

Analytics infrastructure for aggregated data

Citation for published version (APA):

Chembay, V. A. (2017). *Analytics infrastructure for aggregated data*. Technische Universiteit Eindhoven.

Document status and date:

Published: 28/09/2017

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

/ Department of
Mathematics and
Computer Science
/ PDEng Software
Technology

Analytics infrastructure for aggregated data

Vera Chembay

Analytics infrastructure for aggregated data

Vera Chembay

Eindhoven University of Technology
Stan Ackermans Institute / Software Technology

Partners



Medworq

Eindhoven University of Technology

Steering Group

Johan Ruiter
Freek van Keulen
Nicola Zannone

Date

19 September 2017

Document Status

Public

PDEng report no.

2017/048

The design described in this report has been carried out in accordance with TU/e Code of Scientific Conduct.

Contact address	Eindhoven University of Technology Department of Mathematics and Computer Science MF 5.072 P.O. Box 513 NL-5600 MB Eindhoven, The Netherlands +31 402743908
Published by	Eindhoven University of Technology Stan Ackermans Institute
Printed by	Eindhoven University of Technology <i>UniversiteitsDrukkerij</i>
PDEng Report No.	2017/048
Abstract	<p>Medical data aggregated at a national level can bring a large number of insights to medical researchers and pharmaceutical companies. Research on the aggregated data will allow improving patient care, improving preventive healthcare, and measuring the effectiveness of medications and medical treatment. Medworq has developed a solution that aggregates medical data at a regional level from different healthcare sources. The goal of this project is to design and implement a solution that will aggregate healthcare data at a national level. Due to the fact that medical data is highly sensitive information, the main challenge of the data aggregation is to preserve data privacy of individuals. This report describes a project to analyze the data aggregation process, its challenges and use cases within Medworq, and to design a system that provides the data aggregation functionality. An inventory and comparative analysis of existing data privacy preserving techniques are provided in the report. The main deliverable of this project is a system that provides the data aggregation functionality together with a configurable choice of data privacy preserving techniques. The system was verified based on real patient data. The report also gives recommendations for future improvements.</p>
Keywords	data aggregation, data privacy, k-anonymity, l-diversity, t-closeness
Preferred reference	<u>Analytics infrastructure for aggregated data</u> . SAI Technical Report, September 2017. (2017/048)

Partnership	This project was supported by Eindhoven University of Technology and Medworq
Disclaimer Endorsement	Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Eindhoven University of Technology and Company name. The views and opinions of authors expressed herein do not necessarily state or reflect those of the Eindhoven University of Technology and Medworq, and shall not be used for advertising or product endorsement purposes.
Disclaimer Liability	While every effort will be made to ensure that the information contained within this report is accurate and up to date, Eindhoven University of Technology makes no warranty, representation or undertaking whether expressed or implied, nor does it assume any legal liability, whether direct or indirect, or responsibility for the accuracy, completeness, or usefulness of any information.
Trademarks	Product and company names mentioned herein may be trademarks and/or service marks of their respective owners. We use these names without any particular endorsement or with the intent to infringe the copyright of the respective owners.
Copyright	Copyright © 2017 Eindhoven University of Technology. All rights reserved. No part of the material protected by this copyright notice may be reproduced, modified, or redistributed in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the Eindhoven University of Technology and Medworq.

Foreword

Vera's project “Analytics infrastructure for aggregated data” has delivered a big data solution for collecting and aggregating medical data. This is all done within very strict privacy legislation. The aim of the project was to build an infrastructure to collect nationwide medical data, linking it on a patient level while still keeping the data anonymous and complying to national and European regulations.

The assignment posed several challenges, for example, how to anonymize the data without losing the patient level, how to collect large amounts of data and still getting reasonable performance and furthermore how to fit the system in already operational regional data warehouses. The regional systems working with a TTP for anonymization at the source, splitting the identifiers and medical data at the earliest possible moment.

On top of that, when a subset of the data is delivered to a researcher additional measures were taken to ensure the privacy of the patients. Because of the fact that in this use case the data is leaving the organization, you lose full control and extra safeguards have to be in place resulting in an analysis of the data leaving the organization on an array of privacy attacking strategies.

Vera addressed these challenges head on. She emerged herself into the various methods of anonymization, comparing and in the end implementing the most suitable ones for the project. Vera managed the project very well, stayed on top of surfacing issues, consulting the stakeholders and adjusting scope when needed.

It was a pleasure working with Vera on a professional level as well as on a personal level. All other team members learned a lot from her on anonymization and, for the company, other new technologies as Hadoop and Spark. We wish her the best for the future and we are sorry to see her go.

Freek van Keulen
September 2017

Preface

This report summarizes the “Analytics infrastructure for aggregated data” project carried out by Vera Chembay as the final part of the Professional Doctorate in Engineering (PDEng) program in Software Technology, provided by Eindhoven University of Technology, Stan Ackermans Institute. The project lasted for nine months and was conducted in Medworq.

The report describes the architecture, design, and implementation of the solution based on the problem, domain, and requirement analysis. All design decisions are given with the explanation of the rationale behind them. The report provides the description of the verification and validation and gives recommendations for future work.

Vera Chembay
September 2017

Acknowledgments

I would like to express my gratitude to my supervisor Freek van Keulen and manager Johan Ruiters for providing me the opportunity to work on my final project in Medworq. I would like to thank Freek van Keulen for the valuable guidance as well as the necessary domain knowledge you provided to me. I am thankful to Johan Ruiters for your enthusiasm and energy. You helped me to drive my project through all obstacles. I am grateful to Jogchem Dijkstra whose deep knowledge of the domain and technologies were invaluable for the project success. I really enjoyed working with you. It was a great to have discussions with such an intelligent, thoughtful, and broad-minded person as you are.

I owe the success of this project also to my university supervisor Nicola Zannone. Thank you for your guidance, interest, and continuous support during the project. Your feedback and remarks helped me to explore new ideas and directions of the project. I really appreciate your kindness and willingness to help.

I would like to thank the PDEng program management, Ad Aerts, Yanja Dajsuren, Mark van den Brand, and Desiree van Oorschot, for the opportunity to be part of the PDEng program and the guidance during past two years.

I sincere gratitude to my mother for her continuous support and faith in me. I would like to thank my housemates. Masha, Misha, and Dima, thank you for your support and great moments we had together.

Executive Summary

Nowadays all medical organizations collect a large amount of patient data in their information systems. Data related to one particular patient can be spread between different healthcare centers, pharmacies, and hospitals. The process of bringing all this data together at one place is called data aggregation process. Research based on the aggregated data will allow recognizing patterns and trends in the healthcare domain, improving patient care and preventive healthcare, and measuring the effectiveness of medications and medical treatment.

Medworq has developed a solution, called “Population Database,” which aggregates medical data at a regional level from different data sources such as general practitioners, pharmacies, and hospitals. The “Population Database” allows conducting research on the regional level and creating aggregated reports by required medical topic.

However, medical data related to a patient can be located in different regions. In this case, conducting research at a regional level provides a limited overview. In order to create more complete input for medical research, data should be aggregated at a national level.

Aggregating data at a national level brings several challenges:

1. There is large amount of duplicate information.
2. Records related to one particular patient have different regional identifiers.
3. Data privacy regulations require the data aggregated at a national level to be anonymous.

The goal of this project is to create a system that is able to aggregate medical data at a national level from existing “Population Databases.” The solution is called “Cooperative Database.” Currently, there is no solution for the aggregated data at a national level in the Netherlands. Therefore, our system is innovational and will bring added value to the different stakeholders such as pharmaceutical companies, research institutes, and the Dutch society.

The system is delivered as a Java Enterprise Application. The “Population Database” uses Java programming language as well. Therefore, it will be easy to integrate the “Cooperative Database” and the “Population Database” solutions in the future. This integration was designed in the scope of this project. However, it was not implemented due to the limited amount of time.

Below we describe how the system solves the mentioned challenges:

1. The system handles duplicate information and removes redundant data.
2. The system links all records related to one particular patient to one common identifier. This linking process is performed through the Trusted Third Party (TTP) which will generate new identifiers for our system. The communication between “Cooperative Database” and the TTP was designed. However, because of the functionality from the TTP was not implemented by the end of the project, our solution used a synthetically generated response to link records.
3. The domain and problem analysis provide a research and comparison of existing data privacy preserving techniques. The most appropriate for our solution data privacy preserving techniques were chosen and implemented in the “Cooperative Database.”

The system was verified based on data exported from two regional Population Databases. To show the added value of the system, data for medical research on Chronic kidney disease was exported from the system. Before delivering research data to scientists, data was anonymized.

It is recommended that the system be further tested using real data from several regions in order to verify the results, and, if necessary, modify the design and implementation accordingly.

Glossary

AGB code	National code for a health-care provider. This unique code for each provider which is registered in a national database.
AIS	Apotheek informatie systeem (Pharmacy Information System)
Anonymous data	Data which cannot be related back to an individual patient
API	Application programming interface
BSN	Citizen Service Number. BSN is a unique personal number allocated to everyone registered in the Municipal Personal Records Database in the Netherlands.
CDS	Cooperative Database System
Cooperative Pseudonym	Pseudonym which is linking all medical records related one individual patient over all regions. It should be regenerated each time during generation new dataset for Cooperative Database
CRUD	Create, Read, Update, and Delete
CSR	Certificate Signing Request
Data aggregation	process of collecting data from several data sources or regions with intent to expresse data in a summary form for statistical analysis later on
Data source	Source of medical data (such as general practitioner, hospital or pharmacy)
ETL	Extract, transform, load
JAR	Java ARchive
IDE	Integrated development environment
HDFS	Hadoop Distributed File System. Data storage of Hadoop infrastructure
HIS	Huisarts informatie systeem (General Practitioner Information System)
Metadata	Description of Semantically Integrated Model, structure of source data, and mapping between these two
POJO	Plain old Java object
Population Database	Database which stores medical data collected at a regional level
RBAC	Role-based access control is an approach to restricting system access to authorized users
SIM	Semantically Integrated Model. This is a semantically integrated superset of all the different domains (AIS, HIS, ZIS)
SQL	Structured Query Language
TTP	Trusted Third Party. TTP provides pseudonymization process
ZIS	Ziekenhuis informatie systeem (Hospital Information System)

Contents

Foreword	ii
Preface	iv
Acknowledgments	vi
Executive Summary	viii
Glossary	x
List of tables	xvi
List of figures	xviii
1 Introduction	1
1.1 Project context	1
1.2 Problem description	2
1.3 Outline	2
2 Stakeholder Analysis	5
2.1 Medworq	5
2.2 Privacy Company	6
2.3 Eindhoven University of Technology	6
2.4 Other stakeholders	7
2.4.1 Research Institutes	7
2.4.2 General Practitioners and medical specialists	7
2.4.3 Policy makers	8
2.4.4 Dutch society	8
2.5 Stakeholder prioritization map	8
xii Analytics infrastructure for aggregated data	

3	Problem Analysis	11
3.1	Context	11
3.2	Use Cases	13
3.2.1	User Stories	13
3.2.2	Technical Stories	14
3.3	Data privacy	15
3.4	Criteria for the solution	17
4	Domain Analysis	19
4.1	Current Infrastructure	19
4.2	Interaction with other projects	20
4.3	Data aggregation process	21
4.3.1	Duplicated data	21
4.3.2	Linking records related to particular patient	21
4.4	Literature review of anonymization techniques	22
4.4.1	Randomization	22
4.4.2	Generalization	23
4.4.3	Comparative analysis	26
5	System Requirements	27
5.1	Functional Requirements	27
5.2	Non-functional Requirements	29
6	System Architecture	31
6.1	Architecture	31
7	Design and Technology choices	33
7.1	Data storage	33
7.1.1	Hadoop	33
7.1.2	RIAK	34
7.1.3	Data storage decision	35
7.2	Data processing frameworks	36
7.2.1	Hadoop MapReduce	36
7.2.2	Spark	37
7.2.3	Data processing framework decision	37
7.3	SQL support frameworks	38

7.3.1	Hive	38
7.3.2	Pig	38
7.3.3	HBase	39
7.3.4	SQL framework decision	39
8	Feasibility Analysis	41
8.1	Issues and Challenges	41
8.2	Risks	41
9	Implementation and Deployment	43
9.1	Development Environment	43
9.2	Deployment Environment	46
10	Verification and Validation	47
10.1	Verification	47
10.1.1	Unit testing	47
10.1.2	Integration testing	49
10.2	Validation	49
11	Conclusion	53
11.1	Result	53
12	Project Management	55
12.1	Introduction	55
12.2	Work Breakdown Structure (WBS)	56
12.3	Project Planning	56
13	Project Retrospective	59
13.1	Design opportunities revisited	59
	About the Author	63

List of Tables

2.1	Medworq company stakeholder analysis	6
2.2	Privacy company stakeholder analysis	6
2.3	TU/e stakeholder analysis	7
3.1	Cooperative Database Actors	13
7.1	Data storage comparison	36
7.2	Data processing framework comparison	38
7.3	SQL support framework comparison	39
8.1	Risks and mitigation strategies	42
10.1	Integration test data input characteristics	49
10.2	System requirements coverage	50

List of Figures

1.1	Osteoporosis risk dashboard	1
2.1	Stakeholder prioritization map	8
3.1	High level overview of the project goal	12
3.2	Data access process	12
3.3	Use case diagram	14
3.4	Re-identification by data linking (adapted from [1])	17
4.1	Data collection process from different sources to Population Database	20
4.2	Inpatient Microdata (adapted from [1])	24
4.3	4-anonymous Inpatient Microdata (adapted from [1])	25
4.4	3-Diverse Inpatient Microdata (adapted from [1])	25
4.5	Anonymization techniques comparative analysis	26
6.1	System architecture	32
9.1	Development environment of the Cooperative Database in IntelliJ IDEA	44
9.2	Application build output	45
10.1	Unit tests coverage	48
12.1	Work-breakdown structure of the project	57
12.2	The project planning	57

1 Introduction

1.1 Project context

Medworq is an organization that offers solutions in the healthcare domain. In particular, Medworq has developed different EHealth solutions, such as a tool for online communication between patients and caregivers, which allows quick, effective, and efficient communication. Improvement of preventive medical care is another direction of the company interests. Preventive medical care allows finding patients at risk in an early stage to ensure that proper care is provided.

Preventive medical care requires medical data that should be analyzed. In different medical organizations, all kinds of data are available: about individual patients, groups of patients, diagnosis and outcomes or effects of treatment. Aggregating all this data together and analyzing it can bring valuable conclusions for healthcare. One example of the outcomes of the aggregated data is a risk dashboard developed by Medworq. Figure 1.1 shows the risk dashboard, which allows early detection of patients at risk of osteoporosis. Using an action-oriented dashboard, the General Practitioners can track risk groups of patients. Therefore, patients at risk can receive proper medical treatment earlier, in order to prevent more serious consequences for their health.

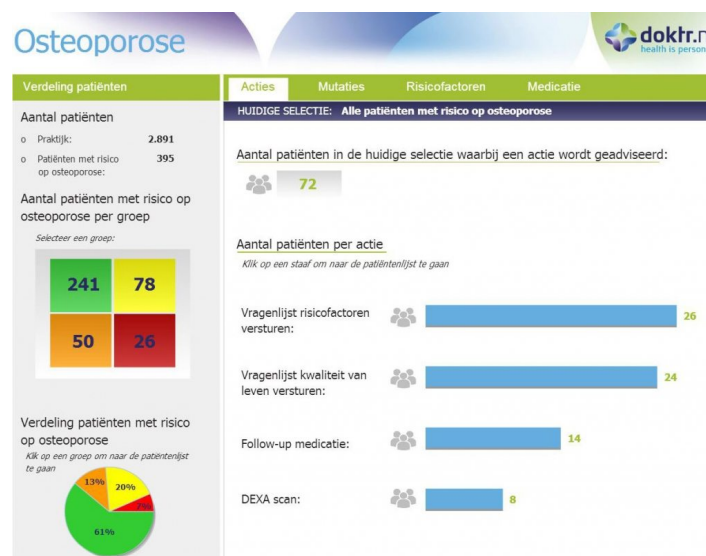


Figure 1.1: Osteoporosis risk dashboard

Medical data is often distributed across different medical organizations, stored in different systems in a variety of formats. The process of bringing all this data together at one place is called data aggregation process. The larger the amount of medical data for research we have, the more precise the conclusions we can make out of it. Therefore, in order to bring even more added value from different medical data, this data should be collected and aggregated from all available medical data sources.

Aggregated data will allow:

- Improving patient care
- Improving preventive healthcare
- Measuring effectiveness of medications and medical treatment
- Processing regional benchmarking

1.2 Problem description

Medworq has developed a solution called “Population Database,” which aggregates data at a regional level from different healthcare sources such as general practitioners, pharmacies, and hospitals. The “Population Database” allows conducting research on the regional level and creating aggregated reports by required medical topic. However, medical data related to a patient can be located in different regions. Thus, conducting research at a regional level provides a limited overview. In order to create more complete input for medical research, data should be aggregated at a national level, i.e. from all Population Databases to one place.

The data collection and aggregation process at a national level brings several challenges:

- Collected data might contain large amount of duplicate information.
- Records related to one particular patient have different identifiers, because data is collected from different systems.
- Data privacy regulations require the data aggregated at a national level to be anonymous.

The goal of this project is to design and implement Cooperative Database System (CDS), which will aggregate healthcare data at a national level. This solution is technically possible, but was not implemented so far in the Netherlands. The design and implementation of the CDS should address all mentioned challenges. Moreover, the CDS should be integrated with Population Database solution later on. Therefore, the design for this integration is in the scope of this project as well.

1.3 Outline

The remainder of the report is structured as follows:

Chapter 2 identifies and describes the main stakeholders of the project and explains their interests, needs, and influence in the project.

2 Analytics infrastructure for aggregated data

Chapter 3 discusses in more detail the problem that the project is focusing on and design criteria for the solution.

Chapter 4 gives the analysis of the domain in which the project is conducted.

Chapter 5 presents the system requirements that were defined from the domain and problem analysis as well as from the interviews with stakeholders.

Chapter 6 provides the system architecture.

Chapter 7 lists the main design and technology choices that were made during the design phase of the project.

Chapter 8 identifies and discusses possible issues and risks.

Chapter 9 describes the implementation and the deployment process.

Chapter 10 presents the validation and verification process.

Chapter 11 formulates the results and provides the description of the remaining work.

Chapter 12 describes the management part of the project.

Chapter 13 gives the retrospective of the project from the author's perspective.

2 Stakeholder Analysis

A project stakeholder is an individual, a group or an organization, who may affect or be affected by a decision, an activity or an outcome of the project. Stakeholder analysis is the process of gathering and analyzing information to determine whose interests should be taken into account during design and implementation of a solution. In this chapter, we describe main stakeholder groups and their interests for this project.

2.1 Medworq

Medworq is a company in the healthcare sector. Medworq develops solutions in eHealth (a solution for communication between doctor and patient) and preventive healthcare (recognizing patients with disease risk). The main interest of the Medworq is the design and implementation of an aggregation infrastructure. Medworq acts as a facilitator of meetings/feedback sessions with important stakeholders, such as pharmaceutical company representatives and Privacy Company (see Section 2.2).

Table 2.1 describes each stakeholder at Medworq in more detail:

Name	Role	Interests
Johan Ruiters	Project Manager	<ul style="list-style-type: none"> • The solution should be compliant with regulations. • The solution should present viable business case. • The solution should be able to operate with other systems such as new data providers and systems which are going to use the data. • The system should be based on market and technology standards. • Medworq should be able to track project progress according to schedule

Freek van Keulen	Company Supervisor	<ul style="list-style-type: none"> • The solution should not interfere with data provider interests. • The solution should use a data-driven approach. • The solution should reuse as many as possible components from current infrastructure.
------------------	--------------------	---

Table 2.1: Medworq company stakeholder analysis

2.2 Privacy Company

Due to the fact that the Cooperative Database processes a large amount of patient medical data, data privacy is a major requirement for the project. In particular, the Cooperative Database solution shall comply with the privacy and data protection regulations in the Netherlands. Privacy Company provides advice on how the privacy requirements should be addressed in the design and implementation of the proposed solution. The main interest of the Privacy Company is to ensure that solution meets all legal requirements from data privacy and security prospective.

Name	Role	Interests
Arnold Roosendaal	Legal expert	<ul style="list-style-type: none"> • Ensure that solution is designed in such a way that it ensures proper level of data privacy and security

Table 2.2: Privacy company stakeholder analysis

2.3 Eindhoven University of Technology

TU/e (Eindhoven University of Technology) offers PDEng (Professional Doctorate in Engineering) programs in Software Technology. As a graduation project, the university provides to a trainee a design project in the industry field. The main interest of the university is to ensure a good quality of design and report accomplished by the trainee and timely project deliverables to the company. Table 2.3 describes each stakeholder of the University in more details:

Name	Role	Interests
Vera Chembay	PDEng trainee	<ul style="list-style-type: none"> • Gain software design experience • Apply design and professional skills • Complete project scope on time
Ad Aerts	Software Technology PDEng program director	<ul style="list-style-type: none"> • Quality of project and process • Continuation of cooperation with industrial companies
Nicola Zannone	TU/e supervisor	<ul style="list-style-type: none"> • Ensure that the project scope meets the level of a PDEng project • Validate that design, implementation, and final report meet the standards of a PDEng project

Table 2.3: TU/e stakeholder analysis

2.4 Other stakeholders

There are some other stakeholders, who do not have direct influence to the project, but have an interest in the final result.

2.4.1 Research Institutes

The main interest of the Research Institutes is to carry out high quality research that has a demonstrable impact. The outcome of this research supports improvement of health and social care for patients and healthcare system sustainability as a whole. With the Cooperative Database solution Research Institutes get access to the data aggregated at a national level, which gives a wide area for research directions.

2.4.2 General Practitioners and medical specialists

The General Practitioners and other medical specialists should be a source of medical knowledge, but also takes into account changes and new methods in healthcare. General Practitioners who want to improve quality and efficiency of healthcare services can find help in the research results. Based on a nation-wide statistics, General Practitioners can encourage patients to follow healthcare advice.

2.4.3 Policy makers

The World Health Organization has recognized the need to use results from the best available research evidence in healthcare recommendations (see e.g. [2]). Therefore, policy makers can promote effective strategies how to bring these results into healthcare. With the Cooperative Database solution, policy makers can have an overview of medical trends at a national scale.

2.4.4 Dutch society

Health research has a high value to society. It can provide important findings about disease and risk factors, results of treatment, healthcare approaches, and healthcare costs. The Dutch society can benefit from healthcare improvements based on these findings.

2.5 Stakeholder prioritization map

Figure 2.1 illustrates the stakeholder prioritization map where all stakeholders are classified by their influence and interest. The stakeholder prioritization map is very important to identify key stakeholders at the beginning of the project. In addition, it is possible to identify stakeholders that have less influence on the project but have interest in the final solution and therefore can support the project.

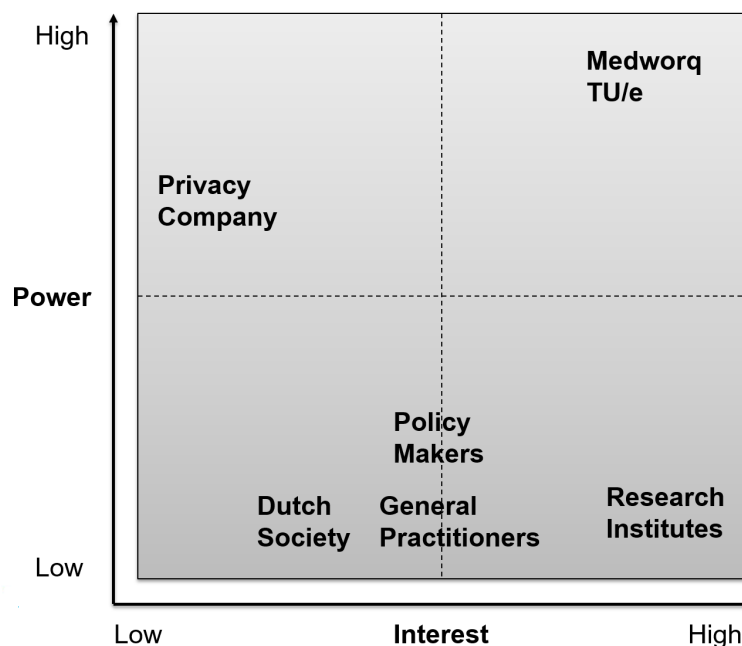


Figure 2.1: Stakeholder prioritization map

The prioritization map also defines how much a stakeholder needs to be aware of project status and progress. Different priorities of groups affect the style of communication. Below we describe different communication scenarios for each prioritization group:

- **High interest/High power**

It is very important to keep up to date stakeholders who have high interest and high power in the project. Because they can easily influence the direction of the project so they need to have the latest information. Therefore, the following scenarios were chosen for the communication with this group:

- Face-to-face meetings
- Regular updates by email or technical document
- Demonstrations of current progress

- **High interest/Low power**

We can use knowledge and expertise from stakeholders who have high interest and low power in the project. They can bring added value to the project by sharing their domain knowledge. Therefore, the following scenarios were chosen for the communication with this group:

- Consultation by face-to-face meeting or email
- Updates about major decisions by email

- **Low interest/Low power**

For stakeholders who have low interest and low power in the project we do not have any specific updates. They will obtain their added value from the project after the Cooperative Database solution will be completed. Currently we do not take into account input from them.

3 Problem Analysis

The analysis of the company context and stakeholder interests allows formulating the problem statement that the project aims to solve. The high level goal of the project is to create framework for aggregated data. In order to understand and analyze the problem deeper, this chapter describes use case analysis, data privacy challenges, criteria for the solution, and the project scope.

3.1 Context

Aggregating data from different health organizations such as general practitioners, pharmacies, and hospitals across several regions make it possible to measure effectiveness of medication and medical treatment, create disease classifications, and improve preventive medicine. As mentioned in Chapter 2 different stakeholders will benefit from this solution: pharmaceutical companies, research institutes, policy makers, general practitioners and the Dutch society as a whole.

Currently, a solution for aggregated data at a regional level has already been implemented by Medworq. This solution is called Population Database. Population Database collects medical data from different medical sources such as general practitioners, pharmacies, and hospitals and aggregates this data by different regions. Medical data related to one particular patient can be spread between different medical organizations. Moreover, these organizations can be located in different regions. In this case, in one region dataset related to one patient could be incomplete. Research based on incomplete datasets could lead to wrong conclusions. In order to avoid this situation, there is a need to aggregate data at a national level. Aggregated data at a national level will bring added value to the pharmaceutical companies as well. They can measure the effectiveness of medication and perform benchmarking at a national scale.

The Population Database stores data collected from different healthcare sources such as general practitioners, pharmacies, and hospitals. The goal of the project is to design and develop a system for aggregating data from a regional level (from the available Population Databases) to a national level. The solution is called Cooperative Database. Figure 3.1 shows a high level view how data is aggregated from different Population Databases to the Cooperative Database. The Cooperative Database will provide the ability to conduct research and generate aggregated medical reports at a national level.

Although a solution for aggregated data at a national level is technically possible, it has not been implemented so far. The design and implementation of such a solution involves a number of legal concerns, which is described in Section 3.3 and Chapter 4. One of these concerns is to preserve

data privacy of individuals while aggregating medical data at a national level. Therefore, the data access process should be defined to address this concern. Figure 3.2 shows an overview of the data access process to the business, research institutes, and policy makers. Data is delivered in two forms: aggregated reports and data subsets of the Cooperative Database dataset. According to the data privacy regulations, aggregated data shall preserve privacy of individuals, i.e. data delivered to a third-party company should be anonymous. Therefore, before delivering a dataset to a third-party company the Cooperative Database performs an anonymization check. In case of aggregated reports data does not reveal any personal identifiable information because it is presented in generalized statistics.

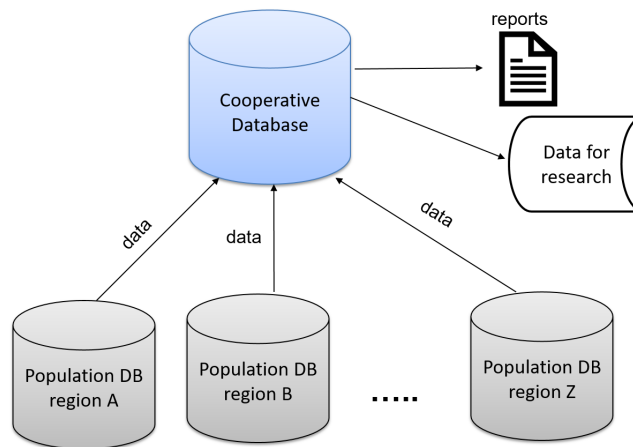


Figure 3.1: High level overview of the project goal

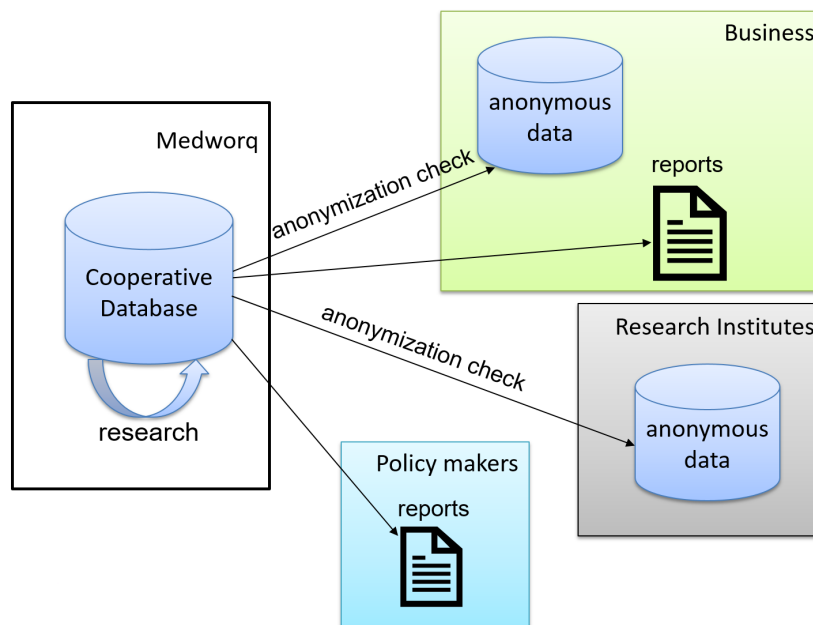


Figure 3.2: Data access process

3.2 Use Cases

In order to understand the scope of the Cooperative Database it is important to analyze how the system will be used by the relevant stakeholders. In the following sections we describe different use cases that have been collected by interviewing different stakeholders. Table 3.1 provides the list of actors which use the system.

Actor	Role, Action or Responsibility
Operator	<ul style="list-style-type: none"> Controlling of data upload process Processing data through the system
Data Manager (Analyst)	<ul style="list-style-type: none"> Detecting and deleting inconsistencies in datasets Creating new classification based on domain knowledge Exporting data for research
Application Manager	<ul style="list-style-type: none"> Maintaining user accounts Processing of an authorization
System Administrator	<ul style="list-style-type: none"> Maintain data server availability
Privacy Officer	<ul style="list-style-type: none"> Checking how data processing complies with regulations. In particular, configure anonymization methods and parameters
End User	<ul style="list-style-type: none"> Retrieve national statistics Carry out research at a national level

Table 3.1: Cooperative Database Actors

3.2.1 User Stories

Figure 3.3 shows a use case diagram representing user stories that involve an End User. The system has one general End User actor, which can be specified as a pharmaceutical company or third-party researcher.

Below we list user stories. Each user story provides information about which action the user can perform and which added value this action brings to the user.

1. As a **User (pharmaceutical company)**, I want to be able to get national statistics, so I can measure the effectiveness of medicine or treatment.
2. As a **User (pharmaceutical company)**, I want to be able to compare different statistics, so I can compare results based on different treatments.
3. As a **User (third-party researcher)**, I want to have access to exported data, so I can carry out research.

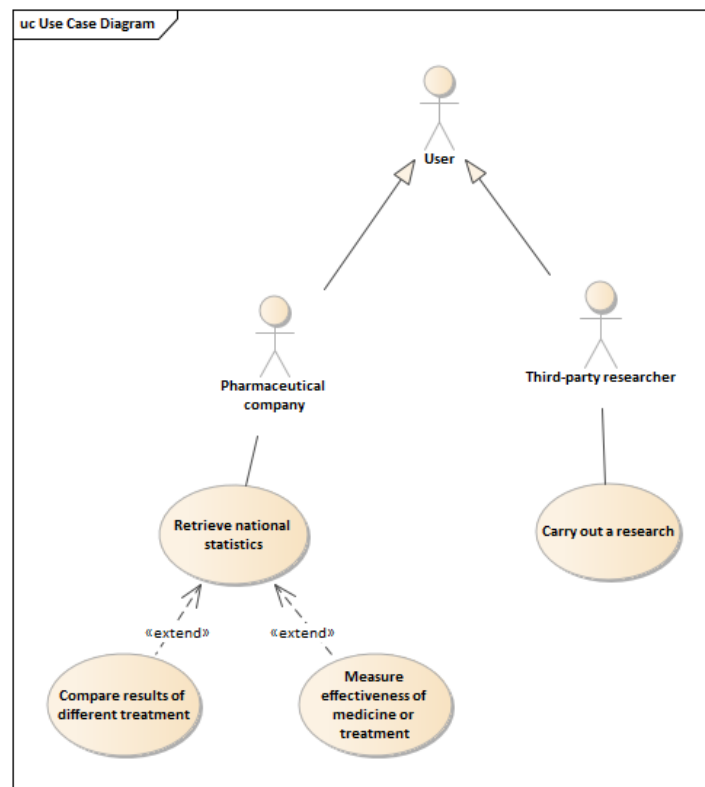


Figure 3.3: Use case diagram

3.2.2 Technical Stories

In order to accomplish the User Stories mentioned in the previous section we need to define several Technical Stories. These Technical Stories give context to the User Stories from a system perspective. Below we describe the actors that are involved in the technical stories, the importance of their role and actions, and arising challenges.

Operator The Operator processes data through the system, i.e., he starts procedures that get data from the Population Databases, anonymizes the data, and uploads it to the Cooperative Database server. He is in control of the full data upload process. Then, he should react to all errors and inconsistencies in the process, restart procedures, or consult with more knowledgeable IT specialists. In order to provide up to date medical information for the stakeholders, it was agreed that the Cooperative Database should collect and aggregate data monthly. Therefore, the success of data aggregation process depends on timely actions of the Operator.

Data Manager The Data Manager is a person who has knowledge about medical domain and database model. Therefore, the Data Manager capable of:

- detecting and deleting inconsistencies in datasets;

- identifying logic for classification based on existing data and domain knowledge. He can update a dataset with new data based on this classification;
- exporting data for third-party research and creates unique pseudonym for each research purpose.

Considering that the Data Manager makes data from the Cooperative Database more precise and meaningful, he brings the added value to all project stakeholders.

Application Manager The Application Manager supports user authentication and authorization process. In order to provide a higher security level of application, the Cooperative Database allows creating different user groups with different data access levels. The Application Manager is in charge of:

- creating and updating user accounts on the server;
- creating different user roles;
- giving permissions/rights to the users to perform different operations.

Due to the fact that data in the Cooperative Database contains a large amount of sensitive medical information, the system additionally protects data through role-based access control (RBAC) [3]. This approach gives to each user group minimal required access to the data. Thus, the Application Manager plays a very important role for the project.

System Administrator The System Administrator starts the data server and tracks the server availability. He should react in a timely manner on unexpected server stop or inaccessibility, which can be harmful for stakeholder's needs.

Privacy Officer The Privacy Officer verifies how data processing complies with European (see [4]) and Netherlands (see [5]) data privacy regulations. Therefore, the Privacy Officer decides which anonymization methods with which parameters the system uses in order to deliver fully anonymous datasets. Because only the Privacy Officer has specific knowledge about both legal and technical sides of the data delivery, his role is indispensable in this process.

3.3 Data privacy

Health research provides valuable benefits to society by improving healthcare. This research also creates added values for different healthcare related organizations as well. However, medical data is sensitive information. Thus, it is very important to preserve the privacy of individuals involved in research.

When data is collected at a regional level, the privacy and data regulations in the Netherlands allow storing this data in one place even though it can contain some re-identifiable information. This situation is allowed because of the connection of care between healthcare organizations and patients within a region. The situation changes when we aggregate data at a national level. It is required by law, that

data aggregated at a national level should not reveal any identifiable personal information. In order to satisfy this requirement two approaches for preserving privacy exist [1]: pseudonymization and anonymization.

Pseudonymization consists of replacing identifiable attributes (usually a unique attribute such as BSN) in a record by another attribute. However, pseudonymization does not provide a high level of data protection. Due to the fact that an individual still has a unique identifier (pseudonymised value) it is still possible to single out individuals' from a dataset and link records related to one individual together.

Data anonymization is defined in [1] as “a technique applied to personal data in order to achieve irreversible deidentification.” Effective anonymization solutions should prevent singling out an individual in a dataset, linking two records between two separate datasets, and inferring any information from the dataset. Therefore, only removing directly identifying elements from a dataset is not enough to ensure that the identification of the data subject is no longer possible.

However, data anonymization techniques can have some drawbacks as well. One of them is re-identification. Data re-identification is the practice of matching de-identified data with other available information in order to find out the individual which the data belongs to. Green et al.[6] define re-identification as “the ability to learn information about individuals that would not otherwise be known. In many cases this new information can lead to a variety of harms for the re-identified individuals.” This is a concern related to all organizations that publish data contained sensitive information. Below we consider several particular cases where individuals were re-identified from an originally supposed anonymous dataset.

Social Networks It was shown in [7] that private information about individuals can be extracted from a social network. A provider of a social network used nicknames as a pseudonymised attributes for publishing data, which was clearly not enough for protecting data privacy.

Locations Montjoye, Hidalgo, Verleysen and Blondel [8] analyzed a pseudonymized dataset containing spatial-temporal mobility coordinates of 1,5 million people spread over a territory within a radius of 100 km. They showed that 95% of people could be singled-out with four location points, and more 50% could be singled-out with two location points, which are most likely “home” and “office” locations.

Individual re-identifications It was shown by Sweeney [9] that 87% of the population in the United States can be uniquely identified based on ZIP code, gender, and date of birth information. Therefore, a released dataset contained such information should not be considered as anonymous.

Sensitive information disclosure Another re-identification example was given in [6] which consisted of the fact that in 2010 Latanya Sweeney merged a dataset released by the Massachusetts Group Insurance Commission with publicly available voting registration records. By doing so, it was possible to identify the individual referenced in many health records that were released under the assumption of anonymity. Particularly, the health records of the Massachusetts governor were disclosed. Figure 3.4 shows how data was merged from two datasets.

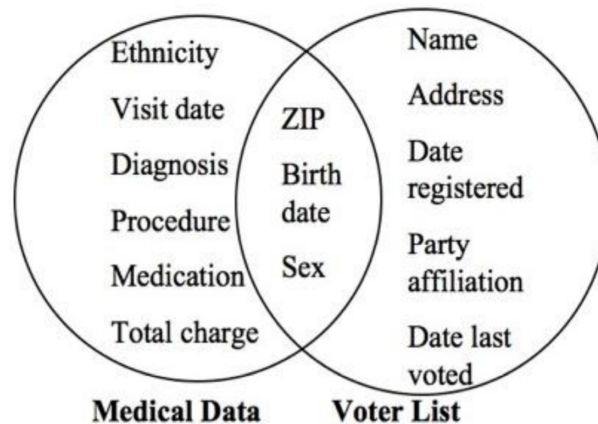


Figure 3.4: Re-identification by data linking (adapted from [1])

All these examples show the importance of choosing the proper anonymization technique for each particular case. There are two families of anonymization techniques: data randomization and generalization. Both approaches with their advantages, disadvantages, and proper use cases are considered in Chapter 4.

3.4 Criteria for the solution

Based on the problem statement described in the current chapter, the design criteria for the project can be identified. Using criteria classification from [10], three design criteria were defined as the important ones for this project:

Functionality. Ease of use The final product should be easy to use by Medworq and end users (e.g. researchers, pharmaceutical companies). It should be easy to export data for further research. Stakeholders who are going to conduct research from the Cooperative Database datasets currently use either SQL for querying data or work with CSV data format. Therefore, the Cooperative Database should provide the similar querying interfaces for them.

Maintainability It should be easy to maintain and extend the functionality of the Cooperative Database by Medworq in the future. Therefore, the design should be modular, which makes the application easier to test, maintain, detect errors, and extend.

Security Security and privacy are very important aspects in the project, because the datasets have large amounts of sensitive medical information that shall not be disclosed. Thus, these aspects should be addressed in the design and implementation phase of the project.

4 Domain Analysis

The goal of this chapter is to provide a better understanding of the domain in which the project is conducted. While the previous chapter has briefly introduced the domain, this chapter discusses it in detail. The chapter describes the current infrastructure in which the project should be integrated and other company projects that can influence the Cooperative Database solution. It also discusses the challenges connected with the data aggregation process. At the end, the chapter provides anonymization technique overview and a comparative analysis of different techniques.

4.1 Current Infrastructure

Medworq has developed a solution called "Population Database," which aggregates data from different datasources (general practitioners, pharmacies, and hospitals) at a regional level. Figure 4.1 shows an overview of the data aggregation process. Initially, data is collected from different information systems such as general practitioners, pharmacies, or hospitals. Further, data is pseudonymized through Trusted Third Party (TTP) in order to remove all personal identifiable information except of four digits of zip code, birth year, and gender. This information remains as necessary statistical information for future research. After pseudonymization all data is standardized to a generic model and stored into the Population Database. Thus, all data from different domains: general practitioners, pharmacies, and hospitals is mapped to a one common model inside the Population Database. This common model is called Semantically Integrated Model (SIM) and it is described in more detail in Chapter 6.

The Population Database provides the ability to conduct medical research and create reports at a regional level. The data in Population Database can be non-anonymous, because data privacy regulations allow storing non-anonymous datasets at a regional level. In the Cooperative Database case, when all data comes to a national level, this situation is not allowed anymore. Therefore, data stored in the Cooperative Database should be anonymous.

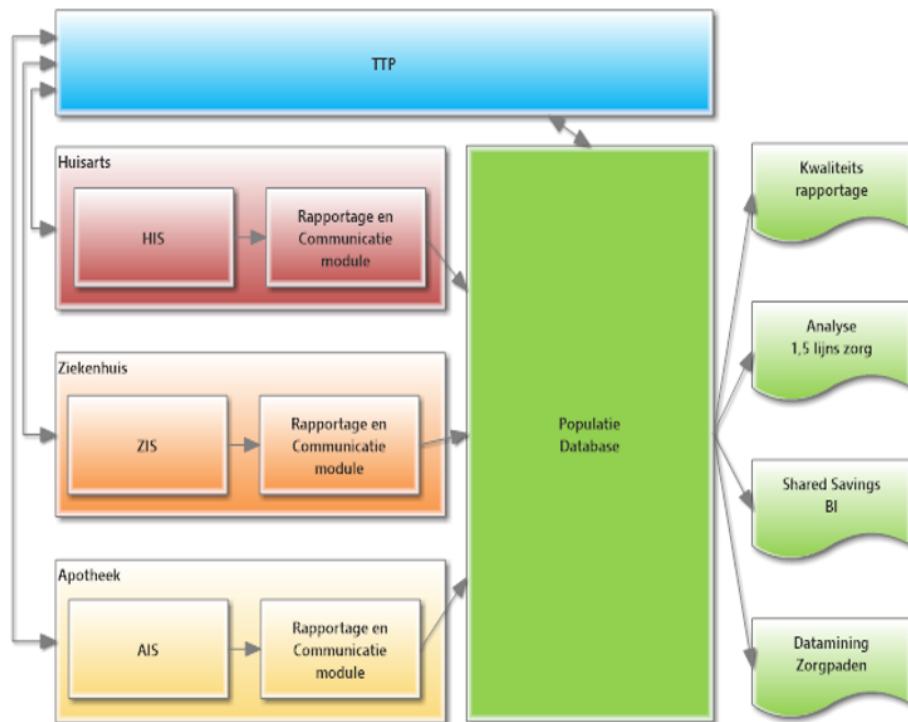


Figure 4.1: Data collection process from different sources to Population Database

4.2 Interaction with other projects

The Cooperative Database has a connection with several other projects:

1. Opt-in/Opt-out

Opt-in policy means that a patient must clearly choose to have his/her data in the dataset for any specific care provider, organization, or data purpose (e.g., research), which can be changed at any specific time. On the other hand, Opt-out policy means that a patient has the ability to forbid adding his/her data in any dataset. The goal of Opt-in/Opt-out project is to define use cases for Opt-in and Opt-out data collection approaches. This project should also check how proposed use cases satisfy data privacy regulations.

2. Authorization

The goal of the Authorization project is to define a role-based access control (RBAC) on the data, which contains sensitive information.

Specifying dynamic access rights to the information is based on three aspects:

- Type of information (for example, information about specific disease)
- Source of information (region)
- Identity of requestor and purpose of data

3. Identifying and developing data model for the Regional Population Database

The goal of this project is to design SIM so that it will cost minimum effort to connect a new

data source (general practitioner, pharmacy, hospital, or other medical data provider) to the system.

Currently, Medworq supports Opt-out approach for the Population Database. Therefore, the Cooperative Database should be designed only for **Opt-out** approach as well. The opt-out Cooperative Database is aiming to collect data for research purposes, national benchmarking and deliver anonymized datasets to the industry.

Although authorization is an important part of the security process in the access to the Cooperative Database, it will be designed separately from the Cooperative Database solution and integrated in a later moment. The design will be accomplished in the scope of “Authorization” project.

SIM is a data model from regional Population Database. The Cooperative Database uses the SIM as a data model as well. The Cooperative Database takes the data description (tables, their fields and purposes) from the SIM. Therefore, changes in the SIM can influence our solution.

4.3 Data aggregation process

In this section we list several challenges that appear during the data aggregation process.

4.3.1 Duplicated data

When data is collected from several regions, it may contain duplicate information. Pharmacies have a mutual agreement between each other to share patient data. Thus, each patient has his data stored at several pharmacies. Later on, when all this data aggregated at the Cooperative Database, we have duplicated records. This situation can also occur when patients visit several medical organizations with the same disease episode. In this case, different medical organizations can have records related to the same patient with the same disease episode, which later on leads to the duplicate records in the Cooperative Database as well. Duplicate information can lead to the wrong statistics obtained from the Cooperative Database. In order to receive truthful results from the research using aggregated data, our system should be able to detect and delete duplicates after data is collected from different regions.

4.3.2 Linking records related to particular patient

The Population Database does not contain a personal identifiable piece of information about patients. Instead, it requests pseudonyms from the Trusted Third Party (TTP) and further uses these pseudonyms as patient identifiers. For each region the TTP returns a different set of pseudonyms, which means that if the same person has his data in two or more regions his records will have different identifiers in the collected dataset. In this case researcher will treat records related to the same patient as records related to different patients. This situation can bring wrong outcomes from medical research. Thus, a process for linking records related to a particular patient should be designed and implemented within our solution.

4.4 Literature review of anonymization techniques

As discussed in Section 3.3 privacy protection is a fundamental enabler for the adoption of technical solutions in the project. In this section we consider effectiveness and limits of existing anonymization techniques and determine which of technique is more suitable to our problem.

The major challenge in the anonymization process is the balance between data utility and the risks of re-identification of an individual. In other words, the goal is to keep the data useful for research as much as possible whilst lowering the risks of revealing the individual to whom the data refers. However, several research studies [11], [12], [7],[9] have shown how difficult it is to create a fully anonymous dataset whilst saving as much of the needed information for the required task.

Randomization and generalization are two families of anonymization techniques. Both of them are divided in several particular techniques. Noise addition, permutation, and differential privacy belong to the randomization family and k-anonymity, l-diversity and t-closeness belong to the generalization. We consider each of these techniques, their strengths and weaknesses, and possible failures.

The data protection working party [1] suggests three criteria to assess the robustness of anonymization techniques:

1. *Singling out*, which corresponds to the possibility to isolate some or all records which identify an individual in the dataset;
2. *Linkability*, which is the ability to link at least two records concerning the same data subject or a group of data subjects (either in the same database or in two different databases);
3. *Inference*, which is the possibility to deduce, with significant probability, the value of an attribute from the values of a set of other attributes.

Knowing the main strengths and weaknesses of each technique allows choosing the proper design of the anonymization process of our problem.

4.4.1 Randomization

Randomization is a family of techniques that alters the veracity of the data in order to remove the strong link between the data and the individual [1]. Randomization by itself does not provide enough protection against singling out and linkability, but may protect against inference attacks. Moreover, it can be combined with a generalization technique to provide more privacy guarantees. We consider several particular randomization techniques: noise addition, permutation, and differential privacy.

4.4.1.1 Noise addition

Noise addition technique adds noise into a dataset. It makes data less accurate (for example, measurement of height is saved with predetermined inaccuracy instead of an exact height). However, it is still possible to single out the records of an individual, even though the records are less reliable. Moreover,

noise addition does not protect against linking records related to an individual, but this linking is less reliable. Noise addition should be used in combination with other anonymization techniques.

4.4.1.2 Permutation

This technique consists of shuffling the values of attributes in a table so that some of them are artificially linked to different data subjects [1]. This technique is useful when it is important that the distribution of attributes remains the same.

Similarly to noise addition, it is still possible to single out the records of an individual, but records are less reliable. Permutation may prevent “correct” linking of attributes, if it affects both attributes and quasi-identifiers¹. Inference may still be possible especially if attributes are correlated. Therefore, similarly to noise addition, permutation should be used in combination with other anonymization techniques.

4.4.1.3 Differential privacy

Differential privacy [11] aims to provide maximum accuracy of queries from a dataset while minimizing the chances of identifying its records. In case of differential privacy, datasets are provided to authorized parties in response to a specific query rather than through the release of a single dataset. Differential privacy requires that each query should return almost the same result (with some specified error) for every two datasets that differ on a single element. To better understand this statement, assume we have two otherwise identical databases, one with information about particular person in it and one without. Differential Privacy ensures that the probability that a statistical query will produce a given result is nearly the same whether it is conducted on the first or second database. In order to accomplish this requirement, noise addition technique applies to the dataset. This fact protects an individual to be singled out from a dataset. Linkability and inference attacks still may be possible using multiple requests and comparing responses.

4.4.2 Generalization

Generalization is another family of anonymization techniques. Generalization replaces individual values of attributes with a broader category, such as region instead of city, or age category instead of concrete age. This family of techniques gives high level of protection against singling out, but it requires more specific approaches to prevent linkability and inference.

4.4.2.1 K-anonymity

K-anonymity, introduced by Sweeney [14], aims to prevent a singling out by grouping each data subject with at least k other individuals. In order to achieve this, attributes values are generalized so that each individual in the dataset shares the same values with at least other k-1 individuals.

¹Variable values or combinations of variable values within a dataset that are not structural unique but might be empirically unique and therefore in principle uniquely identify a population unit [13].

As we mentioned already K-anonymity protects against singling out. K-anonymity makes a linkability attack more difficult, because records can be linked only within the group of k-users. However, k-anonymity is fairly weak against inference attack.

Sensitive information is data that must be protected from unauthorized access to preserve the privacy of individuals. In our case all medical information, such as diseases, different measurements, and medicine prescriptions, is considered as sensitive information.

Considering the following example, Figure 4.2 shows an initial dataset and Figure 4.3 shows 4-anonymous generalized dataset. Each group of individuals contains at least four records. However, if the attacker knows that a specific individual is in the dataset and knows that this person is in his 30s and has “130” as his first digits of zip code he will easily find that this person has cancer.

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Figure 4.2: Inpatient Microdata (adapted from [1])

4.4.2.2 L-diversity

The l-diversity model [12] is an extension of the k-anonymity model that provides protection against inference attacks. L-diversity ensures that in each equivalence class every sensitive attribute has at least L different values. In this situation, the attacker with background knowledge on a specific individual still has a significant level of uncertainty in sensitive information.

Considering the dataset mentioned in the previous section (Figure 4.2), a 3-diverse dataset is shown in Figure 4.4. There are at least three different values of sensitive attributes in each equivalence class.

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Figure 4.3: 4-anonymous Inpatient Microdata (adapted from [1])

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Figure 4.4: 3-Diverse Inpatient Microdata (adapted from [1])

4.4.2.3 T-closeness

T-closeness is a further refinement of the l-diversity anonymization technique. The goal of t-closeness is to create equivalence classes that resemble the initial distribution of sensitive attributes. The distance between the distribution of sensitive attributes in the initial dataset and in each equivalence class should be no more than a threshold t .

4.4.3 Comparative analysis

Figure 4.5 shows a comparative analysis of all mentioned anonymization techniques. As we can see from the figure, the generalization techniques (k-anonymity, l-diversity, and t-closeness) provide higher protection by themselves compared to the randomization family of techniques (noise addition, permutation, differential privacy). As we can see from the overview of different anonymization techniques the choice of techniques should be decided depending on a particular problem, possibly by using a combination of different techniques.

Noise addition is not applicable to our problem because they can significantly harm results of further medical research. For example, in COPD (chronic obstructive pulmonary disease) practice the probability that a patient does not smoke is almost zero. If noise is added to the data, it can bring wrong research results. Permutation can significantly affect veracity of data as well. For example, exact type and number of medicine prescriptions for a patient with a given diagnosis is very important information that should not be altered. Because we do not know in advance which queries researchers will use, differential privacy is not a suitable approach in our case as well.

Generalization family of anonymization techniques does not require preliminary knowledge of future data requests. It also protects data on a higher level. Therefore, generalization techniques seems to be more appropriate for our case compared to randomization techniques.

Technique	Singling out	Linkability	Inference
Noise addition			
Permutation			
Differential Privacy			
K-anonymity			
L-diversity/ T-closeness			

	Does not protect
	May protect
	Protect

Figure 4.5: Anonymization techniques comparative analysis

5 System Requirements

After analyzing the problem and the domain and getting acquainted with stakeholders and their needs a set of requirements is formulated. This chapter describes functional and non-functional requirements of the system.

The priority of the requirements is defined using the MoSCoW [15] method:

- **MUST** – critical to the current delivery time box in order to consider project as a success.
- **SHOULD** – important, but not necessary for delivery in the current delivery time box.
- **COULD** – desirable, but not necessary.
- **WILL NOT** – not planned to be implemented in the current delivery time box.

5.1 Functional Requirements

The following requirements are the high level functional requirements of the project. The detailed design of these requirements is presented in the System Architecture (see Section 6).

ID	Description	Priority
F01	System shall link all medical records concerning one individual patient to one Cooperative Pseudonym.	MUST
F02	Cooperative Pseudonym should be regenerated each time during the generation of a new dataset for the Cooperative Database.	SHOULD
F03	System shall not contain any personal data from free format consult data files.	SHOULD
F04	System shall recognize and delete duplicate medical information related to one individual patient.	MUST
F05	System shall provide the ability to compare the effectiveness of different treatment.	MUST
F06	System shall provide statistics on a multi-regional and national level.	MUST
F07	System shall have SIM as a data model.	MUST

F08	System shall extend SIM model if the existing data model does not satisfy needed statistics.	WILL NOT
F09	System shall not provide possibility to trace records back to the regional level.	MUST
F10	System shall provide the ability to an individual patient to request opt-out.	WILL NOT
F11	System shall provide the ability to export data for private companies to conduct research.	MUST
F12	System shall perform an anonymization check on the datasets, which will be exported for private companies.	MUST

The main goal of the Cooperative Database is to give an opportunity to conduct medical research at a national level. Requirements F01, F04, F05, F06, F11, and F12 serve this main goal. As was mentioned in Section 4.3 each patient should have one pseudonym in our system (requirement F01) and system should not store duplicated records (requirement F04). Data aggregated at a national level will allow comparing effectiveness of different treatment (requirement F05) and provide statistics at a national level (requirement F06). In order to provide the possibility for other companies to conduct research, the Cooperative Database will have an ability to export data (requirement F11). According to the data privacy regulations all data delivered to third-party organizations should be anonymous. Thus, our system will perform an anonymization check before the data delivery (requirement F12).

The importance of having a unique pseudonym for each particular patient was explained in Section 4.3.2. However, it preserves less privacy for individuals. Therefore, the Cooperative Pseudonym will be never the same for a particular person between several generations of Cooperative Database (requirement F02) and it will be not possible to trace records back to the regional level (requirement F09).

Some free text format medical notes, such as medicine prescriptions, can contain personal information. When data is collected at a national level, data privacy regulations do not allow this situation. Requirement F03 captures this domain constraint.

Requirement F07 defines the data model for the Cooperative Database. This model is described in more technical detail in Chapter 6. As was formulated in requirement F07, the SIM is the data model for the Cooperative Database. Therefore, if the SIM structure changes it affects our solution. However, the extension of the SIM is not in the scope of our project. Because of this requirement F08 has “WILL NOT” priority. Requirement F10 states that each individual has the right to request opt-out, i.e., to stop using his medical data for research purposes. This functionality is out of the scope of the current project as well. Both these requirements (F08 and F10) will be designed and implemented in the scope of other Medworq projects (see Section 4.2).

5.2 Non-functional Requirements

Non-functional requirements are requirements that specify system qualities, rather than specific behaviours that are defined by functional requirements. In our report we use non-functional requirement terminology based on the “ISO SQuaRe Product Quality” document [16].

ID	Category	Description	Priority
NF01	Privacy/Security	It must not be possible to identify an individual from the dataset.	MUST
NF01.1	Privacy/Security	System shall provide a configurable choice for an anonymization technique (k-anonymity, l-diversity, t-closeness).	MUST
NF01.2	Privacy/Security	System shall provide an ability to configure anonymization parameters.	MUST
NF01.3	Privacy/Security	System shall provide metrics to measure the level of anonymity.	MUST
NF02	Privacy/Security	Data transfer process from the Population Databases to the Cooperative Database should be secured.	SHOULD
NF03	Security	System shall control the authorization process.	SHOULD
NF04	Interoperability	System should be compliant with other systems/ data sources in the future.	SHOULD
NF05	Efficiency	System should use data-driven design approach.	MUST

As was already mentioned before, preserving data privacy is one of the major project requirements. In order to satisfy this, requirements NF01, NF02, and NF03 were formulated. The Cooperative Database should implement techniques for data anonymization (requirement NF01). Techniques and their parameters should be configurable for the user of the system (requirements NF01.1 and NF01.2). The system should provide metrics to measure the level of anonymity (requirement NF01.3). The data collection process should be secured by design and implementation (requirement NF02). As was mentioned in Section 4.2 the authorization process will be designed and implemented as a separate project inside Medworq and later on will be integrated in our solution (requirement NF03).

Currently, more and more medical organizations are collecting and sharing medical data. Thus, it is very important that our system will be easy to extend in the case of connecting a new medical source. In other words, our system should support interoperability (requirement NF04). With a growing number of Population Databases, the amount of data for the Cooperative Database will grow accordingly. Therefore, it is desired that our system process data in an efficient way. Data-driven design approach (requirement NF05) allows avoiding code duplication, which leads to errors and maintenance issues.

6 System Architecture

The requirements of the system, formulated in the previous chapter, are used to drive the definition of the system architecture. In this chapter we define the main components of the Cooperative Database and describe how these components interact with each other.

6.1 Architecture

The Cooperative Database System (CDS) was designed using a data-driven approach, as was formulated in the requirement NF05 (see Section 5.2). That means that the system recognizes how to process data from a metadata description, which is represented by the SIM in our system. The SIM describes the tables used to store the data along with their fields and purposes. In order to understand how the CDS process data, let us consider the structure of SIM. For each table SIM describes a list of fields. Each field is described with the following information:

- **name**
- boolean value that identifies if it is a **key value** or not
- **type** of a field (pseudonym, string, integer, or other type)
- **location** of a field:
 - **local** means that a field should be present at the data source location
 - **regional** means that a field should be present in the Population Database
 - **national** means that a field should be present in the Cooperative Database

The key value identifier allows distinguishing records in terms of duplicates. If two records store the same information in key value fields it means that these records are duplicates. The type of a field allows detecting a pseudonym field in order to link records by this pseudonym together. Due to the privacy regulations discussed in Chapter 4 some of the fields are allowed to be stored at the Population and Cooperative Database location whilst some of the fields are not allowed to leave medical organization. In order to manage these restrictions each fields specifies a location list.

When the CDS starts any data process, it recognizes from the SIM which entities are needed to be process and in which way. For example, for the data collection process the CDS gets only fields that have **national** location. The data-driven approach serves for decreasing code duplication which leads

to errors and testing issues. It also makes the system easier to extend and maintain.

The core of the architecture of our system is the Extract, Transform, Load (ETL) process. The main steps of the ETL process are:

- extract data from all Population Databases to our system
- transform data, i.e., link records, delete duplicates, anonymize data
- load data to one common storage

Figure 6.1 shows an overview of CDS architecture. Each component in the figure is explained in the following sections. It is worth mentioning that CDS uses a big data technology stack because of the large amount of aggregated data. All grounds for the technologies that we used in the project are described separately in Chapter 7.

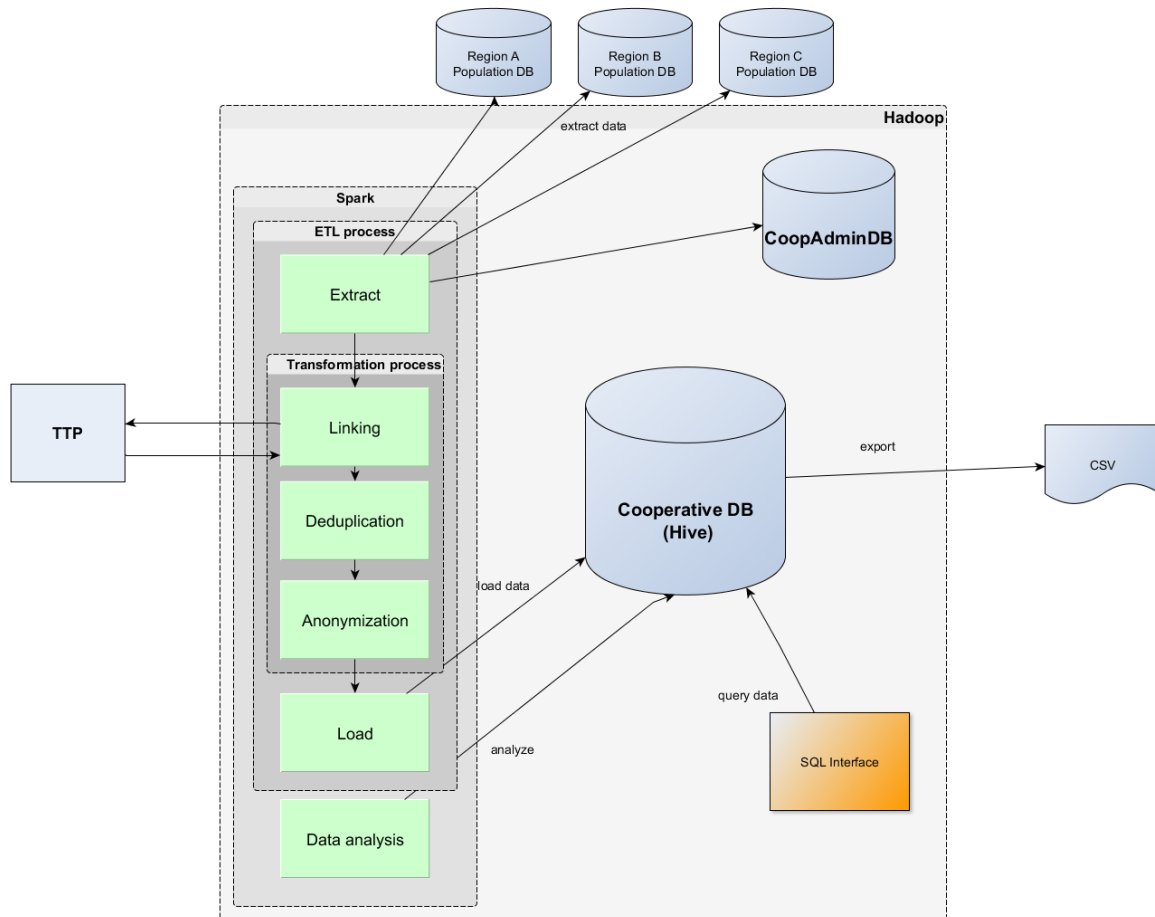


Figure 6.1: System architecture

7 Design and Technology choices

This chapter describes design and technology choices that arose during the design phase of the solution. The chapter presents each choice and alternatives, and then provides arguments for and against each alternative in detail.

The main criteria for the technology choice:

- It should be easy to export a subset of aggregated data into plain text or a relational database.
- It should be possible to conduct research on aggregated data.

7.1 Data storage

Due to the large amount of data that will be collected from all regional databases, we looked into two distributed data storage solutions:

- Hadoop
- RIAK

The next sections describe arguments for and against each of the approaches.

7.1.1 Hadoop

Hadoop¹ is an open-source software framework used for distributed storage and processing of big data sets. Hadoop was designed to support scalability and be resistant to failures. The core of Apache Hadoop consists of a storage component, known as Hadoop Distributed File System (HDFS), and a processing component which can be written in different programming languages such as Java, Python, C++, and Ruby.

Hadoop provides several additional software packages that can be installed on top of Hadoop. These software packages provide functionality of data processing, data analysis, querying data. We describe frameworks choices in Sections 7.2 and 7.3.

¹<http://hadoop.apache.org/>

Currently in the Population Database data is stored on Microsoft SQL Servers. We have two possible options of how to extract data from the MS SQL Server to Hadoop:

- Sqoop streaming solution
- Extract the database to a file and stream the file to the Cooperative Database

Sqoop² is a tool designed for efficiently transferring bulk data between Hadoop and structured data-stores such as relational databases. This approach requires that at the data collection moment, all regional servers will be available to connect. Open connection with the possibility to query all data from the server is not reliable from a security perspective. In order to provide a higher level of security, we chose the extracting database to file approach.

Considering strengths and weaknesses of Hadoop solution:

- **Strengths:**
 - Distributed data storage
 - Distributed data processing
 - Possibility to load to HDFS extracted data from MS SQL
 - SQL interfaces to query data (see more details in Section 7.2)
- **Weaknesses:**
 - Complex configuration. Maintenance for Hadoop cluster from third-party needed.
 - Compatibility of different software packages. Some packages depend on other packages and require a specific version for compatibility.

Hadoop is released under Apache³ license, version 2.0, which allows the user of the software the freedom to use the software for any purpose without concern for royalties.

7.1.2 RIAK

RIAK⁴ is a distributed NoSQL key-value database with advanced local and multi-cluster replication that guarantees reads and writes even in the event of hardware failures or network partitions. RIAK has official drivers for Ruby, Java, Erlang and Python. Riak provides a REST-ful API for basic PUT, GET, POST, and DELETE functions. More complex queries are also possible using MapReduce jobs. In order for the jobs to have high performance, RIAK schema should be adapted as necessary.

RIAK provides **Spark connector** which moves data from RIAK to Spark⁵ (see Section 7.2.3) for in-memory analysis. Further, the results can be stored back in RIAK.

²<http://sqoop.apache.org/>

³<http://www.apache.org/licenses/LICENSE-2.0>

⁴<http://basho.com/products/riak-kv/>

⁵<http://spark.apache.org/>

Functionality that allows retrieving data from the MS SQL Server and putting it into RIAK database should be written explicitly; there is no ready solution for this purpose. Moreover, due to the fact that RIAK is a NoSQL solution database, schema should be structured in a specific way to support complex queries.

Considering strengths and weaknesses of RIAK solution:

- **Strengths:**

- Distributed data storage
- Distributed data processing
- Easy to configure and maintain

- **Weaknesses:**

- There is no ready solution to stream data from MS SQL to RIAK.
- Data should be restructured from relational representation to NoSQL representation.
- In order to increase performance of data querying we need to know possible queries in advance to create appropriate database schema.
- It is not possible to query data in a common SQL way.

RIAK is released under Apache license, version 2.0.

7.1.3 Data storage decision

Table 7.1 provides a comparison of Hadoop and RIAK data storage solutions based on several earlier mentioned criteria. For each criteria we indicate either solution provides needed functionality (✓) or not (✗). Due to the fact that Cooperative Database will be used by data analysts we need a widely used interface for querying data. The most widely used interface is SQL. Hadoop solution provides SQL interface for querying collected data. Hadoop also provides data analysis tools (with support of several programming languages such as Scala, Java, Python) to conduct analysis on the Hadoop cluster in a distributed way.

Data can be exported from Hadoop to plain text files such as CSV and be further analyzed by data scientists who can use the tools that they are more familiar with. For the above mentioned reasons, we chose **Hadoop** as a data storage for Cooperative Database.

Criteria	Hadoop	RIAK
Distributed data storage	✓	✓
Distributed data processing	✓	✓
Out-of-the-box functionality for loading data from relational databases	✓	✗
SQL (or SQL-like) interface for querying data	✓	✗
Easy to configure and maintain	✗	✓
No need for restructuring data from original relational data format	✓	✗

Table 7.1: Data storage comparison

7.2 Data processing frameworks

After we extract data from all regional databases and load it into Hadoop storage, we need to process the data: link records, delete duplicates, and anonymize. This section discusses the possibilities offered by Hadoop and Hadoop software packages.

7.2.1 Hadoop MapReduce

Hadoop by default contains a **MapReduce** component, which is an implementation of the MapReduce⁶ programming model. A MapReduce program is composed of a Map() procedure (method) that performs filtering and sorting and a Reduce() method that performs a summary operation. The "MapReduce System" handles the processing by marshaling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance. Hadoop provides interfaces in several programming languages such as C++, Java, Python, Ruby, and Perl to write MapReduce jobs.

Considering strengths and weaknesses of MapReduce approach:

- **Strengths:**
 - Parallel processing
 - Scalability
- **Weaknesses:**
 - MapReduce programs are not guaranteed to be fast.
 - MapReduce is best suited to batch process huge amounts of data which is already at one place, rather than handle streaming data.
 - You need to rethink/rewrite trivial operations like Joins, Filter to fit it in MapReduce paradigm.

⁶<http://en.wikipedia.org/wiki/MapReduce>

7.2.2 Spark

Apache Spark⁷ is a fast and general engine for large-scale data processing. Spark is designed for fast computation. Spark provides API in Java, Scala, Python, and R. Spark has module for working with structured data called Spark SQL⁸. Spark SQL can connect to any data source such as HDFS storage, Hive (see Section 7.3.1), or JDBC in the same way. It allows query structured data inside Spark programs, using either SQL or a DataFrame API. Spark SQL provides API in the same programming languages as Spark.

Considering strengths and weaknesses of Spark framework:

- **Strengths:**

- Spark runs programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.
- Spark is easy to program as it has a lot of high level operators.
- Spark can process real time data i.e. data coming from the real-time event streams.
- Spark provides Spark SQL module.

- **Weaknesses:**

- Spark requires large amount of RAM to run in-memory (compared to MapReduce that processed data from disk storage).

Spark is released under Apache license, version 2.0.

7.2.3 Data processing framework decision

Table 7.2 provides a comparison of MapReduce and Spark data processing solutions based on several earlier mentioned criteria. For each criteria we indicate either solution provides needed functionality (✓) or not (✗). Apache Spark is a more advanced cluster computing engine than MapReduce. Spark can handle several types of requirements (batch, streaming, iterative) while MapReduce is limited to batch processing. Accordingly, we chose **Spark** as the data processing solution for our system.

⁷<http://spark.apache.org/>

⁸<http://spark.apache.org/sql/>

Criteria	MapReduce	Spark
Parallel processing	✓	✓
Scalability	✓	✓
Execution speed	✗	✓
Ready to use high level operators, such as Join, Filter	✗	✓
Uniform connection to different data sources, such as HDFS or JDBC	✗	✓

Table 7.2: Data processing framework comparison

7.3 SQL support frameworks

As was mentioned earlier aggregated data will be used for data analysis by Medworq. The Cooperative Database should provide an interface for query data in a way that will be the most similar to SQL because Medworq engineers are familiar with SQL. Therefore, they do not need to learn another query language.

There are several frameworks offered by Hadoop that provide this functionality. We consider which of them suits our purposes the best.

7.3.1 Hive

Hive⁹ is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. Hive gives an SQL-like¹⁰ interface to query data stored in various databases and file systems that integrate with Hadoop. Hive converts an SQL query into a MapReduce job which executes in a distributed mode.

The storage and querying operations of Hive closely resemble that of relational databases, which will be convenient for engineers who are already familiar with relational databases and SQL. Hive is released under Apache license, version 2.0.

7.3.2 Pig

Apache Pig¹¹ is a platform for analyzing large datasets. The language for this platform is called Pig Latin. Pig Latin syntax is less similar to native SQL syntax than Hive syntax which means that engineers need to learn a new syntax for querying data. Pig is released under Apache license, version 2.0.

⁹<http://hive.apache.org/>

¹⁰<http://cwiki.apache.org/confluence/display/Hive/LanguageManual>

¹¹<http://pig.apache.org/>

7.3.3 HBase

Apache HBase¹² is an open source, non-relational, distributed database. HBase runs on top of HDFS. HBase is a column-oriented key-value data storage. Unlike relational databases, HBase does not support SQL scripting. Instead, user needs to write a Java application, which is similar to a MapReduce application. By default HBase does not support joins, they are needed to be programmed specifically. Due to the fact that HBase is a NoSQL solution a database schema should be considered in advance (similar to RIAK, see Section 7.1.2).

7.3.4 SQL framework decision

Table 7.3 provides a comparison of Hive, Pig, and HBase SQL frameworks based on several earlier mentioned criteria. For each criteria we indicate either solution provides needed functionality (✓) or not (✗). Due to the fact that the Cooperative Database should provide interface as close as possible to SQL, we chose **Hive** framework as the most suitable for our system.

Criteria	Hive	Pig	HBase
SQL (or SQL-like) interface for querying data	✓	✗	✗
Query execution in distributed mode	✓	✓	✓
Possibility of using SQL JOIN operator	✓	✓	✗
No need for restructuring data from original relational format	✓	✓	✗

Table 7.3: SQL support framework comparison

¹²<http://hbase.apache.org/>

8 Feasibility Analysis

After an analysis of the problem, domain, and stakeholder needs, a feasibility analysis is performed in order to identify the issues and risks that may occur during the project. This chapter discusses the issues and risks identified. It also describes mitigation strategies for the identified risks.

8.1 Issues and Challenges

This section describes issues and challenges that are encountered during the lifetime of the project.

As described in Chapter 3 and Chapter 4 the project is conducted at the combination of several domains: healthcare, IT, and data privacy. This fact requires an input from different groups of stakeholders such as medical specialists, IT specialists, and legal experts. It is a challenge to organize a meeting between all of them. Moreover, it is a challenge to hold specialists from different domain on the same page during the meeting. In our case organizing one common meeting between all stakeholders was not possible. Thus, the trainee organized small local meetings between stakeholders from one domain and after this updated others with the output from the meetings. This was more time consuming than having one common discussion.

The healthcare domain was not familiar for the trainee. Having sufficient domain knowledge is very important to analyze system requirements correctly. Therefore, it was very important to get domain knowledge quickly by communicating with different domain experts.

8.2 Risks

During the project several risks are identified. Table 8.1 describes the risks identified together with their impact on the project and corresponding mitigation strategy.

<i>Risk</i>	The solution uses new technologies that the trainee and company specialists have no experience.
<i>Impact</i>	Becoming familiar with new technology stack is time consuming. This can affect the fulfillment of the project scope.

<i>Mitigation strategy</i>	In order to become familiar with the new technology stack and make correct choices for particular technologies the trainee performed prototyping in order to analyze how particular technology will fulfill needed goal. This analysis requires additional time in the beginning of the project, but later on allows economizing time significantly.
<i>Risk</i>	Communication interfaces that should be provided by third-party companies are not ready by the end of the current project.
<i>Impact</i>	We cannot test real integration.
<i>Mitigation strategy</i>	Create and use synthetic response in order to test all steps of data aggregation process. However, additional time for the generation of synthetic response is needed.
<i>Risk</i>	Not all requirements can be met given the limited time of the project.
<i>Impact</i>	Deliverable of the project would not meet all list of initially formulated requirements.
<i>Mitigation strategy</i>	<ul style="list-style-type: none"> • Prioritize requirements together with stakeholders along the lifetime of the project. Some of the requirements with the “SHOULD” priority will be accomplished as a future work. • Make the design of the system easy to extend with more functionality. • Provide detailed documentation for accomplished work, leave recommendations for future work

Table 8.1: Risks and mitigation strategies

9 Implementation and Deployment

Chapter 6 and Chapter 7 have provided the system architecture and design choices that were made. This chapter focuses on the realization of the system.

The implementation of the Cooperative Database is part of the deliverable of this project. The implementation is based on the architecture presented in Chapter 6 and satisfies the requirements defined in Chapter 5. The system is delivered as Java Enterprise Application.

9.1 Development Environment

For the development of the Cooperative Database the following technologies were used:

- **Java**¹ as a main programming language
- **IntelliJ IDEA**² as an integrated development environment (IDE)
- **Maven**³ a build automation tool
- **Spring**⁴ as a framework that helps building decoupled applications

IntelliJ IDEA is a Java IDE for developing software. It has a full support for developing Java Enterprise applications, i.e. complex business applications that used to satisfy the needs of an organization rather than individual users. Figure 9.1 shows IntelliJ IDEA development environment. The left window shows the project structure. The center window shows the main development environment, where different java classes or configuration files can be created. The right window shows a maven console that is used for building the application.

¹<http://www.oracle.com/java/index.html>

²<http://www.jetbrains.com/idea/>

³<http://maven.apache.org/>

⁴<http://spring.io/>

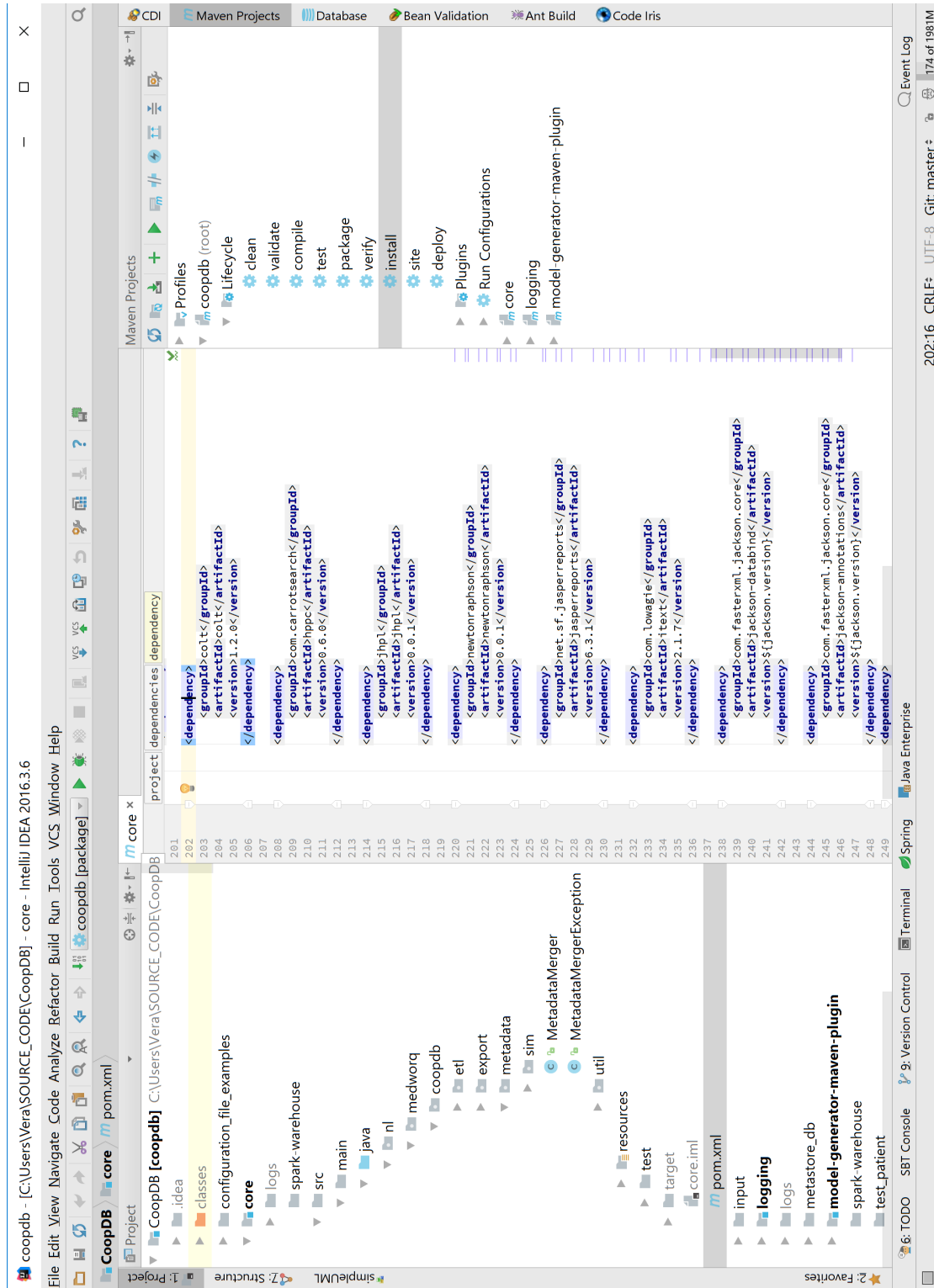


Figure 9.1: Development environment of the Cooperative Database in IntelliJ IDEA

Maven is a build automation tool used for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies. Figure 9.2 shows an output of Maven packaging process. Maven packages our application into a Java ARchive (JAR) file. This JAR file is used to execute our application on Hadoop server. Maven puts all third-party libraries needed for application into a separate folder (*.lib). The configuration file that allows configuring different parameters for our application, such as paths to Hadoop configuration files, anonymization parameters, is located inside “config” folder.




 config	17-Aug-17 3:15 PM	File folder	
 core-1.0-SNAPSHOT.lib	17-Aug-17 3:15 PM	File folder	
 core-1.0-SNAPSHOT	17-Aug-17 3:15 PM	Executable Jar File	114 KB

Figure 9.2: Application build output

Spring is an application framework that helps building decoupled systems by implementing a dependency injection technique. Decoupled systems are easier to maintain and extend in the future. Dependency injection means that the dependencies that a class needs are not fetched by the class itself, but “injected” by a different component, called a container, when the instance is created. Dependency injection technique provided by Spring helps to “wire” different components or processes together. Possibility to wiring different components in different ways increases modularity of the system. In our application we have different processes, such as data linking, data deduplication, that should be combined in one process. However, the order of these steps can change. Therefore, Spring solution is very beneficial for our application.

In addition to the aforementioned development environments, the following libraries are used in the project:

metadata dependency from the Population Database used for retrieving the latest version of SIM.

hadoop as a data storage solution, as was decided in Section 7.1.

spark as a data processing framework, as was decided in Section 7.2.

spring framework for possibility to use dependency injection technique.

hive as an SQL framework, as was decided in Section 7.3.

jasperreports for generating reports.

antlr as a tool for auto-generating different SQL scripts based on SIM description.

javapoet as a tool for auto-generating plain old Java objects (POJO) based on SIM description. These POJOs are data model objects that store data such as patient information, prescribed medicine, or disease cases.

jackson for parsing JSON configuration files.

amazonaws for configuration with Amazon Web Services (AWS). More details about deploying applications on AWS can be found in next section.

junit for unit testing.

9.2 Deployment Environment

As mentioned in Section 7.1.1 Hadoop configuration is a complex task. There are several distributions that provide configured version of Hadoop and other data processing frameworks. The most popular and powerful distributions are provided by Cloudera⁵, HortonWorks⁶, and Amazon⁷. For deploying our application we use Elastic MapReduce (EMR) provided by Amazon Web Services (AWS). EMR contains already configured Hadoop, Spark, and Hive frameworks with their latest versions which makes it suitable for the project.

However, the implementation of our solution is not coupled specifically to the EMR service. Later on, if needed, our application can be deployed to any other Hadoop distribution platform, such as Cloudera or HortonWorks.

⁵<http://www.cloudera.com/products/open-source/apache-hadoop.html>

⁶<https://hortonworks.com/apache/hadoop/>

⁷<http://aws.amazon.com/emr/>

10 Verification and Validation

The previous chapter have described the implementation and deployment of the Cooperative Database System. In order to ensure that the system is being built correctly, the process of verification and validation processes have been performed. This chapter describes the process of verification and validation.

10.1 Verification

The verification process answers to the question “are we building the system right?”. In other words, the verification ensures that the system has been built according to the requirements and design specifications.

10.1.1 Unit testing

In order to verify our system during the development phase unit testing has been performed. Figure 10.1 shows the unit tests coverage for the system. Unit tests were created for every piece of functionality that does not require integration with third-party systems. The remaining part of the functionality was tested with integration tests. The following section describes integration testing in detail.

As shown in the Figure 10.1 the following functionality is covered with unit-tests:

- **nl.medworq.coopdb.etl.extract** package is responsible for loading data to Spark.
- **nl.medworq.coopdb.etl.transform** package is responsible for linking and deduplicating processes.
- **nl.medworq.coopdb.metadata** package is responsible for SIMs merging process that described.
- **nl.medworq.coopdb.etl.load.hive** package is responsible for loading data to Hive storage and generating SQL requests.

Each time the system packages its functionality in a JAR file (see more details in Chapter 9), all unit test are performed. The system does not allow packaging application into a JAR before all unit test are passed.

Package	Class, %	Method, %	Line, %
all classes	39.3% (22/ 56)	8% (54/ 675)	11.4% (222/ 1940)
Coverage Breakdown			
Package	Class, %	Method, %	Line, %
nl.medworq.coopdb.etl	20% (1/ 5)	6.7% (1/ 15)	1.1% (1/ 89)
nl.medworq.coopdb.etl.anonymization	66.7% (2/ 3)	18.8% (3/ 16)	4.3% (5/ 116)
nl.medworq.coopdb.etl.anonymization.report	0% (0/ 1)	0% (0/ 4)	0% (0/ 44)
nl.medworq.coopdb.etl.anonymization.report.model	0% (0/ 2)	0% (0/ 32)	0% (0/ 32)
nl.medworq.coopdb.etl.extract	100% (5/ 5)	29% (9/ 31)	18.5% (23/ 124)
nl.medworq.coopdb.etl.load.hive	85.7% (6/ 7)	57.6% (19/ 33)	48.7% (57/ 117)
nl.medworq.coopdb.etl.transform	100% (2/ 2)	100% (6/ 6)	100% (35/ 35)
nl.medworq.coopdb.export	50% (1/ 2)	31.2% (5/ 16)	25.9% (30/ 116)
nl.medworq.coopdb.export.chronicKidneyDisease	0% (0/ 5)	0% (0/ 188)	0% (0/ 658)
nl.medworq.coopdb.export.model	0% (0/ 2)	0% (0/ 8)	0% (0/ 11)
nl.medworq.coopdb.metadata	100% (2/ 2)	100% (5/ 5)	100% (62/ 62)
nl.medworq.coopdb.metadata.sim	100% (2/ 2)	83.3% (5/ 6)	87.5% (7/ 8)
nl.medworq.coopdb.model	0% (0/ 16)	0% (0/ 311)	0% (0/ 521)
nl.medworq.coopdb.util	50% (1/ 2)	25% (1/ 4)	28.6% (2/ 7)

Figure 10.1: Unit tests coverage

10.1.2 Integration testing

The goal of integration testing is to demonstrate that different modules of the system work together in a proper way. Integration tests cover whole application functionality. During the integration testing the complete functionality of the Cooperative Database was tested.

The system was verified based on two datasets exported from two different Population Databases. The tests have been performed on an AWS EMR cluster with the following configuration:

- 3 instances (1 master, 2 workers). Each instance has the following configuration:
 - 16 virtual Central Processing Units (vCPU)
 - 32 GiB memory

Table 10.1 describes exported dataset characteristics, such data export size and number of records to process. All data from both Population Databases has been processed (linked and deduplicated), anonymized, and stored at the Cooperative Database.

	Population DB region A	Population DB region B	Total
Data export size	11.2GB	12.6GB	23.8GB
Approximate no. records	150 million	152 million	302 million

Table 10.1: Integration test data input characteristics

All data was processed (linked and deduplicated), anonymized, and stored at the Cooperative Database. The risks described in Section 8.2 related to integration between the Cooperative Database and TTP occurred. Integration from TTP side was not ready by the end of the project. Thus, synthetically generated response from TTP was used in order to perform the linking process.

10.2 Validation

The validation process answers to the question “are we building the right system?”. In other words, the validation process ensures that the system actually meets the stakeholders expectations and requirements. The validation process usually is performed at the end of development of a system.

Table 10.2 summarizes how the system requirements defined in Chapter 5 were covered. The table shows requirements with their priority, the status at the end of the project, and the future action (if needed). All requirements with “MUST” priority were designed and implemented during the project. Some of requirements with “SHOULD” priority were designed, but not implemented due to the time limitation of the project. Requirements with “WILL NOT” priority were not in the scope of the project and will be designed and implemented in other projects.

Requirement ID	Priority	Status	Future action
F01, F04, F05, F06, F07, F09, F11, F12, NF01, NF01.1, NF01.2, NF01.3, NF05	MUST	designed and implemented	—
NF04	SHOULD	accomplished	—
F02, NF02, NF03	SHOULD	designed	Implementation based on the design should be accomplished
F03	SHOULD	not accomplished	Third-party library will be used for this functionality
F08, F10	WILL NOT	not accomplished	A design and an implementation will be accomplished in the scope of other projects.

Table 10.2: System requirements coverage

The main goal of the Cooperative Database is to give an opportunity to conduct medical research at a national level. Therefore, in order to validate our system we have exported data from the Cooperative Database for a medical research in area of Chronic Kidney Disease (CKD). Hence, the Requirement F05, F06, and F11 are supported by the system.

Before exporting data to a researcher, data was anonymized. Therefore, Requirement F12 is supported by the Cooperative Database. Another main requirement for the Cooperative Database is preserving data privacy (Requirement NF01). After the anonymization is complete the system generates a report with metrics on anonymity, data utility, and re-identification risks.

The solution allows configuring anonymization techniques together with the needed parameters as was specified in Requirement NF01.1 and NF01.2. The solution provides metrics on anonymity, data utility, and re-identification risks as well. According to all information described above our system supports Requirement NF01.3.

The system links all medical records concerning one individual patient together based on the Cooperative Pseudonym (Requirement F01). After the Cooperative Database finishes this linking process it deletes all connections between Regional Pseudonyms (from Population Database) and Cooperative Pseudonyms. Thus, the Requirement F09 is supported. In order to provide valuable data for medical research, the system recognizes and deletes duplicate medical information (Requirement F04).

The communication process with the TTP in order to generate Cooperative Pseudonyms (Requirement F02) was designed within the project. However, the implementation based on the design should be accomplished in the future from both sides: Cooperative Database and TTP.

The system uses the SIM from the Population Database as a data model as was specified in Requirement F07. The secure data transfer process from Population Databases to the Cooperative Database (Requirement NF02) was designed. However, due to the time limitation of the project it was not implemented. Therefore, it should be implemented based on the design in the future. The authorization

process (Requirement NF03) was designed within the project. Nevertheless it should be implemented based on the design as a future work as well.

A third-party library will be used and integrated in our system in order to delete personal data from free format data files. Therefore, currently, Requirement F03 is not supported in our system.

As described in Chapter 6 the system uses data-driven approach. Therefore, the Requirement NF05 is supported. The system design is modular which makes it easy to connect new data sources to our system in the future (Requirement NF04).

11 Conclusion

This chapter elaborates the results achieved in this project as well as the added value to the stakeholders.

11.1 Result

The end result of the project is an application that provides an opportunity to conduct medical research at a national level. This opportunity creates benefits for different group of stakeholders such as individual patients and the Dutch society, pharmaceutical companies, and research institutes. Findings from a medical research at a national level, such as risk factors, results of treatment, and healthcare approaches, allow improving patient care and preventive healthcare. This is beneficial for both individual patients and the society. Medical research at a national level allows pharmaceutical companies to measure effectiveness of medication and medical treatment. Medical data aggregated at a national level gives to the research institutes a wide area for research directions.

The main challenge of the system design was a trade-off between keeping data utility while preserving privacy of individuals. In the beginning of the project problem and domain analysis was conducted. This analysis provides an analysis and comparison of existing data privacy preserving techniques. The most appropriate techniques were chosen and implemented in the system.

Other challenges are connected with data aggregation process. Data collected from different regions may contain duplicate information. The Cooperative Database handles duplicates and remove redundant data from datasets. Data related to one particular patient may have different identifiers at the aggregated dataset. The Cooperative Database link records related to one individual patient to one common identifier. Handling these two challenges allows avoiding wrong outcomes from medical research.

The system was verified based on real data. To show the added value of the system, data for medical research on Chronic kidney disease was exported from the system.

The application is easy to build and deploy on a cluster for further execution. The application demonstrates quality attributes, such as modularity, which makes it easy to maintain the application and extend with new functionality in the future.

12 Project Management

This chapter describes how the project was carried out from a project management perspective. It describes which methods and approaches were used in order to create the project plan and monitor the project progress. The chapter provides a description of the work breakdown structure and the project planning.

12.1 Introduction

The project was conducted for nine months. The trainee was the main responsible person for the project management as well as for the realization of the project. The Project Steering Group (PSG) helped the trainee to manage the project and provide the trainee with the needed domain knowledge and contacts of specialists who have this knowledge. The PSG consisted of the supervisors from Medworq, Freek van Keulen and Johan Ruiten, and the TU/e supervisor Nicola Zannone. PSG meetings were organized on a monthly basis. The goal of these meetings was to discuss the current status of the project, check the progress since last meeting, define next actions, and discuss issues that need input from all PSG members. The PSG group was involved in all main decisions made for the project including both technical and procedural decisions.

The agile methodology¹ based on Scrum² approach was used for the project management. The agile methodology suggests iterative development, where requirements and solutions evolve through collaboration between IT specialists and project stakeholders. This methodology is flexible and allows adapting to requirements changing easily. As mentioned before the trainee was mostly responsible for all project activities and was not part of the development team. Therefore, it was not suitable to follow scrum approach completely. The following practices from scrum approach were used:

- defining backlog with user stories
- defining two weeks sprint³ as a development unit
- estimating project using planning poker⁴ approach
- reprioritizing features

¹<http://agilemethodology.org/>

²<http://www.scrum.org/resources/what-is-scrum>

³<http://www.scrum.org/resources/what-is-a-sprint-in-scrum>

⁴<http://www.mountangoatsoftware.com/agile/planning-poker>

The project consists of two main parts. The first part focused on problem and domain study and requirements specification. The second part of the project consists of the design and implementation based on the outcomes of the first part. The following sections describe the work breakdown structure of the project and elaborate the project planning.

12.2 Work Breakdown Structure (WBS)

Figure 12.1 shows the work-breakdown structure of the project. Two top-level packages are identified: domain and problem analysis, and design, implementation, and validation of the system. Each package is further decomposed to smaller packages. These packages are included in the project planning described in the next section.

The domain and problem analysis consists of three sub packages:

- **Domain study:** understand and analyze the healthcare domain, analyze use cases of data aggregation at a national level.
- **Data privacy research:** research and analyze existing data privacy preserving techniques, choose more appropriate techniques for our project.
- **Requirements elicitation:** collect, analyze, and formulate the requirements for the project.

The design, implementation, and validation consists of four sub packages:

- **System Design:** architecture and design of the Cooperative Database System (CDS).
- **System Implementation:** implementation of the CDS based on the design.
- **System Deployment:** deployment of the CDS on a cluster.
- **Validation and Verification:** validation and verification of the CDS based on real medical data.

12.3 Project Planning

Based on the breakdown structure defined in the previous section, a project plan was formulated (see Figure 12.2). The project was conducted according to the initial plan with only adjustments in the list of components that should be implemented in the scope of the project.

Initial list of components for the System Implementation phase includes the following components:

1. Set of data processing jobs serving ETL process.
2. CoopAdminDB implementation.
3. Integration with TTP.

However, due to the several factors, such as the time limitation, not accurate estimations (due to the lack of knowledge about big data technology stack), not ready integration from the TTP side, components 2 and 3 were left for future work.

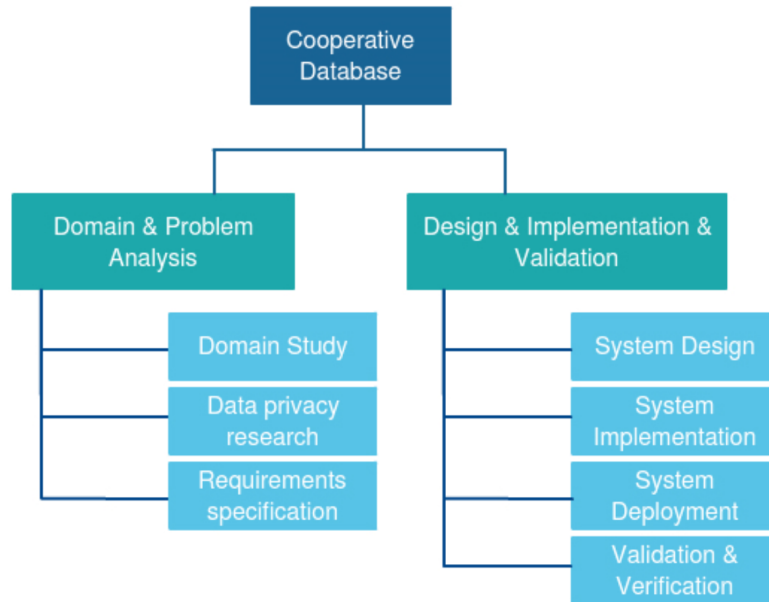


Figure 12.1: Work-breakdown structure of the project

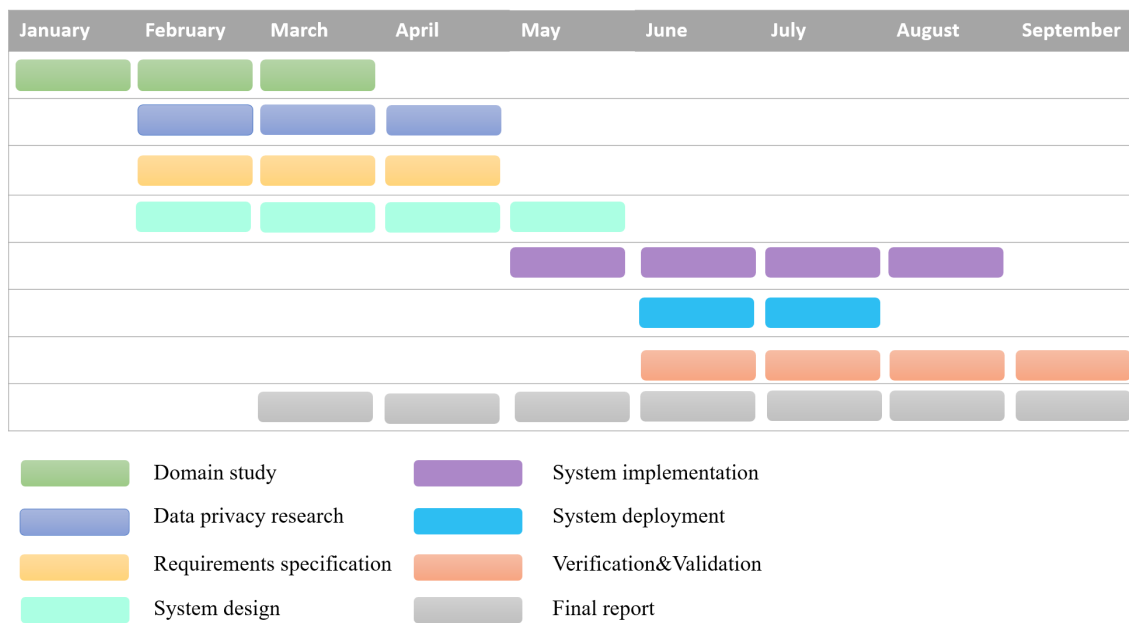


Figure 12.2: The project planning

13 Project Retrospective

This chapter provides the evaluation on the design criteria defined in Section 3.4 from the trainee perspective.

13.1 Design opportunities revisited

In the beginning of the project three design criteria that are important for the project were identified. Below for each of these criteria, it is verified how the system meets that criteria.

- **Functionality**
 - The system satisfies the requirements defined in Chapter 5 that are validated and verified in Chapter 10.
- **Ease of use**
 - The system is easy to use. In particular, it is easy to export data for medical research from the system to the familiar for the stakeholders data format. Hadoop and other frameworks used in the project support exporting data to widely used formats, such as CSV or relational database, by default. Thus, the export is performed in one command.
 - The system is easy to configure and deploy. The cluster for the solution is already configured by Amazon. Further execution of any data processing jobs is performed with one command.
- **Maintainability**
 - The system uses data-driven approach which makes the system design modular. Modularity makes the system easier to test, maintain, detect errors, and extend with new functionality in the future.
 - The system provides logging functionality, which means all main steps or exceptions that are triggered are logged. Therefore, if an error occurred the logs can be used to find out what is the problem.
- **Security**
 - The system design satisfies the security and privacy requirements defined in Chapter 5.
 - The system provides reports and metrics to measure anonymity level of datasets that should be released to the stakeholders.

- The design for secure communication between Population Databases and the Cooperative Database is accomplished within the project scope.

Bibliography

- [1] Article 29 Data Protection Working Party. Opinion 05/2014 on anonymisation techniques. http://cnpd.public.lu/fr/publications/groupe-art29/wp216_en.pdf, 2014.
- [2] Al-Riyami A. Health researchers and policy makers: a need to strengthen relationship. *Oman Medical Journal*, 25(4):251–252, 2010.
- [3] R. Sandhu, Coyne E.J., Feinstein, H.L., and Youman C.E. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [4] Regulation (EU) 2016/679. <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679>, 2016.
- [5] The personal data protection act. http://www.akd.nl/t/Documents/17-03-2016_ENG_Wet-bescherming-persoonsgegevens.pdf, 2016.
- [6] Green B., Cunningham G., Ekblaw A., Kominers P., Linzer A., and Crawford S. Open data privacy. Technical report, Berkman Klein Center for Internet and Society Research, 2017.
- [7] Narayanan A. and Shmatikov V. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy*, 2009.
- [8] de Montjoye Y.-A., Hidalgo C., Verleysen M., and Blondel V. Unique in the crowd: The privacy bounds of human mobility. *Nature*, (1376), 2013.
- [9] Sweeney L. Simple demographics often identify people uniquely. Data privacy working paper 3, Carnegie Mellon University, 2000.
- [10] van Hee K. and van Overveld K. New criteria for assessing a technological design. http://www.4tu.nl/sai/en/testimonials/2012-12-10_2012_April_NewCriteriaSAI.pdf, 2012.
- [11] Dwork C. Differential privacy. *Automata, Languages and Programming. Lecture Notes in Computer Science*, 4052, 2006.
- [12] Machanavajjhala A., Gehrke J., Kifer D., and Venkatasubramanian M. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering*, 2006.
- [13] Statistics Netherlands, Statistics Canada, Germany FSO, and University of Manchester. Glossary of statistical disclosure control, incorporated in paper presented at joint unece/eurostat work session on statistical data confidentiality, 2005.

- [14] Sweeney L. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.
- [15] DSDM Consortium. Dsdm atern handbook. moscow prioritization, 2014.
- [16] ISO SQuare Product Quality. http://maisqual.squoring.com/wiki/index.php/ISO/IEC_SQuaRE.

About the Author



Vera Chembay graduated with honors at the Faculty of Mechanics and Mathematics from the Novosibirsk State University, Russia, in 2012. Her final project was in the field of computational methods applied to geophysics. She worked as a mobile application developer for 1.5 years during her Master program. She has developed several mobile applications published on Google Play. After graduation, she worked in a bank software company “Sberbank Technologies” as a Java Enterprise developer. She participated in design and development of highly-loaded enterprise web applications for financial management of small business.

From September 2015 until September 2017, she worked at the Eindhoven University of Technology, as PDEng trainee in the Software Technology program from the 4TU.Stan Ackermans Institute. During her graduation project, she worked for Medworq on a project focused on creating a solution for aggregated medical data.

