Web-based visualization of guidelines and drug use in epilepsy : a project contributing to smart drug prescriptions

Citation for published version (APA):

Document status and date:
Published: 25/09/2015

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.

Download date: 18. May. 2019
Web-based visualization of guidelines and drug use in epilepsy

Trajche Masinov
September 2015
Web-based visualization of guidelines and drug use in epilepsy
A project contributing to smart drug prescriptions

Eindhoven University of Technology
Stan Ackermans Institute / Software Technology

Partners
Kempenhaeghe
Epilepsy Center Kempenhaeghe

Synergetics Benelux B.V.

Eindhoven University of Technology

ZonMW - The Netherlands Organization for Health Research and Development

Steering Group
Marian Majoie (Kempenhaeghe)
Johan Arends (Kempenhaeghe)
Luk Vervenne (Synergetics)
Jack van Wijk (TU/e)

Date
September 2015

Document status
Public

The design described in this report has been carried out in accordance with the TU/e Code of Scientific Conduct
Abstract  
Epilepsy is a chronic brain disease with unpredictable recurrence of seizures. It is not a curable disease, and the treatment for it is long and often difficult for the patient. Over the years, healthcare professionals have developed a set of guidelines for treating epilepsy. Improving patient’s care is of great importance to Kempenhaeghe. This report describes a project to analyze and interpret epilepsy-related information, and to design a system that provides valuable feedback for healthcare professionals and patients to improve treatment via the use of visualization techniques. A formal model of the epilepsy information and guidelines is developed, and coupled with a prototype. The prototype supports checking conformance of doctors’ prescriptions against the epilepsy guidelines, checking of future prescriptions for conformance and patient-specific applicability, and exploring patient’s care journey with epilepsy. This is supported with visualizations that provide the user with immediate data interpretation and feedback. The project lists also several suggestions for future improvements.

Keywords  
epilepsy, guidelines, visualization, DCM, HL7, three-tier architecture, conformance, decision-support, Kempenhaeghe, Synergetics, TUE, Software technology, PDEng

Preferred reference  

Partnership  
This project was supported by Eindhoven University of Technology, Kempenhaeghe, ZonMW and Synergetics Benelux B.V.

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, Kempenhaeghe, ZonMW or Synergetics Benelux B.V. The views and opinions of authors expressed herein do not necessarily state or reflect those of the Eindhoven University of Technology, Kempenhaeghe, ZonMW or Synergetics Benelux B.V, 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 © 2015, 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, Kempenhaeghe, ZonMW and Synergetics Benelux B.V.
Foreword

The project ‘Web-based visualization of guidelines and drug use in epilepsy’ is part of the new ‘Dutch Epilepsy Register’. In this project we try to provide a database that is able to give advanced feedback to its users, mainly doctors and patients, but also other parties such as health insurance and pharmaceutical companies. This internet-based system should also serve as an intermediate between many hospitals and (epilepsy) institutes and become the communicative backbone between patients, doctors and epilepsy-related organizations.

With this in mind, we asked and received a grant from ZonMW (program ‘Goed Geneesmiddelen Gebruik’, project number 836022001) and asked Trajche Masinov as PDEng to write the medical and visualization part. Apart from acquiring the medical data he also had to translate the Dutch and English Epilepsy Guidelines into logical units that were suited for use in this information system. Finally all the data had to be visualized in a transparent and attractive way. So he had to enter the domain of the doctors and health care regulations without leaving his own.

Now we can say that Trajche has succeeded very well in this mission. This is due to his very intelligent reasoning and tactical attitude towards the health care professionals without using too much the language of an information expert. He continuously translated both domains internally in his head during the discussions. Furthermore he showed much flexibility to adapt to doctors wishes that are not always logical. In the meantime he did not forget his roots as an informatics engineer. His knowledge of medical Dutch language and habits has grown a lot since last year.

Thanks to him we now have a very attractively looking prototype epilepsy program that allows us to give feedback to individual doctors and groups of doctors or other interested parties. The medical data also allow a comprehensive patient view, but since diary data still are lacking, a full feedback system for patients could not be written during Trajche’s presence with us. We are very grateful to Trajche for delivering such an important key element of the future Dutch Epilepsy register. He has done a great job.

prof. dr. Johan Arends
September 2015
Preface

This report summarizes the “Web-based visualization of guidelines and drug use in epilepsy” project executed by Trajche Masinov, as part of his graduation assignment for the PDEng (Professional Doctorate in Engineering) program offered by the Eindhoven University of Technology, Stan Ackermans Institute. The project was executed over the last nine months of the two-year program, under supervision of Synergetics Benelux B.V. and Kempenhaeghe.

The goal of the project is to exploit visualization techniques to improve epilepsy treatment through insight and decision support, and provide a solution based on this findings. This was possible with close collaboration from experts from Kempenhaeghe, and their technology partner, Synergetics Benelux B.V. A prototype was developed that

This report is primarily intended for readers with a technical background in different disciplines, such as visualization, healthcare technology standards and general software engineering. However, certain chapters may be interesting for non-technical readers, such as project managers and doctors.

This table lists the types of readers based on their background and chapters they need to focus on:

<table>
<thead>
<tr>
<th>Reader</th>
<th>Chapters</th>
</tr>
</thead>
</table>
| Healthcare technology specialist| Introduction  
Problem analysis  
Domain analysis  
Conclusions                                      |
| Visualization specialists       | Introduction  
Problem analysis  
User interface design  
Conclusions                                      |
| Software engineer               | System requirements  
System architecture  
System design  
Implementation  
Verification & validation                                      |
| Project managers                | Introduction  
Stakeholders analysis  
Problem analysis  
Project management  
Project retrospective                                      |
| Doctors                         | Introduction  
Problem analysis  
Conclusions                                      |

Trajche Masinov
September 2015
Acknowledgements

I owe the success of this project to the unprecedented help from all involved parties. This project was a great experience, and it gave me the opportunity to work with many wonderful people, which inspired and guided me throughout the project.

First of all, this project was made possible on behalf of the ZonMW, The Netherlands Organization for Health Research and Development. Via their funding, the Epilepsy Register project was initiated (project number 836022001), and my project, as a subproject of the epilepsy register.

I want to acknowledge the help from the specialists from Kempenhaeghe, especially Dr. Marian Majoie and Dr. Johan Arends. I would like to thank you both for your help and support throughout the project, with your knowledge and patience, as well as understanding. I really appreciate everything you did for me.

I would like to thank Luk Vervenne and Paul Imholz from Synergetics Benelux B.V. Your guidance and support was crucial for me, and I would not have make it without you. I enjoyed my internship with you, and the pleasant moments spent together. I would also like to thank you, William and Anneke Goosen, for your devoted cooperation on this project, and your support.

This project would not have been a success without the support from my university supervisor, Prof. Jack van Wijk. I thank you for your critical thinking, advises, and continuous support during this project. I learned a great deal about visualizations from you. I would also like to thank everyone involved in the PDEng programme, especially our programme director Ad Aerts. Thank you for everything, this programme is one of the best experiences of my life.

And last but not least, I would like to thank my family for their continuous support during this programme, and for believing in me. A special thank you goes to my girlfriend, Mimoza, for her patience and support. A special thank you to my mother, Gordana, for everything she has done for me, and to my brother Nenad for his support.

I would also like to thank my friends from the programme, for our shared experience and cooperation during these past two years.

Trajche Masinov
September 2015
Executive Summary

Epilepsy is a chronic brain disease with unpredictable recurrence of seizures. It is not a curable disease, and the treatment for it is long and often difficult for the patient. Like every other treatment, the epilepsy treatment is bound to rules and regulations based on scientific research and experience. Over the years, healthcare professionals have developed a set of guidelines for treating epilepsy.

Improving patient’s care is of great importance to Kempenhaeghe. But improving care is not always done by simply trying new approaches, it often involves looking back and learning from the past. Coupled with the guidelines, Kempenhaeghe aims to learn and derive new insights to improve the treatment process even further. This is the main focus of this project, to utilize existing information to produce insights that will guide caregivers in making better future decisions. The output of this project will be part of the future ‘Dutch Epilepsy Register’.

All these insights derived from the data represent nothing if they are not properly delegated to the end users. Proper delegation means effective and efficient presentation of information in such a way that users can understand it. To address this issue, the use of visualization techniques is chosen to facilitate the data representation.

This report describes a project to analyze and interpret epilepsy-related information, and to design a system that provides valuable feedback to healthcare professionals via the use of visualization techniques. As part of the domain analysis, a formalization of the epilepsy guidelines is created, that serves as a basis for the project. The formalization presents an extendible model for representing guideline information. Another product of the domain analysis is a design of an information model for representing epilepsy-related data by using healthcare technology standards, such as HL7 [4, 5].

As part of the project, a prototype is developed that serves as a decision support tool, intended primarily for healthcare professionals. The support tools provide checking conformance of doctors’ prescriptions against the epilepsy guidelines, checking of future prescriptions for conformance and patient-specific applicability, and exploring patient’s care journeys with epilepsy. The tools are supported by visualizations that provide the user with immediate data interpretation and feedback.

The system described in this report is based on a three-tier web-based architecture (presentation, logic, and data) that is easily maintainable, extendable, and scalable, with particular focus on usability, especially in visualizations and user interface design. The architecture defined supports tier replacements as long as the interfaces between tiers remain the same. Additionally, the system supports multiple presentation tiers due to its web-based architecture and deployment strategy.

The prototype shows that the use of visualization techniques is beneficial for data interpretation and insight extraction. It is recommended that the system is installed on a trial basis, and additional time and resources are spend for creating a production-level system.
# Table of Contents

Foreword ...................................................................................................................... i
Preface ......................................................................................................................... iii
Acknowledgements ...................................................................................................... v
Executive Summary ...................................................................................................... vii
Table of Contents ......................................................................................................... ix
List of Figures .............................................................................................................. xiii
List of Tables ............................................................................................................... xv

1. Introduction .............................................................................................................. 1  
   1.1 Context ............................................................................................................. 1  
   1.2 Rational pharmacotherapy ............................................................................. 1  
   1.3 National Epilepsy Register (NER) ................................................................. 2  
   1.4 Web-based visualization of guidelines and drug use in epilepsy .......... 2  
   1.5 Related work .................................................................................................. 3  
   1.6 Outline .......................................................................................................... 3  

2. Stakeholder Analysis .............................................................................................. 5  
   2.1 Kempenhaeghe .............................................................................................. 5  
   2.2 Eindhoven University of Technology (TU/e) .............................................. 5  
   2.3 Synergetics Benelux BV ............................................................................. 6  

3. Problem Analysis ................................................................................................... 7  
   3.1 Context .......................................................................................................... 7  
   3.2 Epilepsy treatment guidelines ................................................................. 7  
   3.3 Insight and decision support ................................................................. 8  
   3.4 Visualization ............................................................................................... 9  
   3.5 Design opportunities ........................................................................... 9  

4. Domain Analysis ................................................................................................... 11  
   4.1 Introduction ............................................................................................... 11  
   4.2 Health Level 7 (HL7) standard ............................................................... 11  
   4.3 Patient ................................................................................................... 13  
   4.4 Treatment ................................................................................................... 14  
   4.5 Detailed Clinical Models as building blocks .................................. 16  
   4.6 Epilepsy patients .................................................................................... 18
14.2 Work-Breakdown Structure (WBS) ........................................ 91
14.3 Project Planning .................................................................. 92
  14.3.1 Initial ........................................................................... 92
  14.3.2 Final ............................................................................ 93
14.4 Project execution ................................................................. 93

15. Project Retrospective .............................................................. 95
  15.1 Reflection ......................................................................... 95
  15.2 Design opportunities revisited ......................................... 95

Glossary ..................................................................................... 97

References ................................................................................ 99

Appendix A: Types of UML relationships................................. 101
Appendix B: Classification of epilepsy syndromes and seizure types .... 102
Appendix C: Use cases of the system ....................................... 105
Appendix D: The epilepsy guidelines ....................................... 109
Appendix E: Syndrome and seizure mapping ............................. 115

About the Author .................................................................... 119
List of Figures

Figure 1 Healthcare related projects submitted under the Rational Pharmacotherapy program .......................................................... 2
Figure 2 Types of blocks in a Detailed Clinical Model used to represent entity types ................................................................. 12
Figure 3 Detailed Clinical Model representing personal information about a patient ................................................................. 13
Figure 4 Two DCMs representing information about a patient's birth and death 14
Figure 5 A DCM expressing an implementation specification of a Personal Identification block for a patient ........................................ 14
Figure 6 A DCM describing a problem ........................................... 15
Figure 7 A DCM describing a procedure or treatment .............................. 15
Figure 8 DCM representing prescription information (part 1) ...................... 16
Figure 9 DCM representing prescription information (part 2) ...................... 16
Figure 10 Model representing patient treatment. The model is constructed by using separate DCMs .................................................. 17
Figure 11 Model that captures information about side effects and seizures for a patient ............................................................. 18
Figure 12 Proposed structure of a guideline for a single syndrome or seizure type ................................................................. 19
Figure 13 Model of the guidelines. The top section represents the abstract part and the lower one is the epilepsy specific ................................. 22
Figure 14 Structure of the NICE guidelines ........................................... 22
Figure 15 System level use cases for the project .................................... 29
Figure 16 Operating environment for the system .................................... 29
Figure 17 3-tier architecture deployment on separate physical servers .......... 36
Figure 18 Logic tier architecture .......................................................... 41
Figure 19 MVC pattern .................................................................. 43
Figure 20 MVVM pattern ................................................................ 43
Figure 21 Traditional page lifecycle vs. SPA lifecycle. The difference is in the second request to the server, where traditional web pages send a synchronous request, SPA send an asynchronous request (or Asynchronous JavaScript And XML or AJAX calls) to retrieve the data needed, usually in JSON format (image borrowed from https://msdn.microsoft.com/en-us/magazine/dn463786.aspx?f=255&MSPPError=2147217396 )................................. 44
Figure 22 Presentation tier architecture .............................................. 46
Figure 23 Design of a module, following the MVVM pattern ...................... 47
Figure 24 Service and component architecture ..................................... 51
Figure 25 The API section of the logic tier. An API consists of controllers that handle HTTP requests (GET, POST, PUT, and DELETE). The URI to a specific resource is constructed from the entry point to the service, the entry point of the controller, and the entry point to the resource .......................................................... 52
Figure 26 Provider, abstract and concrete service factory. The responsibility of the service factory is to instantiate concrete services that process the request... 52
Figure 27 Sequence diagram representing interaction between the client and the API ........................................................................ 55
Figure 28 Module design. Other modules follow the same style ............... 57
Figure 29 Sequence diagram representing interaction between the user and the client .................................................................. 58
Figure 30 Generic process view for handling a request. Three tiers are involved in the communication ................................................................ 58
Figure 31 Development view of the logic tier ........................................... 59
Figure 32 Development view of the presentation tier ................................. 60
Figure 33 Deployment view of the entire system. A more comprehensive view is offered in the Deployment chapter .................................... 61
Figure 34 Data model regarding patient-related information ...................... 63
Figure 35 Data model representing output information .............................. 65
Figure 36 Default page of the application (also referred to as home page) ........ 70
Figure 37 Template for the feature design. Only the input section is shown since it is common for all the features................................................................. 71
Figure 38 Design of the output/results section for the features .................. 71
Figure 39 Design of the Prescription check feature (input section) .......... 72
Figure 40 Design of the Prescription check feature (output section) ........... 72
Figure 41 Visualization of the conformance results. The top section gives the overall results and the table lists the population subdivision. Color coding of the conformance value is used to emphasize the level of conformance........ 74
Figure 42 Color palette for the conformance value .................................. 74
Figure 43 Visualization of the patient's history ....................................... 75
Figure 44 Line and stacked area chart .................................................... 76
Figure 45 Blood level measurement and tests performed .......................... 77
Figure 46 Timeline for exploring the patient care journey ....................... 78
Figure 47 Calendar for exploring the patient care journey ....................... 78
Figure 48 Adaptation of the interface based on the screen it is viewed on. The figure depicts the web site viewed from a smartphone ........................................ 79
Figure 49 Deployment view revisited. Concrete implementation specifics are taken into consideration ................................................................. 87
Figure 50 Single-machine solution. Everything is hosted on a single network server (not recommended) ............................................................... 88
Figure 51 Work-breakdown structure of the project ................................. 91
Figure 52 Types of UML relationships .................................................. 101
Figure 53 Sample of the syndrome classification (grouped by age) ............. 104
Figure 54 Use cases identified for the system ........................................ 105
List of Tables

Table 1 Kempenhaeghe's stakeholders................................................................. 5
Table 2 TU/e's stakeholders.................................................................................. 6
Table 3 Synergetics' stakeholders......................................................................... 6
Table 4 DCM datatypes.......................................................................................... 12
Table 5 Part of the epilepsy guidelines regarding treatment based on seizure type. .......................................................... 20
Table 6 Treatment process for focal seizures ....................................................... 21
Table 7 Treatment process for absence seizures.................................................... 21
Table 8 Main users and scenarios in the system...................................................... 28
Table 9 Functional requirements for the Conformance check feature................... 30
Table 10 Functional requirements for the Pre-prescription check feature .............. 31
Table 11 Functional requirements for the Advised vs. practical care journey feature .................................................................................................................. 32
Table 12 Functional requirements for the Guidelines management feature ........... 32
Table 13 Matrix of features and roles allowed to use a feature.............................. 66
Table 14 Survey to validate the application ............................................................ 82
Table 15 Initial project planning............................................................................. 92
Table 16 Final project planning............................................................................. 93
Table 17 Glossary .................................................................................................... 97
Table 18 Mapping of seizure types (from old ILAE and Luders classification to new ILAE classification) ................................................................. 115
Table 19 Syndrome mapping between old and new classification....................... 116
1. Introduction

This chapter introduces the project, its context, and the ecosystem where it lives. The scope of the project and its goals are briefly mentioned. The outline section of this chapter gives a brief overview of what is discussed in the following chapters.

1.1 Context

The “Web-based visualization of guidelines and drug use in epilepsy” project was conducted by Trajche Masinov, as part of his Professional Doctorate in Engineering (PDEng) program. The PDEng degree program in Software Technology is provided by the Department of Mathematics and Computer Science of Eindhoven University of Technology in the context of the 3TU.School for Technological Design, Stan Ackermans Institute.

The PDEng program is a two-year, third-cycle (doctorate-level) engineering degree program during which the trainee focuses on strengthening his/her technical and non-technical competencies related to the effective and efficient design and development of software for resource-constrained software-intensive systems, such as real-time embedded or distributed systems, in an industrial setting. In particular the focus is on large-scale project-based design and development of this kind of software.

The project was initiated by Kempenhaeghe, a leading center of medical expertise. They offer diagnosis and treatment to adults and children who suffer or are suspected to suffer from a complex form of epilepsy, a sleeping disorder, and/or neurological learning disabilities. Kempenhaeghe is involved in various software-related projects with the goal of improving patient care both for the patient and the caregiver.

This project is part of a bigger project called the “National Epilepsy Register (NER)” (project number 836022001) which is funded by ZonMW1, under the Rational Pharmacotherapy program. The description and context of that project is stated below.

1.2 Rational pharmacotherapy

The Rational Pharmacotherapy program (in Dutch, Goed Gebruik Geneesmiddelen) aims to make the use of medicines safer, more effective, and more efficient. Patients and care providers, in particular, stand to gain from the best possible use of the medicines available. The program encourages the gathering of reliable, independent data from practice. This information can be useful both for practitioners and for the government, health insurance companies, and the pharmaceutical industry. A number of project proposals have been submitted under this program, such as “Epilepsy Register,” and “Optimizing Drug Therapy in Patients with Epilepsy: An Adverse Drug Reaction Service Tool,” by Kempenhaeghe in collaboration with others (see Figure 1).

---

1 http://www.zonmw.nl/nl/over-zonmw/over-zonmw/
1.3 National Epilepsy Register (NER)

The National Epilepsy Register project is part of the Rational Pharmacotherapy program. It is established in collaboration with the TU/e, the Maastricht University Medical Center, Synergetics Benelux BV, the National Epilepsy Funds, the Knowledge Institute for Medical Specialists, and the Dutch Epilepsy Association or EVN (in Dutch, Epilepsie Vereniging Nederland).

The goal of the register will be to collect and centralize knowledge about the treatment of epilepsy, which will become available for patients and physicians. The registry will support sampling of data by patients, nurses, physicians, pharmacies and hospitals. The registry will be built around the patient. The patient maintains his own data, can use these for diverse applications and decides which healthcare providers and physicians are allowed to have entry to these personal and clinical data. The relevant juridical laws and guidelines are considered during the development of the registry.

Patients and healthcare providers may use the registry as an electronic patient dossier or personal health dossier. The registry will function as an epilepsy healthcare ecosystem.

1.4 Web-based visualization of guidelines and drug use in epilepsy

The project is a part of the NER project (see Figure 1). The idea behind the project is utilization of modern visualization techniques for big data analytics and exploration. The visualizations should provide visual feedback and interpretation of the data gathered by the NER to the stakeholders.

The focus of the project is on epilepsy treatment, as one of the core businesses of Kempenhaeghe. Treatment for epilepsy is defined through a set of guidelines. A guideline merely gives a suggestion about the direction of the treatment when a patient is diagnosed with epilepsy with a specific syndrome or seizure type. Doctors take these guidelines into consideration, but also, the prescribed medicine depends on factors related to the patients, such as age, gender, lifestyle, and current health conditions.

Epilepsy is treated with anti-epileptic medications, which can often cause side effects. Sometimes these side effects can cause severe damage to the patient’s health. With this in mind, the scope of the project extends to managing this information in improving a patient’s care.

The project aims to improve patient’s care by providing the following:

- Visualize the patient journeys during their epilepsy period and give valuable feedback to doctors;
• Provide a tool for the doctors that will allow them to check their issued prescriptions for adherence to the guidelines in a visual way;
• Provide decision support to doctors when prescribing an anti-epileptic medication to a patient based on the patient’s current status as well as the guidelines.

1.5 Related work
Similar work has been conducted in the area of visualizing the prescription behavior [1] [2]. The project was a master’s thesis conducted at the Technical University of Eindhoven. The main focus of the thesis was visualization of the doctor’s prescription behavior, for instance what and how much medication a doctor prescribed during a patient’s epilepsy treatment.

1.6 Outline
The next chapter introduces the stakeholders involved in this project as well as their interests and goal. A brief analysis is presented for each stakeholder. After the stakeholder analysis, a problem analysis is presented (Chapter 3) that depicts the problem of the project in detail and the expectations for the project itself. The story continues with a discussion about the domain of the problem (Chapter 4), its ecosystem, and limitations.

As with any other project, there are always certain risks or issues that may arise during the lifespan of the project. This is explained in Chapter 5 together with mitigation techniques for the issues/risks.

The next seven chapters address the solution to the problem in a structured manner. First, Chapter 6 presents the specific requirements derived from the problem and domain analysis. The functional and non-functional requirements specified in this chapter give a basis for designing the architecture of the system. The architecture of the system is specified in Chapter 7. Moving on, Chapter 8 presents the design of the system based on the architecture defined previously. The next chapter (Chapter 9) explains how the system is implemented following the previous architecture and design. This chapter also explains the decisions taken for using a specific technology for the implementation. Chapter 10 elaborates on the design of the user interface and visualization executed for the project. Chapter 11 discusses verification and validation, since every system needs to be verified and validated. This process ensures that the system meets the requirements and the expectations of the stakeholders. Finally, Chapter 12 depicts the complete deployment of the system as part of an ecosystem for healthcare related projects.

The results and future work for the project are presented in Chapter 13. This chapter also gives a conclusion about the project.

The last two chapters of this report reflect on the management part of the project. Chapter 14 addresses the management process during the nine months of the project as well as techniques used to manage it. Chapter 15 gives the author’s retrospective and reflection on the project.
2. Stakeholder Analysis

The previous chapter introduced the project and its position within a larger project, namely the NER. This chapter analyzes the stakeholders involved in the project and their interests and goals. There are three concerned parties in this project, each with its own interest, namely Kempenhaeghe, Synergetics Benelux BV, and the Eindhoven University of Technology. The following describes each party and introduces the main stakeholders and persons involved.

2.1 Kempenhaeghe

Kempenhaeghe is a leading center of medical expertise in epilepsy, sleeping disorder, and neurological learning disabilities. The members of this organization are the owner and initiator of the project; therefore this makes them the most important stakeholder in the project itself. They have several interests as their key drivers for the project.

The first interest is starting up the NER and making it a showcase product for the entire healthcare industry in The Netherlands. It is already an ongoing project as part of the Rational Pharmacotherapy program. The NER will also be a basis for development of a wide variety of healthcare related systems, and even in other areas (e.g. statistics, insurance).

The second interest is extracting useful information from their data on epilepsy treatment and using it to further improve the treatment itself. This is expressed through a need for tools that can help doctors directly to make better decisions when treating patients. These tools involve analytics and decision support.

As an addition, Kempenhaeghe is interested in including side-effects information registered by the patient themselves through a smartphone app called Eppy. This information can be used to improve patient’s care.

Table 1 lists the specific stakeholders from Kempenhaeghe.

Table 1 Kempenhaeghe’s stakeholders

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Position</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Dr. Johan Arends</td>
<td>Neurologist</td>
<td>Project owner</td>
</tr>
<tr>
<td>Dr. Marian Majoie</td>
<td>Neurologist</td>
<td>Project manager</td>
</tr>
</tbody>
</table>

Dr. Marian Majoie is the coordinator of the epilepsy guidelines’ authors, and her greatest concern is proper interpretation and representation of the guidelines in a certain system.

2.2 Eindhoven University of Technology (TU/e)

The Eindhoven University of Technology is responsible for the educational aspect of this project and fulfilling the requirements for a project of this type. That means certain standards need to be met. The TU/e is concerned with the design process, project management, and implementation.

The stakeholder from the university is prof. Jack van Wijk who has the role of a supervisor and a mentor for this project. He is a full-time professor in visualization at the Department of Mathematics and Computer Science and the group leader of the Visualization Group at Eindhoven University of Technology. Besides his mentorship, his role includes making sure that the design and documentation meet the standard of a PDEng project. Another stakeholder from the TU/e is Trajche Masinov, the PDEng candidate responsible for the implementation of the project.
Table 2 TU/e's stakeholders

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Dr. Jack van Wijk</td>
<td>University supervisor</td>
</tr>
<tr>
<td>Trajche Masinov, MSc.</td>
<td>PDEng candidate</td>
</tr>
</tbody>
</table>

2.3 Synergetics Benelux BV

Synergetics Benelux BV is a service-oriented company that focuses on personal data storage with enhanced security. Another focus is big data analytics and visualization. Synergetics tries to position itself as a utility company providing these personal data stores, coupled with analytics and visualizations, to customers or vendors.

In collaboration with Kempenhaeghe, they are directly involved in the NER project as providers of secure data storage capabilities.

With respect to this project, they have several interests. The first interest is starting the NER on their platform and making it a showcase project for the whole Dutch healthcare industry. The success of this project is clearly a major interest as it will put their name on the map as a healthcare utility company. The second interest of Synergetics is adhering to standards, especially health-related standards, such as HL7 (see section 4.2). This adherence concerns the design process where certain criteria need to be met with respect to software models and interoperability between systems.

Table 3 Synergetics' stakeholders

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Position</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luk Vervenne</td>
<td>Chief Executive Officer</td>
<td>Company supervisor</td>
</tr>
</tbody>
</table>
3. Problem Analysis

The first and second chapter discuss the position of this project in a greater ecosystem as well as the stakeholders with their interests and goals, respectively. This chapter focuses on the problem that the project is trying to solve. First, an introduction to epilepsy is given, followed by the guidelines for its treatment, and finally, reasons why visualizations are needed for this project are given.

3.1 Context

Epilepsy is a chronic brain disease with unpredictable recurrence of seizures. The seizures themselves are caused by sudden and temporary disruption of the electrical equilibrium in the brain. Seizures vary from person to person. An epileptic may fall down and go into shock with contraction of the muscles in the arms and legs. Another may feel strange tingling or hear strange noises, or stare somewhere for a short period of time.

The classification of epilepsy is based on seizure type and syndrome (see Appendix B: Classification of epilepsy syndromes and seizure types). The latest classification on seizure types distinguishes them based on the location of the brain which is affected, namely generalized, focal, and unknown. Generalized seizures affect both hemispheres of the brain, while focal seizures affect only one. The unknown seizures cannot be clearly diagnosed so they are considered unclassified until further information reveals their accurate diagnosis. Each seizure type has different effects on the patient, ranging from harmless to the immediate need of a healthcare professional.

Considering all of these seizures and syndromes, proper diagnosis and treatment significantly improves the patient’s quality of life by reducing the number of seizures. Epilepsy syndromes are grouped by age, starting from neonatal period, to infancy, childhood, adulthood, and some syndromes with characteristic finding or syndromes that cannot be placed in a specific group age.

The treatment for epilepsy is a variety of antiepileptic medications and sometimes psychoactive medications (benzodiazepine) that have sedative, hypnotic, and muscle relaxant properties. Due to the strong effect of the medications, side effects are common for epilepsy patients. Every patient reacts differently to different medications. The side effects that a medication can induce include drowsiness, tiredness, dizziness, or blurriness in vision. These are dose-dependent side effects. There are also non dose-dependent side effects, such as allergic reactions. The treatment usually consists of trying one or more medications to determine the best antiepileptic medications for the patient.

It is estimated that over 100,000 people suffer from epilepsy in the Netherlands. Epilepsy occurs relatively more often in the mentally retarded [3].

3.2 Epilepsy treatment guidelines

Like every other treatment, the epilepsy treatment is bound to rules and regulations on what should be administered to the patient and in what doses. In order to formalize the treatment, guidelines have been developed based on proven scientific research, best practices, and past experiences. The guidelines for epilepsy treatment merely suggest what kind of antiepileptic medication should be prescribed to a patient who suffers
from a specific epilepsy syndrome or experiences specific seizure type(s). The guidelines are a suggestion, because the current condition of the patient determines the proper course of action. Factors that need to be taken into consideration include: age, gender, pregnancy status (if the patient is a woman), allergies, and other ongoing medications.

Patients had been treated for epilepsy prior to the existence of the guidelines. Along the years, progress has been made in both science and medicine, which resulted in the invention of new or improved medications for epilepsy. These discoveries changed how epilepsy is treated, and as a result, the guidelines are ever changing.

Another factor that contributes to these changing guidelines is experience itself. Clinical trials as well as individual reports from professionals influence the guidelines and their content. During their career, healthcare professionals experience various cases of epilepsy and try out different treatments that sometimes lead to unexpected seizure freedom or patterns in treatment that improve a patient’s life. The treatment process is extensively explained in section 4.7.2.

### 3.3 Insight and decision support

As a leading center for expertise and treatment in the field of epilepsy, Kempenhaeghe has almost a hundred years of experience. Throughout these years, more than 30,000 patients have been treated and different treatments have been applied. Over the past forty years Kempenhaeghe has transformed into a specialist hospital (specializing in epilepsy, sleep medicine, and neurological learning disorders) for epilepsy patients.

Throughout these last forty years, in parallel with the evolution of the computer, Kempenhaeghe has gathered a vast quantity of data regarding epilepsy treatment. This data spans from patient information to epilepsy treatment, side effects and other healthcare-related information.

To further improve patients’ care and quality of life, Kempenhaeghe wants to make use of this data and extract meaningful information out of it. This information can be a basis for research and discovery of new insights, identification of patterns in treatment, extracting behavior of professionals, and much more.

Considering the guidelines on one side and the treatment information on the other side, Kempenhaeghe is interested in deriving three kinds of information from these data:

- Doctors’ prescription adherence to the guidelines;
- Decision support to doctors when treating a patient;
- Comparison of the advised care journey versus the patient’s actual care journey.

The first kind of information that Kempenhaeghe is interested in is checking the doctors’ prescription behavior for compliance to the guidelines. This means that a doctor can be evaluated whether his or her treatment for epilepsy is consistent to the guidelines. By addressing this, Kempenhaeghe can examine doctors’ behavior and identify patterns in prescriptions. These patterns can influence the guidelines themselves.

The second kind of information is support rather than insight. Using the guidelines as a basis, Kempenhaeghe wants to have a support tool both for the doctors and for the patients. The goal of this tool is to give suggestions based on the guidelines before the doctor decides to prescribe a medication to an epilepsy patient. This way, Kempenhaeghe wants to steer the direction of prescribing antiepileptic medications according to the guidelines. Important here is that the current condition of the patient is the driving factor for prescribing the right antiepileptic medication as guidelines do not cover every situation or every different patient condition.

The third kind of information is a comparison between two distinct timelines of a patient suffering from epilepsy. The first timeline is called advised care journey and it
represents a form of guidelines or recommended actions a patients must take from the moment the patient is diagnosed with epilepsy to the moment the patient is admitted to a specialized epilepsy center, such as Kempenhaeghe. The second timeline is called practical care journey and it consists of all events that transpired in the same time frame as the advised care journey. These events include visits to the general practitioner, visits to the neurologist, administration of various antiepileptic drugs, and seizure information. Comparison of these two journeys will give Kempenhaeghe a clear picture of what patients are experiencing and provide them with future action points for improving a patient’s quality of life.

### 3.4 Visualization

All these insights derived from the data represent nothing if they are not properly delegated to the end users. Proper delegation means effective and efficient presentation of information in such a way that users can understand it.

This is where visualization comes into play. Data visualization is a science that deals with visual representation of the data, which allows representation of information in a schematic form or uses graphical primitives. It is scientifically proven that the human brain consumes an image much faster than a table with raw data or plain text. Humans are creatures who establish an idea of a thing by its visual perception. They try to apply visual representations such as tables or images to process the increasing amount of information.

From the point of view of epilepsy, proper interpretation of acquired information or insight in a visual form is key to delegating the information to the end user and indirectly improving a patient’s quality of life. If data is properly visualized, the end user is able to comprehend this information and make better decisions for future treatments.

For that purpose, appropriate visualization techniques need to be carefully selected. These techniques need to be coupled with a user interface that must be easy to understand and use. The design of the user interface is an essential part of the project and it must be designed in such a manner that it can be understood by the users within seconds. The interaction between the user and the system needs to be made simple, because the system will be used mostly by people with no technical knowledge. Hence, a complex interaction and interface is out of the question.

### 3.5 Design opportunities

During the study of the epilepsy domain and problem analysis, a single design opportunities was identified, namely usability. The issues discussed in this chapter suggest a need for a system that can improve patient’s quality of life by helping doctors understand patient history and improve future decisions. Section 3.4 discusses visualization techniques as a way of conveying information to the user. Hence, usability should be carefully addressed by means of making the product easy to understand and use, and to be intuitive.
4. Domain Analysis

The problem analysis discussed in the previous chapter reveals the domain in which the project resides, namely healthcare with specialization in epilepsy. The objective of this chapter is to broaden the understanding of the domain and identify the relevant parts that will provide insight for the solution to the problem. This chapter addresses the domain of patients and their epilepsy treatment with anti-epileptic drugs as well as the known guidelines for optimal treatment.

4.1 Introduction

The healthcare domain is vast and has many different disciplines. Each discipline in healthcare represents its own domain with different approaches for handling patients, and treating them. Epilepsy is one of these disciplines. Residing under the neurological diseases discipline, epilepsy has evolved its own domain of knowledge about treatment and handling patients, starting from classification of the epilepsies by syndromes and seizure types, developing care plans for individual patients, to long term hospitalizations of patients with severe forms of epilepsy. Epilepsy is not a curable condition but controllable, which means that the treatment is a long and difficult process that can take many years, even the entire lifespan of the patient.

The analysis of the domain is split into sections that focus on different parts of the domain. Before the sections are explained, an introduction to the HL7 standard is given and its influence in the analysis of the domain is explained. The sections to follow explain the following parts:

- Patient – relevant information about patients;
- Treatment – treatment of patients and the process itself;
- Guidelines – recommendations regarding optimal treatment for epilepsy.

4.2 Health Level 7 (HL7) standard

Health Level Seven International (HL7) is a non-profit, ANSI-accredited standard development organization founded in 1987. The mission of this organization is empowering global health data interoperability by developing standards and enabling their adoption and implementation [4].

HL7 and its members provide a framework (and related standards) for the exchange, integration, sharing, and retrieval of electronic health information. These standards define how information is packaged and communicated from one party to another, setting the language, structure and data types required for seamless integration between systems. HL7 standards support clinical practice and the management, delivery, and evaluation of health services, and are recognized as the most commonly used in the world [5].

The above-mentioned packaging and exchange of information implies existence of a model that represents health related information. The model is a large collection of smaller models called Detailed Clinical Models or DCMs. A Detailed Clinical Model (DCM) is an information model of a discrete set of precise clinical knowledge that can be used in a variety of contexts. In some cases, a DCM offers technical implementation specifications.

The purpose of Detailed Clinical Models is to provide precise semantically consistent data and terminology specifications that are comparable and sharable between multiple care providers, health enterprises, and standards-based Healthcare Information Technology [6].

Examples of items described by DCMs are medicinal products (medication), patient treatment, and description of a problem or diagnosis.
For the purpose of this project, several DCMs are used to describe the domain itself and provide a structure for its representation. The models used are shown in the subsequent sections of the chapter.

Developing such models includes providing types for the entities in the model. Within the DCMs, a number of data types are recognized that are used to label attributes or entire model entities. Table 4 lists the data types used.

Table 4 DCM datatypes

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL</td>
<td>Boolean value</td>
</tr>
<tr>
<td>ED</td>
<td>Encapsulated data (binary information such as an image or a report in a procedure)</td>
</tr>
<tr>
<td>II</td>
<td>Instance identifier (identification of patients or medical products)</td>
</tr>
<tr>
<td>TS</td>
<td>Time stamp</td>
</tr>
<tr>
<td>CD</td>
<td>Coded data. Similar to enumeration. Values usually come from a defined coded dataset.</td>
</tr>
<tr>
<td>CO</td>
<td>Coded ordinal</td>
</tr>
<tr>
<td>INT</td>
<td>Integer value (number)</td>
</tr>
<tr>
<td>ST</td>
<td>String value (text)</td>
</tr>
<tr>
<td>PQ</td>
<td>Physical quantity. Expressed through unit and value. Units come from the unified code for units of measure, UCUM [7]</td>
</tr>
</tbody>
</table>

Throughout the DCMs, different colors are used for the blocks to indicate different types of blocks. Essentially, there are three types of blocks, shown in Figure 2.

Figure 2 Types of blocks in a Detailed Clinical Model used to represent entity types.

The rootconcept notation represents the main entity of the model. The notation is used to represent top level concepts, such as problem, medication, procedure, and person. The container notation is used to represent an entity that holds related information, for example, an address consisting of a street name, number of the house, postal code, city, and a country. Finally, the data notation is used to label entities representing atomic information such as raw text, numbers, Boolean values, and concepts that represent slightly more complex information. One example of this is an enumeration value where besides the value itself, information about the source or the code system used for the enumeration is stored in the entity.

All of these entities are connected with relations. A relation specifies a semantic link between two entities. For example, an entity containing address information has no added value on its own unless it is linked to another entity, such as a person. There are different types of relationships used to make up the semantics of a model. For a com-
plete overview of the available relationships, see Appendix A: Types of UML relationships. Typically, DCMs use composition relationships to build the model, starting from atomic information and going to containers that aggregate the atomic information, leading up to the root concept. A composition is depicted as a solid line connecting two entities with a black rhomboid placed on the beginning of the line on the entity that contains the other entity.

4.3 Patient

The bigger part of the healthcare domain revolves around patients. Throughout their lifespan, patients are typically involved in multiple domains with different specializations, such as general practitioners, and dentists. This adds to the complexity of representing the relevant data accurately and completely.

Patient related information is essential for effective and efficient treatment, especially in epilepsy. During epilepsy treatment, the patient can go through many physical and psychological (mental) changes that the caregiver should be aware of.

The HL7 organization has developed models for this in the form of small building blocks that can be combined. As explained in the section above, each block represents a piece of clinical information. With respect to the patient, these building blocks include personal information, medication use (dispensing), treatments or procedures, and problems or diagnoses.

The information about the patient such as personal, contact, and address data are bundled within a single detailed clinical model called Person. Figure 3 shows the detailed clinical model that captures personal information about a patient. Standard information is noted for a patient, such as birth data, name, and address information.

![Detailed Clinical Model representing personal information about a patient](image)

Figure 3 Detailed Clinical Model representing personal information about a patient

Regarding the birth and death information, the DCM goes into more detail by further specializing the birth and death information about a patient. Figure 4 shows these models.
The DCMs of the HL7 organization also provide specifications for the implementation of the models, regarding their representation within a system. Figure 5 illustrates such a specification. It depicts implementation details for the personal identification block of a patient shown in Figure 3. One can observe that personal identification of a patient is divided over several distinct identification systems, such as the Dutch BSN number and the local identification (within an institution). Other identification systems may also exist.

![Figure 5 A DCM expressing an implementation specification of a Personal Identification block for a patient](image)

**4.4 Treatment**

Treatment of patients is a complicated and often very long process. Different factors influence the setting of a treatment plan for a patient. Proper diagnosis of the problem is merely a starting point for determining a treatment plan. This is due to the variety of factors that a patient can have at that point in time, such as allergies, other conditions/problems, and childbearing period.

Description of a problem should be conducted such that it captures all relevant information about the problem. A DCM has been developed that models this information. As one can observe in Figure 6, a problem is distinguished by many types, each with its own level of complexity. Moreover, the description of a problem involves information about date of occurrence, its frequency, and clinical description.
A typical treatment, especially in epilepsy, involves prescribing medication to patients. A caregiver, usually the treating doctor, issues a prescription, which is realized by the patient in a pharmacy or in a healthcare institution, such as a hospital. The prescription representation is a separate DCM, which includes information about when and what
has been prescribed, by whom, and dosage information. Figure 8 and Figure 9 show the DCM that captures this information.

All of the above mentioned DCMs are used to define a larger model that describes a domain, in this case, the epilepsy domain. This is discussed in the following sections.

4.5 Detailed Clinical Models as building blocks

As explained in the HL7 introduction in the beginning of this chapter, the DCMs are merely building blocks, or specifications of structure for clinical data in healthcare systems and communication between them. A single DCM is not sufficient to model an entire system. For this reason, the DCMs are building blocks that can be combined to form a bigger model for a particular domain in healthcare.
Not all DCMs are available for the purposes of this project. Furthermore, DCMs are constantly being evaluated and changed. The models shown in Figure 10 are DCMs with sufficient level of maturity, provided by Synergetics.

The diagram in Figure 10 represents a model for patient treatment, designed by the candidate for the purposes of this project, and constructed solely of detailed clinical models. For simplicity, only the rootconcept elements are shown to model the domain, otherwise the diagram will be too large. Connections between DCMs are defined by logical relationships that represent real life associations. For example, the relationship between Person and Procedure marked as “treatments” represents the association of treatments related to the patient.

The model designed contains no specifics regarding epilepsy. This is one of the main advantages of using DCMs; models can be abstracted to the level where they are applicable in different domains. However, in many cases, including this one, specifics are inevitable, which is discussed further in this chapter.

For the specifics regarding epilepsy, sections 4.6 and 4.7 present models designed specifically to express the domain knowledge and problem definition of the project. The data models defined are not DCMs, they simply enrich the model defined in Figure 10 by filling in the gaps made by the missing DCMs (see section 5.2.1).
4.6  Epilepsy patients

As mentioned before, epilepsy treatment is based mostly on anti-epileptic medications. Doctors can prescribe one or many concurrent treatments with various medications that are often adjusted to fit the profile of the patient or based on the response of their administration. The reason for this adjustment is the strength of the medications themselves. Almost all of the anti-epileptic medications affect the brain cells, which often leads to other complications. This is essential information for the doctors since it gives them feedback on how a patient is reacting to a specific medication. It is quite often that patients experience side effects or adverse drug reactions (ADR) to anti-epileptic medications. These side effects include sleepiness, tiredness, and gaining weight.

Seizure-wise, epilepsy is manifested differently in every patient, and patients respond differently to medications. In some patients, the seizures are completely withdrawn or lowered to a minimum, whereas in others, the seizures recur often. Due to the complexity of the disease and the patient himself or herself, seizures can also be provoked by an anti-epileptic medication. This is why seizure information is essential. Based on this information, doctors can observe how patients react to a specific medication and adjust or stop a prescription.

Figure 11 presents the model designed that captures this information. Both seizure and side-effect information contain the date of occurrence and the severity of the event. This information can be related to the current treatment plan, which can give the doctor a better perspective of its effectiveness and possible next actions.

Figure 11  Model that captures information about side effects and seizures for a patient

4.7  Guidelines

Based on research and previous experiences, epilepsy specialists around the world have tried to formalize these experiences in the form of guidelines, which in fact are merely suggestions. The reason why these guidelines cannot be further formalized into rules is because of the different conditions of patients regarding their mental state, health state, lifestyle, age, and gender.
In the Netherlands, the Dutch Association for Neurology (in Dutch, Nederlandse Vereniging voor Neurologie) has been developing guidelines as a result of their past experiences and scientific proofs. This resulted in the publication of the document “Epilepsy: Guidelines for diagnosis and treatment” (in Dutch, “Epilepsie: Richtlijnen voor diagnostiek en behandeling”) [8] [9]. As the name suggests, the document elaborates on the diagnosis and treatment, especially the process and medications involved. Since epilepsy is classified by syndromes and seizure types, each of these groups has a separate subset of medications per syndrome and seizure type. Medications can appear as proposed treatment for many syndromes and/or seizure types because of their generic use, regarding epilepsy. Furthermore, patients can experience several syndromes during their lifetime, combined with various seizure types. In some cases, the use of antiepileptic medications is not efficient, and the patients still experience seizures. In these cases, patients can undergo epilepsy surgery, get admitted to specialized epilepsy centers (such as Kempenhaeghe) or have specialized epilepsy treatments. The document focuses on treatment for syndromes and seizure types encountered in the Netherlands, based on experience and research. For the rest, other guidelines are consulted (see section 4.7.4).

An analysis of this document gives a deeper understanding of the guidelines and a structure is identified that can help organize the guidelines. First of all, the guidelines are divided by syndromes and seizure types. Each syndrome is a form of epilepsy that is manifested through different types of seizures. The full classification is elaborated in Appendix B: Classification of epilepsy syndromes and seizure types. For each group, several sub-groups are defined that contain medications that are suggested or recommended for the group. Figure 12 presents the proposed structure for organizing the guidelines per syndrome or seizure type.

<table>
<thead>
<tr>
<th>Syndrome or seizure type</th>
<th>First choice</th>
<th>Second choice</th>
<th>Adjunctive</th>
<th>Do not use</th>
</tr>
</thead>
</table>

*Figure 12 Proposed structure of a guideline for a single syndrome or seizure type*

The structure reveals four groups, each one offering or prohibiting anti-epileptic medications that can be tried based on the seizure types or the syndrome that the patient is experiencing.

The meaning behind each is as follows:

- First choice – best choice medications;
- Second choice – second best choice medications;
- Adjunctive – combination of two anti-epileptic medications;
- Do not use – medication that should never be prescribed to a patient because they can cause side effects or even worsen the seizures.

Table 5 shows part of the guidelines. The full specification of the guidelines separates the guidelines into those based on seizure type, syndrome, status epilepticus for adults, and status epilepticus for children. Status epilepticus is a special seizure, whose duration is longer than five minutes.
Table 5 Part of the epilepsy guidelines regarding treatment based on seizure type.

<table>
<thead>
<tr>
<th>Absence seizures (Generalized seizures)</th>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First choice</td>
<td>Ethosuximide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Second choice</td>
<td>Lamotrigine</td>
<td>Not on pregnant women</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valproate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Adjunctive</td>
<td>1. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vigabatrin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Do not use</td>
<td>Phenobarbital</td>
<td>Do not use this medication in case of childhood absence epilepsy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
</tbody>
</table>

As mentioned earlier, treatment with antiepileptic medications is not always successful. For this reason, several high-level guidelines are defined that help healthcare professionals decide when it is time to try something else, such as surgery, or admittance to a specialized care center. Refer to Appendix D: The epilepsy guidelines for an overview of these guidelines. Because of their high-level definitions, some of the guidelines are difficult to formalize and operationalize. For example, one of the guidelines states: “Refer epilepsy patients to assess the possibility of specialized epilepsy treatments instead of surgery,” which cannot be made operational because it does not reveal any specifics. On the other hand, a guideline that states: “Consider referring the patient to specialized epilepsy care if the epilepsy debuts in the early years” is clear and easy to formalize, if sufficient patient information is available.

4.7.2. The treatment process

Epilepsy is a serious condition, which has proven to be quite resistant to anti-epileptic medications. As mentioned earlier, epilepsy is classified by syndromes, each one manifesting via one or more seizure types. The selection of treatment is chosen primarily by the seizure type, and secondary on syndrome. Some syndromes, however, determine the optimal choice for the anti-epileptic medications, or advise the medications that should not be used, irrespective of the seizure types. Additionally, factors such as the etiology of the syndrome or its time of onset influence the treatment process.

The treatment process differs for each syndrome and seizure type. On top of that, the complexity of the disease itself, combined with the effect of the medications and the state of the patient, makes the process more difficult. The existence of several groups or categories of medications allows doctors to try different approaches in different patients. The goal is to find the optimal treatment that will diminish or at least reduce the number of seizures to a minimum.

The typical treatment involves trying out medication(s) from the before mentioned groups (except the ‘Do not use’ one) based on the seizure types the patient is experiencing. It is often the case that patients still experience seizures after an anti-epileptic medication is administered, and for this reason, doctors try different approaches. These approaches include trying different medications, increasing or decreasing dosages, or starting a combination therapy with two anti-epileptic medications.

Given that the process is different per syndrome and seizure type, the following tables (Table 6 and Table 7) present the treatment process for two different seizure types, namely focal and absence seizures.
Table 6 Treatment process for focal seizures

<table>
<thead>
<tr>
<th>Focal seizures</th>
</tr>
</thead>
<tbody>
<tr>
<td>First attempt</td>
</tr>
<tr>
<td>Second attempt</td>
</tr>
<tr>
<td>Third attempt</td>
</tr>
</tbody>
</table>

Table 7 Treatment process for absence seizures

<table>
<thead>
<tr>
<th>Absence seizures</th>
</tr>
</thead>
<tbody>
<tr>
<td>First attempt</td>
</tr>
<tr>
<td>Second attempt</td>
</tr>
<tr>
<td>Third attempt</td>
</tr>
</tbody>
</table>

The complete overview of the guidelines together with the treatment process of all seizure types and syndromes is elaborated in Appendix D: The epilepsy guidelines.

Naturally, there are deviations from this process due to patient’s state. But the process itself reveals a sequence of actions for a treatment. This can be further abstracted as shown in the next section.

4.7.3. Guidelines model

To capture the model of the guidelines, one needs to take the divisions, groups, and the process into consideration. Looking at a more abstract level, the model or parts of it can also be reused for different domains. The process described in the previous section reveals sequences of actions that can be abstracted to match different healthcare domains. Figure 13 shows the model designed that captures the guidelines information. The elements at the top level of the diagram, Guideline and TreatmentStep, represent the abstraction that can be reused for different domains. As one can observe, a guideline refers to a problem, and the guideline contains a sequence of steps that describe the actions that need to be taken.

For the epilepsy domain, specifics need to be introduced. The bottom part refers to the specifics, by modeling the relevant parts that encapsulate the information about the guidelines. A special form of TreatmentStep is EpilepsyTreatmentStep that encapsulates additional information, such as what kind of seizure it is related, what medication to use, the category (based on the groups identified), and recommended use (based on the process identified) of the medication and its dosage, and the adjunctive medication and its dosage (for combination therapy).
Syndrome is a specialization of a problem. As mentioned earlier, syndrome is a form of epilepsy that is manifested through different types of seizures.

4.7.4. Other guidelines

An often consulted and referred to guideline by healthcare professionals in the field of epilepsy is the guideline defined by the National Institute for Health and Clinical Excellence (NICE) situated in the United Kingdom. We will refer to these guidelines as the NICE guidelines. In their publication, an extensive overview of epilepsy, its classification, diagnosis, treatments, and proof is provided. The use of these guidelines for the purpose of this project is to serve as a backup, when the information in the Dutch guidelines is not present (due to syndrome or seizure not previously encountered within the Dutch healthcare industry).

Compared to the Dutch guidelines, there are small but significant differences between the medication categories as well as the treatment process. Figure 14 shows the structure used in the NICE guidelines.

![Figure 13 Model of the guidelines. The top section represents the abstract part and the lower one is the epilepsy specific.](image)

Syndrome or seizure type

<table>
<thead>
<tr>
<th>First-line</th>
<th>Adjunctive</th>
<th>Other (on referral to tertiary care)</th>
<th>Do not offer</th>
</tr>
</thead>
</table>

![Figure 14 Structure of the NICE guidelines](image)

2 http://www.nice.org.uk/
The meaning of each category is as follows:

- **First-line** – same as ‘First choice’ medications in the Dutch guidelines, these are best choice medications;
- **Adjunctive** – same as ‘Adjunctive’ in the Dutch guidelines, combination of two anti-epileptic medications;
- **Other** – medications that may be considered on referral to tertiary care;
- **Do not offer** – same as ‘Do not use’ in the Dutch guidelines, medications that may worsen seizures.

The NICE guidelines also differ in the treatment process compared to the Dutch guidelines. They do not suggest first or second attempt, although the structure does give some indications on the order of treatment selection.

The correlation between the first-line and first choice medication in both guidelines suggest that the first and subsequent attempts be selected from the first-line medications. The adjunctive category is never a first attempt for an epilepsy treatment, therefore in the NICE guidelines, it is a second attempt and its subsequent attempts. Regarding the ‘Other’ category, according to the NICE guidelines, this category of medications should be considered when a patient is referred to a tertiary center, but it does not reveal how many attempts should be tried before referral. In these guidelines, a patient is advised for referral to a tertiary center if after two years, the epilepsy is not controlled. The ‘Do not offer’ category is the same as ‘Do not use’ in the Dutch guidelines.
5. Feasibility Analysis

After explaining the problem and the domain, a feasibility analysis is performed to identify the issues and risks that may arise or exist. This chapter covers the issues identified, coupled with the risks and mitigation strategies.

5.1 Issues and challenges

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

5.1.1. Detailed Clinical Models

Understanding detailed clinical models and how they can be used in the epilepsy domain is critical in the modeling step of the project. Given their different notations of expressing models and their complexity, DCMs add an extra level of complexity in the project.

5.2 Risks

Several risks are identified within this project. This section explains the risks identified and the corresponding mitigation techniques.

5.2.1. DCM availability

Because the DCM models are under development, their availability and level of maturity is an identified risk to the project. The process of designing a DCM takes a long time, because it involves consulting healthcare specialists of different domains, reading literature, and iterations until the model is ready. The goal of a DCM is to be as general as possible so it can be used in different domains.

In collaboration with experts from Synergetics Benelux BV, a good deal of the DCMs were available, some later than others. On the other hand, some of them were not ready or do not even exist. We assume that these models will arrive, but in order to keep the project going, models were designed that covered the missing information (see sections 4.5, 4.6, and 4.7).

5.2.2. Data variety and availability

Since this project involves working with sensitive data about patients and treatments, a rule was imposed that no personal data will be shared or made available for the needs of this project. Security and privacy issues impose rules about how the data can be used. This means that the user of the data should not be able to identify individuals in any way. These issues also affect the variety of data available for the project.

For this purpose, Kempenhaeghe already owns a historical data set of anonymous data that is used previously for research purposes. An analysis of this data revealed that these were sufficient for the purposes of this project. By consensus it was decided to use this data in the project. However, a limitation of this historical dataset is that it is not complete, especially for the purposes of the model defined in Chapter 4. The dataset contains the very minimum information provided to fulfill the needs of the project. This has a direct influence on the data models used for the system under design.

5.2.3. Formalization of the guidelines

As mentioned in section 4.7.1, some of the guidelines are informal and cannot be made operational. On top of that, the data made available for this project, restricts further...
formalization of other guidelines. This is an identified risk, since this part of the guidelines is still under development.

A decision has been made to formalize and put to use guidelines that can be used within the system and can show meaningful feedback based on the data available.

5.2.4. Platform unavailability

Since the inception of the project, it was decided that NER will be the data provider. This meant that the database setup and data provider is not a concern for this project.

Since the platform was under development and not available in due time for the purposes of this project, a mitigation strategy was imposed to migrate to an existing platform for data storage. A standard relational database was selected as a data provider for the project. The choice was set to MySQL, a relational database management system. MySQL is a popular choice of database for use in web applications.

5.2.5. Structural difference in the guidelines

Two different guidelines are used in the project, namely the Dutch and NICE epilepsy guidelines. As explained in sections 4.7.1 and 4.7.4, there is a structural difference in the categorization of the anti-epileptic medications, as well as the treatment process. This causes a problem in the representation of the guidelines within a model.

In collaboration with experts from Kempenhaeghe, an alignment has been reached that helps to fit the NICE guideline structure into the developed structure for the Dutch guidelines.

---

4 https://www.mysql.com/
6. System Requirements

After analysis of the domain and its problems, a set of requirements are extracted and formulated that have to be satisfied for this project. This chapter presents these requirements, both functional and non-functional ones.

6.1 Requirements gathering process

Both the problem and domain analysis reveal the core of the problem and its position in the healthcare ecosystem. But a simple problem statement is not enough to proceed with solving the problem itself. What is necessary is a problem decomposition into small and traceable sub-problems that address a specific issue. This decomposition leads to the formalized requirements of the project.

The problem decomposition process is conducted by repeated discussions with relevant stakeholders and thorough analysis of the problem and domain. These repeated discussions reveal the fine details of the problem and its subcomponents, which further help to differentiate specific requirements for the project.

After a number of discussions, a set of requirements is defined. Each requirement has a priority assigned to it, which denotes the importance of that requirement. Three categories of priority are identified:

1. High – requirement must be fulfilled;
2. Medium – requirement would be nice to have;
3. Low – requirement is optional (if time permits).

6.2 Product perspective

As mentioned in the introduction, the project is part of a bigger project, namely the National Epilepsy Register. Previous work on a part of the subject has already been done (“Prescription visualization”, P. van der Corput, TU/e, 2013 [1]) in the area of prescription visualization over a historical dataset provided from Kempenhaeghe.

This project takes the visualization to the next level by focusing on the end user ease of use and decision support for doctors. The product will offer tools that help doctors provide better treatment for patients, as well as tools for checking conformance against a set of epilepsy treatment guidelines. This check will allow concerned parties to evaluate how doctors are treating epilepsy patients and will provide feedback for improving the guidelines themselves.

The stakeholders envisioned the product to be a Web-based application, available to both doctors and patients. A clear distinction must be made between the types of users regarding their actions within the application.

Another aspect is data anonymity. Stakeholders are concerned with disclosure of sensitive information, hence the data must be anonymized and used solely for research purposes within this project. This refers to patient and conformance information.

6.3 Main features and use cases

Three main features are identified from the requirements gathering process, namely conformance check, pre-prescription check and advised care journey.

The first feature, conformance check, deals with doctors’ adherence to the epilepsy treatment guidelines. This feature should help doctors identify their prescription patterns and their deviations from the rules. Also, doctors can compare themselves against a certain group of doctors, for example, from another institution.
The second feature, pre-prescription check, addresses the possibility of performing a check prior to the actual prescription. Exploring the patients’ medication history and seizures should help doctors to make better decisions before an actual prescription is made.

The third feature, advised vs. practical care journey, focuses on visualizing the patient journey with epilepsy and its comparison with a set of predefined rules. This feature should enable doctors to point out the similarities or differences of the advised versus the actual patient care journey.

The refinement of the requirements revealed also the main actors or users of the product. Table 8 lists the identified users and their main scenarios within the system. The presented scenarios in the table are expressed from an abstract point of view. Detailed versions are presented in section 6.5.

*Table 8 Main users and scenarios in the system*

<table>
<thead>
<tr>
<th>User</th>
<th>Main scenario</th>
</tr>
</thead>
</table>
| Doctor                | • Check his/her conformance to the guidelines  
                          • Check prescription for a patient  
                          • Inspect patient’s epilepsy treatment and journey throughout the years  
                          • Submit a request to view group data |
| Analyst               | • Check doctors’ conformance to the guidelines  
                          • Compare doctors’ and group’s conformance |
| Patient               | • Check prescription for him/her  
                          • View the epilepsy treatment guidelines |
| Guidelines manager    | • Maintain epilepsy treatment guidelines |
| Group administrator   | • Approve/deny a request to view group data  
                          • Delete a request to view group data |

Figure 15 shows the system level use cases, which are based on the identified main features and actors within the product. Actors are linked to different use cases they can perform within the system. The use cases marked orange represent the main use cases of the system.
Figure 15 System level use cases for the project

6.4 **Operating environment**

One of the main requirements for the system is that it should be easily accessible via the Web from everywhere. Figure 16 depicts the envisioned deployment of the system.

Figure 16 Operating environment for the system
6.5 Functional requirements

This section lists the high-level functional requirements identified after communication with the stakeholders. The requirements are grouped by feature and each of them has an assigned priority.

6.5.1. Feature 1: Conformance check

This feature should provide doctors with the ability to check their conformance against the epilepsy treatment guidelines. The basis for the check are the epilepsy guidelines, and the data that is checked are the doctor’s prescriptions. A prescription is conforming to the guidelines if the prescribed medication is in the list of recommended medications for the syndrome/seizure type that the patient is experiencing.

Conformance should be calculated on subsets of the doctor’s prescriptions based on data about patients, such as age, gender, and level of intelligence.

Visualization techniques should be used to convey the information to the end user.

The feature should also provide means for the doctors to compare their conformance to a group’s conformance of their choosing. However, viewing a group’s conformance results should only be visible after prior authorization by an administrator from the requesting group. Table 9 lists the specific functional requirements connected to this feature.

The output of this feature should provide the doctors with insight on their conformance to the guidelines and point out similarities or differences when compared to a group.

Table 9 Functional requirements for the Conformance check feature

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Priority</th>
<th>Depends on</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR_1.1</td>
<td>The system should enable the user to calculate his/her conformance to the guidelines, using his/her prescriptions as input</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_1.2</td>
<td>The system should visualize the results of the conformance check</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_1.3</td>
<td>The system should enable the user to compare his/her conformance to a group of his/her choosing</td>
<td>High</td>
<td>FR_1.1, FR_1.5</td>
</tr>
<tr>
<td>FR_1.4</td>
<td>The system should enable a specific user type (researcher) to check conformance of any doctor or a group</td>
<td>Medium</td>
<td>FR_1.1</td>
</tr>
<tr>
<td>FR_1.5</td>
<td>The system should enable the user (doctor) to submit a request to view a specific group’s data within a specific time period</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_1.6</td>
<td>The system should enable a specific user to approve, deny or delete a request to view group’s data</td>
<td>High</td>
<td>FR_1.5</td>
</tr>
<tr>
<td>FR_1.7</td>
<td>The system should allow only registered doctors to use this feature</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>
6.5.2. Feature 2: Pre-prescription check

This feature should provide the end user with decision support by means of prescription checking when prescribing medications to an epilepsy patient. Dosages should not be suggested, unless they are strictly defined and apply to every epilepsy patient. The information shown should provide the user with relevant feedback and suggestions about the prescription itself.

The prescription checking includes checking for conformance to the guidelines and checking for reported side effects to the prescribed medication. The results should be presented in a neutral way, meaning the doctors should not be directly advised on their next actions. Other possibilities or alternatives on a medication selection should be provided to the user, based on the experienced syndromes/seizure types of the patient.

The feature should provide a view into the patient history with medications (with their retention times, seizures, and level of medications in patient’s blood. A retention time is the longest period a medication has been prescribed to a patient.

In case information is missing from the Dutch guidelines, the NICE guidelines should be used as a backup. Table 10 lists the specific functional requirements connected to this feature.

The benefit from this feature is decision support for better medication prescription based on previous information about a patient.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Priority</th>
<th>Depends on</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR_2.1</td>
<td>The system should enable the user to check a prescription for compliance to the guidelines</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_2.2</td>
<td>The system should offer alternatives for other medications available for the same condition of the patient in question</td>
<td>High</td>
<td>FR_2.1</td>
</tr>
<tr>
<td>FR_2.3</td>
<td>The system should check for reported side effects based on the selected medication</td>
<td>High</td>
<td>FR_2.1</td>
</tr>
<tr>
<td>FR_2.4</td>
<td>The system should check for reported seizures when the patient had a treatment with the selected medication</td>
<td>Low</td>
<td>FR_2.1</td>
</tr>
<tr>
<td>FR_2.5</td>
<td>The system should visualize the medication and seizure history of the patient</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_2.6</td>
<td>The system should allow only registered doctors and patients to use this feature</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>
6.5.3. Feature 3: Advised vs. practical care journey

This feature should provide the end user with a tool for comparing the advised care journey versus the practical care journey. The patient’s practical care journey is represented by all the actual events that transpired during the patient’s epilepsy related care. The advised care journey is represented in the guidelines in a form of advises with dates referencing from the point of diagnosis of epilepsy to the patient.

Visualization should be used to show the two timelines.

The feature should provide a clear view of the similarities and differences in the journeys, which should give valuable feedback to both doctors and patients. Table 11 lists the specific functional requirements connected to this feature.

The benefit of the feature is a deeper understanding of the history of the patient through graphical representation of his events.

Table 11 Functional requirements for the Advised vs. practical care journey feature

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Priority</th>
<th>Depends on</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR_3.1</td>
<td>The system should enable the user to compare his/her practical care journey against the advised care journey</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>FR_3.2</td>
<td>The system should enable doctors to check a patient’s journey.</td>
<td>Medium</td>
<td>FR_3.1, FR_3.3</td>
</tr>
<tr>
<td>FR_3.3</td>
<td>The system should allow only registered doctors and patients to use this feature</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>

6.5.4. Feature 4: Guidelines management

This feature should provide end users with the ability to manage the data about the guidelines. This includes addition of new data, and removal or modification of existing data. This operation is to be performed by a special role in the system. Viewing the epilepsy treatment guidelines should be enabled for any user, even the ones not registered in the system.

Table 12 lists the specific functional requirements connected to this feature.

Table 12 Functional requirements for the Guidelines management feature

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Priority</th>
<th>Depends on</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR_4.1</td>
<td>The system should enable users with a specific role (administrative role) to make changes in the guidelines information</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>FR_4.2</td>
<td>The system should enable users to add new data, modify or delete existing data</td>
<td>High</td>
<td>FR_4.1</td>
</tr>
<tr>
<td>FR_4.3</td>
<td>The system should enable read-only access to the guideline information to all users to the system</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>
6.6  Non-functional requirements

Besides the functional requirements, a number of non-functional requirements are identified from the requirements gathering process.

6.6.1. Security requirements

The product is meant to be used both by healthcare professionals as well as patients. Each user should have its own profile and access credentials that are for personal use only.

Since the system is dealing with personal data, sharing this data must be controlled via access rights and policies for proper use. Trusted sharing however, can be made possible because the end user is in control and it is he/she that manages who can access their data.

On the other side, law and regulations will overrule the users’ full access and control. As an example, a doctor has the rights and obligations to access all healthcare related data of his/her patients.

6.6.2. Software quality attributes

A number of software quality attributes are identified that the system should satisfy.

Maintainability
The system should be easily maintainable, meaning that the system can undergo changes with a degree of ease.

Performance
The system should respond reasonably fast to user requests. In clinical use, the maximum response time is 3 seconds for any clinical operation. For complex procedures and calculations, information should be shown about the status of the current operation.

Extensibility
System should be both vertically and horizontally extendable. This means that it should be easy to add or modify features.

Usability
The system should satisfy the requirements of the owner and the stakeholders of the project in a way that the system is made usable, intuitive, and friendly.

Upon opening the system, user should immediately understand the interface and its basic navigation. After following a 1 hour course without any prior knowledge of the system, and basic knowledge of computers, the user should be enabled to perform all the features that the product offers.

Localization of the system is important, because the majority of the users speak the Dutch language.
7. System Architecture

The purpose of designing an architecture for a system is solving a problem statement formalized with system requirements. The previous chapter (Chapter 6) elaborates on both functional and non-functional requirements of the system. This chapter elaborates the architectural reasoning and design decisions made that resulted in a design of the system based on the identified requirements.

7.1 Architectural reasoning

A good starting point when designing an architecture is choosing a suitable architectural style or pattern. An architectural pattern is a set of principles that provide an abstract framework for solving frequently recurring problems. Formally, an architectural pattern is a description of element and relation types together with a set of constraints on how they may be used [10].

An architectural pattern gives the backbone of the design. The pattern is typically chosen based on the non-functional requirements. Every pattern exhibits certain quality attributes, but not all can fit the non-functional requirements of a project or are not needed.

For the purposes of this project, several key drivers steer the decision for an architectural pattern and later design of the system (ordered by importance):

1. High-level requirements;
2. Non-functional requirements;
3. Functional requirements.

One of the main requirements of the product (see section 6.2) is for it to be a web application available to end users, such as doctors and patients. Typically, web applications follow the client-server architectural style, where clients (web pages viewed in users’ browsers) make requests to servers (in many cases, server is a database with application logic represented as stored procedures). The client-server architectural style is also referred as 2-tier architectural style.

Although it is not an explicit requirement for this project, one of the main interests of Kempenhaeghe is setting up the NER as a centralized knowledge base. This database should serve as a basis for development of variety of healthcare-related systems. Putting this into perspective, the design of this system should be such that the database is decoupled from the specifics of the system under design.

Starting from a client-server architecture, this decoupling of the database imposes addition of a new tier and hence a move towards a 3-tier architecture, namely presentation, logic, and data tier. The 3-tier architecture is very common in today’s systems, due to the qualities it exhibits, such as maintainability, scalability, flexibility, and availability. The basic idea behind this architecture is to separate the functionality into segments or tiers that can be deployed (not necessarily) on different physical computing platforms.

If we reflect on the non-functional requirements in section 6.6, again the 3-tier architecture suggests a good fit. It adds qualities not envisioned initially, such as flexibility to manage and scale tiers individually and availability of tiers to other systems or components. This is complemented with the functional requirements where different features are identified, each with its own distinctive purpose.
One of the core principles behind the client-server model (the predecessor of the 3-tier model) is separation of concerns. By separation of the tiers’ concerns from each other, an independent evolution of each tier is allowed, as long as communication between them remains the same. This separation of concerns allows any tier to be changed or replaced without affecting the rest of the system as long as the communication interface remains the same.

The deployment of a typical 3-tier web architecture consists of a presentation tier (front-end web server that hosts the web application), logic tier (application server that hosts the business logic of the system), and data tier (database server that hosts the data itself).

The following sections focus on the logical segments of each tier and the reasoning behind the architectural style selected for each of them. The order is bottom-up, starting from the data tier, to the logic and presentation tier.

### 7.2 Data tier

The data tier is responsible for data storage and manipulation. The manipulation involves serving the upper tier by executing its requests. The requests involve retrieving, storing, modifying or deleting data. Sometimes complex processing is also done on this tier in the form of stored procedures for optimization.

The data tier typically consists of a physical computer (usually called server) that runs database management systems, which allow access to the data itself. These servers are high-end processing machines that can support many requests at the same time and heavy processing. Naturally, the NER resides in this tier.

### 7.3 Logic tier

The logic tier is responsible for coordinating the application, processing of commands, and enforcing logical decisions related to business rules for the application in question. The responsibility extends to moving and processing data between its two surrounding layers [11].

![3-tier architecture deployment on separate physical servers](image)
One of the key drivers for designing this tier is the requirement of making the NER the basis for development of other healthcare-related systems. Having not to expose the NER directly to these other systems, a structured and well-defined way of exposing the data is needed. Since the NER is in very early stages, many modifications are expected. With this in mind, this structured and well-defined way must be easily extendable, both vertically and horizontally. Vertical extensibility means that new features can be easily added, and horizontal extensibility means existing ones can be easily extended.

Besides extensibility, ease of modification and the rest of the identified non-functional requirements steer the design choice into a component-based architecture. The main idea behind a component-based architecture is decomposition of the design into individual functional components that expose well-defined communication interfaces. The benefit of using components is that they are reusable, replaceable, easily extendable, and independent [12].

Another driver to be considered in the design of this tier are the functional requirements. As explained in the functional requirements section, several features are identified, which are decoupled from each other. They only share the domain model regarding patient-related information. All features use data from this model to realize their intent. The feature decoupling is in line with using a component-based approach, where each feature represents its own component.

In the world of distributed network systems, a typical way of offering functionalities usable by multiple systems is through services, specifically web services. A web service is a method of communication between two devices over a network. For simplicity, consider a service as a functionality that can be reused for different purposes. One of its purposes is to serve the needs of the system under design.

The idea of the services to be usable by other systems rather than only the system under design imposes additional requirements. Such requirements are:

- Services should be independent of the clients that use them;
- Services should not be aware of the clients that use them;
- Services should provide a uniform interface.

Due to the variety of technologies nowadays, systems are built using different programming languages and different technologies. For the services to be independent and unaware of their clients, they need to exchange data in such a manner that is indifferent to the programming language used to build the client systems.

Such a way is achieved through exposing an Application Programming Interface (API), or in case of the system under design, a web API. A web API is a programmatic interface to a defined request-response message system, typically expressed with languages such as JavaScript Object Notation (JSON) or Extensible Markup Language (XML). These APIs are exposed via the web by using the Hyper Text Transfer Protocol, or HTTP (see section 7.3.2).

From a functional point of view, the web application (presentation tier) will request functionalities offered by the services or APIs (logic tier, or even service tier), which in turn will process the request and access the database (data tier) to retrieve the necessary information, if needed.

There is an architectural style that encourages development of scalable web services (APIs), uses well-known protocol for communication (HTTP), and promotes independence from clients that use them: the Representational State Transfer architectural style, or REST.

The following sections present the architectural styles and protocols that play a role in the architecture, and finally, section 7.3.4 presents the architecture of the logic tier.
7.3.1. Representational State Transfer (REST)

REST is a hybrid style derived from several network-based architectural styles, such as data flow styles (Pipe and Filter), replication styles (Cache), and hierarchical styles (Client-Server, Layers) combined with additional constraints that define a uniform connector interface. The World Wide Web itself represents the largest implementation of a system conforming to the REST style.

The REST style exhibits certain properties and imposes several constraints. Some of the architectural properties of REST are:
- Scalability (support for large number of components and interaction between them);
- Simplicity of interfaces;
- Modifiability to meet changing needs;
- Portability of components.

The properties of REST are realized by applying several constraints to the architecture. The formal constraints of REST are:
- **Client-Server** – separation of concerns is the principle behind this constraint. By separation of the user interface from the application or data logic, an independent evolution to components is allowed;
- **Stateless** – communication must be stateless in nature. Each request from client to server must contain all the information for the server to understand the request and cannot take advantage of any stored context on the server. This improves visibility (no need to get additional info about a request), reliability (easy recovery from partial failures), and scalability (state is not stored between request, less resource usage);
- **Cache** – data within a response should be explicitly labeled as cacheable or non-cacheable, for equivalent requests;
- **Uniform interface** – a distinguishable central feature of REST is a uniform interface between components. Implementations are decoupled from the services they provide. The trade-off is efficiency, uniform interfaces degrade efficiency, since information is transferred in a standardized form rather than format specific to an application’s needs. REST defines four interface constraints: identification of resources, manipulation of resources through representation, self-descriptive messages, and hypermedia as the engine of the application state;
- **Layered system** – encapsulate legacy services and protect new services from legacy clients. A disadvantage is that they add overhead and latency to data processing. The combination of a layered system and uniform interface resemble the pipe-and-filter architectural style;
- **Code-On-Demand (optional)** – REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This improves system extensibility but reduces visibility.

The key abstraction of REST is a resource. Any information that can be named can be a resource, such as documents, images, services, and other resources. REST uses a resource identifier to identify the particular resource involved in the interaction between components. These identifiers are called uniform resource identifiers (URI) [13].

7.3.2. Hypertext Transfer Protocol (HTTP)

HTTP is the foundation of data communication for the World Wide Web. This protocol is an application-level protocol for distributed, collaborative, hypermedia information systems. It is used to deliver data, such as images, services, and other resources. It provides a standardized way for computers to communicate with each other, based on the request-response sequence of messages. HTTP is characterized with several features:
- **Connectionless** – the client initiates a request and after the request is made, the client disconnects from the server and waits for response. After the request
is processed by the server, the server re-establishes connection with the client to send a response back:

- **Media independent** – Any type of data can be sent by HTTP as long as both the client and server know how to handle it;
- **Stateless** – As a consequence of the first feature, HTTP is a stateless protocol. The client and server know about each other only during a current request. After this, they both forget about each other.

A request from a client to a server contains an URI, the version of the protocol used, media type requested, information about the client and possibly some contents in the request. A response from the server to the client includes the protocol version, a success or error code, followed by the server information, metadata information, and request-related data [14].

### 7.3.3. RESTful APIs

Having explained the REST style, APIs, and HTTP, a Web service API that adhere to the REST architectural constraints is called a RESTful API. HTTP based RESTful APIs are defined with these aspects:

- Base URI, such as http://example.com/resources/;
- An Internet media type for the data. This is often JSON but can be any other valid Internet media type (e.g., XML, and images);
- Standard HTTP methods (e.g., GET, PUT, POST, or DELETE);
- Hypertext links to reference state;
- Hypertext links to reference related resources [15].

### 7.3.4. Towards an architecture

It is not uncommon to combine multiple architectural styles in one design. Such a case is the system under design, where component-based style is mixed with the REST style. The component-based design allows features to be independent, while REST allows for the web APIs to be independent of its clients.

Figure 18 shows the logic tier architecture. The functionalities of the system are exposed through the REST API, which uses the components of the system to process the requests.

The goal of this modular design is to separate the concerns of each component. The REST API is only concerned with accepting and acting upon a request. It delegates the processing to the component intended to perform it. As soon as the request is processed, the API component is responsible for replying to the request.

The application components handle the specific processing they are intended to perform. As discussed previously, each feature of the system is independent of each other, with the exception of sharing domain information and data access. As one can observe from the diagram, all of the application components use the Data Access and Domain component. The Pre-prescription check component is responsible for checking and controlling prescriptions before they are issued. The component gives feedback and suggestions based on the information provided in the prescription. The Conformance check component is responsible for processing doctors’ prescriptions and calculating their conformance to the epilepsy guidelines. It also involves conformance comparisons between doctor and a group. The Advised Care Journey component is responsible for determining the similarities or differences between the actual care journey of the patient and the advised one. The Guidelines component offers services related to the epilepsy guidelines, such as modifying, adding, or deleting data. The Benchmark Request component offers services for requesting conformance comparisons against other groups or organizations. It involves submitting, approving or denying a request, as well as deleting an existing request. Finally, the Analytics component is responsible for statistical analysis and insights creation regarding doctors’ prescription patterns.
Figure 18 Logic tier architecture
The application components do not depend on the REST API component, meaning they can be reused in different applications with different technologies.

The Domain component is a representation of the domain objects that are needed for the system under design. It involves generic and domain specific representations, such as patient, treatment, and epilepsy-related representation.

The Data access component is responsible for database access and data manipulation, mainly to retrieve, or issue commands to insert, modify, or delete data. Depending on the database technology used, most of the database management systems support execution of procedures in the database for optimization in processing large datasets. Chapter 8 dives into more detail for the API and the components.

7.4 Presentation tier

The presentation tier is responsible for presenting data to the user and usually allows data manipulation and entry. The two main types of user interface for this layer are traditional applications and Web-based applications.

Over the years, the World Wide Web has evolved dramatically. Traditional web applications involved transitioning from page to page, where each page is generated dynamically on a remote server and sent to the browser where it is rendered. The beginning of the 21st century spawned the term Web 2.0. The term describes World Wide Web sites that emphasize user-generated content, usability, and interoperability. Parallel to the World Wide Web, a lot of progress has been made in modern web development. A shift has been made towards thin client-side Web-based applications that contain only the user interface (UI) and the data to be displayed. All the data processing is done in the logic tier. Page rendering is transferred on the client-side as the browsers became more and more powerful.

Besides the requirement of a web-based application, one of the main non-functional requirements for the system under design is usability. According to an older definition (ISO9126-2001 standard, revised with ISO/IEC 25010:2011 standard), usability is the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions. The new definition states: degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [16].

Utilizing the advances made in modern web development, the choice of the architecture for the presentation tier should maximize the usability. The other requirements should not be excluded as well. Maintainability, performance, and scalability have to be taken into consideration.

7.4.1. Model View Controller (MVC)

One of the most famous and often used architectural patterns for developing user interfaces is the Model View Controller (MVC) pattern. It started as a framework developed by Trygve Reenskaug for the Smalltalk platform in the late 1970s. The pattern divides an interactive application into three components:

- Model – a nonvisual object that contains the core functionality and data;
- Views – display of the model in the user interface;
- Controllers – handler for user input and model manipulation.

The views and the controllers comprise the user interface. A mechanism that propagates changes ensures consistency between the user interface and the model.

The MVC pattern introduces two principal separations, the model from its presentation and the controller from the view. The first one is the fundamental one, because model and view have different concerns. The model represents business or application specific data, while the presentation concerns how this data is presented to the user and the interface mechanics. The pattern also allows for independent development of
presentations of the same model. This is useful when different users want different presentations. The presentation, however, depends on the model, but not vice versa [11].

![Figure 19 MVC pattern](image1)

### 7.4.2. Model View ViewModel (MVVM)

The MVVM architectural pattern is a derivative of MVC and it has been developed in the last decade. It is a well known user interface design pattern used in many applications. MVVM consists of the following parts:

- **Model** – same as the model part of MVC;
- **View** – same as the view part in MVC;
- **ViewModel** – handles UI input and is responsible for propagating changes either one or two way between the view and the model. It can also pass commands from the view to the model.

The main difference between MVC and MVVM is the view model’s responsibility. While controllers can serve more than one view, the view model is coupled only to one view. In fact, the view model is a very specific controller. Moreover, the view model is used as a container for data and commands that the view can use and not as an orchestrator like controllers in MVC [17].

![Figure 20 MVVM pattern](image2)
7.4.3. Single-Page Application (SPA)

Single-Page Application (SPA) is a term that is not new (appeared first around 2004), but has been getting a lot of attention in the past 5 years. A SPA is a web application or a web site that aims to provide a more fluid user experience such as one in desktop applications. The paradigm of SPA is that either all code is downloaded with a single page load or resources are loaded dynamically as they are requested (typically by user actions). The idea is that the page does not reload at any point in the process, nor the control is transferred to another page [18].

In traditional web applications, every time the application makes a request to the server, the server has to create the whole HTML page. This triggers a page refresh in the browser, which causes the page to be reloaded and the user experience the disappearing and appearing of a web page. In a SPA, however, after the first page loads, all interactions with the server are done through asynchronous calls that return only the data needed (usually in JSON format). The SPA uses this data to update the page dynamically, without reloading. This gives a more fluid and responsive experience to the user. From an architectural point of view, this enables the separation of the presentation tier and the logic tier. As mentioned previously, this separation allows independent evolution of each tier, which increases the maintainability and scalability of the system under design.

Figure 21 shows a comparison between a traditional page’s lifecycle and an SPA lifecycle.

![Figure 21 Traditional page lifecycle vs. SPA lifecycle. The difference is in the second request to the server, where traditional web pages send a synchronous request, SPA send an asynchronous request (or Asynchronous JavaScript And XML or AJAX calls) to retrieve the data needed, usually in JSON format (image borrowed from https://msdn.microsoft.com/en-us/magazine/dn463786.aspx?f=255&MSPPError=-2147217396)](image)

In a pure SPA, all the user interface interaction occurs on the client side. After the initial page load, the server acts purely as a service layer. The client just needs to know what requests it needs to send. It doesn’t care how the server implements things on the back end. With this isolation, the client and the server become independent. Both of them could be easily replaced as long as the API remains the same [19].

As the concept of SPA started to become more and more popular, frameworks have been developed that enable development of such SPAs. Almost every framework developed encourages using the MVC pattern for developing SPAs. The pattern serves as a foundation. The reasoning behind it is separation of concerns, where the designer
designs for reusability, maintainability, and testability. If properly designed, MVC provides a clean architecture that allows the SPA to be easily modified or scaled [17].

7.4.4. Towards an architecture
The reasoning behind the architecture is based on the user interface point of view. The system under design should maximize the usability by providing the best user experience. Usability by itself is defined with several other non-functional requirements, such as understandability, learnability, and attractiveness. Maximization of the usability means that the product is easy to learn and use, and is also attractive to the user.

Due to the nature of SPAs and their focus on user experience, the architecture for the presentation tier chosen is MVVM, particularly SPA designed using the MVVM pattern.

Figure 22 shows the high-level architecture of the presentation tier. Again, the component-based or modular style is followed to model the separate features. The architecture consists of six different components that cover the desired functionalities of the system under design and several other components for various purposes, such as communication with the service API that will provide the presentation tier with data, or visualization libraries to support the data visualization.
Figure 22 Presentation tier architecture
The six application-specific components follow the MVVM pattern, with each module having a separate view, view model, and model component. In this way, modularity is achieved on both component and sub-component level.

Chapter 8 ("System Design") elaborates on the specifics of the component design whereas Chapter 9 ("Implementation") dives even deeper with concrete frameworks used and implementation details.

![Diagram of a module following the MVVM pattern.](image)
8. System Design

The previous chapter, “System architecture,” shows the high-level view on the system and its major components. The logic and presentation tier have different responsibilities and therefore different designs are created. This chapter elaborates the two designs and the transition from high-level architecture to component design.

8.1 Introduction

The system design is explained by utilizing the “4+1” view for describing software-intensive systems architecture developed by Philippe Kruchten. The approach suggests using multiple, concurrent views to express different concerns separately to various stakeholders, such as end users, developers, and system engineers, and to handle separately the functional and non-functional requirements. The views in question are as follows:

- Logical view – object model and behavior of the design;
- Process view – concurrency and synchronization aspects of the design;
- Development view – static organization of the software in its development environment;
- Deployment view – software-to-hardware mappings and distribution aspects.

The fifth view of the “4+1” is the scenario view, which serves as a reference point for the decisions made for the design of the architecture [20]. The scenario view is presented through use cases (see section 6.3).

An additional view is introduced in order not to clutter the logical design. The view in question is the data (base) view and depicts the models of the domain, in this case patient, treatment and epilepsy-related information.

The following sections discuss the different views for the logic and presentation tier. The data tier is separately discussed at the end of the chapter.

8.2 Reference use case

Section 6.3 discusses the main features and use cases of the product. Figure 15 in section 6.3 depicts several main use cases and a number of auxiliary ones. The use cases represent the user stories in the system, or what the system is supposed to do in interaction with the user.

As mentioned earlier, several scenarios are identified, such as checking a prescription, checking for conformance, and viewing a patient’s journey. All of these scenarios involve participation of the end user. Furthermore, similarities between them exist. For example, most of the scenarios involve selecting a patient or a doctor. Next, an action is triggered to retrieve an answer. The request is processed, data is generated and returned back to the end user in the form of a visual information or feedback. With this, we can identify common steps that identify a use case. The notation used to describe the use case is:

- **User** – represents the end user, such as doctor or patient;
- **Client-side system** – represents the presentation tier, or the web application;
- **Server-side system** – represents the logic tier, or the services for the web application.
The steps are as follows:
1. User request to perform a feature;
2. Client-side system shows the interface;
3. User enters required information and initiates action;
4. Client-side system verifies information
   a. If entered information is incorrect, user is notified. No action is taken;
   b. If entered information is correct, scenario continues;
5. Client-side system sends request to Server-side system to perform the requested action;
6. Server-side system accepts and processes the request;
7. Server-side system generates the requested information and sends it back to the client-side system;
8. Client-side system receives the information and displays it to the user.

Refer to Appendix C: Use cases of the system for specific use cases and their concrete steps. The steps described in the appendix are at a more abstract level, as they should be, understandable to end users. The steps described in this section serve as a reference for the design of the system and reveal some technology-specific information. A typical use case should be technology independent.

8.3 Logical view

Both designs of the logic and presentation tier follow the component-based approach. Each component in the design has distinctive responsibilities, which makes it easier to be replaced or modified.

8.3.1. Logic tier

The architecture of the logic tier includes a combination of two architectural styles, namely REST and component-based. The REST style is used to design the request handling part and the component-based style to design the business or application logic. A total of six components are designed in the architecture. For sake of simplicity and understandability, the designs shown in the following figures concern one chosen component, namely Prescription checks. If the name Prescription checks is replaced with the name of the other components, such as Conformance check, Patient journey, and Guideline management, one obtains the design of the other components. Figure 24 depicts the logic tier architecture, starting from the REST service on the top and moving towards concrete components in the bottom.
The service part of the architecture (the API) is further modularized to offer services for each component it serves, namely the conformance check service for the conformance check component. This way, both the service and component can be easily maintainable. The API consists of several services, represented with the Application component and its controllers. A service for a component is divided into two parts, a Controller that handles the requests from various clients, and a ServiceProxy that processes the request.

Section 7.3.1 defined an API conforming to the REST style with several aspects, one of them being the Universal Resource Identifier, URI. These URIs are uniquely identifiable access points or endpoints for each service offered by the API. Each controller is identified by a segment of the URI it can handle. In Figure 24, this is shown through the Endpoint section of the concrete controller PrescriptionCheckController, and its endpoint value of "prescription." Combined with the entry point of the whole API (the endpoint value of Application being "/api"), every request with a URI that has a segment equal to "/api/prescription/" will be handled by the PrescriptionCheckController. Furthermore, each controller exposes concrete endpoints that clients can use to request their service.

Each controller has exactly one service proxy. As the name suggest, the service proxy delegates the request handling to the component it is connected to. This delegation helps to further separate the responsibilities of the REST service and the application logic, if one wishes to replace one of the two (with different implementations, for example). Figure 25 depicts the PrescriptionCheckController and its service proxy.
In order to hide details about implementation of the components, each proxy requests a ServiceFactory from a component called FactoryProvider. The role of this provider is to provide factory instances to the service, which in turn can produce concrete components that handle the requests. The REST service knows nothing about their implementation. This is controlled through a set of defined interfaces. Figure 26 shows the responsibilities of the provider and service factories.

Figure 26 Provider, abstract and concrete service factory. The responsibility of the service factory is to instantiate concrete services that process the request

The extra layer of abstractions between the service factories and components and concrete service factories and components is added for extensibility and maintainability purposes. If a different implementation of an existing component has to be made, a change has to be made only in the factory that instantiates that component. Moreover,
even a complete implementation of the application-specific component can be changed by simply changing the service factory that the provider supplies to the API.

As depicted in the architecture in Figure 18 each of the application-specific components uses the DataAccess component to perform some action with data from the database. The same level of abstraction is applied as in the other components, an abstract factory that provides its clients with interfaces for creating data access components, and concrete factories that implement these interfaces. The data access component is the one responsible for communication with the data tier and manipulating the data. The abstraction of the factory and the data access component makes the switch to a different component very easy. If the database is changed, and different technology is used, then only the data access component needs to be adjusted (or re-implemented). To the application-specific components, the data access is available through the provider.

Another component depicted in the architecture is the domain component. This component consists of the domain models representing system-specific information, in this case, patient, treatment, and epilepsy information. A more detailed view of the contents of this component is presented in the data view.

8.3.2. Logic tier behavior
Part of the logical view of the system is to describe its behavior, by means of sequences of communication between components. For this system, the behavior is expressed through a client-server communication via exchanging HTTP messages that contain the data needed for the client, or the web application. The sequence shown in the two tiers’ diagrams represent instances of the reference use case, described in section 8.2, for a particular module.

The behavior of the logic tier is represented with the request handling and response dispatching part of the client-server communication. Figure 27 shows a sequence diagram of an interaction between the client and the API.

The interaction starts with the controller receiving the request from the client. Based on the URL, a specific controller is chosen that handles the requested segment and the concrete service requested by the URL. The controller simply delegates the call to the service proxy. Since each service proxy knows which concrete service it needs to process the request, the factory provider is consulted to provide the proxy with a service factory instance. As depicted in the design of the logic view, Figure 24, the FactoryProvider is marked as singleton. This means that only one instance is created within the system, in this case, all requests use the single instance of the factory provider to create specific components. This also means that the factory instances for both the component and data access are persistent throughout the lifetime of the process. Since the factories have no state and their sole purpose is to create components, a single instance of them is sufficient. Through this instance, the proxy creates the concrete service it needs without knowing its implementation and delegates the request to the service.

If the service requires data from the data tier (usually it does) then again the factory provider is consulted for the data access factory instance. Through this instance, the service creates a data access component that will provide the service with the data needed. The communication starts by establishing a connection to the data tier. When the connection is established, the data access components requests the data needed. After it is received, it is returned to the service, the connection to the data tier is not needed anymore, and therefore it is closed. The service uses this data to process the request, and creates (if needed) a response. This response is returned via the service proxy back to the controller. Since the controller is responsible for accepting requests and dispatching responses, the response data needs to be converted to a standardized format before it is sent back to the client. In this case, the response data is converted to JSON format, and the response is sent back to the client.
Figure 27 Sequence diagram representing interaction between the client and the API
As one can observe from this interaction, no state information is requested or stored anywhere about the client that requests information. Also, the server does not use any context stored on the machine to handle the request. The service is completely stateless, it reacts to requests and issues responses.

8.3.3. Presentation tier

The architecture of the presentation tier is based upon the MVC design pattern. The pattern is very common in user interface development, because of its separation of concerns and ability to independently develop parts of the system, for example views and controllers. The choice of the design is influenced also by the SPA approach, which encourages the use of the MV* patterns (where the star can replace: C for controller, VM for view model, or P for presentation).

To further segregate the responsibilities, the design of the tier is made modular based on the features identified for the product. With this in mind, the design of the tier includes modules, such as conformance check, and pre-prescription check, which are designed following the MVVM pattern.

Figure 28 shows the design of the presentation tier. Again, the design shown, namely for Conformance check, represent a design of a single module. The rest of the modules follow the same approach, if the name is replaced with the other modules’ names, such as Prescription check, Patient journey, and Guideline management. As one can observe, a module consists of a view, a controller and a model. The view is responsible for the user interface, displaying data to the user, and user interaction. The controller is responsible for handling the interaction between the user and the view, hence the connection between them. Both the view and controller have references to the model. This is of course the nature of the MVC pattern itself. In this case, the model represents module-dependent information, specifically conformance check information.

The controller utilizes several other components for the purpose of handling the user-view interaction. The first component is the ApiConnector. As the name suggests, the component is responsible for communication with the logic tier, specifically the REST API. The connector needs to know the web address of the machine where the REST API is provided, or its Uniform Resource Locator (URL). Also, it needs to know the endpoints or the specific services that the API is offering in order to request them.

Another component that the controller is utilizing is the Visualizer. This component is responsible for visualizing the information, and providing the visualizations back to the view. The last component for the controller is the Translator. As the name suggests, the component is responsible for translating the view’s information into a specified language. Since the product is to be used by doctors and patients, understandability is extended to provide the end user with means to choose the preferred language for the product.
Chapter 9 elaborates on the concrete technologies and frameworks used to implement this design.

**8.3.4. Presentation tier behavior**

The behavior of the presentation tier is represented with the request part of the request-response sequence, and the later response handling.

*Figure 29* represents the sequence of actions in the presentation tier. The sequence starts with the user initiating an action on the view. The view reacts to this by invoking an appropriate action/event handler in the controller. In this case, and almost all the others, the controller needs data from the service. For this purpose, the controller asks the API connector to make the request for him. Additionally, the controller provides a reference to a callback function that should be invoked when a response arrives. When the control is transferred to the connector, an appropriate API endpoint is selected, which maps to the service requested initially by the user. The connector uses a component which dispatches requests to the API and waits for the response. As soon as the response arrives, the callback method is invoked, and control is transferred back to the controller. The controller performs the necessary actions, for example, updates the model. After that, the controller updates the view, and the end user can see the result of his action.

The response from the server is not immediate. Due to the client-server communication over the Internet, a delay is introduced which can vary depending on the request and its processing. This is depicted by the separate sequence of actions in the lower part of the diagram. The gap between sending the request and receiving the response is discussed in the section 8.4.
8.4 Process view

The process view is concerned with aspects regarding non-functional requirements, such as performance and availability. It mainly focuses on the runtime behavior of the system. Typically, the process view is explained through the use of activity diagrams.

Figure 30 depicts the process view of the system, showing a request-response sequence. The delay mentioned in the previous section is due to the processing that has to be done on the server side. Since the client does not wait for the response, there is a delay until the server side finishes the processing and send the response to the client.
8.5 Development view

The development view focuses on the actual module organization within the software development environment. This organization is expressed through packages or libraries organized in a way that they provide interfaces between them.

Figure 31 depicts the development view of the logic tier. The organization of the view is layered, so that each layer communicates with the layers next to it. The API is on the top of the diagram composed of two sub-packages, one for the controllers and one for the services.

The layer below the API package is represented by the core package. As the name suggests, it includes the core of the system, meaning the interfaces and key aspects of the design. The package consists of three sub-packages, namely Domain, Interfaces, and Data Access. The domain package contains the model regarding the epilepsy domain. The interfaces package defines the services provided to the API layer, as well as infrastructure elements, such as service factories and providers. The data access package contains the interface definition for accessing the data tier.

The last layer is the coreImpl package. The name stands for core implementation. This package contains the implementations of the interfaces defined in the core package, more specifically, the data access and concrete service implementation.
This organization allows concrete implementations to be replaced and even complete packages, which makes it easily maintainable.

The development view of the presentation tier is much simpler than the one of the logic tier. Figure 32 depicts this view. The tier is split into three distinct packages: Views which contains the user interface specifics; Controllers, which contains user interaction handling; and Factories, which contains utility components, such as connecting to the logic tier, visualizing data, and translating the interface.

8.6 Deployment view

The deployment view concerns the structure of the product after implementation regarding software to hardware mappings and distribution aspects. Given that the very core of the system is client-server communication, client being the end user browser and server being a machine that will host the service, a straightforward approach is followed when designing the deployment view of the system.

Figure 33 depicts the designed deployment view of the system. The client, which is the web application executing on the end user’s personal computer, makes requests via HTTP to the REST API, which is running on a remote machine accessible via the World Wide Web. And last, the service is communicating with the database server, which hosts the database that contains all data.

In order to elaborate the specifics for the deployment view, implementation aspects of the systems need to be taken into consideration. These aspects influence the semantics used in the deployment view. For this reasons, the deployment view is revisited in Chapter 12.
Figure 33 Deployment view of the entire system. A more comprehensive view is offered in the Deployment chapter.
8.7 **Data view**

The goal of the data view is to describe the structure of the data entities and their relationships within the system. The data view is applicable only if the system needs data structures for its implementation. All the features of the system rely on input data, such as patients and their treatments, seizure and syndrome information, and side effects information; as well as output data, such as prescription check results, and conformance check results. Output data can also be entities from the input data. For example, a conformance check about a doctor contains information about the doctor in question.

8.7.1. **Input model**

The following figure, Figure 34, represents the data model regarding information about patients necessary for the realization of the features described for the system under design. This model can be considered as the input model of the system under design. Another name is operational data.

Due to the constraints imposed and risks identified on the project, explained in sections 5.2.1, 5.2.2, and 5.2.4, the initial design of the model presented in Chapter 4 is not used for the purposes of this project. The designed model follows the HL7 standard for expressing clinical knowledge. The design shown in Figure 34 resembles the model designed using DCMs, but due to the restrictions on the variety of data, a simpler model is designed that fits the purposes of the project. Entities found in the domain model can also be found in this model, except that do not contain all the information defined in the DCMs.

The central entity of the model is *Patient*, and all other information is aggregated around it. A patient has information regarding his/her treatments combined with the dose changes that occurred, measurement of medication levels in the blood, reported seizures and side effects, and information about patient’s condition regarding types of seizures experienced and syndromes encountered. The treatment information is connected to specific medications and their active substances. Another entity that is of interest is the *Doctor* entity and its relationships towards treatments and measurements.

Every entity contains attributes that characterize it. For example, a treatment is characterized by the person it was assigned to, the start and end date of the treatment, the medication used, and the list of dose changes for that treatment.
Figure 34 Data model regarding patient-related information
8.7.2. Syndrome and seizure mapping

Figure 34 reveals two other entities in the model, namely SyndromeMapping and SeizureMapping. The role of these entities is to provide a uniform way of representing seizures and syndromes, through the use of the latest classification (and the one used in the guidelines). The reason why these entities are designed is to deal with the different classifications used for distinguishing epilepsy syndromes and seizure types (see Appendix B: Classification of epilepsy syndromes and seizure types). Since the provided dataset contains syndrome and seizure information from different classifications (ILAE, both old and new, and Luders [21]) and the guidelines follow the latest ILAE classification, mapping is necessary to achieve the desired goals.

The SeizureMapping entity provides translation from the different classifications used to the latest ILAE classification. Additionally, treatment for certain seizure types is identical with certain other seizures, thus besides the mapping, an option to specify which actual seizure type to use is allowed.

The SyndromeMapping entity offers similar functionality as the SeizureMapping entity. It maps different syndrome classification, but instead to syndrome classified with the latest ILAE classification, it maps them to seizure types. The reason behind this is the more comprehensive information about seizure types in the guidelines instead of syndromes.

The mapping data is not part of the dataset, but it is created in close collaboration with doctors from Kempenhaeghe. Refer to Appendix E: Syndrome and seizure mapping for a complete view of the mapping between seizure types and syndromes.

8.7.3. Output model

All this information is used to support the features provided by the system under design. Based on this information and the business logic defined behind the features, the system under design outputs certain information that is used in the presentation tier and shown to the end user. This output model is depicted in Figure 35.

Some segments of the model are reused from the model designed in the “Domain Analysis” chapter, specifically information related to epilepsy guidelines. This information is both basis for the logic behind the features, and output information to be shown to the end user. Other information includes conformance results consisting of general results, and population subdivision results based on the information stored for the patients. In this case, the population is subdivided by age, gender, and intellectual level. This is represented with the entities ConformanceResult and ConformanceStatistics.

One of the requirements defined for the system under design is that conformance comparison between a doctor and an organization is to be strictly within a time frame and approved by an administrator from the requested organization. The entity BenchmarkRequest is designed for this reason. It encapsulates the information needed to request, and approve or deny a request to view conformance data.
Figure 35 Data model representing output information
The User and Role entities represent basic user information and support the entire system by providing context of the end user within the system. The role distinguishes users by introducing different roles, such as Doctor, Patient, Group Administrator, and Guidelines Administrator. Based on these roles, certain actions are allowed to be performed within the system. The matrix shown in the table below, shows the actions associated with each role.

Table 13 Matrix of features and roles allowed to use a feature

<table>
<thead>
<tr>
<th>Feature</th>
<th>Epilepsy guidelines</th>
<th>Pre-prescription check</th>
<th>Conformance check</th>
<th>Advised care journey</th>
<th>Submit benchmark request</th>
<th>Control benchmark request</th>
<th>Modify guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doctor</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Group Administrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidelines Administrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Keep in mind that both Group Administrator and Guidelines Administrator are sub-roles of Doctor, meaning a Group Administrator is also a Doctor.

As mentioned, the model for the guidelines and their representation is the one presented in the domain analysis chapter. For more information, refer to section 4.7.3.

The AdvisedEvent entity is concerned with information regarding the Advised Care Journey feature of the system. The entity encapsulates event information about action that the patient should consider based on his/her history. These events include referral to epilepsy surgery, admittance to a specialized care center and other. For a complete overview refer to Appendix D: The epilepsy guidelines.

The last section of the output model refers to statistical information on medication usage. This is part of the conformance check feature, where besides the check, a comparison is offered on the prescription behavior of the doctor and the group, regarding first choice medication for an epilepsy syndrome.
9. Implementation

The previous two chapters discussed the system architecture and design. This chapter elaborates on the realization of the design, specifically the technologies chosen for the development.

9.1 Introduction

An implementation is provided as part of the project. The implementation is based on the architecture and design, elaborated in the corresponding chapters, and satisfies the functional requirements defined in the “System requirements” chapter. The implementation provided is in the form of a prototype, to demonstrate the feasibility of the project.

For the purpose of implementing the design, choices are made on which technologies to use. Due to the client-server nature of this system, several choices are made, one for each tier.

With the evolution of modern Web development, a number of new technologies and frameworks for development have emerged. Choosing a correct technology is both difficult and easy; difficult, because of the many options and their pros and cons, which some fit and some do not, for the purpose of the system under design; easy, because sometimes the choice of technology is dependent on other technologies used in the system under design. The choice can also be easy if the technology and framework one needs to use is the single one available or mature enough. For this project, some decisions are based on the requirements and some on the candidate’s personal preference.

9.2 Presentation tier

The type of user application for this system was decided when the requirements were gathered for this project. The stakeholders explicitly requested a Web application. This means that one decision is already made. The non-functional requirements steered the decision of what type of Web application is to be chosen. The main non-functional requirement being usability, the decision has to maximize this requirement. The choice of the technology is partially decided in the design process, especially when defining the system architecture. Section 7.4.3 elaborates on this latest trend of modern Web development, where these SPAs maximize the user experience through a single page view instead of the entire page being sent back and forward, observed by page flickering in traditional Web applications.

As mentioned, the evolution towards modern Web introduced frameworks for developing SPAs. These all rely on standard Web development technologies, such as Hyper Text Markup Language\(^5\) (HTML), JavaScript\(^6\) and Cascading Style Sheets\(^7\) (CSS).

The selection of the framework for developing an SPA is based on the framework’s popularity, online support, and the personal preference of the candidate. For reasons, AngularJS is selected as a framework for development of an SPA. AngularJS\(^8\) is an open-source web application framework developed by Google for developing SPAs. It provides a framework for development of client-side MVC and MVVM architectures.

---

\(^6\) [http://www.w3schools.com/js/](http://www.w3schools.com/js/)
\(^7\) [http://www.w3schools.com/css/css_intro.asp](http://www.w3schools.com/css/css_intro.asp)
\(^8\) [https://angularjs.org/](https://angularjs.org/)
The idea behind SPAs is that when a user visits the web page through his/her browser, the application itself is downloaded in the user’s machine and it is executed in the browser. For this reason, initially, the web application needs to be hosted on a network server that will stream the application to the clients, when they request it. Chapter 12 discusses this matter.

The design of the user interface is also a crucial part of the project. As explained in the requirements chapter (Chapter 6), the web application should be accessible from everywhere via the web. This means that users can use devices, such as personal computers, laptops, tablets, or smartphones, to access it. In order to provide a consistent design, a specific approach is taken, namely responsive web design (see section 10.5). One of the most famous and often used framework for developing responsive sites is Bootstrap. Chapter 10 reflects in detail about the user interface design.

9.2.1. Visualization
Visualization is a big part of the project. Through the use of visualization techniques, data needs to be presented to the user with high understandability and ease of use. Since the application is web-based, the visualization technologies need to be web-based also. There are many visualization libraries that are web-based and are mainly written using the JavaScript language. For the purposes of this project, a number of visualization libraries are evaluated and several are chosen. The libraries chosen for this project are:

- Highcharts - charting library written in pure JavaScript, offering an easy way of adding interactive charts to a web site or web application;
- TimelineJS - open-source tool that enables building of visually, rich, interactive timelines;
- FullCalendar - a jQuery plugin that provides a full-sized, event calendar.

Another framework for data visualization that is evaluated is D3. This is one, and probably the most powerful data visualization frameworks intended for web standards. The reason for not selecting this tool is the specifics for its usage, meaning that the learning curve is much larger than the other tools, in the given time period.

9.3 Logic tier
The logic tier is represented by a service that supports the Web application and the application logic modules. In order for the service to support different clients, REST style is followed in the architecture, and hence the service is a RESTful API. Again, there are multiple technologies and frameworks that support development of RESTful APIs. One requirement that needs to be supported by the technology chosen is the ability to connect to a database and manipulate data. The choice of the programming language is not difficult because almost all modern languages support it. Based on the candidate preference, Java is selected as a language for development of the API and the modules.

9.4 Data tier
Due to the risk explained in section 5.2.4 a decision had to be taken to implement and set up the data tier. Based on available technologies and platforms provided by stakeholders, a decision is made to use MySQL, a relational database management system, as the data platform and provider. MySQL is an open-source, free-to-use database management system, and it is widely used. This choice is a good fit until the NER is available.

9 http://getbootstrap.com/
10 http://www.highcharts.com/
11 http://timeline.knightlab.com/
12 http://fullcalendar.io/
13 https://jquery.com/
14 http://d3js.org/
10. User interface design

In the previous three chapters, the architecture, design, and implementation details are elaborated. This chapter elaborates on the design of the user interface and presents the visualizations executed as part of the project.

10.1 Introduction

Since the main focus of this project is utilization of visualization techniques to provide insights to the users, primarily doctors, a great amount of effort is spent in the design of the user interface and visualizations that are displayed to them. The requirement of creating a product that maximizes usability imposes additional challenges to the design.

The main driver behind the design process of the user interface is simplicity. Taking into account that the primary users of the product are healthcare professionals and patients, the assumption that the majority of users has little to none technical knowledge is valid. For this reason, simplicity in the design is key for a fast learning curve of the product and its attractiveness.

On the other hand, the design as such needs to be able to convey the information, and provide the feedback to the user as clearly as possible. Depending on the amount of information, it is not always easy to put everything in one screen. Even if it is possible, it does not mean that the user will understand it and extract meaningful information from it. Based on our cognitive abilities, we as humans, understand information better when it is depicted rather than written. Visualization of data helps us to grasp vast amount of data and extract information out of it.

Such is the case for this project. Due to the constantly increasing amount of data in the healthcare domain, visualization is often sought as a solution to battle this information overload. This chapter covers these issues by first addressing the web application design and later the visualizations executed. The design realized as part of this project is validated and verified continuously with the stakeholders. The next chapter (Chapter 11) discusses in details the process of validation and verification.

10.2 Web application

The advances made in today’s modern Web are astonishing. Section 9.2 in the implementation chapter introduces the tools used to build today’s web sites and applications. The interfaces are richer and more intuitive to use, and lately they are becoming responsive. The responsive design of the application is discussed in section 10.5. Regarding the simplicity, several guidelines are followed in the design of the interface. The guidelines are as follows:

- Interfaces should be self-explanatory;
- Short descriptions should be provided for each feature in the form of a manual;
- Users should be enabled to clearly identify what action they are executing;
- The number of steps for completing an action or scenario must be minimized.

The rest of the section shows the interface designed by following the above-mentioned guidelines. A number of screen shots are presented that show the concrete design created for the project.

Figure 36 shows the default page, or home page, of the application. This is what the user sees, when opening the web application from his browser. First thing to notice is the title of the project, with a link to its description. The bottom section shows the
features provided by the web application, combined with a short description and a link to enter and use the feature. In the top-right section of the figure, the possibility of changing the language of the application is provided, coupled with the ability to login to the application, either as a doctor or a patient.

Figure 36 Default page of the application (also referred to as home page)

The design is simple and straightforward. The following sections focus on the design of each feature. The design of each feature is based on a template derived from the guidelines mentioned above, both for the interface and interaction. The template consists of two parts:

- Input – Feature description and data input needed for the feature combined with commands to execute the action
- Output (Results) – Display of the results. This part is different for every feature.

Figure 37 shows the template for the design of the features. The input section differs in the type of data entered for every feature, but the template still holds. In the top part, the title of the feature is clearly presented, and immediately below it, a description of the feature followed by a short manual on how to use the interface are shown. The data input section is located below the description part. In this section the user enters feature-specific data in order to execute the functionality provided by the feature. Depending on the user logged in to the system, the data that is entered includes doctor, patient, medication, or dates. Utilizing rich user controls offered by HTML and JavaScript, the data input is facilitated through specific controls, such as drop-down lists, calendars for date selection, or text auto completion for searching in a big list. Below the data input controls, a set of actions are placed via buttons for the user to press. They initiate the actions offered by the feature.
When the user issues an action, a processing occurs, and the view changes. The input section is hidden, and the output section is shown. The design of the output section is shown in Figure 38. The majority of the features follow this design, but it mainly depends on the results that are displayed. The top part of the output section gives a summary of the input provided in the input section, coupled with actions (via buttons) that the user can perform. The bottom section is where the results of the feature are presented. Due to the amount of information that is usually needed for each feature, this section is split into subsections or tabs. The number of subsections may vary for each feature (typically, three subsections).

The following figures (Figure 39 and Figure 40) show concrete examples, or instantiations, of the template used for the design of the interface. The feature chosen for the concrete example is Prescription check. In the input section of Figure 39 the user has to select a patient, a medication (or substance), and optionally, a concrete brand of the medication. The button Check this prescription invokes the processing and results are displayed, as shown in Figure 40.
10.3 **User interaction**

As mentioned in the guidelines for designing the interface, the design focuses on minimizing the number of actions a user must perform in order to get the results. In the previous section, some of these actions are explained. In general, a total of three steps are required to perform an action or utilize a feature:
• Input data – user inputs data needed for the feature;
• Invoke action – user invokes the action provided by the feature;
• Explore results – user explores the results obtained by the action invocation.

10.4 Visualization
A number of visualization techniques are applied in order to interpret and present data to the user. The visualizations provide insight and decision support to the main three features of the product, namely Conformance check, Prescription check, and Patient care journey. Depending on the data that is visualized, the techniques range from elementary, such as tables with no interaction, to advanced, such as a timeline with multiple different datasets. The choice for elementary or simple visualizations is driven by the fact that this type of techniques suits best for small datasets where there is not much information to extract.

The techniques chosen for data visualization are validated and verified with the stakeholders with an iterative approach. This is explained in Chapter 11. The following sections discuss the visualizations applied to each of the three features.

10.4.1. Conformance check
The main goal of the feature is to calculate to what extent doctors are conforming to the epilepsy guidelines by inspecting their prescription dataset. Additionally, this calculation extends to subsets of data based on patient’s attributes, such as age, gender, and intellectual level. The output of the feature is primarily a percentage, showing how much a doctor is conforming to the guidelines. Added to that, conformance values of the subsets is provided.

An extension to the feature is comparison between the conformance value of the doctor and another organization. The idea is to provide enough data for the user to be able to extract meaningful information.

Figure 41 shows a conformance result of a comparison between a doctor and an organization. In the top section, the overall conformance value is presented, for both the doctor and the organization. The bottom section consists of several subsections, each one presenting different aspects. In Figure 41 the second subsection is shown which uses a table to plot the results. The table is split into sections equivalent to the population subdivision. Each section of the table contains two or more categories, and for each of them, a conformance value is presented. To emphasize and easily interpret this value, color coding is used. Depending on the value, a different background color is used for the table cell where the value is displayed.

When it comes to choosing which color palette to use, a decision towards something that is familiar to everyone has to be made. We as humans, have become accustomed to relate certain colors to certain situations. For example, red color means danger, yellow means warning, and green means ok or good. Hence, the color palette is based on these conventions. Figure 42 shows the colors used for coding the conformance value.

The goal of using a color palette is not to judge the doctors’ decisions, it simply relates to how close a doctor is conforming to the guidelines. As mentioned multiple times in this report, the choice of the medication is not based simply on what the guidelines suggest, it is mostly influenced by the condition of the patient.
10.4.2. Prescription check

This feature provides the user with a tool to check a prescription before it is issued. Prescription issuing is not a part of this system. The check includes: checking a prescription for conformance to the guidelines, and checking whether there are reported side effects to the medication selected in the prescription. Additionally, the feature provides alternatives to the selected medication, based on the guidelines. In order to further facilitate the decision process, the feature is extended with the possibility to explore patient history regarding events, such as prescribed medications, seizures reported, and medication levels measured in the blood. This is where visualization techniques are applied.

As with the other features, the output is split into several sections, one for the check, one for the alternatives, and one for exploring patient’s history. The remaining of this section focuses on the exploration of a patient’s history, as the most interesting part and it is where visualization is applied. Figure 43 shows the visualization of the patient history.
Figure 43 Visualization of the patient's history
The entire visualization is one big timeline representing different types of events. It is split into four different sections (top to bottom):

- Medication dosages – information about prescribed medication and their corresponding dose changes over the years;
- Blood level measurements – measurements on the medication level in patient’s blood;
- Seizure frequency – frequency of seizures grouped by month;
- Navigator – exploration of the timeline, zooming and panning.

The horizontal axis represents time, and the vertical axis is different for each section. This type of visualization allows for exploring different kinds of data with respect to time. The four sections are vertically aligned to match the time axis. Below the four sections there is a legend that color-codes the different medications used in the selected patient.

The first section visualizes the medications prescribed represented by their respective dose changes over time. Due to different stakeholders’ opinions on the expression of the dosage levels, a configuration is added that lets the user choose what kind of technique to use to visualize this information. Two different representations of dosage levels are offered: total, meaning the exact amount of medication prescribed (usually in milligrams), and normalized by the Defined Daily Dose (DDD) of the medication. The DDD is the assumed average maintenance dose per day for a drug used for its main indication in adults [22]. This normalization exists in order to compare prescribed dosages, where one medication may be prescribed in the range of thousands and other in the range of tens of milligrams. By normalizing the values, the user can see if the dosage prescribed is low or high, based on the medication’s DDD. The formula for normalizing the medication is simply dividing the actual prescribed dosage by the DDD of the medication.

Besides the dosage level configurations, stakeholders requested a configurable view of the dosage levels over time. Based on this, two techniques are offered: stacked area charts (as the one shown in Figure 43), and individual line charts. A line chart is a type of chart that displays information on a 2-dimensional system in the form of markers or data points that are connected with straight lines. The area chart is based on the line chart with the difference that the area below the line connecting two markers is typically painted, as an area. Stacked area charts are based on area charts, and as the name suggests, a data series is stacked on top of the previous one. Figure 44 explains the difference between the two visualization techniques, line charts and stacked area charts. The same dataset is visualized using the two techniques.

![Figure 44 Line and stacked area chart](image)

By using a stacked area chart a user can observe the total amount of medication that was prescribed in a particular instance of time. As shown in Figure 43, a small window is shown if the user hovers the mouse anywhere in the chart. This window shows the concrete brands of the medications prescribed and their respective dose at that point in time.
The second section visualizes the medication levels in the blood. These levels are simple data points or markers on the timeline that provide information on the tests performed on the patient to measure the medication level in the blood. Figure 45 shows the information window that is displayed when the user hovers with the mouse over one of the markers. The total level is defined by summing the individual tests performed for a medication at that point in time.

![Figure 45 Blood level measurement and tests performed](image)

The third section visualizes the seizure information and aims to provide feedback on the frequency of seizures based on the medications prescribed. The yellow block in the seizure graph represents that no seizures are recorded or there is no information about it in the data set. That does not mean that there were no seizures in that period. The seizure information is presented in the form of a column chart, where the height of the column is determined by the number of seizures the patient has had in that month.

The last section of the visualization is the navigator. This is part of the tool used to visualize the data and it facilitates in navigation in the timeline. The user can zoom in and out, and pan through the timeline. The line represented within the navigator represents the total dosage of all medication. This gives the user perspective when the view is zoomed in to explore what is before or after the current view, and keep in mind the overall picture.

Looking at the whole picture, this visualization allows the user to explore the patient history from different perspectives to support decisions.

### 10.4.3. Patient care journey

The initial idea behind this feature was to compare the patient’s actual care journey versus the advised one. Since the information in the advised care journey is not complete (see section 5.2.3), a decision was made to use only the ones that can be made operational (in this case, only one).

The guideline states that if a patient still experiences seizures after trying two or three antiepileptic medications, then the patient should be referred to evaluation for epilepsy surgery (see Appendix D: The epilepsy guidelines for the complete list of guidelines for the advised care journey). After consultation of the stakeholders, the application of the other guidelines was planned as future work when the guidelines are more mature.

With this in mind, this feature aims to create a story based on the events that transpired for a patient. Creating a story indirectly implies a form of a timeline. For this feature, two visualizations of the patient journey are offered, a timeline and a calendar. The timeline lets users jump backwards and forwards in the patient history, and the calendar is pretty self-descriptive. Both visualizations plot the same data, just in a different way.

Figure 46 shows the timeline visualization for exploring the patient’s events, such as treatments, and reported seizures. The visualization is in a form of a slide show. Users can navigate back and forth through the events (by clicking on the arrows on the side of the top part), and the central section displays information about the event. The bottom part is the timeline, where all the events are plotted and represented by markers in the form of a label spanning from the start to the end of the event. The user can zoom

---

in and out, and pan through the timeline. This visualization creates a story for the user, by providing a path of what the patient has been through during his epilepsy treatment.

The other type of visualization, the calendar, is simply a tool for the user to explore the events in a well-known manner. A calendar is something that everyone uses and is familiar with. For this reason, this simple visualization was chosen. Figure 47 shows the visualization.

Patient events are plotted on the calendar the same as the ones for the timeline. At the top part, a familiar set of controls are offered. These controls provide navigation
throughout the calendar. Additionally, the view of the calendar can be changed from the default month view to a week or day view. The user can get current medication information by selecting any date in the calendar (top section, left of the calendar), or information about a certain event when it is selected (bottom section, left of the calendar).

The use of a calendar facilitates the user with exploring a patient history in a familiar way and extracting information, such as how many seizures a patient has had, what is the time between seizures, do they occur right after a medication is prescribed or a specific dose is reached, and do the seizures stop after a medication is prescribed.

### 10.5 Responsive design

Responsive web design is a web design approach that aims to provide optimal viewing and interaction experience across a wide range of devices [23]. Hence the web application is accessible from everywhere, the design has to be such that the experience and interaction is natural based on the device used. Users today browse the Internet through their devices, such as personal computer, laptop, tablet, or smartphone, and these devices differ from each other in their screen size. Since the user accesses the site via a browser, an interface should adjust itself based on the right size that it will be shown, so the users have a smooth experience on the site.

A site developed using responsive web design principles adapts to the user’s needs and device capabilities. It does not mean simply rescaling to fit the content on the screen size, it also involves decisions about what will be shown first to the user, which may lead to different presentations and layout of the information [24].

Such an approach is taken for the design of the web application, as part of the deliverables of the project. As explained in section 9.2, a framework is used to facilitate a responsive application, namely Bootstrap. Bootstrap is a collection of tools and templates for developing mobile first user interfaces. It adopts the responsive web design principles, allowing adjustable interfaces based on the screen size. The use of this framework allowed the web application to follow the responsive design principles. Figure 48 shows the interface when viewed from a mobile device (smartphone).

![Figure 48 Adaptation of the interface based on the screen it is viewed on. The figure depicts the web site viewed from a smartphone](image)

79
The previous chapters explain the design and development of the project. In order to confirm that the right product is being built, verification and validation techniques need to be applied. This chapter discusses these techniques and explains the process of verification and validation.

11.1 Validation

Validation of a software product is the process of checking whether the product built satisfies the stakeholder’s requirements. As explained in Chapter 6, the goal of the project is to create a set of tools, for both healthcare professionals and patients, which will provide decision support and insight for improving epilepsy care.

Two aspects need to be taken into consideration in the validation process. First, does the product built satisfies the functional requirements; does it do what it says in the functional requirements? Second, does the product satisfy the non-functional requirements? For the functional requirement, the validation should address whether the product offers the features defined. As for the non-functional requirements, the main one being usability, the validation should address the user interface design and how easy it is to learn and use the product.

Following an iterative development approach (see chapter 14), the results were continuously validated by the stakeholders. At least once per month, a meeting was scheduled that included a progress report and a demo of the current version of the product. During the demo, feedback was provided on the completeness and correctness of the product under development.

In order to formalize the validation process, a survey is created and dispatched to a group of users (doctors) for assessment. This assessment includes providing the users with access to the web application and letting them use it for a while. After that certain questions need to be filled out in the survey that address the functionality and usefulness of the product.

11.1.1. Survey

The table below shows the survey created for validation purposes of the product. The questions address the user interface and functionality. Users can answer each question with an answer scaling from:

- 1 (bad) – Feature is not easy to use or don’t understand it / User interface is bad;
- 2 (not good) – Feature takes time to learn, can be made more intuitive / User interface is a bit difficult to use
- 3 (good) – Feature is fairly easy to use / User interface is easy enough to use
- 4 (very good) – Feature is very easy to use / User interface is easy to use
- 5 (excellent) – Feature exceeds expectations / User interface is very easy and intuitive

The time required for completing the survey is less than 5 minutes after spending about 30 minutes on the application. The survey and the results are shown in Table 14. The dots that mark the answers represent an average grade from all the answers collected from the survey.
Table 14 Survey to validate the application

<table>
<thead>
<tr>
<th>Question (mark the answer with an X)</th>
<th>1 (bad)</th>
<th>2 (not good)</th>
<th>3 (good)</th>
<th>4 (very good)</th>
<th>5 (excellent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1 (Conformance check)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy it is to use this feature?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the feature performs as is should?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How clear are the results displayed?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Did the results provide you with meaningful information?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 2 (Prescription check)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy it is to use this feature?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the feature performs as is should?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How clear are the results displayed?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Did the results provide you with meaningful information?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature 3 (Patient care journey)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy it is to use this feature?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the feature performs as is should?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How clear are the results displayed?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Did the results provide you with meaningful information?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy it is to learn the application?</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How did you like the overall design of the application?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How helpful is the application in your daily work?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results show that the features satisfy the stakeholders’ needs to a certain level, but there is still room for improvement. The average grade shows the difference of opinions and needs of each user, where some features are graded better than others. These improvements are laid out in Chapter 13, “Conclusions,” in section 13.2.

11.2 Verification

Verification of a software product is the process of evaluating whether the system is engineered properly. This means that verification should check how well the product is developed, tested, and documented. To ensure the quality of the product, verification is performed by software testing and code quality measurements.

Since the product is split into two parts, namely web application and web service, different testing is applied to them. In the web application, the user interface needs to be tested, as well as the code, and in the web service besides the code, the API needs to be tested.
11.2.1. Testing the user interface

Manual testing is conducted on the user interface. This revealed several bugs that were fixed, and identified some missing features. Some of them were implemented, some of them are left as future work (see section 13.2). A set of testing tools exist that can automate the testing of the user interface. The tools range from using languages such as JavaScript to write tests, to browser application or plugins that record the user actions on the screen and create tests from them to be executed later. Such tools include PhantomJS\(^\text{16}\), Selenium\(^\text{17}\), and Watir\(^\text{18}\). Due to lack of time, they were not used. Automated user interface testing is also listed as future work (see section 13.2).

11.2.2. Unit testing

Due to time constraints, unit testing is not performed on the system developed. This is identified as a future work for the project (see section 13.2).

\(^{16}\) http://phantomjs.org/
\(^{17}\) http://www.seleniumhq.org/
\(^{18}\) http://watir.com/
12. Deployment

Based on the architecture and design of the system, a prototype is implemented that satisfies the stakeholders’ requirements. This chapter elaborates on the specifics of the deployment considering the implementation choices regarding technologies and programming languages.

12.1 Deployment view

Based on the decisions for client and server implementation (described in Chapter 9), the development view defined in the “4+1” model is revisited to take these choices into consideration and provide a clearer picture of the entire system. Figure 49 depicts the revisited deployment view of the system.

Since the web application is executed on the end user’s personal computer it first needs to be provided or delivered to him/her. The easiest way to do this is through content delivery networks. Content Delivery Network (CDN) is a distributed system of servers with a sole purpose of providing content to end users with high availability and high performance. The use of CDN increases the availability of the application by distributing the load to multiple servers, instead of using a single server that hosts the application and all the end users accessing it to retrieve the application.

The application server that hosts the RESTful API is connected to the Internet and is publicly visible. The web application sends HTTP requests and the application server responds with HTTP response. In order to run the API and the services behind it, an environment that hosts a Java Virtual Machine is required.

Last, a database server that hosts the database is required for the deployment of the system. A management service is needed to handle connections to the database as well as management of the database itself. The JDBC acronym stands for Java Database Connector.
Figure 49 Deployment view revisited. Concrete implementation specifics are taken into consideration.
A single-machine solution is also possible, meaning, this single machine will distribute the web application on request, will host the RESTful API and the database itself. This, of course, breaks the maintainability and scalability of the entire system. First of all, changing the any part may result in stopping the complete system, whereas the first solution requires only a redirection to a backup or a secondary machine. Also, upgrading the hardware on the machine will most probably initiate a long setup process of the entire system. Changing the database, however, requires even more attention since this is the source of information. Still, there are mitigation techniques, such as database replication. Figure 50 depicts the single-machine solution.

Figure 50 Single-machine solution. Everything is hosted on a single network server (not recommended)
13. Conclusions

This chapter elaborates the results achieved by this project and the added value to the stakeholders. It also discusses future work related to the project.

13.1 Results

Epilepsy is not a curable disease, and the treatment for it is long and often difficult for the patient. Improving patient’s care is of great importance to Kempenhaeghe. Improving care involves trying new approaches, as well as looking back and learning from the past. That was the main focus of this project, to utilize existing information to produce insights that will guide caregivers in making better future decisions.

Utilizing past data to produce new insights involves analysis and interpretation. The requirements defined for the project included the analysis that had to be done in order to produce feedback. The main challenge was how to convey this feedback to the end users. By applying data visualization techniques, data can be made easy to read and effective to use. By visualizing data, one can identify patterns or grasp an enormous amount of data by simply looking at a visual representation of it. In healthcare, this is important, because often time is of the essence. Choosing the correct visualization technique is a difficult process, since preferences and needs of users are different. Some are prone to simpler views than others. This was a challenge during the project, and the reason behind it is that end users are often non-technical people with some to very little knowledge in data visualization. Conveying the proper message through data creates challenges that were addressed throughout this project.

The end result is a decision support tool for the doctors. All this is achieved with simple and easy to use design of the user interface. The visualization techniques chosen offer clear data representation, are easy to understand and use, and are configurable based on user preferences. Chapter 10, “User interface design” presents the results and visualizations executed for the project.

From the software architecture point of view, the designed system exhibits properties, such as modularity, maintainability, and scalability, which makes it easily extendible and usable by other systems.

13.2 Future work

This section discusses future possibilities or extensions for the product that were identified along the way as well as work that was not performed due to various constraints, such as time, or unavailability of knowledge, data, and technology (see section 5.2).

The possible extensions are as follows:

i. Currently, the list of medications used throughout the project is an internal list by Kempenhaeghe. This can be improved by retrieving a list of medications from the Royal Dutch Association for the Advancement of Pharmacy (in Dutch, Koninklijke Nederlandse Maatschappij ter bevordering der Pharmacie or KNMP). They provide a service to retrieve full information regarding a specific medicine and their database is regularly updated.

ii. A lot of information about seizures is stored in the form of free text in the doctors’ personal notes. This information can be processed using natural language processing techniques to complement the data about seizures. By doing this, data will be enriched and more accurate.
iii. As mentioned several times, the advised care journey guidelines are not at a mature level. This means that not all guidelines can be formalized and made operational, which caused a partial implementation of this feature. Further improvement of the guidelines is a future extension of the product. In addition, appropriate data need to be provided for this to work, just like the current project has strongly benefit from the historical dataset on medicine prescriptions that was available.

iv. In discussion with the stakeholders from Kempenhaeghe, it was revealed that some of the prescribed medications for the patients were actually ongoing medications that the admitting doctors logged in the system under their name, just to have information on what the patient is administering. With the provided dataset, there was no possibility to distinguish which medications are prescribed by them, and which were ongoing medications. This influences the conformance check feature, where these ongoing medications were part of the doctors’ prescriptions. Clearly, a richer dataset would eliminate this.

v. The model used to represent the domain knowledge was built partially using Detailed Clinical Models. A possible extension is to finish the model when the remaining DCMs are available and use this model as a data model. This way, the system will comply with HL7 standards.

vi. As soon as the NER is completed and started up, the prototype needs to be revised and adjusted so that eventually it can connect to the NER and use live patient data.

vii. Currently, the user management part of the prototype is developed pretty straightforward, in order to present the functionalities offered to different types of users. An extension would be to design and develop an extensive user management and authentication module that will control every action and aspect of the application. This is essential, since the site will be available from everywhere and data privacy and sharing is becoming a pressing issue.

viii. Due to time constraints, manual testing was conducted on the developed software. As a future work, proper unit testing has to be conducted using test frameworks for every aspect of the system, especially the API.

ix. The information in the guidelines is somewhat contradictory when it comes to advising medications for seizure types and syndromes. A patient may be diagnosed with a specific syndrome, and experience different seizure types. There are cases when a medication is forbidden for a seizure type, but it is advised as a first choice medication in the guidelines for a syndrome. This is not taken into consideration in the prescription and conformance check, since a mapping was created to align the different classifications used in the data from Kempenhaeghe. The system can be extended to take this into consideration and provide a more complete feedback to the user.
14. Project Management

This chapter elaborates on the management conducted throughout the project’s lifetime.

14.1 Introduction

The management of the project was conducted under the agile methodology\(^\text{19}\) by using the Scrum approach. The agile methodology suggests iterative and incremental development through so-called sprints. Sprint is a short time span of usually two weeks, where a deliverable needs to be produced. This methodology is applicable in different domains, but it was initially forged for software development as an alternative to the traditional waterfall approach.

The project consisted of two parts. The first part of the project focused on epilepsy domain study and analysis, model definitions, and formalization of the epilepsy guidelines. The second part of the project focused on design and implementation based on the outcomes of the first part.

The following sections dive into the details of the project, such as the work-breakdown structure, project plan and execution.

14.2 Work-Breakdown Structure (WBS)

The work-breakdown structure for the project is presented in Figure 51. Two top-level packages are identified to match the two parts discussed earlier. Each package is further decomposed to smaller packages. These packages are later on referenced in the project plan and execution section.

\(^\text{19}\) http://agilemethodology.org/
The Epilepsy domain analysis part consists of three packages:

- **Domain study** – research and study about epilepsy and treatment;
- **HL7** – research and study about the standard including DCMs, and building an epilepsy model using DCMs;
- **Epilepsy guidelines** – research and analysis on the epilepsy guidelines followed by a formalization and concise presentation.

The Design and implementation part consists of two packages, each with its own set of sub-packages. It includes:

- **Database** – setting up the data source for the product
  - **Data import** – importing historical data in the database;
- **Application** – design and implementation of a client-server application. The sub-packages defined under the Application package represent separate work packages for each feature. They include design and implementation of both client and server side software.

### 14.3 Project Planning

Based on the breakdown structure defined in the previous section, a project plan was formulated accordingly. As expected, the initial project planning devised in the beginning did not match the actual project execution due to the incremental approach. Several adjustments were introduced along the way, as the knowledge and understanding deepened. The next two sections show the initial and final version of the project plan.

#### 14.3.1. Initial

The initial version of the planning involved four major parts: the epilepsy domain analysis, for which the first three months were reserved; the design and implementation, for which five months were reserved; and preparation for the final presentation, one month. The reporting on the project is considered an ongoing task for the entire lifespan of the project. Table 15 shows a Gantt chart for the initial planning for the project.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Epilepsy domain analysis</td>
<td>06/01/2025</td>
<td>20/03/2025</td>
<td>50d</td>
</tr>
<tr>
<td>2</td>
<td>Domain study - Epilepsy and treatment</td>
<td>05/01/2025</td>
<td>30/01/2025</td>
<td>30d</td>
</tr>
<tr>
<td>3</td>
<td>Basics of sign platform</td>
<td>12/01/2025</td>
<td>06/02/2025</td>
<td>20d</td>
</tr>
<tr>
<td>4</td>
<td>Historical data import</td>
<td>12/01/2025</td>
<td>28/01/2025</td>
<td>30d</td>
</tr>
<tr>
<td>5</td>
<td>Dutch epilepsy guidelines - structure and model</td>
<td>29/02/2025</td>
<td>23/02/2025</td>
<td>45d</td>
</tr>
<tr>
<td>6</td>
<td>Design and implementation</td>
<td>09/03/2025</td>
<td>07/04/2025</td>
<td>90d</td>
</tr>
<tr>
<td>7</td>
<td>System architecture</td>
<td>06/04/2025</td>
<td>16/04/2025</td>
<td>10d</td>
</tr>
<tr>
<td>8</td>
<td>Backbone development (web + svc)</td>
<td>06/04/2025</td>
<td>17/04/2025</td>
<td>10d</td>
</tr>
<tr>
<td>9</td>
<td>Database design and implementation</td>
<td>20/04/2025</td>
<td>02/05/2025</td>
<td>10d</td>
</tr>
<tr>
<td>10</td>
<td>Feature 1: Conformance check for doctors</td>
<td>29/06/2025</td>
<td>01/07/2025</td>
<td>20d</td>
</tr>
<tr>
<td>11</td>
<td>Conformance checking analysis and design</td>
<td>01/06/2025</td>
<td>29/06/2025</td>
<td>31d</td>
</tr>
<tr>
<td>12</td>
<td>Feature 2: Pre-prescription check</td>
<td>04/06/2025</td>
<td>29/06/2025</td>
<td>20d</td>
</tr>
<tr>
<td>13</td>
<td>Feature 3: Advice on patient care journey</td>
<td>10/07/2025</td>
<td>07/08/2025</td>
<td>20d</td>
</tr>
<tr>
<td>14</td>
<td>Testing and validation</td>
<td>21/08/2025</td>
<td>07/09/2025</td>
<td>61d</td>
</tr>
<tr>
<td>15</td>
<td>Technical report - Writing and Receiving</td>
<td>26/09/2025</td>
<td>28/09/2025</td>
<td>155d</td>
</tr>
<tr>
<td>16</td>
<td>Final presentation preparation</td>
<td>29/09/2025</td>
<td>18/10/2025</td>
<td>30d</td>
</tr>
</tbody>
</table>
14.3.2. Final

After the first two months, several adjustments were introduced. The adjustments were reflection on the understanding of the project and refinement of the stakeholders’ requirements. New tasks were added, several modified or deleted. Table 16 shows a Gantt chart of the final project planning.

Table 16 Final project planning

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Epileps domain analysis</td>
<td>05/01/2015</td>
<td>03/04/2015</td>
<td>55d</td>
</tr>
<tr>
<td>2</td>
<td>Domain study - Epilepsy treatment</td>
<td>05/01/2015</td>
<td>30/01/2015</td>
<td>20d</td>
</tr>
<tr>
<td>3</td>
<td>Basics of Spine platform</td>
<td>12/01/2015</td>
<td>22/02/2015</td>
<td>20d</td>
</tr>
<tr>
<td>4</td>
<td>Historical data import</td>
<td>12/01/2015</td>
<td>22/01/2015</td>
<td>10d</td>
</tr>
<tr>
<td>5</td>
<td>Design and implementation</td>
<td>02/02/2015</td>
<td>03/02/2015</td>
<td>10d</td>
</tr>
<tr>
<td>6</td>
<td>System architecture</td>
<td>30/03/2015</td>
<td>03/04/2015</td>
<td>10d</td>
</tr>
<tr>
<td>7</td>
<td>Model parser</td>
<td>23/03/2015</td>
<td>12/03/2015</td>
<td>10d</td>
</tr>
<tr>
<td>8</td>
<td>Gulf Mockup design</td>
<td>06/04/2015</td>
<td>17/04/2015</td>
<td>20d</td>
</tr>
<tr>
<td>9</td>
<td>Database design and implementation</td>
<td>20/04/2015</td>
<td>18/04/2015</td>
<td>10d</td>
</tr>
<tr>
<td>10</td>
<td>Sp-ih database</td>
<td>20/04/2015</td>
<td>24/04/2015</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>MySQL database</td>
<td>27/04/2015</td>
<td>01/05/2015</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>Backbone development (client-server)</td>
<td>04/05/2015</td>
<td>08/05/2015</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>Feature 2: Pre-prescription plan</td>
<td>11/06/2015</td>
<td>28/06/2015</td>
<td>15d</td>
</tr>
<tr>
<td>14</td>
<td>Benchmark request</td>
<td>06/06/2015</td>
<td>06/07/2015</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>Feature 1: Conformance check</td>
<td>18/06/2015</td>
<td>20/07/2015</td>
<td>20d</td>
</tr>
<tr>
<td>16</td>
<td>Guidelines management</td>
<td>18/07/2015</td>
<td>18/07/2015</td>
<td>50</td>
</tr>
<tr>
<td>17</td>
<td>Interface: Advanced up practical care journey</td>
<td>27/07/2015</td>
<td>27/07/2015</td>
<td>10d</td>
</tr>
<tr>
<td>18</td>
<td>Testing and validation</td>
<td>20/08/2015</td>
<td>28/08/2015</td>
<td>60d</td>
</tr>
<tr>
<td>19</td>
<td>Technical report - write &amp; review</td>
<td>28/08/2015</td>
<td>28/10/2015</td>
<td>105d</td>
</tr>
<tr>
<td>20</td>
<td>Final presentation</td>
<td>31/08/2015</td>
<td>17/09/2015</td>
<td>14d</td>
</tr>
</tbody>
</table>

Since the technology for the selected database platform was not available in due time, a mitigation strategy was introduced in order to keep the project going. This strategy included transition to a more familiar and mature database platform. The negative side of the strategy was that the database design was not following the HL7 standards for representing clinical knowledge.

The long running tasks are the overlaps that can be observed in the charts mean that time was spent on both tasks during a day’s work.

14.4 Project execution

The execution of the project followed a structured path based on the project planning. The first month comprised of meeting stakeholders and reading literature about the domain. It also included getting to know the high-level requirements of the stakeholders.

Following the agile approach of iterative software development, every month a Project Steering Group (PSG) meeting was held where the progress of the project was presented by the candidate and the direction of the project was maintained. During these meetings, the candidate presented the current status as well as demonstration of the product. The stakeholders gave feedback and to some extent validation of the product. If necessary, additional meetings were scheduled by the candidate to obtain additional information regarding visualization or epilepsy-related knowledge. For each meeting,
notes were taken by the candidate (meeting minutes), which were later put in a document and set to the stakeholders for review and comment.

By following the project plan, stakeholders could transparently observe how the project is progressing, and if the deliverables satisfy their requirements. Since the product includes several features, an ordered approach was followed in their development, one at a time.

As mentioned in section 5.2, a number of risks were identified throughout the project which caused slight changes in the direction of development. These changes were clearly presented to the stakeholders coupled with a proposed mitigation strategy from the candidate. Some of these risks are mentioned as future work for the product.
15. Project Retrospective

This chapter gives a reflection on the project based on the candidate’s perspective. At the end, the design opportunities set in the beginning of the project are revisited and evaluated.

15.1 Reflection

The project conducted in the past nine months exhibited both familiar and unfamiliar, yet challenging, characteristics. On the familiar side, there was already sufficient knowledge in web technologies and web development, as well as knowledge in data visualization. The challenging parts include working with doctors, getting to know the domain, and translating their requirements into tangible results.

As with every project, a sufficient level of understanding of the domain is required to be able to translate requirements into a result. Having a slight medical background, made the process a bit easy, but the complexity of the domain has its own weight. Luckily, the stakeholders from Kempenhaeghe were more than helpful and provided needed information and feedback whenever it was requested. This is important because it gives a two-way feedback, for the candidate to understand them, and for them to understand whether the candidate understands the domain.

One of the biggest concerns of the project was the formalization of the guidelines, and their applicability in a system. Significant time was spent on this, especially in the first three months. An additional overhead was that the guidelines were written in Dutch. Using translation tools and the candidate’s beginner knowledge in the Dutch language a translation was made, followed by a formalization with a model. In the beginning, this language barrier introduced a minor problem, but the continuous effort and the use of tools mitigated this problem. This of course was constantly evaluated by stakeholders from Kempenhaeghe for preciseness and completeness.

Having to work with Detailed Clinical Models proved to be not that difficult except the notation used to express the models. This was different than the typical use of UML, so it took a little time to get a grasp on it.

Overall, the project was a great experience and a change to exercise many skills related to software development. Cooperating with people and managing expectations is crucial and this was repeatedly exercised along the project. On top of that, several new technologies were used that broadened the knowledge and opened new horizons for the future.

15.2 Design opportunities revisited

During the problem analysis process, a single design opportunity was identified, namely usability. Since the goal of the project was to make a tool for end users, such as doctors and patient, and the tool had to make use of visualization techniques, special consideration was taken to ensure that usability is maximized. Following the iterative development approach and receiving constant feedback, stakeholders were able to evaluate the usability, look, and feel of the product. Formally, the use of a survey provided a feedback from a larger group of users.

■
This chapter presents the terminologies used throughout this report. Table 17 lists the terminologies with their definition.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDEng</td>
<td>Professional Doctorate in Engineering</td>
</tr>
<tr>
<td>NER</td>
<td>National Epilepsy Register. Database for centralizing knowledge about epilepsy</td>
</tr>
<tr>
<td>TU/e</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>HL7</td>
<td>Health Level Seven</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>ADR</td>
<td>Adverse Drug Reaction</td>
</tr>
<tr>
<td>NICE</td>
<td>National Institute for Health and Clinical Excellence</td>
</tr>
<tr>
<td>Cloud</td>
<td>General term for anything that involves delivering hosted services over the Internet</td>
</tr>
<tr>
<td>Server</td>
<td>A computing platform whose purpose is to serve other computing platforms</td>
</tr>
<tr>
<td>Web server</td>
<td>Same as server, available over the Internet</td>
</tr>
<tr>
<td>Database server</td>
<td>Same as server, used to host a database</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer (architectural style)</td>
</tr>
<tr>
<td>RESTful APIs</td>
<td>Interfaces that adhere to the REST style</td>
</tr>
<tr>
<td>MVC</td>
<td>Model View Controller (architectural style)</td>
</tr>
<tr>
<td>MVVM</td>
<td>Model View ViewModel (architectural style)</td>
</tr>
<tr>
<td>SPA</td>
<td>Single Page Application</td>
</tr>
<tr>
<td>Client-side</td>
<td>Operations that are performed by the client in a client–server relationship in a computer network. Typically, a client is a computer application, such as a web browser, that runs on a user's local computer or workstation and connects to a server as necessary.</td>
</tr>
<tr>
<td>Server-side</td>
<td>Operations that are performed by the server in a client–server relationship in a computer network. Typically, a server is a computer program, such as a web server, that runs on a remote server, reachable from a user's local computer or workstation.</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>Factory</td>
<td>Component responsible for providing instances of classes</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java database connectivity technology that defines how a client may access a database</td>
</tr>
</tbody>
</table>
References


Appendix A: Types of UML relationships

The UML relationships are thoroughly specified in the UML specification, but this section gives a brief and simple overview of them.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>![Dependency Diagram]</td>
</tr>
<tr>
<td>Aggregation</td>
<td>![Aggregation Diagram]</td>
</tr>
<tr>
<td>Inheritance</td>
<td>![Inheritance Diagram]</td>
</tr>
<tr>
<td>Composition</td>
<td>![Composition Diagram]</td>
</tr>
<tr>
<td>Association</td>
<td>![Association Diagram]</td>
</tr>
<tr>
<td>Directed Association</td>
<td>![Directed Association Diagram]</td>
</tr>
<tr>
<td>Interface Type Implementation</td>
<td>![Interface Type Implementation Diagram]</td>
</tr>
</tbody>
</table>

*Figure 52 Types of UML relationships*

For explanation, consider two entities A and B, and relationships between them according to Figure 52. The relationships user are:

- Dependency – A uses B for its implementation;
- Aggregation – A has an B;
- Inheritance – B is an A (or B extends A);
- Composition – A owns B;
- Association – similar as dependency, represents logical links between A and B (for example, A is linked to B, Passenger is linked with Airplane);
- Directed association – see Association. It involves additional direction that constrains the reading direction of the relationship;
- Interface Type Implementation – B realizes interface A.
Appendix B: Classification of epilepsy syndromes and seizure types

ILAE seizure type classification of 1981
The epilepsy classification is based on seizure type and syndrome. The first widely accepted classification of epileptic seizures was published in 1981 by the International League Against Epilepsy (ILAE\(^2^0\)).

- **Focal (local) seizures**
  - Simple focal seizures (consciousness not impaired)
    - With motor signs
    - With somatosensory or special-sensory symptoms (simple hallucinations, for example, tingling, light flashes, buzzing)
    - With autonomic symptoms or signs (for example, epigastric sensation, pallor, sweating, flushing, piloerection and papillary dilatation)
    - With psychic symptoms (disturbance of higher cerebral function) (for example, déjà vu, distortion of time sense, fear. NB these rarely occur without impairment of consciousness and are much more commonly experienced as 1.2 complex focal seizures)
  - Complex focal seizures (with impairment of consciousness)
    - With simple partial onset followed by impairment of consciousness
    - With impairment of consciousness at onset
  - Focal seizures evolving to secondarily generalized seizures (may be generalized tonic-clonic, tonic, or clonic)
    - Simple focal seizures evolving to generalized seizures
    - Complex focal seizures evolving to generalized seizures
    - Simple focal seizures evolving to complex focal seizures and then evolving to generalized seizures

- **Generalized seizures (convulsive or non-convulsive)**
  - Absence seizures (impairment of consciousness alone or with: mild clonic, atonic or tonic components, automatisms and/or autonomic symptoms or signs)
  - Atypical absence
  - Myoclonic seizures
  - Clonic seizures
  - Tonic-clonic seizures
  - Atonic seizures

- **Unclassified seizures**

ILAE syndrome classification of 1989
In 1989, a classification of epilepsies and epilepsy syndromes was published.

- **Localization-related (focal, local, ) epilepsies and syndromes**
  - Idiopathic (listed in order of age of onset)
    - Benign childhood epilepsy with centrotemporal spike
    - Childhood epilepsy with occipital paroxysms
  - Symptomatic
  - Cryptogenic

- **Generalized epilepsies and syndromes**
  - Idiopathic (listed in order of age of onset)
    - Benign neonatal familial convulsions

- Benign neonatal convulsions
- Benign myoclonic epilepsy in infancy
- Childhood absence epilepsy (pyknolepsy)
- Juvenile absence epilepsy
- Juvenile myoclonic epilepsy (impulsive petit mal)
- Epilepsy with grand mal (generalized tonic-clonic) seizures on awakening
  o Cryptogenic or symptomatic (listed in order of age of onset)
    - West syndrome (infantile spasms)
    - Lennox-Gastaut syndrome
    - Epilepsy with myoclonic-astatic seizures
    - Epilepsy with myoclonic absences
  o Symptomatic
    - Non-specific etiology
      - Early myoclonic encephalopathy
      - Early infantile epileptic encephalopathy with suppression burst
      - Other symptomatic generalized epilepsies not defined above
    - Specific syndromes
      - Epileptic seizures may complicate many disease states. Under this heading are included diseases in which seizures are a presenting or predominant feature
- Epilepsies and syndromes undetermined whether focal or generalized
  o With both generalized and focal seizures
    - Neonatal seizures – excluded from G/L
    - Severe myoclonic epilepsy in infancy
    - Epilepsy with continuous spike-waves during slow wave sleep
    - Acquired epileptic aphasia (Landau-Kleffner syndrome)
  o Without unequivocal generalized or focal features

All cases with generalized tonic-clonic seizures in which clinical and EEG findings do not permit classification as clearly generalized or localization related, such as in many cases of sleep-grand mal are considered not to have unequivocal generalized or focal features.

- Special syndromes
  o Febrile convulsions
  o Isolated seizures or isolated status epilepticus
  o Seizures occurring only when there is an acute metabolic or toxic event.

**ILAE classification of 2010 (seizure types and syndromes)**

After years of research and experiments, a new classification emerged in 2010. This classification, however, is not recognized globally, so both classifications are still used. The 2010 ILAE seizure classification is listed below [3] [25]:

- **Generalized seizures**
  o Tonic-clonic (in any combination)
  o Absence
    - Typical
    - Atypical
    - Absence with special features
      - Myoclonic absence
      - Eyelid myoclonia
  o Myoclonic
    - Myoclonic
    - Myoclonic atonic
    - Myoclonic tonic
  o Clonic
- Tonic
- Atonic

**Focal seizures**
- Without reduced level of consciousness
  - Motoric
  - Autonomous
  - Subjective sensory or psychic phenomenon
- Dyscognitive (change of awareness/consciousness)
- Becoming bilateral convulsive seizures
  - Motoric
  - Autonomous
  - Subjective sensory or psychic phenomenon
- Unknown (epileptic spasms)

**Unknown seizures**
- Epileptic spasms

Generalized seizures affect both hemispheres of the brain, while focal seizures affect only one. The unknown seizures cannot be clearly diagnosed so they are considered unclassified until further information reveals their accurate diagnosis. Each seizure type has different effects on the patient, ranging from harmless to the immediate need of a healthcare professional.

The number of syndromes is greater than the number of seizure types. They are grouped by age, starting from neonatal period, infancy, children age, adults, and some syndromes with characteristic finding or syndromes that cannot be placed in a specific group age. Figure 53 depicts part of the classification, where the syndromes are grouped based on the age of onset.

| Neonatal period | Benign familial neonatal epilepsy (BFNE)  
|                 | Early myoclonic encephalopathy (EME)  
|                 | Ohtahara syndrome |
| Infancy         | West syndrome  
|                 | Myoclonic epilepsy in infancy (MEI)  
|                 | Dravet syndrome |
| Childhood       | Benign epilepsy with centrotemporal spikes (BECTS)  
|                 | Lennox-Gastaut syndrome  
|                 | Childhood absence epilepsy (CAE) |
| Addolescence    | Juvenile absence epilepsy (JAE)  
|                 | Juvenile myoclonic epilepsy (JME) |
| Variable debut age | Familial focal epilepsy with variable foci (childhood to adult)  
|                 | Reflex epilepsies |
| Distinctive constellations | Rasmussen syndrome  
|                 | Gelastic seizures with hypothalamic hamartoma |

*Figure 53 Sample of the syndrome classification (grouped by age)*
Appendix C: Use cases of the system

This section lists the identified use cases in the system and their concrete steps.

Figure 54 Use cases identified for the system

The use cases described start from top to bottom. In the listing of the concrete steps of the scenarios, characters representing the actor and the system are used. For the system, the character is “S”, and depending on the type of actor, it can be “P” for patient, “D” for doctor, “GR_A” for group administrator, “GL_A” for guidelines administrator, or simply “U” for an general user.

**Use case 01: View the epilepsy guidelines**

<table>
<thead>
<tr>
<th>Description</th>
<th>The user wants to view the epilepsy guidelines. This scenario is available to registered and non-registered users.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>User-goal</td>
</tr>
<tr>
<td>Primary actor</td>
<td>Patient, Doctor, Anonymous user</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>System has Internet access and the application is opened</td>
</tr>
</tbody>
</table>

**Main scenario**

1. U: Requests to view guidelines
2. S: Retrieves the guidelines
3. S: Displays the guidelines

**Alternative flow**

2a. System cannot retrieve guidelines
    1. S: An error message is displayed to the user
### Use case 02: View patient’s care journey

<table>
<thead>
<tr>
<th>Description</th>
<th>User wants to explore patient’s care journey with epilepsy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>User-goal</td>
</tr>
<tr>
<td>Primary actor</td>
<td>Patient, Doctor</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>System has Internet access and the application is opened</td>
</tr>
</tbody>
</table>

#### Main scenario
1. U: Selects patient to view  
2. S: Retrieves data  
3. S: Displays patient’s care journey

#### Alternative flow
1a. User is patient  
   1. U: Selects himself as patient

---

### Use case 03: Check prescription

<table>
<thead>
<tr>
<th>Description</th>
<th>User wants to check a prescription for conformance to the guidelines and for registered side effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>User-goal</td>
</tr>
<tr>
<td>Primary actor</td>
<td>Patient, Doctor</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>System has Internet access and the application is opened</td>
</tr>
</tbody>
</table>

#### Main scenario
1. U: Enters information about patient, doctor, and medication  
2. S: Validates the information  
3. S: Checks the prescription  
4. S: Presents results

#### Alternative flow
1a. User is doctor  
   1. U: Enters information about patient and medication  
1b. User is patient  
   1. U: Enters information about doctor and medication  
2a. Information is invalid  
   1. S: Displays error information  
   2. U: Corrects the entered information

---

### Use case 04: Check for conformance

<table>
<thead>
<tr>
<th>Description</th>
<th>Doctor wants to check his conformance to the guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>User-goal</td>
</tr>
<tr>
<td>Primary actor</td>
<td>Doctor</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>System has Internet access and the application is opened</td>
</tr>
</tbody>
</table>

#### Main scenario
1. D: Chooses guidelines for comparison and range of prescriptions based on date  
2. D: Chooses a group to compare to (optional, Use case 05).  
3. S: Validates entered information  
4. S: Checks for conformance  
5. S: Presents results

#### Alternative flow
1a: Range information is not entered  
1. D: Chooses guidelines to compare against  
3b: Information is invalid  
1. S: Displays error information  
2. D: Corrects the entered information
Use case 05: Compare doctor to group

**Description**
Part of the check conformance scenario. User can optionally select a group to compare conformance results

**Level**
Sub-function

**Primary actor**
Doctor

**Precondition(s)**
System has Internet access and the application is opened

**Main scenario**

1. D: Chooses a group to compare against

**Alternative flow**

Use case 06: Request to view data

**Description**
Doctors requests to view data about a group’s conformance for comparison

**Primary actor**
Doctor

**Precondition(s)**
System has Internet access and the application is opened

**Main scenario**

1. D: Enters required information to request for comparison (group, date from, date to)
2. S: Validates the information
3. S: Sends the request
4. S: Displays confirmation

**Alternative flow**

2a: Information is not correct

1. S: Displays error message
2. D: Corrects entered information

4a: Error in saving request

1. S: Displays error message

Use case 07: Manage request to view data

**Description**
A group administrator manages requests to view the group data. The administrator can approve, deny, or delete a request.

**Primary actor**
Group Administrator

**Precondition(s)**
System has Internet access and the application is opened

**Main scenario**

1. GR_A: Requests to view benchmark requests for the group
2. S: Displays requests
3. GR_A: Selects a request
4. S: Presents details of the request
5. GR_A: Approves a request
6. S: Updates the request

**Alternative flow**

5a: Administrator denies a request

1. GR_A: Denies a request

5b: Administrator deletes a request

1. GR_A: Deletes a request
<table>
<thead>
<tr>
<th>Use case 08: Modify the epilepsy guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Primary actor</strong></td>
</tr>
<tr>
<td><strong>Precondition(s)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Main scenario</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  GL_A: Selects an entry from the guidelines</td>
</tr>
<tr>
<td>2.  S: Displays details about the entry</td>
</tr>
<tr>
<td>3.  GL_A: Makes changes to the entry</td>
</tr>
<tr>
<td>4.  GL_A: Requests to modify the entry</td>
</tr>
<tr>
<td>5.  S: Modifies the entry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Alternative flow</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1a: User adds new entry in the guidelines</td>
</tr>
<tr>
<td>1.  GL_A: Enters required information</td>
</tr>
<tr>
<td>2.  S: Validates information</td>
</tr>
<tr>
<td>3.  S: Saves the entry</td>
</tr>
<tr>
<td>3a: User deletes an entry</td>
</tr>
<tr>
<td>1.  GL_A: Requests to delete an entry</td>
</tr>
<tr>
<td>2.  S: System deletes the entry</td>
</tr>
<tr>
<td>5a: System cannot modify entry</td>
</tr>
<tr>
<td>1.  S: Displays error information</td>
</tr>
</tbody>
</table>
### Appendix D: The epilepsy guidelines

#### Seizure types

<table>
<thead>
<tr>
<th>Focal seizures</th>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First choice (first &amp; second attempt)</strong></td>
<td></td>
<td>Carbamazepine Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Levetiracetam Oxcarbazepine Valproate</td>
<td>Not on pregnant women</td>
</tr>
<tr>
<td><strong>Second choice (third attempt)</strong></td>
<td></td>
<td>Phenytoin Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lacosamide Pregabalin Topiramate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clobazam Perampanel</td>
<td>Only as adjunctive AED</td>
</tr>
<tr>
<td><strong>Adjunctive (third attempt)</strong></td>
<td></td>
<td>Zonisamide</td>
<td></td>
</tr>
<tr>
<td><strong>Do not use</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Absence seizures (Generalized seizures)</th>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First choice (first attempt)</strong></td>
<td></td>
<td>Ethosuximide</td>
<td></td>
</tr>
<tr>
<td><strong>Second choice (second attempt)</strong></td>
<td></td>
<td>Lamotrigine Valproate</td>
<td>not on pregnant women</td>
</tr>
<tr>
<td><strong>Adjunctive (third attempt)</strong></td>
<td></td>
<td>1. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td><strong>Do not use</strong></td>
<td></td>
<td>Phenobarbital Carbamazepine</td>
<td>Do not use this medication in case of childhood absence epilepsy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Phenytoin Gabapentin Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
</tbody>
</table>
### Tonic-clonic seizures combined with myoclonus (*Generalized seizures*)

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First choice</strong> (first &amp; second attempt)</td>
<td>Levetiracetam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td>Not on pregnant women</td>
</tr>
<tr>
<td><strong>Second choice</strong> (second attempt)</td>
<td>Clobazam</td>
<td>Only as adjunctive AED</td>
</tr>
<tr>
<td></td>
<td>Topiramate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Zonisamide</td>
<td></td>
</tr>
<tr>
<td><strong>Adjunctive</strong> (third attempt)</td>
<td>Clonazepam</td>
<td>Only as adjunctive AED</td>
</tr>
<tr>
<td></td>
<td>1. Clobazam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Clonazepam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td><strong>Do not use</strong></td>
<td>Vigabatrin</td>
<td>Do not use in case of absences and/or myoclonus</td>
</tr>
<tr>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
</tbody>
</table>

### Tonic-clonic seizures without myoclonus (*Generalized seizures*)

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First choice</strong> (first &amp; second attempt)</td>
<td>Levetiracetam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td>Not on pregnant women</td>
</tr>
<tr>
<td></td>
<td>Clonazepam</td>
<td>Only as adjunctive AED</td>
</tr>
<tr>
<td><strong>Second choice</strong> (second attempt)</td>
<td>Topiramate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Zonisamide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clobazam</td>
<td>Only as adjunctive AED</td>
</tr>
<tr>
<td><strong>Adjunctive</strong> (third attempt)</td>
<td>1. Clobazam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Clonazepam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td><strong>Do not use</strong></td>
<td>Vigabatrin</td>
<td>Do not use in case of absences</td>
</tr>
<tr>
<td></td>
<td>Phenobarbital</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
</tbody>
</table>
### Unclassifiable seizures

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>First choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(first &amp; second attempt)</td>
<td>Levetiracetam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td>not on pregnant women</td>
</tr>
<tr>
<td></td>
<td>Clobazam</td>
<td></td>
</tr>
<tr>
<td>Second choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(second attempt)</td>
<td>Topiramate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenytoin</td>
<td>Caution when used because of paradoxical</td>
</tr>
<tr>
<td></td>
<td>Gabapentin</td>
<td>effect in myoclonus or absence seizures</td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
<tr>
<td>Adjunctive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not use</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: Third attempt – refer to specialized center*

### Syndromes

#### Childhood absence epilepsy (CAE)

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>First choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(first &amp; second attempt)</td>
<td>Ethosuximide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td></td>
</tr>
<tr>
<td>Second choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjunctive</td>
<td>1. Ethosuximide</td>
<td></td>
</tr>
<tr>
<td>(third attempt)</td>
<td>1. Lamotrigine</td>
<td></td>
</tr>
<tr>
<td>Do not use</td>
<td>2. Valproate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenobarbital</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tiagabine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregabalin</td>
<td></td>
</tr>
</tbody>
</table>

#### Juvenile myoclonic epilepsy (JME)

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>First choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(first &amp; second attempt)</td>
<td>Valproate</td>
<td>not on pregnant women</td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Levetiracetam</td>
<td></td>
</tr>
<tr>
<td>Second choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjunctive</td>
<td>1. Valproate</td>
<td>may increase absence and/or myoclonic</td>
</tr>
<tr>
<td>(third attempt)</td>
<td>2. Clobazam</td>
<td>seizures</td>
</tr>
<tr>
<td></td>
<td>2. Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td>Do not use</td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gabapentin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tiagabine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vigabatrin</td>
<td></td>
</tr>
</tbody>
</table>
### Benign epilepsy with centrotemporal spikes (BECTS) (rolandic)

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>First choice (first &amp; second attempt)</td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Levetiracetam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td>Not on pregnant women</td>
</tr>
<tr>
<td></td>
<td>Clobazam</td>
<td></td>
</tr>
<tr>
<td>Second choice</td>
<td>Topiramate</td>
<td>Not on children, can cause cognitive side-effects</td>
</tr>
<tr>
<td>Adjunctive (third attempt)</td>
<td>1. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Any first or second choice</td>
<td></td>
</tr>
<tr>
<td>Do not use</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Panayiotopoulos syndrome

<table>
<thead>
<tr>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>First choice (first &amp; second attempt)</td>
<td>Carbamazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lamotrigine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oxcarbazepine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valproate</td>
<td></td>
</tr>
<tr>
<td>Second choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjunctive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not use</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Status epilepticus

<table>
<thead>
<tr>
<th>Status epilepticus in children</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tonic-clonic seizures longer than 5 minutes or series of seizures without the patient regaining consciousness</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td>Step 1: t=0 (for medical intervention) Attack duration of 5 minutes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Step 2: t=5min. Insufficient results after step 1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Step 3: t=10min. Insufficient results after 5 minutes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Step 4: t=15min. Insufficient results after 10 minutes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Refractory status (Failure of benzodiazepine and antiepileptic drug)</td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td>Step 5: t=25-35min. Consultation with intensive care (consider intubation and ventilation)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Special case Children under 2 years with little or no effect of AEDs</td>
</tr>
</tbody>
</table>
### Status epilepticus in adults

**Tonic-clonic seizures longer than 5 minutes or series of seizures without the patient regaining consciousness**

<table>
<thead>
<tr>
<th>Steps</th>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>First choice</strong></td>
<td>Lorazepam</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Midazolam</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Phenytoin</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Second choice</strong></td>
<td>Valproate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Levetiracetam</td>
<td></td>
</tr>
</tbody>
</table>

**Refractory status**

*(Failure of benzodiazepine and antiepileptic drug)*

<table>
<thead>
<tr>
<th>Steps</th>
<th>Category</th>
<th>AED</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td><strong>First choice</strong></td>
<td>Midazolam</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thiopental</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Propofol</td>
<td></td>
</tr>
</tbody>
</table>

### Advised care journey

<table>
<thead>
<tr>
<th>#</th>
<th>Advice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Refer epilepsy patients to evaluation for surgery if the patient is drug resistant and keeps getting attacks despite treatment with two or three antiepileptic drugs in adequate dosage</td>
</tr>
<tr>
<td>2</td>
<td>Refer epilepsy patients to assess the possibility of specialized epilepsy treatments instead of surgery</td>
</tr>
<tr>
<td>3</td>
<td>Refer to a pediatric neurologist or epilepsy center if the clinical symptoms and/or seizures on children with benign Rolandic epilepsy change</td>
</tr>
<tr>
<td>4</td>
<td>Refer the patient to specialized epilepsy care if within two years from the debut of the epilepsy and/or failure of two antiepileptics the patient is still not seizure-free</td>
</tr>
<tr>
<td>5</td>
<td>Consider referring the patient to specialized epilepsy care if the epilepsy debuts in the early years</td>
</tr>
</tbody>
</table>
Appendix E: Syndrome and seizure mapping

This section lists the mapping developed to translate the different classifications used into a single one, namely ILAE classification of 2010. Table 18 lists the Lunders and ILAE classification of seizure types (names are in Dutch) and their respective mapping to the new classification. Seizure types marked light red are from the Lunders classification, and seizure types marked light yellow are from the ILAE classification.

Table 18 Mapping of seizure types (from old ILAE and Lunders classification to new ILAE classification)

<table>
<thead>
<tr>
<th>Original seizure name (as recorded in the dataset)</th>
<th>Seizure (ILAE classification)</th>
<th>Combined with</th>
<th>Treat as</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialeptische aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Automotore aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Somatosensorische aura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Hypermotore aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Autonome aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Psychische aura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Buitaura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Afatische aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Autonome aura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Smaakaura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Gelastische aanvalen</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Visuele aura's</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Tonisch-clonische aanvalen</td>
<td>Tonic-clonic seizures</td>
<td></td>
<td>Tonic-clonic seizures</td>
</tr>
<tr>
<td>Tonische aanvalen</td>
<td>Tonic seizures</td>
<td></td>
<td>Tonic seizures</td>
</tr>
<tr>
<td>Clonische aanvalen</td>
<td>Clonic seizures</td>
<td></td>
<td>Clonic seizures</td>
</tr>
<tr>
<td>Versiee aanvalen</td>
<td>Tonic seizures</td>
<td></td>
<td>Tonic seizures</td>
</tr>
<tr>
<td>Myoclonische aanvalen</td>
<td>Myoclonic seizures</td>
<td></td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td>Atone aanvalen</td>
<td>Atonic seizures</td>
<td></td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td>Astatiche aanvalen</td>
<td>Myoclonic seizures</td>
<td></td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td>Epileptische spasmen</td>
<td>Tonic seizures</td>
<td></td>
<td>Tonic seizures</td>
</tr>
<tr>
<td>Epilepsie-aanval niet nader gespecificeerd</td>
<td>Unclassified seizures</td>
<td></td>
<td>Unclassified seizures</td>
</tr>
<tr>
<td>Aanvalsgewijze incidenten niet-epileptisch</td>
<td>Non-epileptic seizures</td>
<td></td>
<td>Non-epileptic seizures</td>
</tr>
<tr>
<td>Complex Partiele Aanval (Cpa)</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Tonisch-Clonische Aanval</td>
<td>Tonic-clonic seizures</td>
<td></td>
<td>Tonic-clonic seizures</td>
</tr>
<tr>
<td>Eenvoudig Partiele Aanval (E.P.A.)</td>
<td>Focal seizures</td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Clonische Aanval</td>
<td>Clonic seizures</td>
<td></td>
<td>Clonic seizures</td>
</tr>
</tbody>
</table>
The next table, Table 19, lists the syndrome translation from the different classification to the ILAE classification of syndromes. It also lists the different types of seizure types that the syndrome is manifested through.

**Table 19 Syndrome mapping between old and new classification**

<table>
<thead>
<tr>
<th>Syndrome (registered in Kempenhaeghe)</th>
<th>Syndrome (ILAE classification)</th>
<th>Manifested with (seizure type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partiele Aanval, Overgaand In Secundair Gegeneraliseerd</td>
<td>Focal seizures</td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Tonische Aanval</td>
<td>Tonic seizures</td>
<td>Tonic seizures</td>
</tr>
<tr>
<td>Absences</td>
<td>Absence seizures</td>
<td>Absence seizures</td>
</tr>
<tr>
<td>Niet Geclasseificeerde Aanval</td>
<td>Unclassifiable seizures</td>
<td>Unclassifiable seizures</td>
</tr>
<tr>
<td>Myoclonische Aanval</td>
<td>Myoclonic seizures</td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td>Psychogene Aanval</td>
<td>Focal seizures</td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Syncope</td>
<td>Focal seizures</td>
<td>Focal seizures</td>
</tr>
<tr>
<td>Atonische Of Val-Aanval</td>
<td>Atonic seizures</td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td>Drop Attack</td>
<td>Myoclonic-astatic seizures</td>
<td>Myoclonic-astatic seizures</td>
</tr>
<tr>
<td>No Epilepsy</td>
<td>Benign epilepsy with centrotemporal spikes (BECTS)</td>
<td>Focal seizures</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Idiopathic (Primary) Localization-Related (Focal, Local, Partial)</td>
<td>Late-onset childhood occipital epilepsy (Gastaut type)</td>
<td>Focal seizures</td>
</tr>
<tr>
<td></td>
<td>Epilepsies with unknown cause</td>
<td>Unclassifiable seizures</td>
</tr>
<tr>
<td>Undetermined Epilepsies With Both Generalized And Focal Seizures</td>
<td>Lennox-Gastaut syndrome</td>
<td>Tonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Atonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Atypical absence seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Atypical seizures</td>
</tr>
<tr>
<td>Cryptogenic Or Symptomatic Generalized</td>
<td>Dravet syndrome</td>
<td>Myoclonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Atypical absence seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Focal seizures</td>
</tr>
<tr>
<td></td>
<td>West syndrome</td>
<td>Tonic-clonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Infantile spasms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tonic-clonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other seizure types</td>
</tr>
<tr>
<td></td>
<td>Ohathara syndrome</td>
<td>Tonic-clonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tonic seizures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other seizure types</td>
</tr>
</tbody>
</table>
About the Author

Trajche Masinov received his Bachelor of Electrical Engineering and Information Technologies from the Faculty of Electrical Engineering and Information Technology (Skopje, Macedonia), in October 2009, specializing in Informatics and Computer Engineering. He received his MSc degree in Software Engineering of the Faculty of Computer Science and Engineering (Skopje, Macedonia) in November 2013. His master thesis involved utilizing modern data visualization techniques as a tool for displaying news, offering different ways a user consumes everyday news. In parallel with the master studies he worked for almost 5 years as a Software Developer at Edusoft, Macedonia, developing Web-based information systems in ASP.NET and SQL Server/Oracle RDBMS.

From September 2013 until September 2015, he worked at the Eindhoven University of Technology, as PDEng trainee in the Software Technology program from the 3TU.Stan Ackermans Institute. During his graduation project, he worked for the Epilepsy Center Kempenhaeghe (under supervision of Synergetics Benelux BV) on a project focused on decision support via data visualization for improving epilepsy care.
3TU. School for Technological Design, Stan Ackermans Institute offers two-year postgraduate technological designer programmes. This institute is a joint initiative of the three technological universities of the Netherlands: Delft University of Technology, Eindhoven University of Technology and University of Twente. For more information please visit: www.3tu.nl/sai.