-analytics-services/

Fasel, D., & Meier, A. (2016). *Big Data - Grundlagen, Systeme und Nutzungspotenziale.* Wiesbaden: Springer Verlag.

Fuchß, T. (2009). *Mobile Computing. Grundlagen und Konzepte für mobile Anwendungen.* Munich: Carl Hanser Verlag.

Gansor, T. (2016). *Opitz Consulting*. Retrieved from http://www.opitz-consulting.com/ueber-uns/

Gartner. (2019). *Gartner*. Retrieved from https://www.gartner.com/it-glossary/self-service-business-intelligence

Gerken, W. (2018). *Data-Warehouse-Systeme für Dummies.* John Wiley & Son.

Gluchowski, P. (2001). *Business Intelligence: Konzepte, Technologien und Einsatzbereiche.*

Gluchwoski, P., & Chamoni, P. (2016). *Analytische Informationssysteme – Business Intelligence-Technologien und –Anwendungen.* Berlin: Springer Gabler.

Kemper, H. e. (2016). *Business Intelligence – Grundlagen und praktische Anwendungen.* Wiesbaden: Springer Vieweg.

Kroker, M. (2015). *Wirtschaftswoche*. Retrieved from http://blog.wiwo.de/look-at-it/2015/05/05/big-data-sorgt-schon-2016-fur-speicher-engpass-2020-fehlt-speicher-volumen-von-6-zetabytes/

Kroker, M. (2015). *Wirtschaftswoche*. Retrieved from http://blog.wiwo.de/look-at-it/2015/05/05/big-data-sorgt-schon-2016-fur-speicher-engpass-2020-fehlt-speicher-volumen-von-6-zetabytes/

Manhart, K. (2008). *Computerwoche*. Retrieved from https://www.tecchannel.de/a/bi-datenmanagement-teil-1-datenaufbereitung-durch-den-etl-prozess,1746250,3

Manyika, J., Chui, M., & Brown, B. (2011). *Big data – The next frontier for innovation, competition, and productivity.*

Mayer, J. H., & Quick, R. (2015). *Business Intelligence for New-Generation Managers.* Cham: Springer International Publishing.

Mertens, P. (2002). *Business Intelligence: Ein Ueberblick.* Nuernberg.

Müller, R. M., & Lenz, H. (2013). *Business Intelligence – Grundlagen.* Heidelberg: Springer Vieweg.

Patel, R. (2014). *Enterprise Mobility Strategy & Solutions.* Partridge Publishing.

Rouse, M. (2015). Retrieved from https://www.computerweekly.com/de/definition/Business-Intelligence-Dashboard

Rouse, M. (2016, December). *TechTarget*. Retrieved from https://searchbusinessanalytics.techtarget.com/definition/Qlik

Schön, D. (2016). *Planung und Reporting – Grundlagen, Business Intelligence, Mobile BI und Big-Data-Analytics.* Wiesbaden: Springer Gabler.

Sisense. (2019, Febuary). *sisense*. Retrieved from https://www.sisense.com/gartner-magic-quadrant-business-intelligence/

University of Agder. (2018). Data lakes in business intelligence: reporting from the trenches. Norway.