Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Tormod Lysvold Sjømæling
Håkon Johansen Finstad

Efficient Data Visualization in Healthcare: A User-Friendly Sepsis Dashboard for Clinicians

Masteroppgave i Datateknologi (MIDT) og Informatikk (MSIT)
Veileder: Rajeev Bopche
Medveileder: Øystein Nytrø

Juni 2023

NTNU
Kunnskap for en bedre verden

Tormod Lysvold Sjømæling
Håkon Johansen Finstad

Efficient Data Visualization in Healthcare: A User-Friendly Sepsis Dashboard for Clinicians

Masteroppgave i Datateknologi (MIDT) og Informatikk (MSIT)
Veileder: Rajeev Bopche
Medveileder: Øystein Nytrø
Juni 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Kunnskap for en bedre verden

Assignment

Electronic health records have been stored for the past 40 years in Norway. This data provides many opportunities for research and development of healthcare applications. The infection ward at St. Olavs in Trondheim, uses Helseplatformen today. The system has however been under much criticism for various reasons by clinicians.

The projects objective is to gain insights into the design and implementation of a sepsis dashboard that visualizes relevant patient medical data in an organized and understandable fashion, aimed towards the needs and expectations of clinicians. The dashboard should utilize timelines, graphs and other data visualization techniques. Clinicians should be able to observe and interact with the dashboard and gain a better understanding of the patients condition, trends and disease progression.

Abstract

As the healthcare sector grows and large software systems are implemented in hospitals, there is a growing need to store and visualize medical data. With such complicated data it becomes increasingly important to visualize it in an understandable and organized fashion. It therefore becomes crucial to ensure effective and appropriate data presentation. The objective of this thesis is to gain insights into the design and implementation of a sepsis dashboard. This will be achieved through qualitative methods, including semi-structured interviews with clinicians and experts, as well as review of existing literature. Expert reviews will be conducted provide valuable feedback on issues and opportunities for improvement.
Valuable insight and ideas from the interviews, emphasizing the importance of a simple, understandable, and user-friendly interface is used in the design of the dashboard. Additionally criteria, guidelines and best practices from existing literature to ensure effectiveness and usability in the dashboard is also used. The final result is a dashboard that contains patient medication history, visual representations of vital signs, and sample results through timelines and graphs. In conclusion, the thesis provides valuable insights into effective presentation of data in time-based graphs and timelines on a sepsis dashboard.

Abstrakt

Mens helsesektoren vokser og store programvare systemer implementeres på sykehusene, er det en økende nødvendighet å lagre og visualisere medisinske data. Med slik kompleks data blir det stadig viktigere å presentere det på en forståelig og organisert måte. Det er derfor avgjørende å sikre effektiv og passende presentasjon av data. Målet med denne oppgaven er å få innsikt i design og implementering av et sepsis dashboard. Dette vil bli gjort gjennom kvalitative metoder, inkludert semistrukturert intervjuer med klinikere og eksperter, samt gjennomgang av eksisterende litteratur. Ekspertvurderinger vil bli gjennomført for å gi verdifull tilbakemelding om utfordringer og muligheter for forbedring.

Verdifull innsikt og ideer fra intervjuene, som vektlegger betydningen av et enkelt, forståelig og brukervennlig grensesnitt, brukes i utformingen av dashbordet. I tillegg brukes kriterier, retningslinjer og bestepraksis fra eksisterende litteratur for å sikre effektivitet og brukervennlighet i dashbordet. Sluttresultatet er et dashbord som inneholder pasientens medikamenthistorikk, visuell fremstilling av vitale tegn og labsvar gjennom tidslinjer og grafer. Konklusjonen er at oppgaven gir verdifulle innsikt i effektiv presentasjon av data i tidsbaserte grafer og tidslinjer på et sepsis dashbord.

Table of Contents

List of Figures **ix**

List of Tables **xi**

1 Introduction **1**

1.1 Motivation 2

1.2 Research Problem 2

1.3 Contributions 3

1.4 Thesis Structure 3

2 Background **5**

2.1 Dataset and REK 5

2.2 Electronic Health Records 6

2.3 Medical Concepts 7

2.3.1 Sepsis 7

2.3.2 International Classification of Diseases Version 10: ICD-10 . . 7

2.4 The CoSeM Project 8

3 Methods **11**

3.1 Experimental Design 11

3.1.1 Design Alternatives 11

3.1.2 Why Development Was Chosen 13

3.2 Design Plan 13

3.3 Project Process 14

3.4 Literature Review 16

3.5 Interview with Experts 17

3.6 System Development 19

3.6.1 Data Exploration and Data Preprocessing 19

3.6.2 Data Selection 21

3.6.3 Backend Architecture and Database 22

3.6.4 Selecting Technology 29

4 Results **35**

4.1 Literature Review Results 35

4.2 Interview Results 42

4.2.1 Interview One - Expert Opinion 43

4.2.2 Interview Two - Doctor 45

4.2.3 Interview Three - Nurses 47

4.2.4 Interviews Round-Up 48

4.3 Expert Feedback and Review 49

4.4 Implementation 50

4.4.1 Description of Dashboard Interface 51

4.4.2 Architectural Design 52

5 Discussion and Evaluation **55**

5.1 Project Process 55

5.2 Interviews 56
5.3 Expert Review and Feedback 56
5.4 Implementation 57
5.4.1 Technology 57
5.4.2 Dashboard 58
5.5 Guidelines for Timeline Visualization 62

6 Conclusion **65**
6.1 Answers to Research Questions 65
6.2 Future Work 67

Bibliography **69**
Appendix 73
A SQL Queries for Data Selection 73
B Graph Criteria Evaluation 73
C Interviews 73
D FHIR Patient 73

List of Figures

2.1 Screenshot of ICD-10 hierarchy and codes from Direktoratet for e-helse 8

3.1 Image of the iterative process done during this project 14

3.2 An overview of the process for the thesis 15

3.3 A gantt diagram 16

3.4 Screenshot of ICD-10 codes marked as Explicit sepsis 21

3.5 Data selection visualisation 22

3.6 Summarization of common API HTTP and naming conventions 24

3.7 Screenshot of the API in Swagger 24

3.8 ER model - Restructured database of the provided dataset 28

3.9 Stack Overflow trends 30

3.10 Overview of MVVM pattern 33

4.1 Horbrugger et al. graphical interface. 36

4.2 VisuExplore: visualization tool that displays patient data in timelines. 37

4.3 Lifelines: visualization tool. 38

4.4 LifeFlow: Interactive visualization tool. 39

4.5 Overview of evaluation criteria. 40

4.6 Image of dashboard after feedback 51

4.7 Image of the system architecture 53

5.1 Alternative medication timeline 61

5.2 Graph showing normalized values of sample results 62

1 SQL queries 73

2 Visualization of criteria evaluation from Sitting et. al.s paper. 73

3 Interview guide and questions 76

List of Tables

3.1 Patient Table 20
3.2 Nimesaktivitet Table 20
3.3 Aninopphold Table 20
3.4 Trfl Table 21
3.5 Statistics over frameworks as of 29th May 2023 29

5.1 Timeline Visualization Guidelines 63

Glossary

Dataset A collection of data provided by Helse-Midt Norge IT.

Member A member of the team, whom is co writing this thesis.

Nuget NuGet is a package manager, primarily used for packaging and distributing software written using the .NET framework.

RESTful Obeys the rules of a REST API.

Team The students writing this thesis.

WebAPI Enables HTTP Requests to an API.

Acronyms

API Application Programming Interface.

BSI suspected bloodstream infection.

CoSeM Computational sepsis mining and modelling.

CQRS Command and Query Responsibility Segregation.

CSS Cascading Style Sheet.

EHR Electronic Health-Record.

FHIR Fast Healthcare Interoperability Resources.

HEMIT Helse Midt-Norge IT.

HP Helseplatformen.

HTML Hypertext Markup Language.

HTTP Hypertext Transfer Protocol.

ICD International Classification of Diseases.

ICD-10 International Classification of Diseases Version 10.

ICU Intensive Care Units.

JSON JavaScript Object Notation.

JSX JavaScript extendable mark-up language.

MVP Minimal Viable Product.

MVVM Model View View-model.

NDA non-disclosure agreement.

NTNU Norwegian University of Science and Technology.

REK Norwegian Regional Committees for Medical and Health Research Ethics.

REST Representational state transfer.

SFC Single-File Component.

SOAP Simple Object Access Protocol.

UI User Interface.

URI Uniform Resource Identifier.

UWP Universal Windows Platform.

UX User Experience.

WPF Windows Presentation Foundation.

Acknowledgements

We would like to thank Associate Professor Øystein Nytrø and PhD Candidate Rajeev Bopche whom have been supervisors throughout this project. Particularly Birgitta Ehrnström and Jan Kristian Damas for providing valuable help in understanding medical concepts and data. We would also like to thank the members of the CoSeM group whom have allowed us to work on this project and given good insight in many related topics.

CHAPTER 1

Introduction

The healthcare industry is a complex and extensive system that comprises various privately and state owned departments. The efficient flow of information within the system is not always streamlined, resulting in a range of challenges that could be addressed through innovative solutions. With the expansion of technology and society, there is an increasing demand for healthcare software that aligns with the evolving landscape. As a result, continued efforts have led to the development of new software and solutions aimed at promoting information sharing and ensuring reliable and straightforward technology.

Helseplatformen (HP) has recently been introduced in the Helse-Midt region, with mixed reviews. The monitor dashboard at the infection ward at St. Olavs Hospital in Trondheim has been described as cluttered and confusing, which highlights the need for a clear, concise and effective dashboard. The development of a well designed dashboard is crucial not only for monitoring and detecting possible dangerous signs but also for ensuring patient safety. This thesis aims to explore the needs and preferences of clinicians at St. Olavs infection ward regarding a dashboard that visualizes patient data.

The focus of the thesis is on patients who have either had sepsis or are currently experiencing it. Sepsis is a significant cause of mortality in Norwegian hospitals, underscoring the importance of developing effective solutions to combat this condition (Kossman and Scheidenhelm, 2008). To address this issue, the project has conducted interviews with nurses, doctors, and experts to identify the most important information that should be displayed on the dashboard. By considering their feedback, the project aims to develop a user-friendly dashboard that provides the necessary information for clinicians to make informed decisions about patient care.

1.1 Motivation

Sepsis is a critical health condition that poses a significant challenge to healthcare providers worldwide. Effective management of sepsis can have a profound impact on patient outcomes, yet the complexity and variability of patient data make this task difficult. With the launch of HP this project digs deeper into how it could be improved upon, based on expert and clinician opinion.

This thesis aims to develop a sepsis dashboard that will provide healthcare providers with a intuitive visualization of critical patient data. The dashboard will include information from multiple sources, including electronic health records and vital signs, into a single, user-friendly interface. The dashboard will enable healthcare providers to monitor sepsis patients and facilitate early recognition of worsening clinical status.

The development of a sepsis dashboard will address sepsis care by providing healthcare providers with a tool for data visualization. The dashboard has the potential to improve quality and efficiency of sepsis care, reducing the burden on healthcare systems and ultimately benefiting patients.

1.2 Research Problem

The purpose of this project is to visualize patient data in the context of sepsis in a dashboard. With this a goal was made.

> Goal: *Develop a dashboard which visualizes important patient data in a clear, concise and orderly fashion, giving the clinician a good overview of the patients current situation.*

This system is made for clinicians observing a patient, and the dashboard should display important medical data that enables the clinician to identify if there is ongoing sepsis. Three research questions(RQ) were formulated:

> RQ1: *What are the clinicians expectations of a system like this and what medical information is vital to display?*

In order to develop a dashboard it is important to understand what the experts want, their thoughts and expectations of key data, design and functionality of a dashboard. This question sets the foundation of the design of the dashboard and the functionality of the graphs and timelines. This also allows for displaying only relevant information, leading to less clutter, and allowing clinicians to quickly gain an overview of the patient.

RQ2: *How can patient data be efficiently visualized in timelines?*

To further display the medical data it is important to understand how to best visualize timelines. How does the clinicians want it to be displayed , and how has existing literature addressed the visualization of patient data in timelines?

RQ3: *Is showing previous episodes in a timeline important for a clinician?*

A further question for us to understand what is relevant information to a clinician observing a patient.

1.3 Contributions

The master thesis aims to make a significant contribution to the field of sepsis research by providing novel insights into the development and implementation of data-driven visualization tools in healthcare. Specifically, the thesis proposes a solution for a sepsis dashboard, which offers valuable suggestions and guidelines for the inclusion and functioning of a dashboard dedicated to sepsis.

The dashboard proposed in this project is expected to enhance clinical decision making and patient outcomes by providing real-time,and personalized information to healthcare professionals. This contribution has the potential be used in further research in design and development of a sepsis dashboard and ultimately improving patient care.

1.4 Thesis Structure

The thesis has both technical and medical aspects, which are described in chapter 2. Further the methods used to conduct the project is described in chapter 3. In chapter 4, the results from the literature review, interviews, feedback and implementation are presented. Lastly in chapter 5, discussion and evaluation of the project is done.

CHAPTER 2

Background

To properly visualize patient medical data, technical and medical knowledge is required. It is crucial to understand the data in order to present accurate information and to display relevant data that is applicable to the context. In section 2.1, the data utilized by the dashboard is introduced. Section 2.2 delves into the concept of Electronic Health-Record, while section 2.3 elaborates on medical concepts relevant to this project. Finally, section 2.4 presents the CoSeM project this thesis has been collaborating with.

2.1 Dataset and REK

For the project, a dataset from Helse Midt-Norge IT (HEMIT) was provided. However, the data did not meet the level of richness and comprehensiveness desired by the clinicians, in terms of their expectations and needs for the dashboard. As a result, a combination of the provided dataset and synthetic/mocked data was utilized to fulfill the clinicians requirements, more on this in later chapters.

The dataset consists of complete lifetime medical histories of patients that had at least one episode of suspected bloodstream infection (BSI) at St. Olavs hospital between 2015- 2020. The medical histories comprises of demographics, diagnosis codes, medical and procedural codes, and laboratory tests and results. It contains records of 35,599 patients with information about hospital visits and admissions. Diagnosis for each visit is coded using International Classification of Diseases Version 10 (ICD-10) codes, more on this in section 2.3.

An application was submitted to the Norwegian Regional Committees for Medical and Health Research Ethics (REK) seeking approval to utilize a dataset. In January 2023, the REK ethical approval for the usage of the dataset was granted, REK approval 2018/1201. A non-disclosure agreement (NDA) was also signed.

2.2 Electronic Health Records

In recent years hospitals has gone from using full paper journals for patients to the use of electronic papers, Electronic Health-Record (EHR). The Nordic countries were among the first to start using EHRs, because of this the importance of visualizing the patient data in the correct manner and context has increased (Nøhr et al., 2018). At its simplest a EHR is a electronic version of the paper, however it has now expanded to showing trends of patient history in timelines and graphs. Managing a patient across multiple departments, or even hospitals heightens its complexity (Sridharan et al., 2021).

The impact of EHRs on healthcare delivery has been substantial and holds great promise for the future of patient care. With the right EHR system in place, there is potential for significant improvements in patient outcomes and quality of care (Bowman, 2013).

One of the primary advantages of EHRs is the ability to maintain a centralized patient record that can be accessed by multiple healthcare providers across different locations. This feature allows patients to receive seamless care regardless of geographical location. (Bowman, 2013). EHR also helps to save time and effort when obtaining an overview of the patients medical history. Quick access to patient information can even be a life saving factor in crucial situations (Sridharan et al., 2021).

Furthermore, EHRs offer numerous benefits, including improved care coordination, better medication management, reduced medical errors, and enhanced patient engagement. These features contribute to better patient outcomes and a higher quality of care. Therefore, the implementation of effective EHR systems has become increasingly vital for healthcare providers to remain competitive and deliver high-quality care (Campanella et al., 2015).

The use of EHRs comes with potential threats that should not be overlooked. One of the major concerns is poor design. Poor EHR design can lead to mistakes, reduced levels of care, and endangering patient safety. As noted by Bowman, these risks should not be taken lightly, and it is essential to design EHR systems with attention to detail (Bowman, 2013).

Therefore, it is crucial to gather input from experts to understand what they consider important in an EHR system. This effort will help identify key elements that should be used in the system to enhance its usability, efficiency, and effectiveness in improving patient care. By doing so, it can minimize the risks associated with poor EHR design, and the benefits of using EHRs can be fully realized in healthcare delivery (Bowman, 2013).

2.3 Medical Concepts

This section aims to offer a short and concise introduction to certain medical concepts that are relevant to the project and may facilitate the readers understanding of the following chapters. The intention is not to provide an in depth explanation of these concepts, but rather to present a foundational understanding of their fundamentals.

2.3.1 Sepsis

Sepsis, also referred to as blood poisoning, is a life threatening condition caused by the bodys response to an infection. The condition is typically triggered by a bacterial infection that causes a widespread inflammatory response in the body (Singer et al., 2016), and it is one of the leading causes of death in Norwegian hospitals (Sintef, 2019).

2.3.2 International Classification of Diseases Version 10: ICD-10

The International Classification of Diseases (ICD) coding system is a medical classification system that includes codes for various medical conditions, including diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease. The system aims to establish a universal and standardized understanding of disease classification and diagnostic interpretation. This global standardization enables more effective and efficient medical research, improved healthcare delivery, and better healthcare outcome for patients worldwide. ICD-10 has multiple revisions, the latest being ICD-11, but this version has not yet been rolled out in Norway (Malt and Braut, 2022).

The ICD-10 classification system arranges diseases into distinct groups, such as I, IV, and X, with each group assigned a chapter letter, such as A, B, and P, that denotes its overall classification. Within a group, a digit identifies the specific disease group, which may be further specified by a second digit that distinguishes a particular disease within the group. A period separates this identification system from the additional use of one or two digits to describe subgroups of a specific disease (Malt and Braut, 2022). Figure 2.1 below shows a screenshot of the ICD-10 hierarchy and codes of the system, taken from Direktoratet for e-helse.

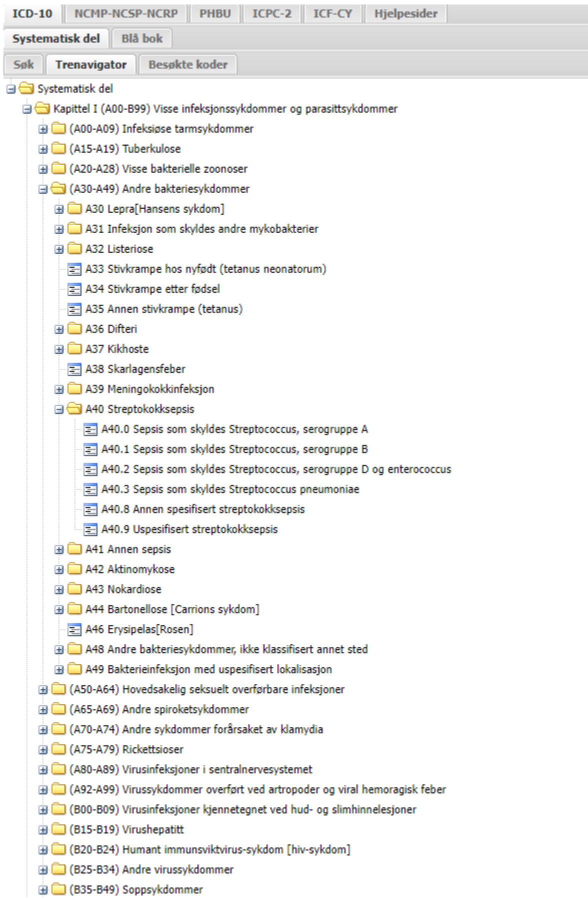

Figure 2.1: Screenshot of ICD-10 hierarchy and codes from Direktoratet for e-helse

2.4 The CoSeM Project

Computational sepsis mining and modelling (CoSeM) project is a research collaboration between Norwegian University of Science and Technology (NTNU), St. Olav Hospital, Helse Norge-Trøndelag HF, BigMed, and Gemini Centre for Sepsis Research. The project is a 5-year initiative that started in 2020. The objective of the project is to phenotype and characterize patients at increased risk of bloodstream infections and sepsis (NTNU, 2020).

Some of the projects in the CoSeM group includes capturing and preserving individual case histories using novel technology for integrating temporal interpretation of clinical text with structured data, infection and sepsis monitoring, demonstrate a clinical dashboard and more, (KIN, 2020).

CHAPTER 3

Methods

In this chapter, the methods used in the project will be presented. The chapter is divided into six sections, comprising of different strategies of prototyping section 3.1, and the methodologies of organizing and executing the design in section 3.2. Next the overall structure and process of the project section 3.3, methods for data collection in sections 3.4 and 3.5, and finally the methods for developing the system in section 3.6.

3.1 Experimental Design

The design of the dashboard is used to show various features and elements to experts, who will provide their feedback and thoughts on each component. From this feedback and the results from the literature review, a guideline for timeline visualization will be derived and presented.

3.1.1 Design Alternatives

Before creating a design, the first step is to choose a method for designing it. There are multiple directions to consider, each with their own advantages and disadvantages.

Paper Prototype

The most primal solution is to create a paper prototype. For the paper prototype, a design will be drawn on paper, then presented to experts. There could be several paper pages, and each one might have slight differences depending on the features, for example navigating between pages on a website could be presented as two separate paper slices. A tooltip could be presented as a small paper slice that is put on top of the current paper. An advantage to paper prototyping is that it is fairly cost effective (Holzinger, 2004). This means a prototype can be made quickly without investing too much time into it.

Paper prototype is a great starting point to sort out rough UI/User Experience (UX) design choices, however it does not provide a realistic feeling of what it is supposed to represent, as Holzinger mentiones in Rapid prototyping for a virtual medical campus interface about testers not taking the test seriously enough (Holzinger, 2004). Because of this it might not be the best option as one of the focus points of the thesis is complicated timelines and graphs.

Sefelin et al. conducted a study comparing a computer prototype and a paper prototype. While the amount of feedback was the same, the subjects said they felt they had more freedom to explore during computer prototype testing. The study stated that the paper prototype made them feel like they caused extra work, and maybe most importantly, they feel more observed while testing a paper prototype (Sefelin et al., 2003).

Figma

Another option is to use a design tool such as Figma. Figma is essentially an online version of paper prototyping, that offers visualization, interactability and navigation[1]. Figma can be used for visualization of presentations, applications, websites and more. Similar to paper prototyping, Figma is cost effective to use. In addition it is more scalable as the prototype grows in size and features, and can handle change better (Sharma and Tiwari, 2021).

As the prototype scales it might be harder to maintain changes that are made. Figma solve this problem by enabling the user to use assets. For example a tooltip can be saved as an asset and copies can be used in multiple places. When a change in the saved asset is done, all of the copies will receive this change.

Figma could be suited for a research problem such as presented in this thesis, however the dataset received from HEMIT made it challenging. The dataset would need to be included in the Figma prototype, which would be a time consuming and redundant

[1] https://www.figma.com/design/

task. Therefore Figma might be better suited for prototyping UI/UX design. This leads to the next section about development.

Development

The last option that will be presented is developing a prototype. Out of the alternatives presented so far, development is by far the most cost inefficient to use for a prototype (Sedano et al., 2017). However it does come with a few advantages. Once a good foundation has been made, it can visualize a lot of data and it is more scalable than the other two options.

Since development is the most time inefficient option and big changes is cost heavy, the preceding work and analytics to determine what needs to be developed is more important than the other alternatives.

3.1.2 Why Development Was Chosen

The decision ultimately landed on development. This is because the data can be handled more easily programmatically, rather than by hand. The natural next step was to develop a program that can make use of analyzed data as well. Lastly, another advantage of developing a prototype is that it is scalable and can be easily tested in the future, which is not easily achievable with either of the two other options.

The alternative methods of how the dashboard can be developed, and how it will be developed for this thesis will further discussed in section 3.6.

3.2 Design Plan

When choosing development methodology there are multiple choices. The two main pillars is a linear approach like waterfall, or a more cyclic method, like agile methodology. Thesing et al. describes that there is no silver bullet that will always fit a project, and the methodology used may vary based on the criteria of the project (Thesing et al., 2021).

Furthermore Thesing et al. describes waterfall as being characterized to fit long term projects, backed by holistic planning and stability. Which is in stark contrast to agile project managements incremental, flexible and short-term oriented, based on a long term vision.

Based on the differences between the two, and the nature of the thesis, as every aspect can not be planned, specifically feedback given from experts based on the prototype, and changes based on that feedback. An agile methodology will fit the process of the project better, than a linear carefully planned approach such as waterfall, where each step should preferably be known.

During the process of developing the prototype, agile iterative design methodology was used. This methodology would give more data to be analyzed during the feedback loops. Feedback loops is a big reason of why agile development fits with the qualitative process. Feedback loops are sessions where as the name says feedback is given, which can help the project further by discovering changes quicker (Poppendieck and Cusumano, 2012). The feedback loop for this project is the expert reviews/demonstrations. As can be seen in figure 3.1, iterative design is a circular process of analyzing the data from a demonstration (RQ1 and RQ2), and using the analyzed data to improve and finally circling back to demonstration.

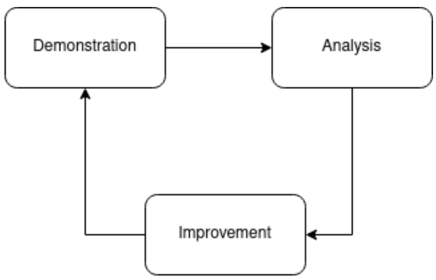

Figure 3.1: Image of the iterative process done during this project

3.3 Project Process

For efficient work, various research methodologies were discussed, among them were a process guideline presented by Kothari (Kothari, 2004), which can be seen in the figure 3.2.

This methodology was chosen on the grounds that the segments lined up with the iterative process, as well as the qualitative method that was chosen to be used during this project, more on this in section 3.5.

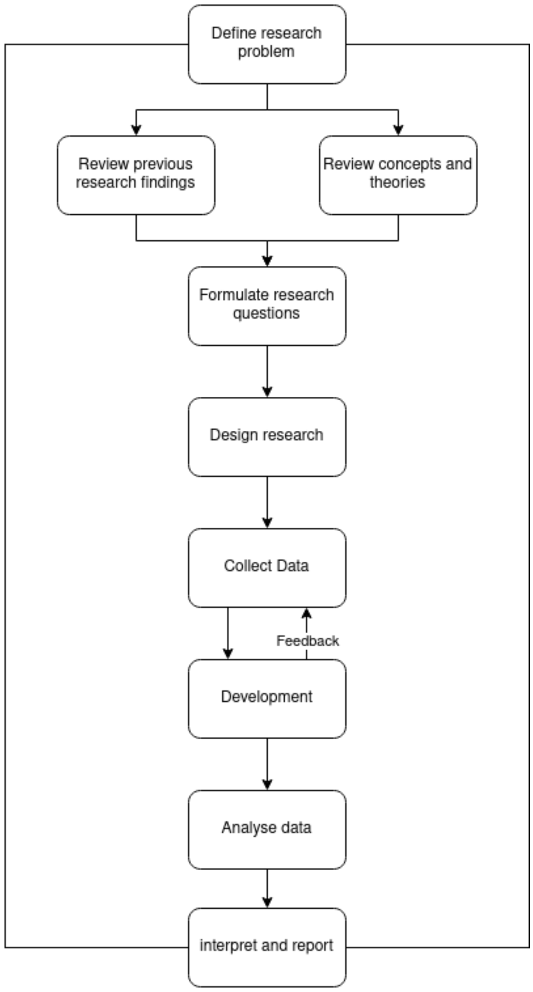

Figure 3.2: An overview of the process for the thesis

In the guideline, Kothari presents a seven steps in a research process. The first step defines the research problem. This is what the thesis is going to conduct the research on, which is how patient medical data visualization can be improved upon. From the research problem the next two steps, which are done in parallel, is to review previous research findings, concepts and theories. This is essentially the literature review, where similar existing literature and solutions are reviewed.

Based on the literature review, the next step of formulating the research questions can be done, which is the problems the thesis tries to answer. The next step is designing the research which involves how to retrieve information in the available time for research.

Once this has been established, the next step is collecting data. This is where interviews are conducted, to proceed with qualitative research with clinicians, more on this methodology in section 3.5. As stated by Kothari, the process is a guideline that can be modified to the needs of the project (Kothari, 2004). With this flexibility a development step was added to the process. During development there is feedback to collect data, which in turn gives more room to analyze data later in the process. Analyzing the data is an essential step to get a deeper understanding of the needs of the clinicians (RQ1 and RQ3), and thus come closer to a conclusion for the research questions. Lastly, interpreting and reporting the results is the final step, which will be presented in chapter 4 and 5.

The defined steps in the process guideline were used to plan each of the segments during the duration of the thesis. This was used to create an approximate time schedule. A gantt diagram was made to further plan the steps and distribute tasks throughout the time span of the project, as seen in figure 3.3. The figure shows an image of how long each period should roughly last to get the thesis done in time for delivery. When the planning phase of the project was done, the next phase of the project started by collecting data, which prompted development of the prototype. The development itself gave more data for which could be further analyzed in the next step. Lastly, the research questions were revisited to assess the overall project.

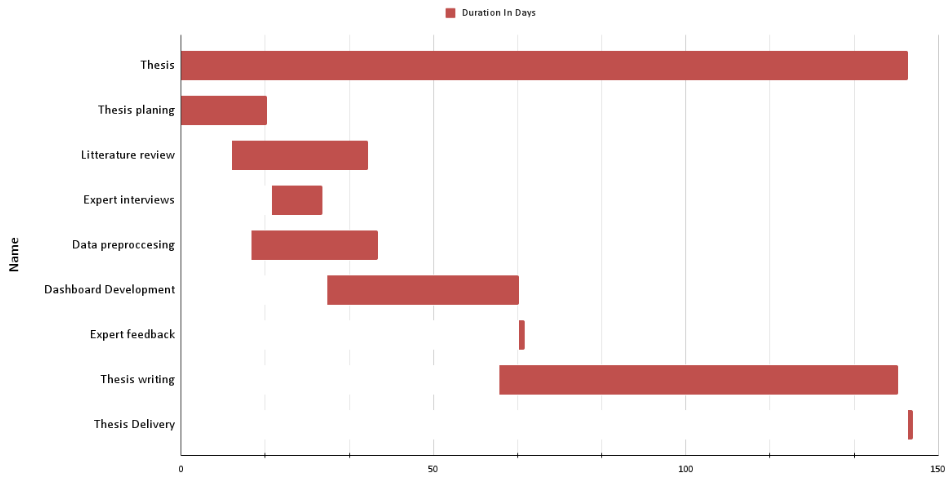

Figure 3.3: A gantt diagram

3.4 Literature Review

In order to gain insight and knowledge about similar solutions relevant to the project, particularly in the context of efficiently visualizing patient data in timelines, it becomes important to conduct a literature review Budgen and Brereton, 2006. The literature review serves multiple purposes. It allows to find inspiration, learn best practices, and obtain knowledge in user centered design. The literature review also helps discover techniques and strategies that contribute to quality improvement in healthcare through effective data visualization. In addition, valuable insights on how certain principles have been applied in previous solutions can be gained, learn from failures encountered, and draw meaningful insights from their findings. This literature review allows to address RQ2, and propose innovative solutions for visualizing patient data in timelines.

The first step was to establish keywords to search for, then where to look for it. Keywords selected was "Visualizing patient data", "visualization of electronic health records" and "data visualization in timelines and graphs".

The publishing databases used were Google Scholar [2], PubMed [3] and NTNU's university library [4]. Once an article was found, the decision to do a thorough read through or not was taken based on the abstract. If a publication was deemed to have a value for the thesis it would be selected for further review.

3.5 Interview with Experts

One of the methods selected for data collection in the project was semi-structured interviews, which allowed for qualitative research to be included into the study.

Semi-structured interviews are in-depth interviews where the interviewee answer open-ended questions. These interviews are characterized by a flexible and adaptable format, allowing for a open and relaxed interaction between the interviewer and the interviewee. Typically semi-structured interviews are conducted only once, and are used with individuals and/or groups of people. These types of interviews generally lasts 30 to 60 minutes (Jamshed, 2014). The main advantage of semi-structured interviews is their flexibility and adaptability, while still being considered a structured interview (Eriksson et al., 2005).

Semi-structured interviews were chosen as the preferred data collection method for several reasons. First and foremost they allow for in depth exploration of clinicians expectations of a sepsis dashboard (RQ1) and the importance of displaying previous episodes (RQ3). They can also contribute with valuable information which can be used to answer RQ2. Secondly, semi-structured interviews creates a flexible and interactive environment where the participants are able to express their thoughts, experiences, and preferences, as mentioned above. They can also help discover confusion and/or challenges with previous/current solution. Overall these types of interviews gives personalized, detailed information that helps enhance the relevance and validity of the research findings.

[2] https://scholar.google.com/
[3] https://pubmed.ncbi.nlm.nih.gov/
[4] https://www.ntnu.no/ub

Surveys were considered as an alternative data collection method. However it was deemed that this approach was not be suitable since collecting data by providing participants with a paper based or online questionnaires would not yield sufficiently informative responses. The complexity and details needed to answer the RQs simply did not suit the nature of surveys. Surveys lack the interactivity and flexibility in terms of being able to ask follow up questions and encourage an open discussion, which can restrict the very much important detailed elaboration and additional context of the participants answers (Oates et al., 2022). With this decision, interviews with experts, doctors and nurses were set up. Therefore the decision was made to use interviews as the data collection method.

Three interviews where conducted with five total participants. Two separate interviews with two different "overleger" (senior physicians), one of whom works at the infeksjonsavdeling (infection ward) and medisinsk overvåking (medical monitoring), while the other works at infeksjonsavdeling (infection ward) and conducting research. The last interview was conducted as a group interview with three nurses, with varying degree of education and experience, working at infeksjonsavdelingen at St. Olavs. The interviews were conducted in a controlled environment, specifically a closed meeting room, to ensure privacy and minimize distractions. All interviews were recorded with participants permission, allowing for accurate transcription. The recordings were later deleted, while the transcriptions were digitally saved for analysis and reference.

Review and Feedback of Dashboard

A decision was made not to conduct user testing on the dashboard due to its limited functionality. The dashboard is more a tool to monitor, analyze and observe, rather than containing a large set of features. With this it was decided that conducting heuristic evaluation through expert reviews would be more beneficial to the project.

Shneiderman et al. says in Designing the User Interface: Strategies for Effective Human Computer that a danger with expert reviews is that the experts may not have a good enough understanding of the domain. To avoid this however one can increase the chance of a successful expert review by inviting experts who have insight with the project and have good knowledge within the subject area. Another option was to conduct the dashboard review internally, but studies have shown that formal expert reviews are more effective (Shneiderman et al., 2016).

By the end of the project, expert reviews were set up with experts that had been a part of the project, either through participating in the interviews or as mentors withing the field. The primary objective of the review was to gather valuable feedback on the developed dashboard. The feedback obtained from the experts would serve as a basis for further work and improvements on the dashboard, aligning with the goal and research questions. It provides valuable insights into whether the implementation has met its objectives or deviated from the intended path.

Before the review started, the objective and reasoning of the dashboard was presented, to further reinforce the end goal and to avoid any confusion of its purpose. During the review of the dashboard, open ended questions were asked about the implemented features, in addition they were given the opportunity to comment at any point during the review.

3.6 System Development

After the literature review was conducted, various visualization techniques for best practices, user friendliness and more, were discovered which is important because having a good visual and responsive interface may result in better patient outcomes (Kossman and Scheidenhelm, 2008). To choose an appropriate technology for the project, several technologies were evaluated and assessed. Ultimately, the outcome was a dashboard that displays a timeline of different patient data, which was used for further analysis to identify possible improvements. More on final outcome of the dashboard in section 4.4.

The dashboard was developed with Vue, a JavaScript framework that is built on top of standard Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) and JavaScript. In addition to this, a webAPI was developed with ASP.NET Core and C# to handle processing and fetching of all patient data. The patient data is stored in a postgreSQL database on a HUNT cloud lab. HUNT cloud is a cloud service at NTNU, which offers research and a environment to explore sensitive data [5]. Hunt Cloud access was supported by REK approval 2018/1201, and effectuated by REK owner Lise T. Gustad.

3.6.1 Data Exploration and Data Preprocessing

As mentioned in section 2.1, the dataset provided by HEMIT was accessed from the internal kista in hunt cloud late January. The data was thoroughly explored to understand the meaning of different columns and their relationships. It was realized that arranging a meeting with HEMIT and contacts at Helsedirektoratet, Per Bjarte Sjømæling and Asbjørn Mikkelsen from LOGEX (the creators of the software NIMES), would be necessary to gain a complete understanding of the data. After several meetings, a rough overview of relevant tables and a description of columns was made. An overview of relevant tables, columns and an explanation can be seen below.

[5] (https://www.ntnu.edu/mh/huntcloud)

Pasient	
Column Name	**Explanation**
ppid	Unique patient identifier
fødtår	Year of birth, date and month not given
kjønn	Gender of the patient

Table 3.1: Patient Table

Nimesaktivitet: Table of episodes/interactions with healthcare	
Column Name	**Explanation**
ppid	Unique patient identifier
episodeid	Episode identifier, multiple episodes can have the same identifier
inndatotid	Date and time of episode start
utdatotid	Date and time of episode end
stdeptreshnavn	Location of episode, e.g. Ortopedisk prepoliklinikk Levanger
hastegradbeskrivelse	Urgency of the episode e.g. Akutt
fagområdebeskrivelse	Subject area of episode e.g. Generell kirurgi
pdxkoder	Code that represents the primary diagnosis for the episode
sdxkoder	Code that represents the bi-diagnosis
ncmpkoder	Code that represents the medication that was used in the episode

Table 3.2: Nimesaktivitet Table

Aninopphold: Table of hospital addmissions	
Column Name	**Explanation**
ppid	Unique patient identifier
aninoppholdstart	Episode identifier, multiple episodes can have the same identifier
aninoppholdslutt	Date and time of admission start
avdelingkode	Name of the ward e.g. StOlav Anestesi Gastro

Table 3.3: Aninopphold Table

Trfl: Table of blood sample results	
Column Name	**Explanation**
date_coll	Year of requisition
reqs_no	Requisition identifier
ppid	Unique patient identifier
ande_name	Name of sample value
anly_result	Value of the sample result

Table 3.4: Trfl Table

3.6.2 Data Selection

After gaining a comprehensive understanding of the dataset, the focus shifted towards identifying patients who were diagnosed with explicit sepsis. This involved using ICD-10 codes that had been previously used as explicit sepsis codes in other CoSeM projects. A screenshot of these is provided below, figure 3.4.

	Explicit sepsis	
Type	**ICD-10_CM**	**Description**
Explicit sepsis	A02.1	Salmonella sepsis
Explicit sepsis	A20.7	Septicemic plague
Explicit sepsis	A21.7	Generalized tularemia
Explicit sepsis	A22.7	Anthrax sepsis
Explicit sepsis	A24.1	Acute and fulminating melioidosis
Explicit sepsis	A26.7	Erysipelothrix sepsis
Explicit sepsis	A28.2	Extraintestinal yersiniosis
Explicit sepsis	A32.7	Listerial sepsis
Explicit sepsis	A39.4	Meningococcemia, unspecified
Explicit sepsis	A40	Streptococcal sepsis
Explicit sepsis	A41	Other sepsis
Explicit sepsis	A42.7	Actinomycotic sepsis
Explicit sepsis	A54.86	Gonococcal sepsis
Explicit sepsis	B00.7	Disseminated herpesviral disease
Explicit sepsis	B37.7	Candidal sepsis
Explicit sepsis	O85	Puerperal sepsis
Explicit sepsis	R65.2	Severe sepsis
	Implicit sepsis (infection code + organ dysfunction code required)	
Type	**ICD-10_CM**	**Description**
Infection	A00	Cholera
Infection	A01	Typhoid and paratyphoid fevers
Infection	A02	Other salmonella infections

Figure 3.4: Screenshot of ICD-10 codes marked as Explicit sepsis

It was discovered from the meetings, contacts at HP and LOGEX, that occurrences of sepsis could be identified through the table, nimesaktivitet, and its columns pdx and sdx-koder. Additionally, due to the hierarchical structure of the ICD-10 code system, sepsis could potentially be identified by querying three, four, and five-digit codes, depending on the level of detail recorded. This was confirmed by the company behind NIMES, Asbjørn Mikkelsen.

The codes were used to query the nimesaktivitet table and pdxkoder/sdxkoder columns in order to identify matching codes. When a match was found, the patient was labeled as having explicit sepsis. The results obtained from the queries varied depending on the approach used. When querying towards pdxkoder or sdxkoder, a total of 3726 patients were identified. However, when querying towards only pdxkoder (primary diagnosis), the number of patients identified decreased to 1794. More on how the database was altered to get these results can we seen in section 3.6.3. Below is a visualization of the data selection, figure 3.5. The SQL queries of how these results were achieved can be seen in appendix A.

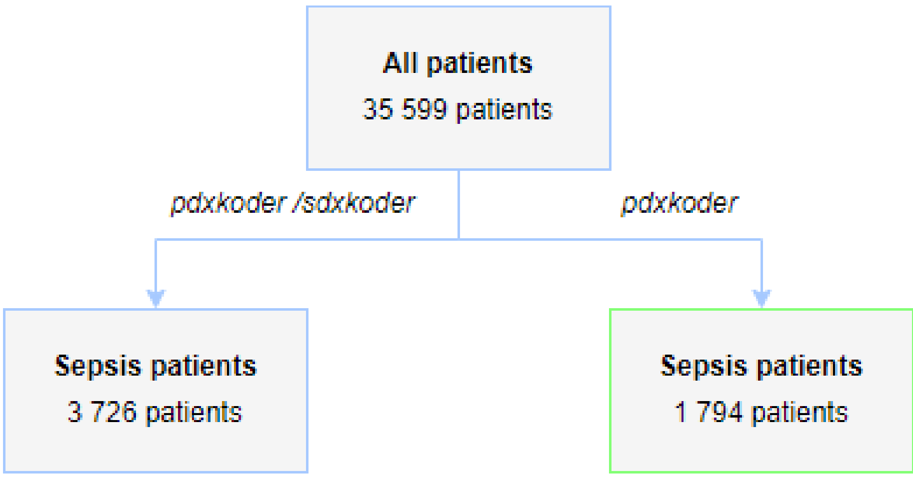

Figure 3.5: Data selection visualisation

3.6.3 Backend Architecture and Database

API

There are several benefits of developing an Application Programming Interface (API). One key advantage is the decoupling of the database schema from the clients, providing flexibility for modifications and enhancing security (Patni, 2017). Directly exposing the database to users can pose potential risks, especially when dealing with sensitive patient data, which is very important factor to take into consideration. However it is worth noting that building an API also requires additional effort, adds an extra step, and increases the overall implementation timeline. However, when implementing an API the application code does not need to change. The backend simply depends on a URL to retrieve the data from the API. This is advantageous, especially in the case where there may be changes in the database technology. Furthermore, integrating new applications with an API is a simpler process, offering increased convenience. Moving forwards, the choice was to develop an API.

The API was built using ASP.NET Core and C#. This technology was chosen due to the teams experience with it. Additionally, ASP.NET Core provides an out-of-the-box API template that requires minimal to no configuration to get up and running. The backend and API also uses dependency injection which promotes low coupling, making the code easier to test and improves the code maintainability (Biju, 2012). The API template also includes built-in support for Swagger, a web-based User Interface (UI) that makes documentation and endpoint testing easier.

Designing the API

There are several design patterns to consider when designing an API. Possibly the three most well known and documented are Representational state transfer (REST), Simple Object Access Protocol (SOAP) and GraphQL. Paul Logan talks about the advantages and disadvantages of these patterns in his blogpost. He says that REST is ideal for fetching data, when compared with SOAP. SOAP is often regarded as the oldest of the three choices, while GraphQL is recognized for combining the strengths of both SOAP and REST (Logan, 2018).

However, considering the use of FHIR and JavaScript Object Notation (JSON), the choice becomes clear. More on FHIR in subsection 3.6.3. REST and JSON goes hand in hand and work seamlessly together. Additionally, REST is widely considered as the most familiar and extensively documented pattern (Logan, 2018).

The design of the API follows RESTful architectural style and principles. The resources for the project primarily revolves around patients, but also vital signs, sample results, and others. According to REST principles, resources should be named using nouns and plurals in Uniform Resource Identifier (URI)s. Additionally, the URIs are hierarchically structured, for example /resource/id and /resource1/id/resource2. This approach helps to keep the web API intuitive, clean and understandable (Microsoft, 2023).

Microsoft includes a summarization of common conventions in terms of Hypertext Transfer Protocol (HTTP) methods and naming, on their documentation website which is open-sourced, that are adopted by most RESTful implementations (Microsoft, 2023). This can be seen in figure 3.6.

Resource	POST	GET	PUT	DELETE
/customers	Create a new customer	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1	Error	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1
/customers/1/orders	Create a new order for customer 1	Retrieve all orders for customer 1	Bulk update of orders for customer 1	Remove all orders for customer 1

Figure 3.6: Summarization of common API HTTP and naming conventions

The API implemented in this application only handles HTTP GET requests, and therefore none of the other HTTP methods needed to be considered. A screenshot of the application API Swagger page is provided below in figure 3.7.

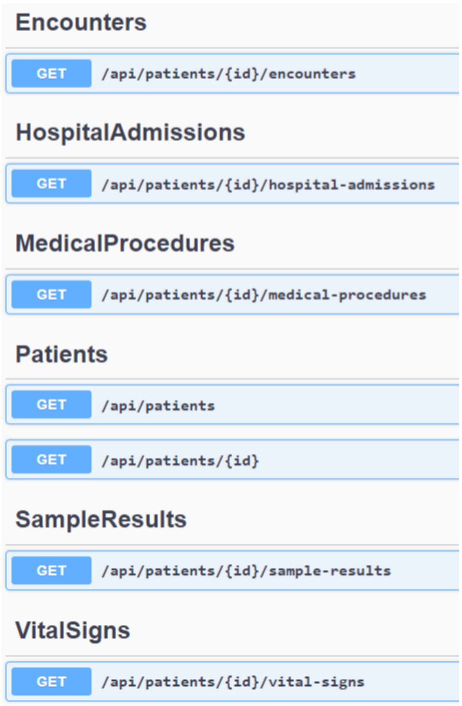

Figure 3.7: Screenshot of the API in Swagger

Architecture Design Pattern

The backend of the application was designed using the Command and Query Responsibility Segregation (CQRS) architectural pattern. CQRS is a pattern that separates the read and update operations. This is typically done by splitting up methods for reading and writing data so they are not kept in the same interface or class (Rajković et al., 2013).

While it might seem excessive to use CQRS in an application that only retrieves data, one could argue that the potential benefits of reusing the code in future projects justifies the implementation. By adopting a CQRS architecture, writing code that supports adding or updating data would be straightforward and no changing of existing code would be needed. Another advantage of using CQRS pattern includes better scalability, flexibility and adaptability of the application (Betts et al., 2013).

Fast Healthcare Interoperability Resources

Fast Healthcare Interoperability Resources (FHIR) is a standard that describes data formats and resources for information flow of health data. It was created by HL7 health-care-standards organization. The FHIR standard makes it so however data is stored on a system, one can expect the API to return the data in a predictable, structured and concise format. FHIR data is broken down into categories, for instance patients, encounters and observations and each of these categories is further defined within a FHIR resource (for e-helse, 2019).

There are tools to communicate with servers and databases that are FHIR compliant, however the database used for this project was postgreSQL, which was not made FHIR compliant. For that reason it was necessary to manually define and map the data into FHIR resources. The Hl7.Fhir.R4 nuget package [6] was used, which is Firelys official support SDK for working with HL7 FHIR in .NET. This package allowed for the utilization of already defined class models representing resources in FHIR and the JSON serializer. On the next page an example of the extraction of the data from the database and the convertion into FHIR resources is shown. Specifically, data from the trfl table is converted into the DiagnosticReport resource, which is an appropriate resource for storing blood sample data.

[6] https://www.nuget.org/packages/Hl7.Fhir.R4

```
public DiagnosticReport ConvertToDiagnosticReportResource(DbDataReader reader)
{
    var diagnosticReport = new DiagnosticReport
    {
        Id = $"{reader["reqs_no"]}",
        Subject = new ResourceReference($"Patient/{reader["ppid"]}"),
        Effective = new Period { Start = $"{reader["date_req"]}" },
        Contained = ConvertToListOfObservations(reader)
    };

    return diagnosticReport;
}

private List<Resource> ConvertToListOfObservations(DbDataReader reader)
{
    var obervations = new List<Resource>();
    var ande_name = $"{reader["ande_name"]}".Trim();
    string anly_result = $"{reader["anly_result"]}".Trim();


    var observation = new Observation
    {
        Id = $"{reader["reqs_no"]}",
        Code = new CodeableConcept(null, $"{reader["ande_code"]}", $"{ande_name}"),
        Value = new FhirString(anly_result)
    };

    obervations.Add(observation);

    return obervations;
}
```

To define the resources in the API, the HL7 FHIR website [7] was used to see a detailed description of elements included in the resources. The documentation gives an overview which elements one could include in the resources, cardinality, type and more. Validating the FHIR data was simply done using a web based FHIR Validator [8].

Database - PostgreSQL

The selection of the database was more or less set from the start, due to the nature of the dataset, which was provided as a PostgreSQL backup file. The process of transferring all the data into a database of a different technology would have been too time consuming for little gain.

Upon reviewing the dataset, it was noted that none of the tables had established primary and foreign keys, nor were there any indexes defined. It was decided that it would be beneficial to implement primary and foreign keys and introduce indexing.

While the primary keys in the nimesaktivitet table were based on a surrogate value/running number, the primary key in the pasient table was simply based on the patient ID, ppid. The primary key was created to enable the transfer of values from columns with repeating values to separate tables. This was necessary because searching for a value in a column with repeating values, such as the pdxkoder column containing 'A41,A40,O85', would be inaccurate. To address this issue, the values in these columns were moved to separate tables, such as the Diagnose table containing pdxkoder and sdxkoder columns, the medisinskprosedyre table containing ncmpkoder column, the kirurgiskprosedyre table containing ncspkoder"column, and the Skode table containing skoder column.

Creating distinct tables for repeating values enabled accurate searching for specific diagnoses in the database. In addition, indexes was created on foreign key columns to increase query performance in a relatively large database. Finally, tables for ICD10 codes were created to facilitate for easier querying for sepsis codes.

In summary, database normalization were used to address the issue of repeating values in the database, creating distinct tables for each set of values, and indexing foreign key columns to optimize query performance. Separate tables for ICD-10 codes were also added and populated to facilitate searching for specific codes. An ER-model of the restructured database can be seen below, figure 3.8.

[7] https://build.fhir.org/patient.html
[8] https://fhir.healthit.gov/validator/

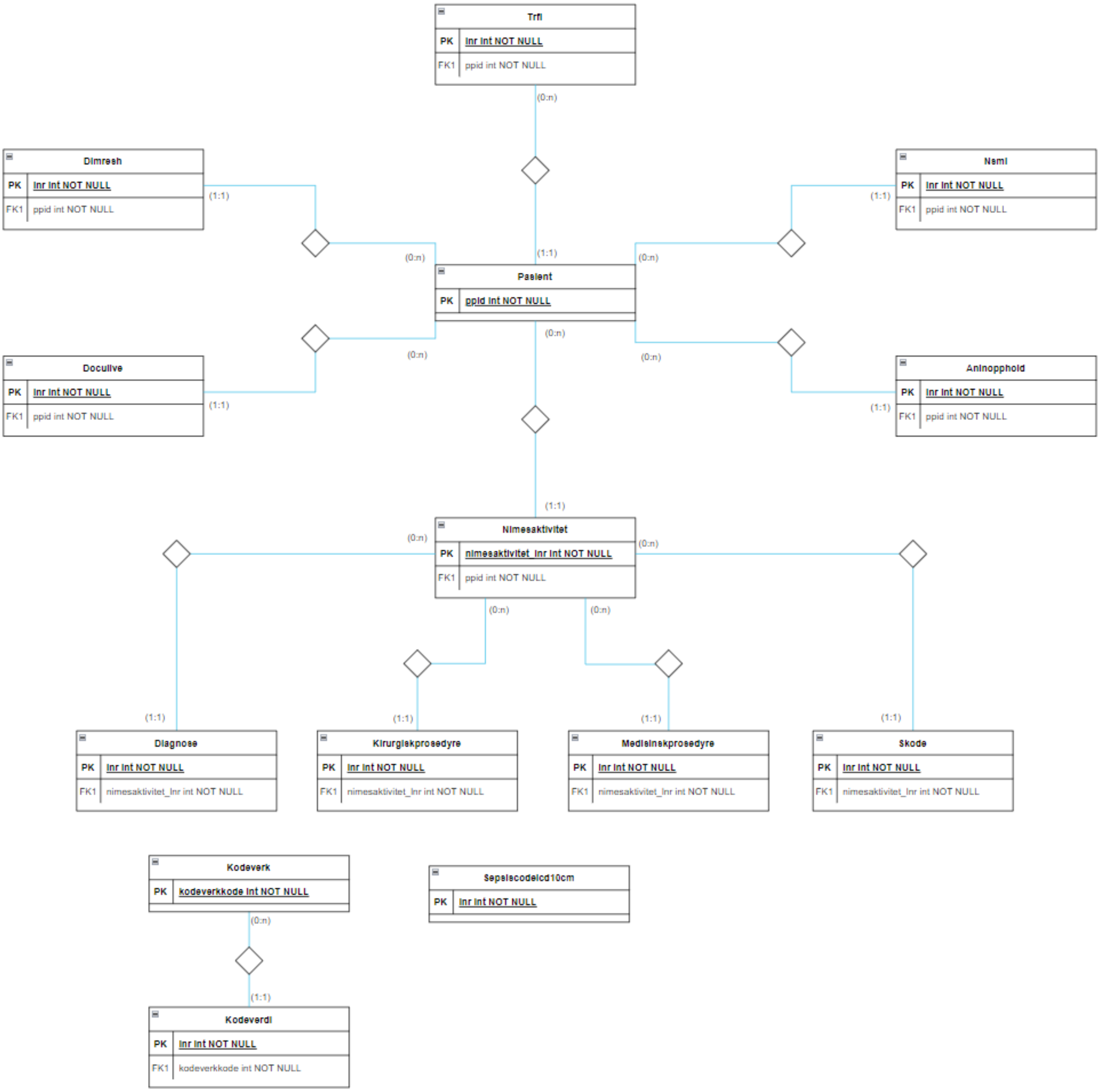

Figure 3.8: ER model - Restructured database of the provided dataset

3.6.4 Selecting Technology

As mentioned in section 3.1, the decision was to develop a prototype, and this leads to the next question of deciding what kind of technology should be used to develop the prototype.

There are many alternatives, ranging from web technology to desktop applications, and as each of these also have a considerable amount of options, it is virtually impossible to go through all of them. To make it easier the most popular options will be looked at in table 3.5.

> *A framework generally provides a skeletal abstraction of a solution to a number of problems that have some similarities. A framework will generally outline the steps or phases that must be followed in implementing a solution without getting into the details of what activities are done in each phase* (Mnkandla, 2009).

A framework will significantly speed up the development process, as it lays a foundation of code that can be used. Some of the criteria when choosing a framework, is how evolved or complete the framework is, amount of third party packages available, and previous experience with the framework. The completeness of the framework is measured by Github stars and percentage of trending posts on stack overflow, which gives a good indication of the amount of users using the framework. The criteria for having third party packages available, is to speed up the development process by using preexisting packages of timelines that can be used and customized to the needs of the thesis. Lastly, as a time constraint, a framework that the members of this project have previous experience with is a bonus but not a deciding factor.

Statistics over frameworks		
Framework name	**Github stars**	**Third party packages** [9]
React	208k	230000
Vue	37.9k	77000
Electron	108k	8400
Tauri	65k	150

Table 3.5: Statistics over frameworks as of 29th May 2023

[9] External packages found on Npm https://www.npmjs.com, keyword used to search is the framework name.

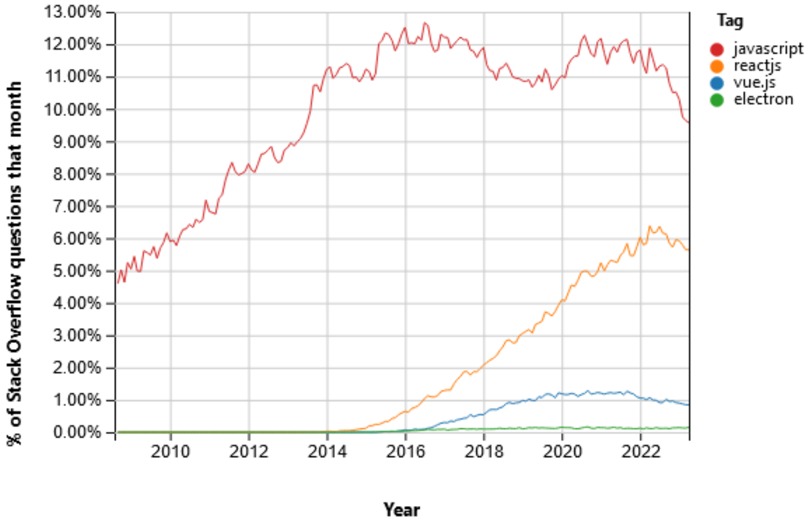

Figure 3.9: Stack Overflow trends

Vue and React are both JavaScript frameworks, and while it is not needed to use a framework when developing a website, it can potentially speed up the process. Javascript has roughly a 10% of the posts on stack overflow as can be seen in figure 3.9. While React is the most popular framework out of the options, with over 200 thousand github stars, and around 5% of new posts in stack overflow. Electron is a framework that allows developers to build cross-platform desktop applications, while still using HTML, CSS and Javascript. It uses the Chromium rendering engine combined with Node.Js runtime, which provides a environment to create desktop applications with web technologies. Notable usage of electron is Visual Studio Code, Discord, Twitch [10]. Electron has a fairly wide usage, and a is powerful tool. Lastly, Tauri is a new open source framework that is used to build standalone apps. It is built on top of Rust, and can use HTML, CSS and Javascript to build the frontend and is known for being a quick framework.

Why Vue

While the members have competence with react, it was decided not to use it. The biggest reason is the fact that React does not manipulate HTML directly, but instead uses JavaScript extendable mark-up language (JSX). This means if the HTML is going to be manipulated directly, the whole website will be re-rendered, which eventually causes slow downs. Vue on the other hand, allows to write HTML directly. This is advantageous since it can combine with any JavaScript third party package. Vue also has a healthy amount of users, and many third party packages available, which is a good indication that it is a thriving framework. The team initially had little

[10] https://www.electronjs.org/apps/

to no experience using Vue specifically, however the members has experience with JavaScript, HTML and CSS. Comparatively, the team had no competence using either Electron nor Tauri and its architecture. Because of time constraints and fewer third party packages, they were eventually taken out of consideration.

Leaving the last two options Vue and React. Vue is advertised as a lightweight and progressive framework. Vue employs Single-File Component (SFC), which combine HTML, CSS, and JavaScript in a single *.vue* file. SFCs can be reused multiple times throughout a project, resulting in more efficient development and less duplicate code.

Building the Timelines

To create the graph timelines used in the dashboard, the decision was made to use a third party package. The library used is called ChartJs [11], which is a free to use open source project that can also be found on Github.

The library can be used to create multiple different types of charts. The one used in the dashboard is a Line Chart. The line chart has a multitude of options that can be tweaked to design the chart in a way that suited the projects needs.

The code below describes the graph used for vital signs and sample results. It defines the chart as a time chart, which is a type of line chart, with the time unit defined as hours. The tick is the short black line with the label for every interval, which has a callback if the text should be bold.

The tick label will be bold if the tick is a major tick, as seen in line 7. Whether the tick is major or not is defined in the next callback function on line 13. This callback keeps track of if the interval has gone to a new day, if so it should be formatted as a date with day/month/hour, if not it should have the formatting as hour:minute.

```
1  const options = {
2      scales: {
3          xAxis: {
4              type: 'time',
5              position: "top",
6              ticks: {
7                 font: (context: any) => {
8                      const bolded = context.tick && context.tick.major
                        ↪  ? 'bold' : '';
9                      return {
10                         weight: bolded
11                     }
```

[11] https://www.chartjs.org/

```
                },
                callback: function (value: number, index: number, ticks: Tick[]) {
                    var newDay = false;
                    if (index == 0)
                        newDay = true;

                    const day = moment(value).day();
                    var lastTickDate = 0;
                    if (index !== 0) {
                        lastTickDate = moment(ticks[index-1].value).day();
                    }

                    if (day !== lastTickDate) {
                        newDay = true;
                        ticks[index].major = true;
                    }
                    if (newDay) return moment(value).format("DD/MM HH")
                        return moment(value).format("HH:mm")
                }
            }
            ,
            time: {
                unit: "hour"
            }
        }
    },
};
```

The next code segment describes how to add data set to the chart. This is done with the method *createDataSet* which takes a parameter key which is the name of the variable, and an array of records which is the data itself.

The data is sorted based on dates, which is then added to a new list, then added to the chart itself.

```
    createDataSet(key: string, rawData: Array<Record<string, {}>>) {
            const data: Array<{ y: number, x: string }> = [];

```

```
        rawData.sort((a, b) => new
        ↪ Moment(a.date).format('YYYYMMDD') - new
        ↪ Moment(b.date).format('YYYYMMDD'))

        rawData.forEach((d, index) => data.push({ y:
        ↪ (d.value.toString().replace(',', '.') as number), x:
        ↪ (moment().subtract(index, 'hour').format('YYYY-MM-DD
        ↪ HH:mm') as string) }));

        const dataset =
        {
            label: key,
            backgroundColor: this.colors[this.index],
            borderColor: this.colors[this.index],
            pointStyle: 'circle',
            fill: true,
            data: data
        };
        this.index++;
        this.chart?.data.datasets.push(dataset);
    },
```

Architecture

For the project Model View View-model (MVVM) has been used, this is an software architectural pattern used when developing user interfaces, as shown in figure 3.10. The model represents the data and business logic of the dashboard, while the view represents the user interface. The ViewModel acts as a mediator between the model and view, and it provides separation of concerns. Lou compared different architectural patterns, and one of the advantages with MVVM was testability, the business logic is separated from the view it makes it easier to test (Lou et al., 2016). Which in the context of EHRs is very important, since errors can cause an unknown amount of damage.

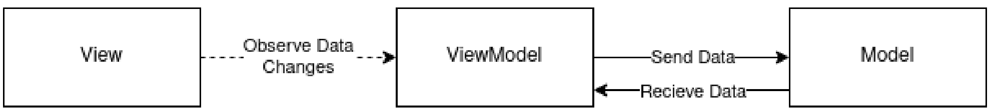

Figure 3.10: Overview of MVVM pattern

The ViewModel exposes data and actions to the model and also provides additional data and behaviour specific to the View, which reduces coupling between the View and Model. This results in being able to easily modify either view or model without breaking the other.

To keep the dashboard performant, the use of asynchronous requests to the backend when requesting patient data. An example here is in the PatientView.vue. Three requests are sent to the backend to get data about a patient, however these three data points are unrelated to each other which means it can be processed in parallel.

```
const data = await
↪ Promise.all([this.vitalSignService.GetVitalSign(this.patientId),
↪ this.medicationService.GetMedication(this.patientId),
↪ this.sampleService.GetSampleData(this.patientId)]);
```

This means each of the graphs and timelines can immediately display the data that has been requested once it is processed, instead of waiting until all of them are completed before it can display data. [12]

[12] Note when using Promise.all will fail immediately if one of the requests fail. Consider using Promise.allSettled for better error handling, as it will wait for all requests to finish.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/allSettled

CHAPTER 4

Results

The results of the project will be presented in this chapter. Section 4.1 covers the findings of literature review. The results from the interviews will be presented in section 4.2. Section 4.3 outlines the feedback from the expert review and lastly a prototype of the dashboard will be presented in section 4.4.

More on how the results from the literature review and interviews was applied to the sepsis dashboard in section 5.4.

4.1 Literature Review Results

A considerable amount of literature has been conducted on visualization of patient data. In the following section, a few relevant studies will be presented that share similar methodologies and/or project focuses, before going over relevant literature regarding visualizing of data in timelines and graphs, and finally going over relevant literature in terms of interface design. This section will mostly address RQ2 by gathering findings in existing literature. However inspiration and findings that support clinicians expectations and needs will also be considered in the prototype.

Review of Previous Solutions

Horbrugger et al. in the research paper Comprehensive Visualization of Longitudinal Patient Data for the Dermatological Oncological Tumor Board, used a visual approach where the interface is divided into three areas representing the respective underlying data. The three areas were, visualization of therapy data , visualization of important patient parameters and visualization of general patient data (Hörbrügger et al., 2020). The interface of their system was designed and evaluated with the help of domain experts in a qualitative user study. Figure 4.1 includes a screenshot of their solution.

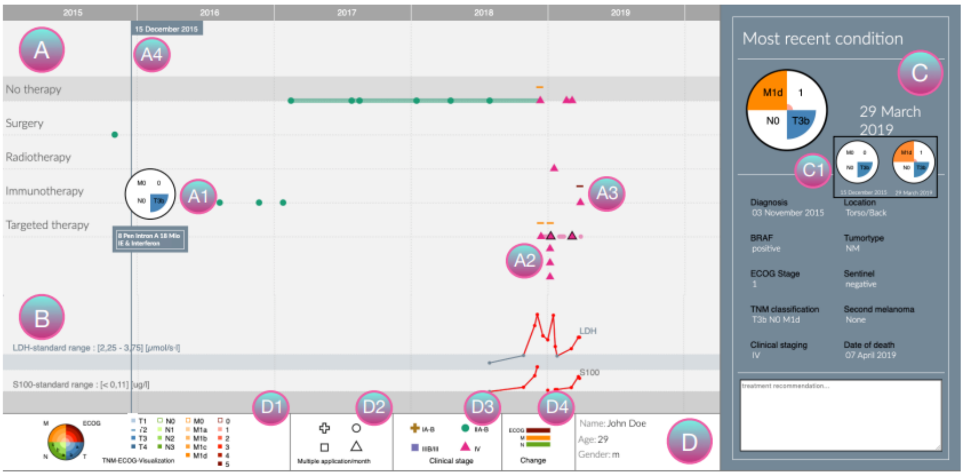

Figure 4.1: Horbrugger et al. graphical interface.

VisuExplore displays patient data with horizontal timelines. The system makes use of well known methods of visualization through event charts and timeline charts. Pohl et al. involved physicians in the development of VisuExplore through interviews and workshops (Pohl et al., 2011). There was an agreement among the physicians in their paper, that line plots where the best for visualizing data and that line plots are easily interpreted and can show development over time in a convincing manner, making it easy to identify outliers. It was found in their study that a great advantages of their system was that it allowed physicians to see the change and development of the patients at one glance and that the clinicians preferred an easy to learn and understandable design and familiar visualizations (Pohl et al., 2011). Figure 4.2 includes a screenshot of VisuExplore.

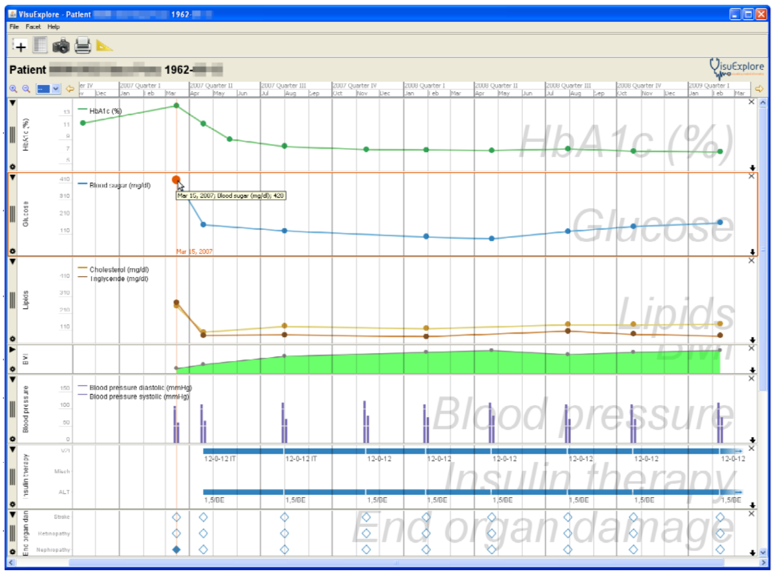

Figure 4.2: VisuExplore: visualization tool that displays patient data in timelines.

Lifelines uses a single screen interface approach for presenting their system. Lifelines features a zoomable timeline displaying patient history, allowing for the display of individual patient records. Their system enables the recognition of trends and the recognizing of important events (problems, allergies, diagnoses, etc). Colors have also been used to differentiate between the severity or type, for example severe allergies have been marked as red (Plaisant et al., 2003). Figure 4.3 includes a screenshot of Lifelines.

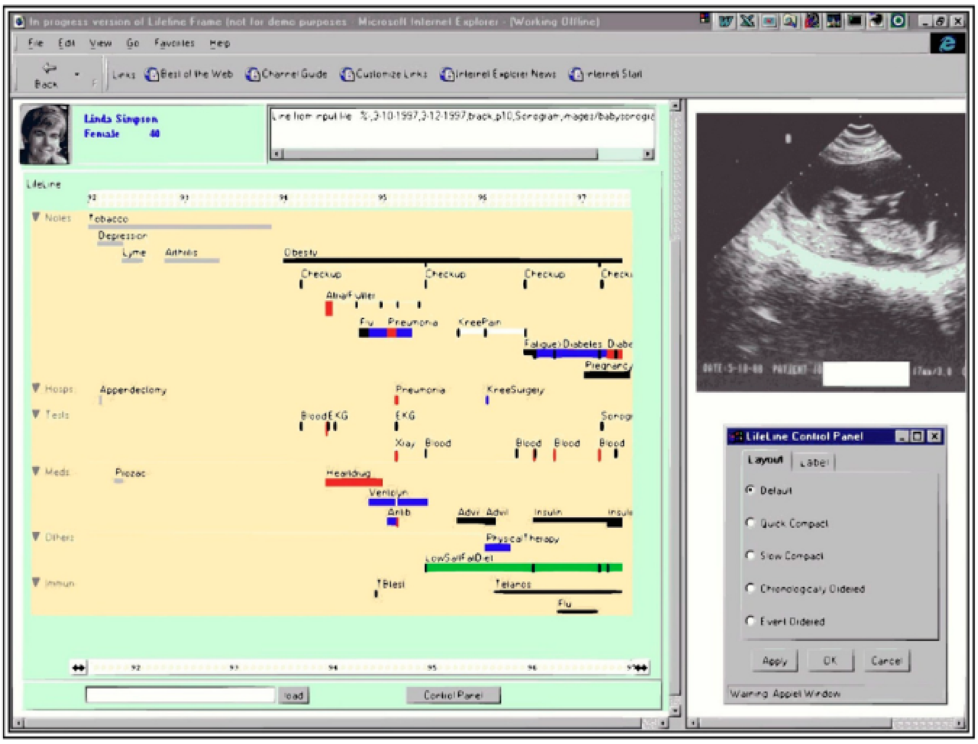

Figure 4.3: Lifelines: visualization tool.

Wongsuphasawat et al. developed LifeFlow, in which many also developed Lifelines, is a system for visualizing patient records on a single page that allows the user to see trends and evaluate quality of care. In their research paper Wong et al. mentions that new users can easily explore the data to understand patterns and trends at a high level. They found that their domain expert was able to understand LifeFlow quickly and that their system was useful to provide an overview, compare and measure the patient record (Wongsuphasawat et al., 2011). Figure 4.4 includes a screenshot of LifeFlow.

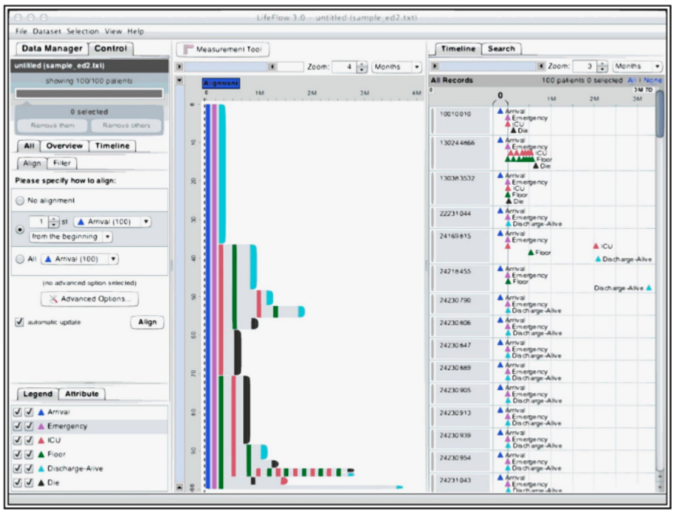

Figure 4.4: LifeFlow: Interactive visualization tool.

Data Visualization Through Timelines and Graphs

In Graphical summary of patient status by Powsner and Tufte, it is suggested that displaying patient test results and treatment data in graphs is of great benefit. One of the advantages is that graphs have tremendous potential for uncovering important and useful information in the data that might otherwise go unnoticed (Powsner and Tufte, 1994. Powsner and Tufte mentions that illustrating data through the use of graphs is common and that graphs make it easier to understand and explore comparisons, trends and associations. Visualization of data in healthcare through graphs and timelines has been increasing, and many tools have been created using longitudinal data, for example the most widely reported tool, LifeLines, as talked about earlier in the section.

A research paper published in 2015 by Sitting et al. focused on the graphical representation of diagnostic test results in EHRs. The authors evaluated and compared the graphical displays of laboratory test results in eight different EHR systems, against objective criteria for optimal graphs based on existing literature and expert opinions. In the paper it is mentioned that it is crucial to display and interpret clinical laboratory test results accurately to ensure patient safety. They highlight that a poorly designed interface could have serious consequences for clinical decision-making. The paper presents an overview of the criteria used to evaluate the graphical displays in EHRs. All of the criteria were aimed at ensuring compliance with widely recognized principles of effective data presentation (Sittig et al., 2015). Figure 4.5 presents the evaluation criteria used in the paper. A visual representation of the evaluation can be found in appendix B.

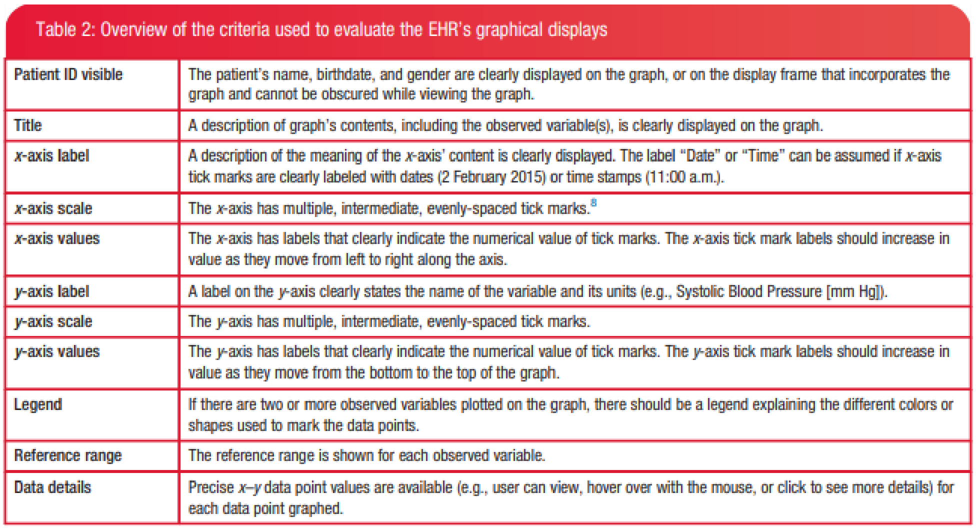

Table 2: Overview of the criteria used to evaluate the EHR's graphical displays

Patient ID visible	The patient's name, birthdate, and gender are clearly displayed on the graph, or on the display frame that incorporates the graph and cannot be obscured while viewing the graph.
Title	A description of graph's contents, including the observed variable(s), is clearly displayed on the graph.
***x*-axis label**	A description of the meaning of the *x*-axis' content is clearly displayed. The label "Date" or "Time" can be assumed if *x*-axis tick marks are clearly labeled with dates (2 February 2015) or time stamps (11:00 a.m.).
***x*-axis scale**	The *x*-axis has multiple, intermediate, evenly-spaced tick marks.[8]
***x*-axis values**	The *x*-axis has labels that clearly indicate the numerical value of tick marks. The *x*-axis tick mark labels should increase in value as they move from left to right along the axis.
***y*-axis label**	A label on the *y*-axis clearly states the name of the variable and its units (e.g., Systolic Blood Pressure [mm Hg]).
***y*-axis scale**	The *y*-axis has multiple, intermediate, evenly-spaced tick marks.
***y*-axis values**	The *y*-axis has labels that clearly indicate the numerical value of tick marks. The *y*-axis tick mark labels should increase in value as they move from the bottom to the top of the graph.
Legend	If there are two or more observed variables plotted on the graph, there should be a legend explaining the different colors or shapes used to mark the data points.
Reference range	The reference range is shown for each observed variable.
Data details	Precise *x*–*y* data point values are available (e.g., user can view, hover over with the mouse, or click to see more details) for each data point graphed.

Figure 4.5: Overview of evaluation criteria.

Several other papers mentions important features in timelines to be, zooming, more details on hover/click, filtering and more (Plaisant et al., 2003;Shneiderman, 1996;Wongsuphasawat et al., 2011). According to Shneiderman in his paper, filtering is one of the key ideas in information visualization, where users can quickly focus on what they find important by removing unwanted items. He also mentions that all of the features above are great ideas because they present information quickly and allow for user controlled exploration (Shneiderman, 1996).

Additionally many of these papers refer to the well known and often cited principle, visual information seeking mantra, including Nørdbø in his thesis (Nordb, 2006). This principle describes overview first, zoom and filter, then details on demand. To begin, it is important to provide an overview to guide the user on what to expect. After that it is important to organize relevant information and make sure that all details are available when the user needs them.

Interface Design and Principles

Shneiderman et al. talks about the eight golden rules of interface design. The focus is primary towards interfaces in general but principles from these rules can be applied to the sepsis dashboard as well. The eight golden rules of interface design is a set of principles that are applicable in most interfaces. These eight rules are defined and evaluated from experience, validation and design domains over 20 years. Shneiderman et al. mentions that these rules have been well received and can prove favorable as a guideline for designing user interfaces (Shneiderman et al., 2016).

Strive for consistency

This rule talks about the importance of striving for consistency throughout the interface. It can be consistency in terms of layout, spacing, naming conventions, interactions, fonts, colors and so on. Being consistent is important to reduce uncertainty, and it helps the user get a sense of familiarity of certain element, features and their purposes (Shneiderman et al., 2016).

Inclusive design

The second rule is about acknowledging and recognizing the needs of different users. It is important to cater to experienced users by providing shortcuts and methods for more efficient task completion. Catering and being considerate in design when in comes to users with disabilities and different technological expertise is a central idea in this rule.

Offer informative feedback

The rule of informative feedback involves giving the user feedback on user action. For example when hovering something that is clickable, appropriate feedback could be that the text changes color, opacity or gets outlined. This is important as it helps the user understand the consequences and confirmation of their actions (Shneiderman et al., 2016).

Design dialogs to yield closure

The interface should give clear indication on user action as to whether the sequence of actions is finished or not. The user should understand the flow of tasks and get the feeling of accomplishment and completion.

Prevent errors

The system and interface should be designed in such a way that it does not give the user the ability to make serious errors. The interface should anticipate and detect errors and give detailed, yet simple, explanation as to what went wrong and how to prevent said error.

Permit easy reversal of actions

Users should be able to reverse their step of actions. Meaning that users should be allowed to undo or redo their actions if they want. This relieves anxiety, knowing that the user can undo their actions, which also encourages the user exploration of lesser known features (Shneiderman et al., 2016).

Support internal locus of control

The seventh rule of supporting internal locus of control is about offering experienced users with features that enhances their sense of control of the interface. This can include offering customization options and adjustable setting. Shneiderman et al.

says that offering the sense of user control in the interface helps prevent dissatisfaction and it is encouraged to make users the initiators of actions rather then the responders (Shneiderman et al., 2016).

Reduce short-term memory load

The eight and final rule is about minimizing the cognitive load on the users. Interfaces should be kept simple and a single page interface is preferable compared to having the interface spread over multiple windows and pages. A few considerations have to be taken into account with this approach however. What is the purpose of the application, complexity of the content and user tasks are all factors to consider.

All of these eight principles form a guideline which can be followed, however the relevancy of each rule must be evaluated towards the needs of the application. Shneiderman et al. mentions that each rule has limitations and flaws, but they do however make a good starting point for interface design (Shneiderman et al., 2016).

Usage of Colors

Color is an important tool for interface design, aiding in presentation and visualization. It serves as a visual cue and enables intuitive variation, making it easier to identify patterns. Using colors effectively has numerous benefits, such as displaying multiple attributes simultaneously without causing confusion, facilitating the recognition of patterns and trends, and distinguishing labels from other elements (Einakian and Newman, 2019).

In Norbøs masters thesis on Information Visualization and the Electronic Health Record from 2006, he discusses the importance of color selection. He provides relevant guidelines for the project, emphasizing the need for a noticeable contrast in brightness between the foreground and background of the interface. Norbø also notes that the red-green color difference is generally regarded as most effective for individuals without color blindness. Additionally, he suggests using only five to ten color codes at a time for efficient information visualization (Nordb, 2006).

4.2 Interview Results

This section provides an overview of the interviews conducted, highlighting key points discussed by each interviewee. Similarities and differences in opinions are then analyzed and summed up. The section will primarily address RQ1 and RQ3 by gathering information from the interviews, however relevant answers for RQ2 can also be found here.

The results will be presented in the form of questions in cursive and summarized bullet-point responses and more in depths answers in nested squared bullet points. The summarizing of the interviews does not give the full picture of how the interviews was conducted, for a full overview of all of the planned questions, see appendix C.

To encourage a natural flow during the interviews, the questions during the interview may be slightly varied.

4.2.1 Interview One - Expert Opinion

Interview one was the first interview conducted. The participant is a senior physician and is known in this paper as an expert.

Question 1: *What specific information in terms of general patient information, do you consider important?*

- Name, date of birth, id
- Admission date of patient
  - ■ This information is helpful because there are often multiple doctors involved in a patients care.
- Preliminary diagnosis (what could be the possible problems)
  - ■ Knowing if there is a preliminary diagnosis or what the possible problems may be can be helpful. While the hypothesis or preliminary diagnosis may be incorrect or incomplete, having this information can help the doctor form a hypothesis about what is wrong with the patient whilst it can also be updated/changed.

Question 2: *What type of patient history do you want to be able to see and what type of data is important for you to see in a timeline?*

- Blood sampling results, vitals signs and medication/treatment
  - ■ Showing sequential blood tests in a timeline would be helpful, including CRP, leukocytes, creatinine, Hb and other parameters, depending on the type and location of the infection.
  - ■ It may be necessary to have different timelines for different types of data, such as one for vital signs and another for blood test values.
  - ■ Several parameters are important in sepsis, including blood pressure, heart rate, temperature, and oxygen saturation. It may not be necessary to display all parameters at the same time.

  - In terms of treatment, it is important to display the fluids and antibiotics being administered to the patient on a timeline, as well as when they started receiving the treatment.
  - Showing all medications the patient is on can be overwhelming. Could have a bigger focus on fluids, antibiotics and other important medication that the patient is currently on and has previously been taking. Knowing when treatment started and how long its been ongoing is relevant for assessing whether a change in dosage or medication is necessary.
- The expert discusses whether previous illnesses and episodes should be displayed on the dashboard and notes that although it may not be very relevant to have it on the first window, it could be included in a separate window.

Question 3: *How would you want to interact with the timeline?*

- Changing of intervals, zoom, colors, lines, hover information and selection of parameters (filter).
  - Choosing how much information the user want to see in each window, either hour by hour or with longer time intervals such as 2 hours, 6 hours, or 1 day, would be a good feature to include. This type of feature is useful when the patient is very ill over different time periods.
  - Would like if the user could zoom. Zooming could come in the form of scrolling or changing intervals.
  - For the vital sign and blood sample timelines a good implementation would be if the user could choose themselves which parameters to include and exclude.
  - Differentiating the graphs by using different colors for different parameters.
  - Include lines between points. To clearly indicate which data point is connected to each other, connecting the related dots to each other would be helpful and visually stimulating.
  - Enable detailed information to appear when hovering or clicking on a data point.

Question 4: *Is there a difference in needs across departments for such a system, e.g. intensive care vs infection ward?*

- There certainly are differences in needs across departments for a sepsis dashboard. For medical clinics and regular hospital wards, the needs for sepsis monitoring would be similar however.

- The needs for sepsis monitoring may be more complex for specialized departments like Intensive Care Units (ICU).
  - In ICU, blood gas analysis is a commonly used parameter for monitoring patients respiratory function and metabolic status. In these types of wards blood pressure is continuously measured. In infection ward blood pressure is usually checked sporadically.

Question 5: *What are your overall thoughts on HP and/or previous systems? What do you think could be improved upon?*

- There is a lot of information in HP, and it can be difficult to navigate. There is a desire for a dashboard to be simple and not overwhelming.
- There are too many clicks needed to perform certain actions.
- The module is not very intuitive.
- Should put emphasize on prioritizing simplicity and ease of use.

To summarize, the expert suggests that patient information such as name, date of birth, and admission date is important. It is mentioned that showing blood sampling results, vital signs, and medication/treatment in a timeline would be helpful. The user should be able to choose how much information they want to see in each window, zoom in, select parameters, and hover over a point to get detailed information. The expert suggest that HP can be overwhelming and not intuitive, and it is important to prioritize simplicity and ease of use in the design of the sepsis dashboard.

4.2.2 Interview Two - Doctor

Interview two was conducted with a senior physician that works 50/50 in the infeksjonsavdeling and medisinsk overvåking.

Question 1: *What specific information in terms of general patient information, do you consider important?*

- Name, age, gender, allergies and isolation due to infection.
- Admitted ward of patient.
  - It would be helpful to see an indication of which department the patient is admitted to.
- Only display relevant information.

  - Important to have only display necessary information on the screen without any unnecessary noise or information.

Question 2: *What type of patient history do you want to be able to see and what type of data is important for you to see in a timeline?*

- Blood sampling results, vitals signs, respiration and fluid balance.
  - Blood pressure, heart rate, temperature, alertness/mental state (Glashow Coma Score), and urine production are important factors to monitor.
  - Urine production is important because it indicates if the blood pressure is ”good enough” to produce urine and whether the kidneys are functioning well, as kidney failure is common with sepsis.
  - CRP, leukocytes, creatinine, and electrolytes are also important parameters to monitor.
- HP displays previous illnesses in a different tab, but it is not working very well. It is not important to display this kind of information on a sepsis dashboard. In the case that a clinician need to see this type of information he/she can access it elsewhere. It is therefore not necessary to display on this dashboard as it would be disrupting and would only create noise.

Question 3: *Is there a difference in needs across departments for such a system, e.g. intensive care vs infection ward?*

- There is likely a significant difference between general wards (sengepost) and monitoring wards (ICU). For example general wards do not have 'arteriekraner' which measures values in the blood every couple hours, and it is not important to monitor patients as closely in a infection ward.

Question 4: *What are your overall thoughts on HP and/or previous systems? What do you think could be improved upon?*

- The user interface is cluttered and noisy.
- The system is disorganized, and there are many ways to find the same information, making it difficult to understand the system.
- There could be fewer clicks, and information is sometimes located in unexpected places.

To summarize the interviewee says that important patient information includes name, age, gender, allergies, and isolation due to infection. Blood sampling results, vital signs, respiration, and fluid balance are important factors to monitor. It is discussed that there is a significant difference between general wards and monitoring wards in terms of system needs. HPs interface is cluttered and noisy and could be improved with fewer clicks and more organized information.

4.2.3 Interview Three - Nurses

The third and final interview was conducted as a group interview. The participants consists of three nurses, all of whom work at the infection ward at St.Olavs. They all have varying degree of experience ranging from 2-12 years of working as nurses.

Question 1: *What specific information in terms of general patient information, do you consider important?*

- Preliminary/relevant diagnosis (what could be the possible problems)
- Alertness/mental state (Glashow Coma Score)

Question 2: *What type of data is important for you to see in a timeline?*

- Vital signs, fluid balance and blood sampling results and medication/treatment.
    - Graphs are useful, and it is good to be able to see trends over several days.
    - Want to see antibiotics the patient is on, and previous medication.

Question 3: *How would you want to interact with the timeline?*

- Hover/click data points, changing of interval in graph and timeline and selection of parameters
    - Want to be able to interact with the system by hovering/clicking to see more detailed information on the data points in the graph.
    - Would be beneficial to be able to select and exclude parameters.
    - The time interval in HP is set too narrow, and one has to change it every time to get what they want to see. If not changed, one only gets the last 12 hours, which is too narrow on a general ward and might be better suited on ICU. It would be better to view the timeline over a longer period, 1-2 weeks, so that a trend in the graph can be seen.
    - Information should be sorted in a logical order, for example alphabetically.

Question 4: *What are your overall thoughts on HP and/or previous systems? What do you think could be improved upon and what should we avoid in our solution?*

- The platform is too cluttered and overwhelming with too much information.
- It is difficult to find relevant information as there are too many ways to access the same information in different places.
- The system could be more structured and organized to make it easier to navigate.
- Intervals on the graphs are too narrow in terms of the time interval shown.
- In case of sepsis patients, it would be beneficial to have the most important information gathered in one place to avoid searching for that information.

To summarize, they prefer to see vital signs, fluid balance, blood sampling results, medication/treatment, and antibiotics the patient is on in the patient history and data in a dashboard. They want to be able to interact with the timeline by hovering/clicking on data points and changing the time interval. They suggest having the most important information for sepsis patients gathered in one place to avoid searching for information.

4.2.4 Interviews Round-Up

The information needed to answer RQ1 and RQ3 are provided below, however for the sake of coherence in the thesis, a dedicated section has been devoted to directly addressing these research questions, section 6.1.

Patient information: All interviewees emphasize the importance of including patient information such as name, age, gender, and admission date in the sepsis dashboard. There were some differences in answers here, for example the first interviewee, emphasizes the inclusion of specific details such as the patients name, date of birth, and admission date. However, the second interviewee, a doctor, highlights the importance of additional information such as the patients allergies and isolation due to infection. The third interviewees, a group of nurses, does not explicitly mention these details but highlight the importance of a preliminary/relevant diagnosis and the patients alertness/mental state.

Previous illnesses and episodes: The expert suggests that although it may not be highly relevant for the first window, previous illnesses and episodes could be included in a separate window. The first and second interviewees both agreed that previous illnesses and episodes is not essential information to be shown on the dashboard. It could however be considered to be included in a separate window.

Vital signs and monitoring: Vital signs, including blood sampling results, and fluid balance/medication, are considered critical parameters to monitor in sepsis cases.

These measurements provide valuable insight into the patients condition and response to treatment.

Timeline visualization: A timeline graph format was suggested as a useful way to present data, particularly for blood sampling results, vital signs, and medication/treatment information. The ability to interact with the dashboard and timeline was highlighted as important. Users should be able to zoom in, select parameters, and hover over data points to access detailed information.

Simplified and organized interface: The interviewees express concerns about the current cluttered and noisy user interface in HP. They recommend reducing the number of clicks required and having better organizing information to improve usability.

Based on the common themes and answers in each interview, it is suggested that an effective sepsis dashboard should include detailed patient information, prioritize monitoring of vital signs, blood sampling and medication through timeline visualization, offer user interactivity for data exploration, and feature a simplified and organized interface. By addressing these points the sepsis dashboard can provide healthcare professionals with a valuable tool for managing sepsis cases efficiently and effectively.

4.3 Expert Feedback and Review

This chapter describes the results of the review conducted with the experts. The feedback that was given was used for improvements to the dashboard and future improvements that could be made. The final results after the review can be seen in chapter 4.4.

The review started with demonstrating the function of each part of the dashboard at the time and having a discussion on the respective features. The first prototype had a timeline, with a singular y-axis with multiple graphs. The y-axis ranged from the minimum value of the data set, to the maximum. The expert mentioned the need for the graph to display trends in order to enhance clarity and readability. The expert gave an idea of presenting the timeline with multiple y-axes. This was to better display changes in lab results that inherently have low values, ranging from 1-10, which was not easily observed if the maximum value was 500. The graphs with low values would therefore look like a flat line.

Before the final review, changes to the timeline was done to display more y-axes and to show trends of each of the values. The change is discussed further in chapter 5.4.2. Firstly, all of the features presented in the dashboard had high relevancy for monitoring patients, both general information and the graphs that may indicate ongoing sepsis, or the patients overall condition. The expert stated that there are 3 essential features that the prototype incorporated:

- Clinical examination
- Vital Signs
- Results of various samples

Even though the solution that exists today at St. Olavs hospital has some of the same features, the experts believed this solution presented the data better more effectively. They highlighted that having all the medical data presented on a single screen, instead of the information being spread across multiple tabs, improved the data presentation. This supports the thesis goal of presenting data in a clear manner, that avoids unnecessary clutter and information.

In addition to this, the experts was positive and happy that trends were presented in the timelines. This change and feature enables the user to view changes over time, compared to when the timeline displayed absolute values where changes may be hard to observe. Some of the other features such as being able to toggle graphs to hide or show lines, and the ability to change the time interval, was also a plus for the expert.

Improvements that could have been done is to better show which graph the y-axis belongs to. A suggestion that was given was to give the y-axis the same color as the graph it was meant to represent. Secondly, a graph could be highlighted and others be more transparent when a graph was hovered.

Another concern was abnormal values, meaning values outside of the normal range for vital signs or sample results. Currently the solution presents a graph with minimum and maximum values that is in the normal range. This means a value at the top of the graph means it is in the high end of the range, but still within the acceptable range. If the graph scales with the maximum range, it could still indicate that the value is in the high end of the normal range. Therefore the expert suggested that values outside the accepted range could be marked with a warning label or symbol, either in the graph and/or in the graph selector on the right-hand side of the graph.

4.4 Implementation

This chapter will present the results of the dashboard that was developed during the project. As discussed in earlier chapters, the implementation is a product of an iterative process. The final solution is presented in figure 4.6, while the iterative changes that has been made is presented in section 5.4.2.

4.4.1 Description of Dashboard Interface

The design of the dashboard was made after the interviews, which was based on expectations, needs and wishes of clinicians and experts (RQ1 and RQ3), and existing literature (RQ2). One interview participant envisioned three graphs each displaying a timeline of history of medications, a timeline of vital signs and lastly results of various lab tests, which was partly the final result of the dashboard, figure 4.6.

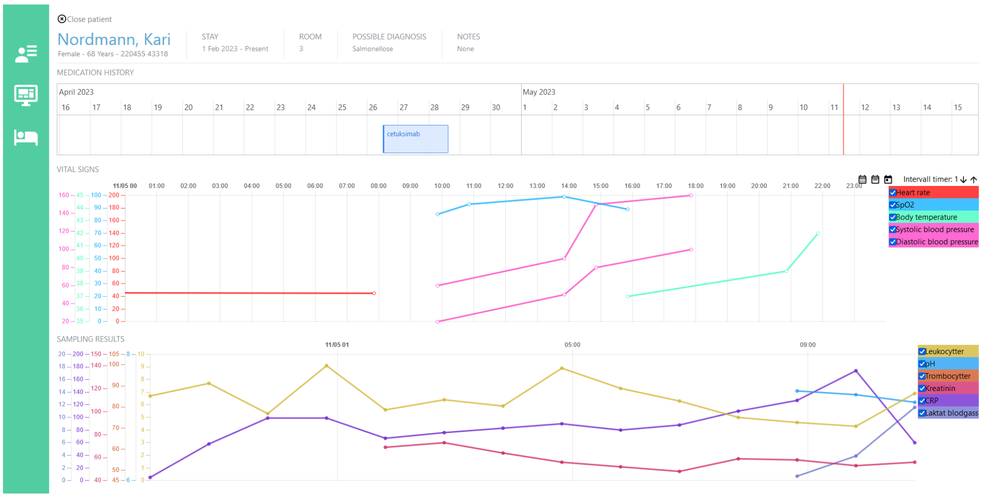

Figure 4.6: Image of dashboard after feedback

The top section of the dashboard features the patient card, which displays general patient information. To optimize space, the information is presented horizontally, based on similarities in wishes found in the interviews. The patients name is highlighted with a larger font size and a different color to provide instant recognition of the active patient.

Below the patient information card, a medication history timeline is included. This timeline visualizes the patients past and current medications, fluids, antibiotics etc. Medication events are represented by blue boxes, which can be clicked or hovered over to access more detailed information. One interviewee expressed the desire for timeline scrolling, which was implemented through the CTRL + Up/Down scroll functionality. This feature allows users to scroll within the medication timeline, dynamically adjusting the displayed dates and months to provide a comprehensive and easily understandable overview of the timeline.

Moving on to the vitals signs timeline graph. Much like a graph the y-axis displays the values of the parameters and the x-axis displays the date/time. A point brought up by experts is the ability to hide data that is currently not needed, this is done in the graphs by a list of toggles shown in the right hand side of a graph to toggle lines. The toggle has the same color as the line to indicate which line it belongs to. The intervals in the timeline can be changed by either clicking the interval arrow up or down, or clicking one of the three buttons to change the view from daily, to weekly or monthly.

Positioned at the bottom of the page, the timeline graph for sampling results mimics the layout of the previous timeline. It illustrates data points based on the date and their respective y-axis values. Just like the vitals signs timeline, the data points are connected by lines, fulfilling the request expressed in the interviews. Moreover, each line in the vital sign and sampling timeline graph is assigned a specific color. These color choices were influenced by existing solutions and the preferences of clinicians. For example, the heart rate line is displayed in red, aligning with clinicians expectations and making it easy to quickly identify. Further explanations regarding the choice to incorporate multiple y-axes will be provided in section 5.4.2.

Finally, on the left side of the screen, there is a navigation bar consisting of icons. The first icon is used to access the patient list, allowing the user to activate/view a new patient. The icon below it represents the dashboard, which is the current screen being displayed. Initially there was an icon intended for navigating to view previous illnesses or episodes, but this feature was later removed. Further details on this decision will be mentioned in in section 5.4.

4.4.2 Architectural Design

A visualization of the architecture of the system was made, figure 4.7 shows the relation between the dashboard itself, the webAPI and the database. The dashboard has a couple of services that may request certain data from the webAPI, which then does a query to the database. When the database returns a set of data, the API processes this data, and turns it into a FHIR format then sends it back to the service that requested the data. Once the service has received the data, the view will be notified and update the website accordingly.

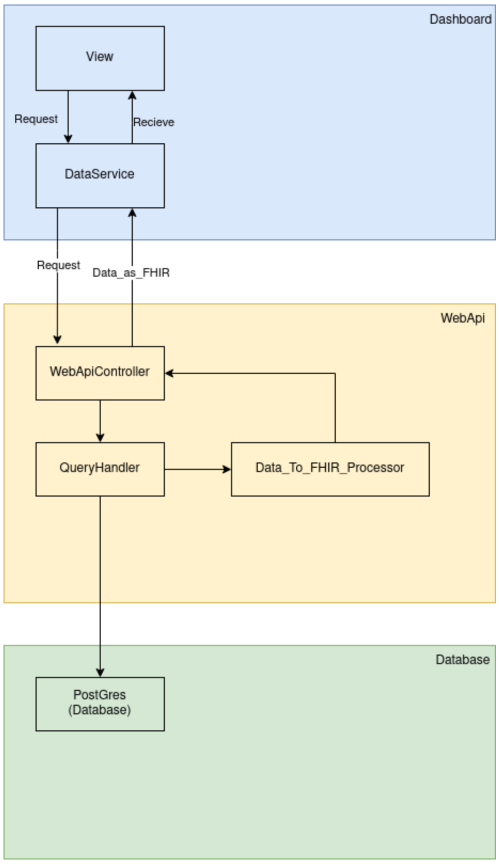

Figure 4.7: Image of the system architecture

CHAPTER 5

Discussion and Evaluation

During the project various methods has given valuable results, some of which were expected and some unexpected. To get a better understanding of the project, this chapter will discuss and evaluate the results. Section 5.1 will talk about deviations that happened during the process, and the effects of them. Section 5.2 and section 5.3 discusses results from the interviews and feedback from the expert reviews. An evaluation of the final product that was implemented in section 5.4 and lastly, the introduction of guidelines that was made based on key findings from the interviews and literature review, section 5.5.

5.1 Project Process

Overall, the scheduled project timeline and process were mostly completed as planned. However there were slight variations in the timing of the interviews, with the last interview taking place two weeks after the first one. Similarly, the development phase was delayed by one week which caused the expert review phase to be delayed as well, which was a phase that was inherently tough to schedule as it had to accommodate for the experts available times.

5.2 Interviews

The interview process proved to be successful, with the chosen semi-structured format allowing participants to freely express their expectations and needs on various topics. The pre-designed interview questions were generally appropriate, although some of the questions could have been phrased better to avoid leading the respondents answers. In Interaction Design: Beyond Human Computer Interaction by Sharp et al., it is emphasized that questions should not be framed in a way that influences/guides the answers in a certain direction (Sharp et al., 2007). For instance, one of the questions raised was related to important general patient information. When specific examples of various parameters were included in the questions, it is possible that the interviewees might have been influenced to respond using those same examples. This could have potentially introduced a leading element into the interviews.

Another approach that could have been taken was to avoid conducting the third interview as a group interview involving three nurses. During this interview, it became noticeable that the interviewees tended to quickly agree with each others answers without expanding on their own thoughts. To gain a better overview of the nurses expectations and gather more comprehensive data for analysis, separating the group interview into three individual interviews would have been preferable. Generally, it is acceptable for participants to agree with each other in group interviews, but it would have been more insightful to have each nurse respond independently in separate interviews rather than relying on agreement without further elaboration in a group setting. On the other hand, group interviews can have the advantage of being less threatening to the participants than individual interview when discussion among the participants is involved (Tjora, 2012).

Regarding the sample size of the interview participants, a total of five participants took part in three separate interviews. According to an article by Townsend about interview sample size in qualitative research, it is suggested that semi-structured interviews should ideally have a minimum sample size of between 5 and 25 (Townsend, 2013). Positioned at the lower end of this range, it would have been preferable to have a larger number of participants. Data saturation from the interviews was therefore not fully achieved as more insight from additional participants would have been preferable.

5.3 Expert Review and Feedback

The expert review phase included two experts, which may not have provided a comprehensive analysis. Research suggests that involving a few additional experts, typically around five evaluators, can identify roughly 75% of usability problems (Sharp et al., 2007). Therefore additional reviews with experts with different backgrounds would have been preferred. Another option is to organize a group session where the experts can openly discuss the dashboard among themselves. This approach encourages more extensive and in depth conversations that may not happen during one-to-one interviews (De Ruyter, 1996).

During the reviews, feedback was given on ensuring clarity of the dashboard, assessing the relevance of information presented by each feature, and evaluating the readability of the timelines. Ideally multiple rounds of reviews could have been initiated after changes was made to the dashboard. This could further reinforce the point of having a more in depth discussion about the dashboard, and would also present more data for analysis. Regardless, valuable data was given by the experts that was used to improve the design, and discussion for further work.

5.4 Implementation

From the start of the project it was a priority for the users to be involved during the project which in this case are the clinicians of a hospital. Involving the clinicians resulted in a reduced development time, as many of them presented their vision of such a dashboard. This meant less rounds of expert reviews were needed to get to the right track.

5.4.1 Technology

As mentioned previously, the frontend technology used for the dashboard was Vue. Vue supports using Typescript which ensures type safety. One advantage of type safety is that the properties of the object is known. Given the example below, line 5 will give an error that patient has no property called name, as the property has a capital N in *Name*.

```
1   class Patient {
2       Name: string = ;
3   }
4   const patient = new Patient();
5   const patientName : string = patient.name;
```

The biggest difference between TypeScript and JavaScript is that the latter will compile, and the variable patientName will be undefined, whilst TypeScript will not let it compile. Because of this, TypeScript made it easier to scale the dashboard, when combined with FHIR resources. The FHIR objects have many properties [1], which can

[1] https://www.hl7.org/fhir/patient.html

be seen in appendix D. These properties where exposed and known during compile time, thus avoiding errors related to undefined properties.

Type safety ensures scalability as changes to an objects property will be reflected during compilation, while JavaScript can let this fail silently. Which was an issue that arose was when using JavaScript third party resources, which does not have type safety. This produced compile time errors resulting in forcing to either fix the errors within the third party packs, or disable type safety for certain objects. Ultimately this enabled quick development, while also adding technological debt where errors on objects that does not have a defined type.

5.4.2 Dashboard

As previously mentioned, the design of the dashboard is a result of the interviews with clinicians and previous solutions and literature. This section focuses on how the literature, principles, guidelines and inspiration presented in section 3.4, was applied and used in the design of the dashboard. Firstly, this section goes into previous solutions and explores how specific elements from those systems were implemented into the sepsis dashboard. Finally, principles and criteria applied to the dashboard are presented.

One of the approaches that was followed was Horbruggers visual approach of dividing the interface into parts. The sepsis dashboard interface was divided into two main parts, the patient card (top part) and the patient data (middle and bottom part of the dashboard). Arguably the navigation section of the left side of the dashboard could be considered a part on the interface, however it does not hold any real value to the dashboard, as it is purely meant for navigation.

VisuExplore and Pohl et al. have a similar approach in their solution, however they do not explicitly mention it. They have a main information display consisting of graphs and a top section containing brief patient information. It does however appear that they fall short in effectively highlighting crucial information, particularly when it comes to the patients name. Plaisant et al., in Lifelines addresses this by highlighting the patient name in bright dark blue against a light greenish background, which draws the eyes towards the name, highlighting the importance of identifying the patient. As mentioned in the the literature review, usage of colors, section 4.1, Einakian and Newman supports that distinguishing labels from other elements is of great benefit. In the sepsis dashboard the patient name was not only differentiated from the rest of the content with colors, the font size was also increased to provide users with a clear indication of the patient they are viewing.

One of the main findings from the interviews was that the HP solution had too many number of clicks, making it cluttered and disorganized. To address this issue a decision was made to create an interface that prioritizes simplicity, a minimalist style which promotes user friendliness. The goal was to condense all the necessary information onto a single page, eliminating the need for excessive navigation and clicks. However fitting everything onto one page was a challenge, as having too much information could lead to clutter. Therefore, careful selection was made regarding what to include in the dashboard. Effective use of margins and spacing was also crucial to create a sense of spaciousness and openness.

During the interviews, all participants mentioned that timelines and graphs are the preferred methods for data visualization. This fits the finding from the literature and existing solutions reviewed in the literature review. The requested features mentioned in the interviews included the ability to zoom and scroll on the timeline to adjust the time interval for viewing less specific or more specific information, as well as filtering and accessing additional details through hovering or clicking. This approach is widely supported by existing literature, as referred to by the works of Plaisant, Shneiderman, and Wongsupohasawat in the literature review.

Principles and Criteria

When considering Shneidermans eight golden rules of interface design, some of the principles were not fully followed to certain extents. Specifically the inclusive design principle, which focuses on catering to users with disabilities, received less attention. To cater to users with disabilities, extensive research would have been necessary to understand various disabilities and find appropriate solutions, and due to time limitations it was not prioritized. It is worth noting that none of the interviews mentioned this aspect, however the responses might have been biased if none of the participants had disabilities. On another note, it was not specifically asked about either.
One feature that could have been easily implemented was a colorblind mode. In a dashboard where clinicians rely on different colors to identify trends and patterns in graphs, having a colorblind mode could be highly valuable for individuals struggling with colorblindness. Exploring this further in future projects could be interesting. A feature for shortcuts were implemented on the medication timeline, where the user can click CTRL+Scroll(Up/Down) to change interval/zoom.

The principle of designing dialogues to yield closure was deemed not relevant for the dashboard and was therefore not considered. Less focus was placed on error prevention in the dashboard due to the limited number of features. Restrictions were implemented on scrolling and changing of intervals in the timelines to prevent users from scrolling too far into the future, for example jumping ahead by several months or years.

Support internal locus of control was also arguably less considered. The user cannot customize and adjust many settings. However, one could argue that the filter of parameters is customization and adjusting settings. Dark mode could have been implemented as a customization tool, however it was not considered important based on the results from the interviews.

When comparing the developed system to the criteria given in figure 4.5, the dashboard scores a 9 out of the 11 possible points. The points which is not included into the timelines was *y-axis label and unit* and *Reference range.*

- Patient ID visible
  - Arguably, the criteria states having a patient id visibly at all times. While the timelines does not have that, the dashboard will always present a patient details card at the top of the window, while that patient is activated. Thus having a patient id in each timeline was decided to be unnecessary.
- y-axis label and unit
  - In a graph where there are multiple y-axes, the space the labels would occupy would drastically increase, which means a decrease of area where the graph itself takes place. To promote readability it was decided that the legend (filter box) on the right hand side does display the name of each graph and could therefore be considered name label for each y-axis. It does not however display the unit of the value, which could have been introduced, and therefore this point was not given.
- Reference range
  - It was concluded that including a reference range for every graph would be too much information on very little space and ultimately was deemed to stray away from this point to reduce unnecessary clutter.

Medication Timeline

The decisions around the design of the medication timeline was a discussion point. There were multiple ways that the medication timeline could have been designed. Ultimately, the design of the timeline was influenced by the need to fit all the necessary information on the page and the concern of it becoming overwhelming with too much details.

A possible approach would have been to organize related medications and fluids into groups on the timeline. By arranging multiple groups vertically a more detailed timeline could be achieved. An example illustrating this concept is provided in figure 5.1.

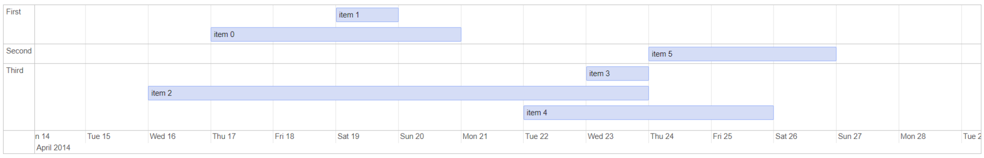

Figure 5.1: Alternative medication timeline

However when trying to fit everything into one screen this becomes a problem. It takes up too much space. What we also could have done it simply reduce the space the graph timelines takes and give the medication timeline a bit more space. However, it was discovered that implementing this change resulted in users needing to scroll vertically within the medication timeline to view all the groups. Avoiding this was important, as users typically prefer to see the entire picture without the need for manual scrolling. This aligns with Plaisants perspective in Lifelines, where he says:

> *Scroll bars are the common answer to pixel shortage but scrollbars are inadequate if not harmful - when presenting overviews as users often forget to browse the complete image, or worse are even unaware that part of the image is hidden.* (Plaisant et al., 2003)

Therefore the decision was made to not pursue this approach. Although no negative feedback was received regarding the current design of the medication timeline, it is worth exploring the potential benefits of grouping relevant medications into group vertically further.

Changes Based on Feedback

As previously stated, the implementation underwent some changes, based on the feedback given during the expert review. Changes that was done are presented below.

Trend graphs

The most significant change that was implemented in the system revolved around how the graph displayed the values. In scenarios where a graph has multiple axes, each with different value scales, the aim was to gather feedback from experts to identify their priorities. Surprisingly the experts expressed that knowing the exact value of each data point was not necessarily their primary concern. Instead they highlighted the importance of understanding the trend and how the condition of a patient evolved over time. By focusing on the trend rather than the precise value at each point, the experts argued that they could gain valuable insights into the patients changing health status. Trends provide a more intuitive understanding of the patients condition, enabling healthcare professionals to make informed decisions and monitor progress more efficiently.

To illustrate this point further, consider a graph displaying multiple lines representing various data values with the same scale, thereby reflecting absolute values. In such cases, data values that fluctuate within the range of 1-10 could exhibit a substantial increase from three to eight. However, if the axis scale were set from 1-1000, this increase might appear as a flat line on the graph, obscuring the magnitude of the change.

To overcome this limitation and ensure accurate representation of the data, the dashboard implemented a new approach. It introduced additional axes for each sample, effectively expanding the graphs visual space. This design choice allowed for a more precise depiction of the values within the normal range for each respective sample. Consequently when a data point appears near the top of the graph, it indicates that the value is close to the maximal value within the normal range for the respective sample. By adopting this approach, the system addresses the challenge of displaying values within varying scales and provided a clearer understanding of the evolving patient condition over time.

In addition to the change to present trends, the colors of each axis tick and text were changed to represent which graph it belonged to, as can be observed in figure 5.2.

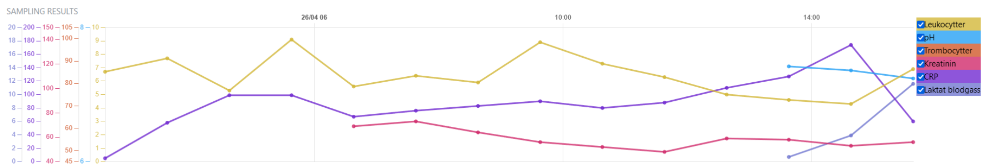

Figure 5.2: Graph showing normalized values of sample results

5.5 Guidelines for Timeline Visualization

Throughout the project it became clear that there was no specific guideline available that catered directly to this projects needs. Relevant theories, criteria, and guidelines had to be gathered from various research papers and evaluated against the project. Additionally, input from clinicians was taken into consideration when developing the dashboard. A decision was therefore made to contribute in making guidelines for similar projects related to sepsis timeline visualization.

Based on the findings from the literature review and the analysis of interview results, the most relevant and significant points were identified and put together into a set of guidelines. These guidelines can serve as a starting point for future projects and can be further refined and enhanced. It is crucial to evaluate each point and consider its relevance to the project. The guidelines can be seen in table 5.1.

Timeline Visualization Guidelines	
Granularity	The time axis should be able to be re-scaled, to show smaller or bigger date range.
Icon consistency	A single icon should always mean the same thing.
Axis to graph link	A graph line should have the same color as the y-axis line.
Detail	Timeline should express more details on demand. Either through click or hover on data point
Filtering and Legend	The ability to toggle unwanted graphs. One way to present this information is by using a filterable legend.
Highlighting	When hovering above a graph, the other graphs should have a lower transparency.
Title	The timeline should have a title which is informative to what it represents.
Multi y-axis	When the timeline includes multiple graphs with different value scales, it is important to use multiple y-axes. This ensures better accuracy and precision of the graph, and help user observes trends and patterns better. It is important to evaluate and consider the goal of the timeline. For example, when clinicians observes a timeline, they want to see a trend.
x and y-axis labels	Displaying labels for the x and y-axis is recommended, but it is crucial to consider the impact on clarity and clutter of the timeline when including these labels. If x-axis is Date or Time, it can be considered to not include the label as long as the tick marks as labeled with dates or timestamps
y-axis units	It is recommended to include units on the y-axis, providing users with clear context of the data being presented.

Table 5.1: Timeline Visualization Guidelines

Some of the points in the guidelines were not implemented into the dashboard. This was not due to negligence, but rather because some of the points were identified as issues during the expert interviews, which occurred after the development phase was over. As a result, these points emerged after the development period had ended.

These points were;

- Icons
- Highlighting
- y-axis units and labels*

Arguably, there are y-axis labels, however they were moved from the axis to the filtering box, to avoid clutter and increase readability, as mentioned in section 5.4. The color of the y-axis, graph and background of the box is the same, to be able to visually identify these as being linked to the same information. Highlighting and Icons was discussed in the expert reviews, and included into the guidelines, but not implemented in the dashboard due to time restrictions.

CHAPTER 6

Conclusion

The implementation of EHRs has undoubtedly been useful for the health personnel, especially when documenting medical information about patients. As development within this area continues to grow and is implemented in hospitals, the need to present medical data efficiently also increases. Throughout the years there have been many different systems developed to utilize medical data. A risk with dashboards is bloated software that contains a ton of features, with medical data that may be found in multiple places. This thesis focused on developing a dashboard along with clinicians, where they have input on how the dashboard can effectively and efficiently present data. This method allowed the dashboard to expose the most relevant patient data to clinicians. While also avoiding unnecessary features that the experts deemed non important. The end result is a sepsis dashboard that displays a trajectory of a patients current status and trends.

6.1 Answers to Research Questions

This section revisits the research questions mentioned in section 1.2 and provide concise answers to them.

RQ1: What are the clinicians expectations of a system like this and what medical information is vital to display?

The overall consensus among clinicians regarding their expectations of a sepsis dashboard is primarily focused on having a user-friendly interface. All of the participants interviewed emphasized the importance of having a well organized dashboard that is not cluttered and avoids too many clicks. They expressed the need for all relevant information in terms of sepsis patients to be gathered in one place rather than spread across multiple windows. Additionally, they highlighted the importance of prioritizing simplicity and ease of use in the dashboard design.

In addition, it is expected that the dashboard will offer interactive features. This includes the ability to change intervals and scroll through the timeline, as well as the option to click or hover over data points for more detailed information. Clinicians also anticipate the ability to filter parameters in the timeline graphs, allowing them to choose which specific parameters they want to view. Furthermore, it is expected that the vital sign and sample results graphs use different colors to easily distinguish between different parameters.

When considering the essential medical information to display, there is a general an agreement among the clinicians. They prioritize having vital signs, sampling results, and medication histories available on the dashboard. Specifically, regarding medication history, clinicians desire the ability to view both current and past medications and fluids administered to the patient. Furthermore, displaying general patient information is also crucial. This includes key data such as the patients name, date of birth, submission date, mental state, known allergies, and preliminary or relevant diagnosis.

RQ2: How can patient data be efficiently visualized in timelines?

Based on the reviewed literature there are several important factors to consider for effective and efficient data visualization using timelines and graphs. These factors include providing clear information on the timeline, such as labeling the x and y-axis, including units for the x and y-axis, and using evenly spaced tick marks. It is also essential to have a legend that explains the graph lines used. Adding a descriptive title to the timeline helps users understand the content of the graph. Additionally, allowing users to access more details by hovering or clicking on data points in the graph enhances their experience. Features like zooming and filtering, which are fundamental in information visualization, should also be included.

From clinicians perspective it is important to get a quick overview of the patient, humans have great perceptual ability, which can be utilized greatly in a timeline, where changes occur. In a timeline, multiple graphs representing specific parameters can be included, each assigned a designated icon and color. Over time, clinicians can recognize a specific icon and/or color with certain parameters. Timelines improves understanding of clinical data, which helps clinicians recognize patterns. Data should be presented as a trend, where granularity may be changed by the user. By viewing a trend, the clinician can quickly assess how the patients status has changed.

When dealing with multiple parameters with different value ranges it is important to use multiple y-axis. This ensures that each graph is readable, as opposed to if all data sets shared a single y-axis. Unless data sets shares roughly the same normal ranges, they should all have their own y-axis. Additionally, to enhance readability and facilitate a quick understanding of graph values, the y-axis lines and value labels should match the color of their respective graph lines.

Furthermore, a timeline can become crowded with multiple graphs. To isolate a specific graph, it is ideal to hide other graphs using filters or by reducing their transparency when hovering the mouse over a particular graph. This approach improves visibility and clarity when analyzing trends.

Additionally we created guidelines that were based on the similarities in the literature review and interviews, that was relevant to the project. These guidelines are meant as a rough starting point for relevant future projects and aim to guide how one can visualize patient data efficiently in timelines. The guidelines table can be seen figure 5.1.

RQ3: Is showing previous episodes in a timeline important for a clinician?

During the interviews it was discovered that the current solution, HP, utilized at St. Olavs Hospital, attempts to address these requirements. However, it was reported that the current implementation is not functioning effectively. Multiple interview participants mentioned that this information is not considered particularly relevant to include in the dashboard. While it could potentially be included in a separate window, there were concerns that incorporating it into the sepsis dashboard might introduce more noise and clutter, potentially causing more harm than benefit in terms of usability and clarity.

6.2 Future Work

In order to enhance the usability and effectiveness of the dashboard, future work should center around analyzing how clinicians interact with it and addressing specific concerns related to data presentation, clarity, and readability.

One area of focus for future work is understanding how clinicians prefer values outside the normal range to be presented. This concern arises from the need to ensure that clinicians can quickly and accurately identify data points that fall outside the expected ranges. By conducting interviews and gathering feedback from clinicians, valuable insights into their preferences and expectations regarding the visual representation of abnormal values can be obtained. This information can then guide the development of design strategies that effectively highlight and differentiate these outliers, ensuring that clinicians can easily spot and interpret them.

To improve the clarity and readability of the timelines within the dashboard, instead of the current value points which is a circle, can be changed to a specific icon for each graph. Icons can serve as visual cues to understand trends. By associating meaningful icons with certain data points, clinicians can quickly grasp the relevant information without solely relying on color, or hovering above points to read the value. Icons can provide a visual shorthand, enabling clinicians to interpret the data at a glance and make informed decisions more efficiently.

Furthermore, it is worth considering that there are three distinct timelines, each potentially operating on different timescales. This raises the question of whether the timescales should be connected or adjustable individually. Resolving this issue requires further research and user testing of the dashboard to observe user behavior and identify potential problems.

Bibliography

Betts, D., Dominguez, J., Melnik, G., Simonazzi, F., & Subramanian, M. (2013). *Exploring cqrs and event sourcing: A journey into high scalability, availability, and maintainability with windows azure* (1st). Microsoft patterns amp; practices.

Biju, S. M. (2012). Dependency injection for loose coupling of objects.

Bowman, S. (2013). Impact of electronic health record systems on information integrity: Quality and safety implications. *Perspectives in health information management*, *10*(Fall).

Budgen, D., & Brereton, P. (2006). Performing systematic literature reviews in software engineering. *Proceedings of the 28th International Conference on Software Engineering*, 1051–1052. https://doi.org/10.1145/1134285.1134500

Campanella, P., Lovato, E., Marone, C., Fallacara, L., Mancuso, A., Ricciardi, W., & Specchia, M. (2015). The impact of electronic health records on healthcare quality: A systematic review and meta-analysis. *European journal of public health*. https://doi.org/10.1093/eurpub/ckv122

De Ruyter, K. (1996). Focus versus nominal group interviews: A comparative analysis. *Marketing intelligence & planning*, *14*(6), 44–50.

Einakian, S., & Newman, T. S. (2019). An examination of color theories in map-based information visualization. *Journal of Computer Languages*, *51*, 143–153.

Eriksson, E., Bälter, O., Engwall, O., Öster, A.-M., & Kjellström, H. (2005). Design recommendations for a computer-based speech training system based on end-user interviews. *Proceedings of the Tenth International Conference on Speech and Computers*, 483–486.

for e-helse, D. (2019). *Anbefaling om bruk av hl7 fhir for datadeling*. https://www.ehelse.no/standardisering/standarder/anbefaling-om-bruk-av-hl7-fhir-for-datadeling (accessed 02/05/23)

Holzinger, A. (2004). Rapid prototyping for a virtual medical campus interface. *IEEE Software*, *21*(1), 92–99. https://doi.org/10.1109/MS.2004.1259241

Hörbrügger, N. S. M., Braun, A., Tüting, T., Oeltze-Jafra, S., & Müller, J. (2020). Comprehensive visualization of longitudinal patient data for the dermatological oncological tumor board. *Proc. of EuroVis*.

Jamshed, S. (2014). Qualitative research method-interviewing and observation. *Journal of basic and clinical pharmacy*, *5*(4), 87.

KIN. (2020). *Computational sepsis mining and modelling*. https://www.ntnu.edu/cosem#/view/publications (accessed 11/04/23)

Kossman, S. P., & Scheidenhelm, S. L. (2008). Nurses' perceptions of the impact of electronic health records on work and patient outcomes. *CIN: Computers, Informatics, Nursing*, *26*(2), 69–77.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*. New Age International.

Logan, P. (2018). *Rest, soap, graphql — gesundheit!* https://medium.com/better-practices/rest-soap-graphql-gesundheit-6544053f65cf (accessed 27/05/23)

Lou, T., et al. (2016). A comparison of android native app architecture mvc, mvp and mvvm. *Eindhoven University of Technology*.

Malt, U., & Braut, G. S. (2022). Icd-10 – Store medisinske leksikon [[Online; accessed 2023-03-23]].

Microsoft. (2023). *Restful web api design*. https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design (accessed 03/05/23)

Mnkandla, E. (2009). About software engineering frameworks and methodologies. *AFRICON 2009*, 1–5. https://doi.org/10.1109/AFRCON.2009.5308117

Nøhr, C., Koch, S., Vimarlund, V., Gilstad, H., Faxvaag, A., Hardardottir, G. A., Andreassen, H. K., Kangas, M., Reponen, J., Bertelsen, P., et al. (2018). Monitoring and benchmarking ehealth in the nordic countries. *MIE*, 86–90.

Nordb, S. (2006). Information visualisation and the electronic health record. *Norwegian University of Science and Technology*.

NTNU. (2020). *Cosem*. https://ehealthresearch.no/kin/prosjekter/cosem (accessed 11/04/23)

Oates, B. J., Griffiths, M., & McLean, R. (2022). *Researching information systems and computing*. Sage.

Patni, S. (2017). *Pro restful apis*. Springer.

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., & Shneiderman, B. (2003). Lifelines: Using visualization to enhance navigation and analysis of patient records. In *The craft of information visualization* (pp. 308–312). Elsevier.

Pohl, M., Wiltner, S., Rind, A., Aigner, W., Miksch, S., Turic, T., & Drexler, F. (2011). Patient development at a glance: An evaluation of a medical data visualization. *Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part IV 13*, 292–299.

Poppendieck, M., & Cusumano, M. A. (2012). Lean software development: A tutorial. *IEEE Software*, *29*(5), 26–32. https://doi.org/10.1109/MS.2012.107

Powsner, S. M., & Tufte, E. R. (1994). Graphical summary of patient status. *Lancet*, *344*(8919), 386–389.

Rajković, P., Janković, D., & Milenković, A. (2013). Using cqrs pattern for improving performances in medical information systems. *Proc. of the 6th Balkan Conference in Informatics*, 86–91.

Sedano, T., Ralph, P., & Péraire, C. (2017). Software development waste. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 130–140. https://doi.org/10.1109/ICSE.2017.20

Sefelin, R., Tscheligi, M., & Giller, V. (2003). Paper prototyping - what is it good for? a comparison of paper- and computer-based low-fidelity prototyping. *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, 778–779. https://doi.org/10.1145/765891.765986

Sharma, V., & Tiwari, A. K. (2021). A study on user interface and user experience designs and its tools. *World Journal of Research and Review (WJRR)*, *12*(6).

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction design: Beyond human computer interaction*. John Wiley Sons, Inc.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE symposium on visual languages*, 336–343.

Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson.

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., et al. (2016). The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, *315*(8), 801–810.

Sintef. (2019). *Den er ukjent, unødvendig og dreper millioner*. https://www.sintef.no/siste-nytt/2019/den-er-ukjent-unodvendig-og-dreper-millioner/ (accessed 07/06/23)

Sittig, D. F., Murphy, D. R., Smith, M. W., Russo, E., Wright, A., & Singh, H. (2015). Graphical display of diagnostic test results in electronic health records: A comparison of 8 systems. *Journal of the American Medical Informatics Association*, *22*(4), 900–904.

Sridharan, S., Peters, C., Newcombe, S., Jephson, C., Robinson, R., Mulder, B., Houghton, W., Visram, S., & Sebire, N. J. (2021). The essence of healthcare records: Embedded electronic health record system microblogging functionality for patient care narrative. *Future Healthcare Journal*, *8*(3), e709.

Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus waterfall project management: Decision model for selecting the appropriate approach to a project. *Procedia Computer Science*, *181*, 746–756. https://doi.org/10.1016/j.procs.2021.01.227

Tjora, A. (2012). *Kvalitative forskningsmetoder i praksis* (Vol. 2). Gyldendal akademisk Oslo.

Townsend, K. (2013). *Saturation and run off: How many interviews are required in qualitative research?*

Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., & Shneiderman, B. (2011). Lifeflow: Visualizing an overview of event sequences. *Proceedings of the SIGCHI conference on human factors in computing systems*, 1747–1756.

Appendix

A SQL Queries for Data Selection

```sql
--  Denne her gir 3726 treff. Vi søker da både på hoveddiagnoser og bidiagnoser
SELECT DISTINCT pasient.ppid, pasient.fødtår, pasient.kjønn  /* distinct her, pasient kan ha fleire sepsis-koder */
FROM public.nimesaktivitet
INNER JOIN public.pasient ON pasient.ppid = nimesaktivitet.ppid
INNER JOIN public.diagnose ON diagnose.nimesaktivitet_lnr = nimesaktivitet.lnr
INNER JOIN public.sepsiscodeicd10cm ON diagnose.diagnosekode LIKE CONCAT(sepsiscodeicd10cm.icd10cm_code, '%')
WHERE sepsiscodeicd10cm.typecode = 'ExplicitSepsis'
ORDER BY pasient.ppid

--  Om vi spør kun på hoveddiagnoser får vi 1794 treff. (kjøretid hos meg størrelsesorden ett minutt)
SELECT DISTINCT pasient.ppid, pasient.fødtår, pasient.kjønn  /* distinct her, pasient kan ha fleire sepsis-koder */
FROM public.nimesaktivitet
INNER JOIN public.pasient ON pasient.ppid = nimesaktivitet.ppid
INNER JOIN public.diagnose ON diagnose.nimesaktivitet_lnr = nimesaktivitet.lnr
INNER JOIN public.sepsiscodeicd10cm ON diagnose.diagnosekode LIKE CONCAT(sepsiscodeicd10cm.icd10cm_code, '%')
WHERE sepsiscodeicd10cm.typecode = 'ExplicitSepsis'
AND diagnose.er_primardiagnose = 1
ORDER BY pasient.ppid
```

Figure 1: SQL queries

B Graph Criteria Evaluation

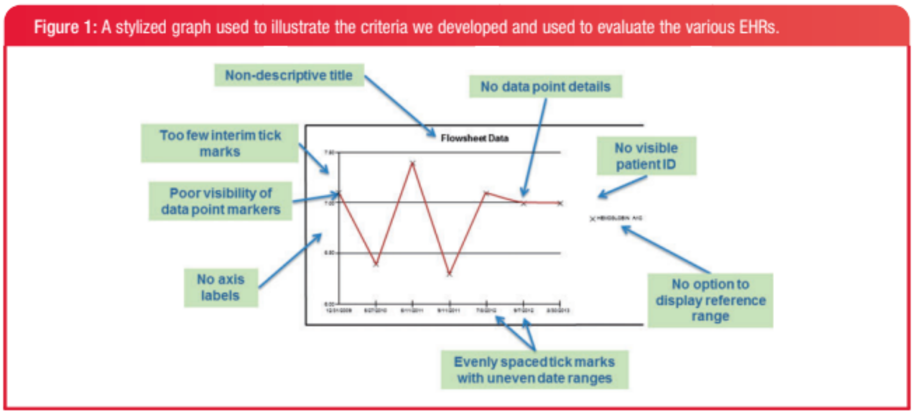

Figure 2: Visualization of criteria evaluation from Sitting et. al.s paper.

C Interviews

D FHIR Patient

```xml
<Patient xmlns="http://hl7.org/fhir"> doco
 <!-- from Resource: id, meta, implicitRules, and language -->
 <!-- from DomainResource: text, contained, extension, and
   modifierExtension -->
```

```xml
<identifier><!-- 0..* Identifier An identifier for this patient --></identifier>
<active value="[boolean]"/><!-- 0..1 Whether this patient's record is in active use -->
<name><!-- 0..* HumanName A name associated with the patient --></name>
<telecom><!-- 0..* ContactPoint A contact detail for the individual --></telecom>
<gender value="[code]"/><!-- 0..1 male | female | other | unknown -->
<birthDate value="[date]"/><!-- 0..1 The date of birth for the individual -->
<deceased[x]><!-- 0..1 boolean|dateTime Indicates if the individual is deceased or not --></deceased[x]>
<address><!-- 0..* Address An address for the individual --></address>
<maritalStatus><!-- 0..1 CodeableConcept Marital (civil) status of a patient --></maritalStatus>
<multipleBirth[x]><!-- 0..1 boolean|integer Whether patient is part of a multiple birth --></multipleBirth[x]>
<photo><!-- 0..* Attachment Image of the patient --></photo>
<contact>  <!-- 0..* A contact party (e.g. guardian, partner, friend) for the patient -->
 <relationship><!-- 0..* CodeableConcept The kind of relationship --></relationship>
 <name><!-- I 0..1 HumanName A name associated with the contact person --></name>
 <telecom><!-- I 0..* ContactPoint A contact detail for the person --></telecom>
 <address><!-- I 0..1 Address Address for the contact person --></address>
 <gender value="[code]"/><!-- 0..1 male | female | other | unknown -->
 <organization><!-- I 0..1 Reference(Organization) Organization that is associated with the contact --></organization>
 <period><!-- 0..1 Period The period during which this contact person or organization is valid to be contacted relating to this patient --></period>
</contact>
<communication>  <!-- 0..* A language which may be used to communicate with the patient about his or her health -->
```

```
  <language><!-- 1..1 CodeableConcept The language which can be used to communicate with the patient about his or her health --></language>
  <preferred value="[boolean]"/><!-- 0..1 Language preference indicator -->
 </communication>
 <generalPractitioner><!-- 0..* Reference(Organization|Practitioner|
   PractitionerRole) Patient's nominated primary care provider --></generalPractitioner>
 <managingOrganization><!-- 0..1 Reference(Organization) Organization that is the custodian of the patient record --></managingOrganization>
 <link>  <!-- 0..* Link to a Patient or RelatedPerson resource that concerns the same actual individual -->
  <other><!-- 1..1 Reference(Patient|RelatedPerson) The other patient or related person resource that the link refers to --></other>
  <type value="[code]"/><!-- 1..1 replaced-by | replaces | refer | seealso -->
 </link>
</Patient>
```

Gjennomføring av intervju

1. Forklare prosjektet
    a. Vi skal visualisere relevant data av en pasient som har eller har hatt sepsis i et dashboard og vise sykehistorie av pasienten i en interaktiv tidslinje.
    b. Vi har fått et dataset fra HEMIT.

1. Forklare hensikten med intervjuet
    a. Vi ønsker å kartlegge hvilket behov og forventninger dere som klinikere har til et slikt dashboard.
    b. Før vi går i gang med spørsmålene så lurer vi på om du/dere har dere noen spørsmål til oppgaven vår?

Intervju Start:

- Er det greit at vi tar opptak av intervjuet? Det blir bare brukt til å transkribere senere. Vi kommer ikke til å sitere det noen har sagt og deltakerne i intervjuet forblir anonyme selvfølgelig.
- Vi kan jo starte med å spørre deg tittelen din er hvilken utdanning og bakgrunn du har.

1. Når dere som klinikere går inn i systemet å finner frem en pasient, hva er det viktig for dere kommer godt frem av litt mer generell pasientinformasjon, f.eks navn, fdato, smittetrekant etc.
2. Som nevnt har vi planlagt å lage en interaktiv tidslinje. Hvilken type historikk om pasienten er viktig for dere å kunne se i en slik tidslinje?
3. Hvilken data er det som er viktig for dere å kunne se og interagere med i et slikt system?
4. Hva er det viktig for dere å kunne se av prøveresultat? Hvilke verdier er viktige å kunne se på en pasient med sepsis?
5. Er det noe annet som er viktig å få frem i et slikt dashboard mtp pasienter med sepsis?
6. Er det en forskjell i behov på tvers av avdelinger for et slikt system, f.eks intensiv vs infeksjonsavdelinger?
7. Er noe dere ikke er fornøyde med i helseplattformen eller tidligere løsninger som dere ikke vil se i det hele tatt i vår system og evt. hvorfor?
    a. Hva mener dere kunne blitt gjort bedre i HP/andre løsninger?
8. Hva synes dere HP/andre løsninger har gjort bra?
9. Hva mener du er viktig å unngå i vår løsning for å unngå at løsningen kan virke forvirrende og rotete?

Figure 3: Interview guide and questions

NTNU
Kunnskap for en bedre verden