Stava, Edvin Aamot
Vadet, Henrik Moen
Vestby, Petter Aleksander

Development of growth chart plugin in DHIS2

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Besmir Tola
May 2024

Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Norwegian University of
Science and Technology

devotta

Stava, Edvin Aamot
Vadet, Henrik Moen
Vestby, Petter Aleksander

Development of growth chart plugin in DHIS2

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Besmir Tola
May 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Sammendrag av Bacheloroppgaven

Tittel	Utvikling av vekstkurve plugin i DHIS2
Dato	21.05.2024
Deltakere	Edvin Aamot Stava Henrik Moen Vadet Petter Aleksander Vestby
Veiledere	Besmir Tola
Oppdragsgiver	Devotta
Kontaktperson	Eirik Haugstulen, eirik@devotta.no
Nøkkelord	Vekstkurver, WHO, Capture
Antall sider	60
Antall vedlegg	10
Tilgjengelighet	Åpen

Sammendrag:

Denne bacheloren undersøker utviklingen av en digital løsning for overvåkning av barns vekst ved å lage en vekstkurve-plugin for Capture web-applikasjonen på DHIS2. Vekstovervåkning er tradisjonelt gjort manuelt, noe som begrenser effektiviteten og nøyaktigheten i analyse og innhenting av data. Oppgaven gitt av Devotta etterspør en digital løsning som forbedrer overvåkningen av vekst samt kan forbedre barns helse globalt.

Vi utførte en grundig kravinnhenting ved å samarbeide med interessenter for å hente deres krav og utfordringer med vekstovervåkning. Innsikten vi fikk la grunnlaget for utviklingen av en plugin som sømløst kan integreres på DHIS2 plattformen. Løsningen ble utviklet med fokus på tilpasningsdyktighet for at den skal kunne integreres i helsesystemer verden over.

Vi har samlet kvalitative og kvantitative resultater som indikerer en brukervennlig vekstgraf-plugin som forbedrer vekstovervåkningsprosessen. Resultatene har blitt samlet med brukertesting og tilbakemeldinger fra interessenter, med faktiske brukere i et reelt miljø. Selv om resultatene indikerer at løsningen svarer på forskningsspørsmålene, identifiserer vi noen mulige forbedringer.

Oppgaven konkluderer med en vekstgraf-plugin som svarer på forskningsspørsmålene, men med noen mulige forbedringer som krever framtidig arbeid. Pluginen har muligheten til å møte framtidige krav og bidra til god barnehelse gjennom videre utvikling, forskning og samarbeid med interessenter. Gjennom videre utvikling og tilpasning kan løsningen forbedre og forenkle vekstovervåkning, og dermed bidra til bedre helse og velværelse for barn globalt.

Summary of Graduate Project

Title	Development of growth chart plugin in DHIS2
Date	21.05.2024
Participants	Edvin Aamot Stava Henrik Moen Vadet Petter Aleksander Vestby
Supervisors	Besmir Tola
Client	Devotta
Contact Person	Eirik Haugstulen, eirik@devotta.no
Keywords	Growth charts, WHO, Capture
Number of pages	60
Attachments	10
Availability	Open

Summary: This thesis explores the development of a digital growth monitoring solution via a growth chart plugin intended for the Capture web application on the DHIS2 platform. Growth monitoring is traditionally done manually, which limits the efficiency and accuracy of data collection and analysis. The task commissioned by Devotta requested a digital solution to improve the growth monitoring process and enhance child health globally.

We conducted a thorough requirement elicitation process by collaborating with stakeholders to gather their needs and challenges with growth monitoring. The insight gained established the foundation for developing a plugin that can be seamlessly integrated into the DHIS2 platform. The solution was developed with a focus on adaptability, to enable the product's ability to be integrated into health systems worldwide.

We gathered qualitative and quantitative results indicating a user-friendly growth chart plugin improving the growth monitoring process. Results were gathered with user testing and feedback from stakeholders, with actual users in a real-world environment. Even though the results indicate a solution that answers the research questions, we identify some possible improvements.

The task concludes with a growth chart plugin answering the research questions, but still with some improvement potential requiring future work. The plugin has the potential to meet future demands and contribute to child health through further collaboration with stakeholders, development, and research. Through further development and adaptation, the solution could improve and simplify growth monitoring, thereby contributing to better health and well-being for children globally.

Preface

We wish to express our gratitude to everyone who has contributed to our bachelor's thesis. First of all, we would like to thank our supervisor, Besmir Tola, for good and useful guidance throughout the project. Our appreciation also goes to our task provider, DevOtta, with special thanks to Eirik Haugstulen for his consistent support and assistance. We are deeply appreciative of the financial support provided by Innlandet Fylkeskommune through the FORREGION program, which enabled our research journey to Sri Lanka. We are particularly grateful to Dr. Pamod Amarakoon and HISP Sri Lanka for their hospitality and support during our time in Sri Lanka, making our stay in Sri Lanka successful by facilitating our research and enhancing our understanding and insight. Lastly, we acknowledge the guidance and support of Hanin Sa'adah and HISP MENA. They have provided valuable insights that have significantly contributed to the quality and depth of our work This thesis stands as a reflection of the collaborative effort and support of all those mentioned above.

Trondheim, May 2024

Edvin Aamot Stava   Henrik Moen Vadet   Petter Aleksander Vestby

Contents

Figures

Tables

Code Listings

Glossary

Capture is an application within the DHIS2 platform designed for data entry, collection, and display, facilitating efficient management and analysis of health information. 2, 3, 7, 8, 11, 13, 25, 28, 43, 47, 49, 50, 54, 58

Indicator is used to describe the type of growth chart displayed, in the context of this thesis. 26, 28, 36, 37, 42, 44–48, 57

Linter is a static analysis tool that scans code for potential errors and helps maintain code readability. 23

Open source refers to software with source code accessible to the public for viewing, modification, and distribution, typically under a license allowing for collaborative development and innovation. 4

Plugin is a software component that adds specific features or functionality to an existing application or system without modifying its core code. 2–4, 6–8, 11, 13–17, 21, 25, 30, 31, 33–37, 39, 41–50, 54–58

Acronyms

AI Artificial intelligence. 23

CI/CD Continuous Integration/Continuous Delivery. v, 21, 30, 31

DHIS2 District Health Information Software 2. 1–7, 11, 13–16, 18, 26, 33, 35, 42–44, 46–51, 54–58

E2E End-to-End. 31

HISP Health Information Systems Program. vii, viii, 2, 3, 11, 13, 14, 17, 25, 41–46, 49, 55, 57, 58

HMIS Health Management Information System. 2, 42, 58

MENA Middle East and North Africa. viii, 11, 41, 43, 44, 55, 57, 58

PR pull request. 21, 30

SD standard deviation. 2, 28

SDLC software development lifecycle. vii, 10

SUS System Usability Scale. v, 11, 15, 16, 41, 51–56, 58

UI user interface. v, vii, viii, 13–16, 21, 41, 46–48, 51–55, 57, 58

UiO University of Oslo. 2, 3

WHO World Health Organization. vii, 1, 26–28, 33, 34, 36, 45

Chapter 1
Introduction

1.1 Background

Child growth monitoring is a global practice. In 2004, a report surveyed 202 countries to describe growth monitoring practices (de Onis et al., 2004, p. 461). Out of 178 respondents, 154 reported using growth charts for child growth monitoring.

The use of growth charts is essential in pediatric practice as they offer a systematic approach to monitoring and evaluating children's growth (Khadilkar & Khadilkar, 2011, p. 166). By comparing individual children's growth data with standardized growth charts, healthcare professionals can identify deviations from expected growth patterns at an early stage. Early detection of such abnormalities is critical, as they may indicate potential underlying health problems or malnutrition. This allows healthcare professionals to take action before potential problems escalate. To enable this early detection, the World Health Organization (WHO) has developed a *growth chart standard*.

The use of WHO's growth charts is widespread. In 2011, 125 countries had adopted the WHO's growth standard (Yang et al., 2015, p. 1). The standard results from extensive population studies documenting average growth in children of different ages and sexes. These charts serve as reliable reference points for assessing average growth and development in children everywhere, regardless of ethnicity, socioeconomic status, and type of feeding (de Onis et al., 2008, p. 48).

1.2 Social Context

Today, many countries still monitor growth using pen and paper. Several healthcare facilities, especially in resource-poor areas, lack access to digital tools for growth monitoring and must rely on traditional methods. We experienced this during a field trip to Sri Lanka, where parents brought their children to health clinics for regular check-ups, and the growth data was manually recorded in paper-based home records.

However, tracking growth in the traditional way with pen and paper has several disadvantages. Human error can occur during data entry and make the data unreliable, whether due to misunderstandings, inaccuracies, or unclear writing. Handwritten copies are also subject to several risks that threaten the integrity of the information, including loss and damage of documents.

To avoid the disadvantages of traditional data processing with pen and paper, many low- and middle-income countries today use the *District Health Information Software 2* platform (DHIS2). The DHIS2 platform allows health institutions to digitize their health data, thus simplifying collection, analysis, and reporting. However, the functionality of growth charts on the DHIS2 platform is currently limited. Therefore, many countries still carry out growth monitoring traditionally with pen and paper, even though they have other health data recorded digitally with the DHIS2 platform.

1.3 Problem Statement

The limited functionality of growth charts in DHIS2 constrains accurate and effective monitoring of children's growth and health. This limitation prevents the transition from manual to digital systems for growth monitoring. Therefore, there is an urgent need to develop and integrate growth charts on DHIS2, adapted to the requirements and specifications of different countries' health systems. Addressing this shortcoming will streamline the growth monitoring processes and enhance the quality and reliability of health data collection and analysis.

Therefore, the problem statement revolves around the necessity to bridge the gap between manual methods of growth monitoring and DHIS2 by developing a generic and adaptable growth chart *plugin* that can seamlessly integrate into the platform. In this context, *generic* refers to a plugin that covers the needs of all of the countries using the DHIS2 platform. A plugin is a software component that adds specific functionality to an existing system. DHIS2 has plugin functionality, enabling seamless integration of external applications to run on the platform. In cooperation with stakeholders from Sri Lanka, the Middle East, and North Africa, we identified three core requirements for a growth chart plugin on the DHIS2 platform:

- *The growth chart plugin needs to show a child's growth over time.*
- *Countries can change the dataset to their country-specific growth references.*
- *The growth chart plugin supports both standard deviation (SD) and percentiles*

We aim to develop a plugin within the *Capture application*. Capture is an application on DHIS2 that facilitates individual-level data collection, thereby enabling the storage of relevant measurement values for monitoring growth. Additionally, the recent introduction of plugin support in Capture further strengthens our motivation to implement the growth chart within this application. This support and Capture's individual-level data collection has the potential to ensure effective and accurate growth monitoring practices for all countries using Capture in DHIS2. Considering the existing problem, we have defined research questions to be able to explore the topic more thoroughly:

***Research question 1*:** How can we design a generic, user-friendly growth chart encouraging the target audience to adopt the solution over traditional pen-and-paper-based methods?

***Research question 2*:** How can a growth chart plugin be seamlessly implemented into existing instances in the Capture application on DHIS2?

1.4 Definition of Terms

District Health Information Software 2 (DHIS2) is an open-source platform for collecting, aggregating, visualizing, sharing, and analyzing data. The platform is developed by the *Health Information Systems Program* (HISP) at the University of Oslo (UiO) and is the world's largest *Health Management Information System* (HMIS) used in over 80 countries (DHIS2, n.d.). DHIS2 is primarily used as a health information system in low- and middle-income countries, as the open-source platform works well for environments with limited resources. It offers easy customization without the need for coding and emphasizes local ownership. In addition to healthcare use, the platform is used to manage education, supply chains, logistics, agriculture, and other sectors. The platform offers several included core applications developed by the DHIS2 community. One of these core applications is Capture.

The Capture application in DHIS2 is an essential component that allows users to record and collect data. It offers a flexible platform for customizing data registration forms that suit different needs and program areas within health information systems and other sectors that use DHIS2. Additionally, Capture enables users to quickly and efficiently collect and visualize data via web browsers or the *Capture's Android application*. In this assignment, we will specifically focus on the *Capture web app*, and it is essential to clarify that this should not be confused with the Capture Android app. While the Capture app for Android primarily handles event recording and offers offline functionality, the Capture web app has similar functionality but is intended for online use.

Health Information Systems Program (HISP) is a global movement working to adapt and configure DHIS2 locally, support implementation, and offer user training at both national and regional level (UIO, n.d.). With its origins at UiO, HISP has grown to become an extensive collaborative network that brings together experts and stakeholders from all over the world. Utilizing partnerships with local authorities, healthcare professionals, and international organizations, HISP is dedicated to promoting health information as a global public resource.

1.5 Project Overview and Implementation Approach

In this task, we aimed to develop functionality for growth monitoring in the Capture app on the DHIS2 platform. We conducted qualitative interviews with experts from HISP with technical and medical backgrounds. The purpose of the interviews was to map real-world requirements and challenges for digital growth charts. The results of these interviews lay the foundation for our *requirement specification*, which served as a guide through the development process. Following the completion of the product and its alignment with the requirements outlined in the requirement specification, we conducted a trip to Sri Lanka to test the product in a real-world environment. Observations and user tests conducted during this trip played a central role in elucidating the problem statement.

1.6 Structure

The structure of the thesis consists of eight chapters.

Chapter 2 - DHIS2 Software: Includes an introduction to relevant theory about DHIS2 software in the context of this project.

Chapter 3 - Methodology: Elaborates on the method used to develop the product in accordance with the problem. We provide insight into the choice of approach and strategy.

Chapter 4 - Development Process: Examines aspects of the development process behind the growth chart plugin.

Chapter 5 - Results Evaluation: Evaluate and present the results of the project. We demonstrate how the product addresses requirements and expectations in a real-world test environment and present feedback from our stakeholders. Then we will evaluate these results to determine whether the plugin developed effectively addresses the problem statement.

Chapter 6 - Conclusion: Ssummarize the findings from the evaluation and conclude according to the problem introduced in problem statement.

Chapter 2
DHIS2 Software

This chapter provides a brief overview of DHIS2, introducing core concepts and relevant applications used for this project.

2.1 Introduction to DHIS2

In the context of implementing a growth chart plugin on DHIS2, it is essential to understand the fundamental principles of the platform. DHIS2 is a flexible system, allowing specific adaptation for various countries (Dehnavieh et al., 2019, p. 72). This adaptability provides the possibility to customize the platform based on local needs. Being an *open source platform*, DHIS2 further allows for flexibility and customization (Dehnavieh et al., 2019, p. 66). The open architecture facilitates custom applications, enabling the developers to improve and customize the platform to meet specific needs, such as implementing a growth chart. However, implementing a growth chart plugin necessitates a further understanding of the relevant data model for DHIS2.

2.2 Data Model

Understanding the core concept of DHIS2 is essential for the implementation of the growth chart plugin. Figure 2.1 illustrates the core concepts of the data model. The figure displays a simplified data model with relevant components from the much more complex DHIS2 platform, and the relationship between these components.

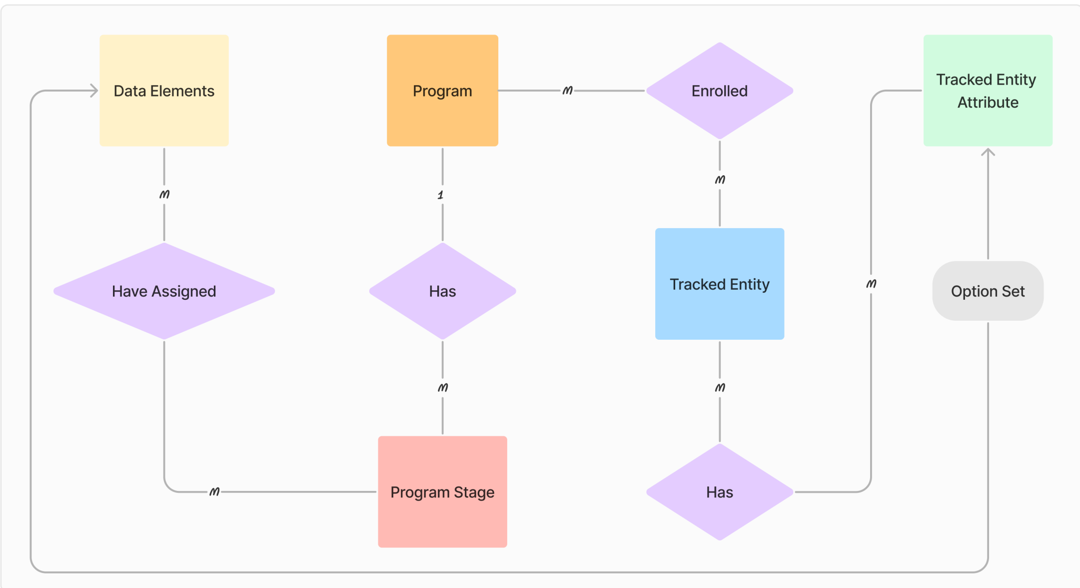


Figure 2.1: Diagram illustrating components in the relevant data model of DHIS2 software.

In Figure 2.1, the cardinalities *1* (one) and *M* (many) indicate the type of relationship. For example, program and program stages have a *one-to-many* relationship, where a program can have multiple program stages, but a program stage can only belong to one program. In the following section, we explore the different core components present in the data model.

2.2.1 Program

DHIS2 provides the possibility to create a wide range of programs. In DHIS2, a program defines the workflow and the parameters for a specific service, such as disease monitoring, maternal monitoring, or child health monitoring. These programs are fundamental for the platform's functionality in this project's scope, as seen in Figure 2.1, illustrating the data model.

2.2.2 Program Stage

A program typically includes one or more *program stages*. Program stages allow the assignment of data elements collected within each stage. A child program, for example, can have program stages for birth, visits, and interventions. It thereby gives the possibility to configure and group data elements collected within a program.

2.2.3 Data Elements

Within a program stage, users can collect *data elements* representing individual data recorded in the system. They contain the specific variables used and tracked in the program. Figure 2.2 illustrates relevant data elements in the context of this project. For example, *Child visit* refers to a program stage where the relevant data elements such as *head circumference*, *weight*, and *length* are assigned.

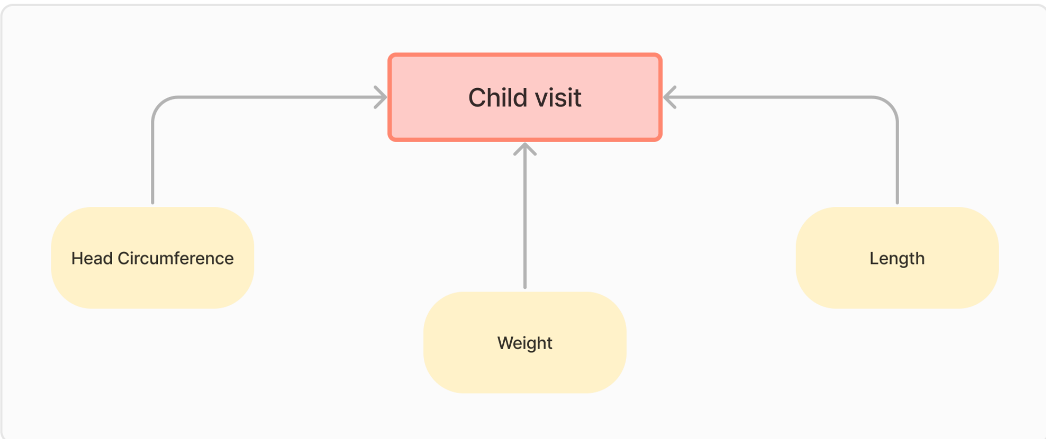


Figure 2.2: Program stage Child *visit with assigned data elements*

2.2.4 Tracked Entity

A *tracked entity* is a unique object within the system, such as a health center, location, individual, etc. In addressing the lack of growth charts on the DHIS2 platform, the tracked entity refers to an individual, specifically a *person*, with relevant attributes assigned.

2.2.5 Tracked Entity Attribute

Tracked entity attribute refers to attributes linked to an individual. These attributes rarely change and are only entered at the registration of a new individual. In this project, essential attributes for the tracked entity include *date of birth*, *sex*, *first name*, and *last name*. Figure 2.3 illustrates the essential attributes.

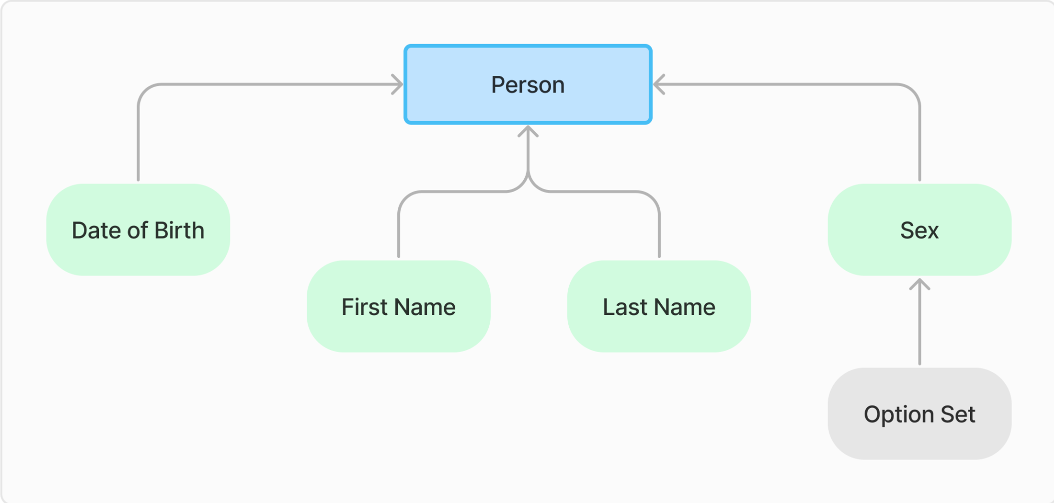


Figure 2.3: Tracked entity Person *with assigned attributes*

2.2.6 Option Set

Option sets allow validating and standardizing tracked entity attributes and data elements. This feature is beneficial to ensure data consistency, especially since DHIS2 is used in more than 80 different countries, each with *distinct data format* and language. Figure 2.3 and Figure 2.4 illustrate the use case of option set for the attribute *sex*, which validates the sex of a person by enabling a standardized format. As illustrated in Figure 2.4, the *sex* is retrieved by the growth chart plugin as the defined variables ”CGC_Female” or ”CGC_Male”. With option sets, the plugin knows for sure what the *sex* is and does not have to take into account how the *sex* is written in each country. For instance, in England, it would be ”boy” or ”girl”, in Norway ”gutt” or ”jente”. With the defined option set a child is ”CGC_Female” or ”CGC_Male” for both countries.

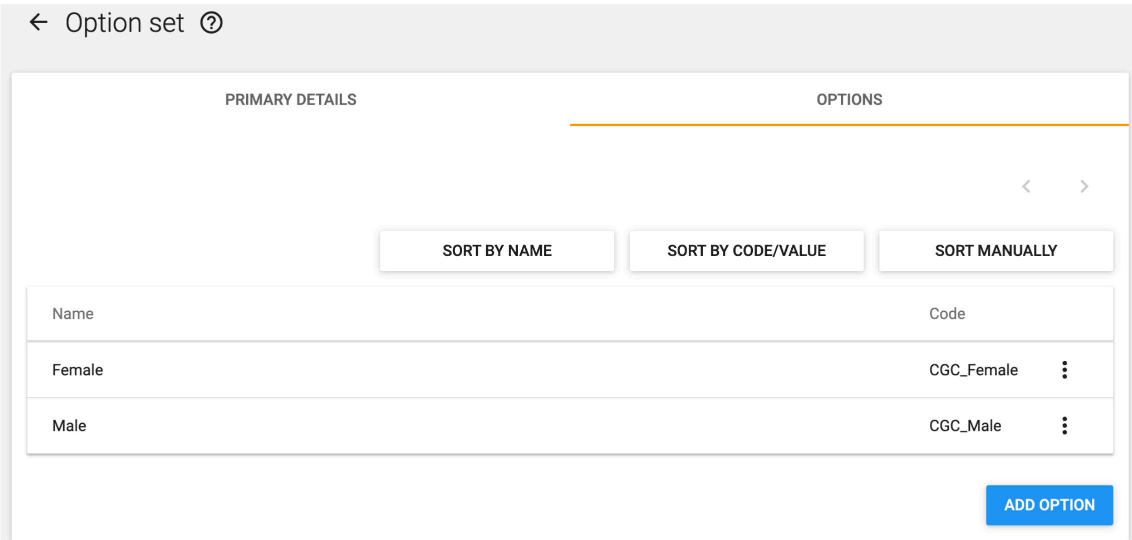


Figure 2.4: Option set for sex

2.3 Core Applications

Several *core applications* on DHIS2 are used for this project. These core applications include *Users*, *App Management*, *Capture*, and *Datastore Management*. This section explore relevant aspects of these applications and how they are important for addressing the limitations associated with the availability of growth charts on DHIS2.

2.3.1 App Management

The *App Management application* facilitates custom application upload to a DHIS2 instance. This feature is essential for addressing the problem statement, as it allows the integration of the growth chart plugin, which must be uploaded as an application.

2.3.2 Capture

The *Capture application* provides the capability to inspect a *tracked entity* enrolled in a program. This is illustrated in Figure 2.5, showing the Capture dashboard in the *Growth Monitoring program*, with the tracked entity ”Michael Taylor” selected. Additionally, the figure shows two program stages: ”Child visit” and ”Person profile” with respective data elements and tracked entity attributes, as shown in Figure 2.2 and Figure 2.3. Given the problem statement, we want to develop and implement a plugin to enhance this dashboard by integrating growth chart functionality.

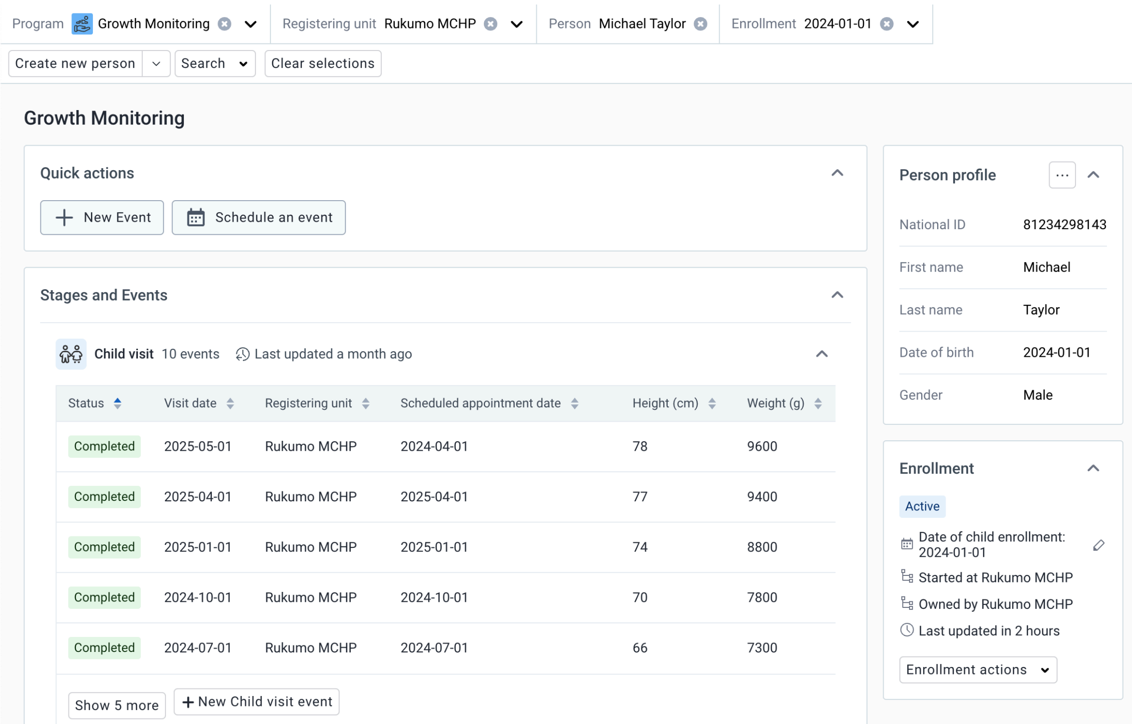


Figure 2.5: Capture dashboard.

The Capture application also allows capture of new events within a program stage. An example is shown in Figure 2.6, where the ”Enrollment: New Event” form for the ”Child visit” program stage can be seen. This form allows Capture to track and display *individual-level data*, an essential foundation for monitoring a child’s growth over time.

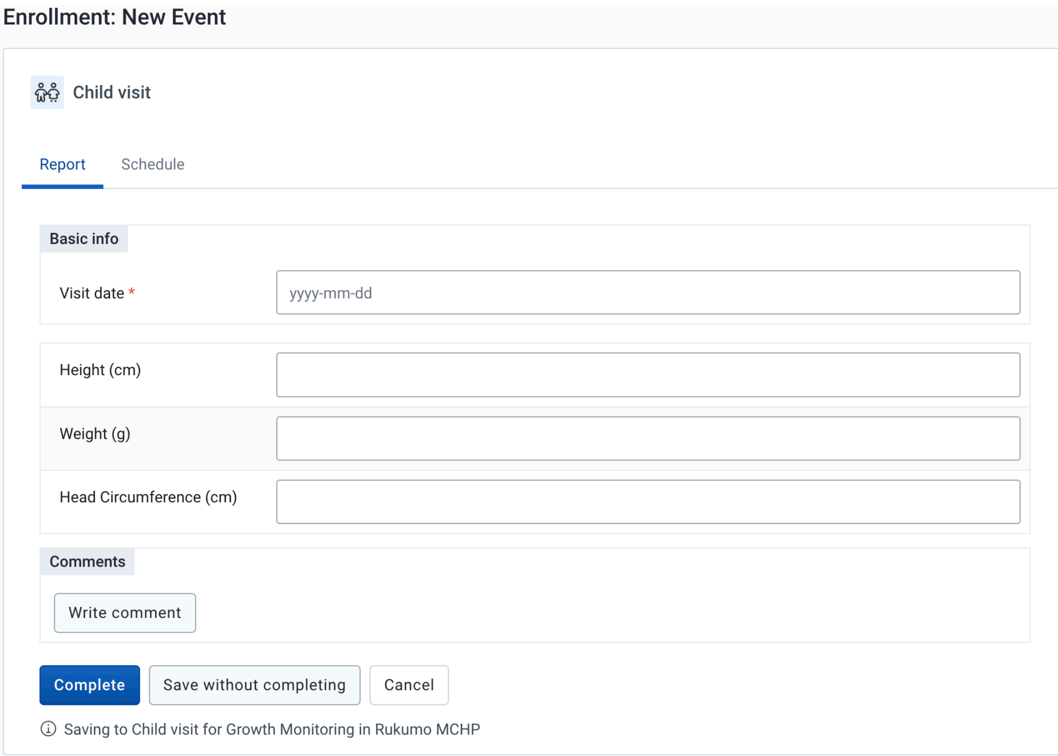


Figure 2.6: New Event page in the Capture application.

Furthermore, the Capture application supports the *plugin feature*. The plugin feature enables custom applications, uploaded with App Management, to be implemented and rendered in the Capture application. An example is shown in Figure 2.7, where a *tic-tac-toe* game is rendered as a plugin in Capture.

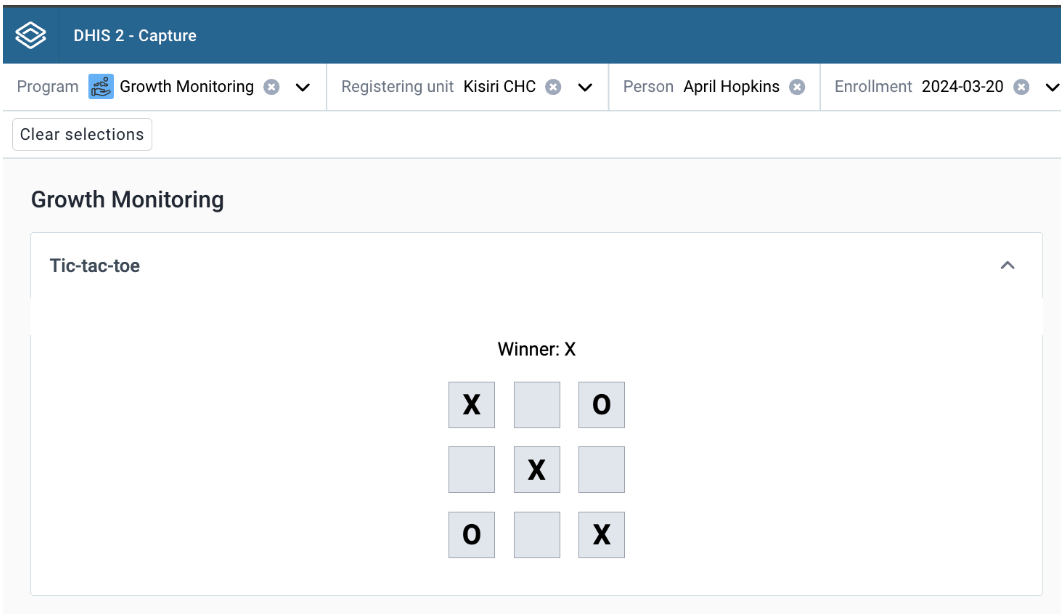


Figure 2.7: Tic-tac-toe rendered as a plugin on the Capture dashboard.

To render the plugin, it is necessary to create a custom layout for the Capture dashboard, which is accomplished using the Datastore Management application. The Capture dashboard is customizable, allowing countries to insert functionality and design the dashboard according to their needs.

2.3.3 Users

User management and access control functionalities are available and managed in the *Users* application. The application allows distinct user privileges based on the user's role or group. Roles and groups can be configured to control access to data or applications. The *Users* application is therefore beneficial for the project, as it allows for regulating access to the Datastore Management application, introduced in 2.3.4.

2.3.4 Datastore Management

Datastore Management provides a storage hub for applications. It facilitates the preservation of configurations and other necessary data related to various applications. The Datastore Management is a convenient location for storing various data relevant to addressing the limitation of growth charts functionality on DHIS2, such as a configuration file. This makes Datastore Management essential within the project's scope. Moreover, access to this application can be limited using the *Users* application by altering the user role or user group access setting. The Users application can thereby restrict access to configuration files and sensitive data stored in Datastore Management.

Chapter 3
Methodology

This chapter deals with the methodological approach of the project. It analyzes and evaluates the methodology to provide an understanding of how these employed methods help address the research questions.

3.1 Selection of Methodologies

This section explores the methods used in various aspects of this project. During the project, we have used the *Waterfall methodology* shown below in Figure 3.1. The Waterfall methodology is applied at a high level throughout the project, to guide our overall project management and planning. Additionally, this section provides the relevance of these methods for data collection and development.

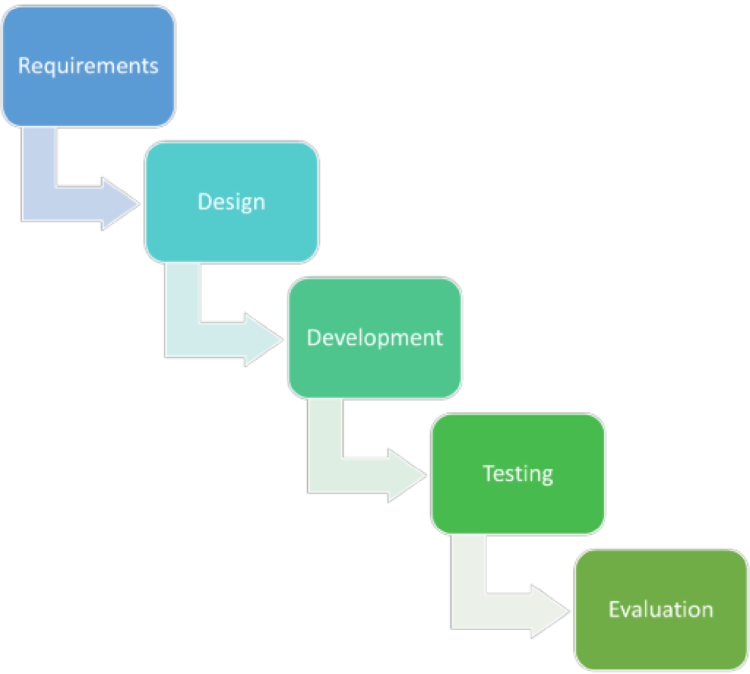


Figure 3.1: Software development lifecycle (SDLC) - Waterfall model, (Maulana et al., 2021, Figure 1)

- ***Requirements***: For the requirements element of the Waterfall methodology, we used qualitative interviews with relevant stakeholders.
- ***Design & Development***: For the design and development phase in Figure 3.1, we utilized the agile methodology frameworks Srum and Kanban, instead of the Waterfall methodology. These agile methodology frameworks are further introduced in *3.1.2 Design and Development - Development Methodology* on page 11.

- ***Testing***: In the Waterfall methodology's testing element, we conducted application and implementation testing of the plugin in Sri Lanka.

- ***Evaluation***: For the evaluation element in the Waterfall methodology, we hosted product demonstrations with stakeholders to gather feedback to evaluate the plugin. We also conducted user tests and calculated System Usability Scale (SUS) scores to evaluate the usability of the plugin and its implementation process.

3.1.1 Requirements - Qualitative Interviews with HISP-experts

To elicit requirements and necessities for the project, we conducted qualitative semi-structured interviews with representatives from diverse *Health Information Systems Program* (HISP) groups as well as the task provider, Devotta. We used these interviews to explore the perspectives, insights, and experiences of the HISP representatives. Specifically, we chose Hanin Sa'adah, DHIS2 Implementation Specialist for the Middle East region at HISP Middle East and North Africa, and Dr. Pamod Amarakoon, Lead at HISP Sri Lanka, based on their expertise with the DHIS2 platform and their insight into key requirements and needs for growth monitoring.

By employing a semi-structured approach, we took advantage of predefined questions with relevant follow-up questions. The discussion topics of these qualitative interviews included:

- *What are the current uses of DHIS2 and the Capture app?*
- *How is growth monitoring currently conducted in your region?*
- *How are growth charts currently used?*
- *What requirements must be met for the growth chart functionality in the Capture app on the DHIS2 platform to be useful?*

The interview discussion topics were selected to cover various aspects of DHIS2, growth monitoring, and the requirements for the growth chart functionality. These interviews were conducted to establish a solid foundation of understanding and insight that would guide the project development.

3.1.2 Design and Development - Development Methodology

Our development methodology is based on principles derived from agile software development. Despite using waterfall methodology to guide our project, we applied the agile *Scrum* framework for the design and development stages in the waterfall model. However, after completing two sprints, we switched to the agile framework *Kanban*.

Scrum

Scrum is a development framework including planning, sprints, and closure (Schwaber, 1997, p. 126). The planning and closure phases are linear defined processes, while the sprint phase is an iterative process. Figure 3.2 illustrates the entire process.

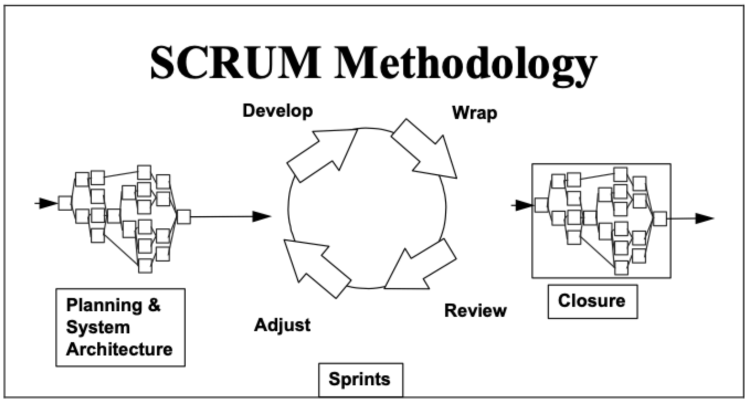


Figure 3.2: Scrum framework stages (Schwaber, 1997, Figure 6)

We employed this framework by starting with a linear planning phase of the development process. This planning phase included creating a backlog of issues in *Jira Software*, a project management tool intended for agile methodologies, further explained in *4.2.1 Jira Software* on page 20. More detailed information about the planning phase of the development process is provided in *4.3 Planning* on page 24.

During the development process, we conducted two sprints lasting two weeks each. A *sprint* is a recurring cycle of product development, thereby being an *iterative process*. The sprints conducted include the following key components:

- ***Sprint Planning***: A meeting is held at the start of each sprint, where the team selects issues from the backlog to complete during the sprint.
- ***Stand-up Meetings***: Daily meetings to ensure the entire team is synchronized and up to date on issue progress, as well as possible challenges or delays.
- ***Sprint Retrospective***: A reflection meeting is held at the conclusion of each sprint. The team reviews various sprint aspects, such as progress, challenges, and areas for improvement.

The closure stage of the Scrum framework includes the necessary steps to prepare the developed product for release (Schwaber, 1997, p. 130). The steps include testing, documenting, integrating, and marketing. In the closure phase, our primary focus was preparing for field testing in Sri Lanka and creating an implementation guide using Kanban.

Kanban

The agile methodology framework *Kanban* functions similarly to Scrum but without the use of sprints. Nearly all growth chart functionality was complete at the end of the sprints, except for a few issues that remained in the backlog. The remaining issues mainly concerned application bugs, potential code improvements, and some features that had not yet been completed.

We decided to employ principles from the *Kanban framework* because we needed to progress beyond the development phase and since the remaining issues did not fill up the workload of a sprint. Using Kanban allowed us to continuously address the issues from the last sprint and the remaining backlog items without the fixed sprint length. As a result, we could work more flexibly and address the remaining issues whenever we had time available.

3.1.3 Testing - User Testing

User testing is a central part of our project methodology. The purpose of the user tests was to ensure that the developed growth chart plugin met the needs and expectations of the intended users.

User testing was chosen based on the method's ability to identify problems and improvements for the plugin directly from possible end users. By allowing users from the target group to interact with the product, we gained insights that we otherwise would not obtain with internal tests. In addition to uncovering errors and shortcomings, user testing provides valuable information about the user's experience and needs. The user testing results allow us to evaluate whether the product meets the users' expectations and preferences.

Because of these advantages, we consider user testing a critical method in the process of developing a product that can close the gap between DHIS2 and child growth monitoring.

Rationale for Testing UI and Implementation Process

The user testing of the growth chart plugin is separated into two parts: testing of the *user interface* (UI) and testing of the *implementation*. As explained in Chapter *2 DHIS2 Software* on page 4, the plugin is designed to be used and implemented in the Capture application on the DHIS2 platform. Given the widespread use of DHIS2, the growth chart plugin must be designed with functionality to meet different needs globally. Additionally, since countries operate DHIS2 applications on customized instances tailored to their specific needs, the implementation has to be generic to adapt to all the different requirements. Therefore, a comprehensible implementation guide is needed to facilitate the implementation process.

UI and implementation testing are conducted to ensure a thorough project evaluation. Testing of the UI focuses on the *functionality* and *usability* of the plugin within the DHIS2 platform, ensuring it performs as intended and meets the requirements specified. On the other hand, implementation testing evaluates the *efficiency* and *intuitiveness* of the implementation process through the provided implementation guide. The aim is to create an implementation guide that helps implementers with diverse backgrounds to successfully integrate the growth chart plugin into their DHIS2 instances with the correct settings covering their country-specific needs. By conducting tests on the UI and implementation guide, we aim to validate the usability of the growth chart plugin across various DHIS2 instances, thereby maximizing its potential impact and adoption.

- **UI Testing in Sri Lanka**

  User testing of the application took place in Sri Lanka, where we collaborated with HISP Sri Lanka to test the growth chart plugin on healthcare professionals at different clinics in the country. By testing the plugin with healthcare professionals at the clinics, we gained insight into how well our product integrated into their workflow and whether it effectively supported their growth monitoring practice. This test environment provided real-world feedback on usability.

  Participant Recruitment

  We selected participants who met specific criteria for testing the UI to ensure a diverse group with valuable perspectives. We tested people who had direct experience or knowledge of growth monitoring and health services for children. This included health workers who regularly worked with growth data using the current system in Sri Lanka and had in-depth knowledge of the topic. In addition to testing two health workers at the clinics, we included a doctor in the test group. Although she did not necessarily carry out measurements and mapping growth data as part of her daily tasks, she still had valuable knowledge of growth curves and provided a different perspective to the testing.

Test Design

We developed user tests for the application to assess the plugin's functionality and usability in conjunction with the Capture app. We designed tests to cover various aspects of the plugin's performance by formulating *ten tasks* the participants had to complete. The test tasks were designed to recreate real-world scenarios, providing participants with situations similar to their daily experiences. Thus, we aimed to measure how seamlessly the growth chart plugin integrates into their workflow and supports their tasks, providing valuable insight into its functionality and usability. The tests were conducted at the clinics where the participants work with growth monitoring. Despite the inherent challenges of noise and potential distractions, this approach provided a valuable opportunity to observe the plugin's performance in a realistic context.

Figure 3.3: Testing of the UI in clinics in Sri Lanka

The test document with the user scenario is attached in the *Appendix A* on page 63.

- **User testing - Implementation**

  We conducted user testing focusing on the implementation process of the plugin. This involved testing employees from various HISP groups with knowledge of the DHIS2 platform. By involving representatives from different HISP groups, we aimed to assess the efficiency and intuitiveness of integrating the growth chart plugin into DHIS2. User tests allowed us to identify any challenges or areas for improvement in the implementation guide.

Participant Recruitment

The implementation guide contained specific steps related to the DHIS2 platform. Therefore, one requirement for the participants was to have knowledge about the DHIS2 platform. The implementation guide is available in the project's repository on GitHub: *Implementation Guide*. It was essential to simultaneously select participants who could provide different perspectives and valuable input. Therefore, we selected participants with various backgrounds in medicine, programming, and implementation work. A total of three tests were conducted.

Test Design

The implementation tests investigated whether the implementation guide was intuitive and effective. We prepared a test plan to evaluate the implementation guide. The implementation tests were conducted on an already configured environment facilitated for user tests. The approach replicates a situation the users will face if they want to implement the plugin on their instance. By using an already configured instance, we isolated the tests to focus only on our implementation guide.

During the tests, the participants were given the plugin's implementation guide and instructed to follow it step by step. We observed the participants while they performed the steps. The test document with the user scenario is in the *Appendix A.6 Implementation Test Document* on page 68.

Analysis

Both the UI tests and the implementation tests were thoroughly analyzed. Analyses of the tests were conducted in *two rounds*. First, we briefly assessed the results after each test. After completing all the user tests, we conducted another more comprehensive analysis. This analysis involved a thorough review of aggregated feedback to identify patterns and recurring issues. *Cross-checking* the results from the individual tests were useful for identifying what changes would improve the product's usability and ensure that it meets stakeholders' expectations and the project's goals. As part of this process, we also calculated the *System Usability Scale* (SUS) score for each test, which provided a quantitative measure of the user experience.

3.1.4 Evaluation - Demonstrations for Stakeholders and SUS score

To collect further data about the result of the product, we conducted a demo and evaluation with the stakeholders. We calculated SUS scores for everyone who interacted with the growth chart plugin through user testing.

Product demonstrations

Product demonstrations were conducted after the product was fully developed to map strengths and weaknesses. The purpose was to identify deficiencies that had to be rectified. With product demonstration, we presented functionalities, properties, and use cases to the stakeholders. As a part of the demonstration, participants were also allowed to interact with the solution in a simulated environment and provide direct feedback. This method was useful for validating the solution's relevance and usability. By gathering feedback from stakeholders and observations during the demonstration, we could assess how well the solution aligned with the project's goals and how it addressed the research questions.

Furthermore, we were also able to identify technical or user-related challenges that needed to be resolved. Thus, the product demonstration was an important part of the process to ensure the solution fulfilled the stakeholders' expectations and to evaluate the user-friendliness.

System Usability Scale (SUS)

System Usability Scale (SUS) is a reliable tool used to measure the usability of the product. It has become a popular questionnaire for end-of-test subjective assessments of usability (Lewis & Sauro, 2009, p. 2). By administering the SUS to everyone who interacted with the plugins during user testing, we were able to quantify the user experience and identify specific areas in need of improvement. The quantitative data complemented the qualitative feedback from the demonstrations, providing a comprehensive overview of the product's usability.

The SUS score method is conducted using a questionnaire with *ten statements*. In response to the statements, the test participant chooses a number between *1-5*, where *1* is "strongly disagree" and *5* is "strongly agree". The SUS form is attached in the *Appendix A.1 System Usability Scale (SUS)* on page 62.

The SUS score is calculated based on the results of the questionnaire. A SUS score is an integer from *0-100*, where *100* is the best, and *0* is the worst. A high SUS score suggests good usability, while a low score indicates poor usability. The approach used for calculating SUS score is as follows:

- For odd-numbered statements: subtract *1* from the response score.
- For even-numbered statements: subtract the response score from *5*.

The *response score* is the value of the box the participant checked off. If the participant checks off a *5* (response score), it will be calculated as *4* if the statement number is odd or *0* if the statement number is even. By summing the resulting scores together and multiplying by *2.5*, we gain a score ranging from *0* to *100*, representing the SUS score. The calculation of our user test SUS scores are shown in *5.2.1 System Usability Scale (SUS) for UI* on page 51. SUS scores are used to evaluate both the plugin and implementation guide.

3.2 Validity and Reliability

To ensure the validity and reliability of our results, we used several measures throughout the process. The aim was to ensure *accuracy* and *consistent* data.

3.2.1 Validity

We strengthened the validity of the methods by including relevant individuals according to the product's target group for both the UI of the growth chart plugin and the implementation. By doing this, we ensured that the results were as accurate as possible.

- ***Selection of Participants***: We chose interviewees with experience and knowledge within the DHIS2 platform and growth monitoring. This ensured that we collected data from informants with in-depth knowledge of the subject who provided valuable insights.
- ***User testing***: During user testing in Sri Lanka, we included representatives from the target group of the product. This ensured that we received feedback from relevant end users of the product, increasing the validity of our results.

3.2.2 Reliability

We assessed the reliability of the data collection process by looking at the consistency of the results from interviews and user tests. By doing this, we could explore whether there was consistency in the results. Significant disparities in the feedback from user tests, stakeholders, or the methods used may imply that the collected data is unreliable, while minor variations imply the opposite.

- ***Method consistency***: We followed the same procedure during the interviews with the HISP groups. In addition, the user tests were conducted using the same test and procedure to ensure consistency in data collection. Consistent testing procedures means the same method throughout the user testing stage, and with the same tasks for all the test participants. Consistency in the user testing helped minimize variations in the results.

- ***Comparison and Cross-checking***: We compared results from different interviews and user tests to identify consistency and reliability in the data. Comparing helped validate our findings and conclusions by cross-checking information from various sources. Through these measures, we ensured the validity and reliability of our results and findings, strengthening confidence in the conclusions presented in our research work.

3.3 Evaluation of Methods

The methods employed in the project were chosen based on their ability to be effective and precise. With a tight project time frame, it is crucial to use methods that maximize productivity without compromising the quality of the work. We found that the methods we chose achieved their goals effectively. However, we also acknowledge the importance of critically evaluating our approaches. Therefore, we will address some potential limitations and weaknesses of our chosen methods in the following section.

- ***Diversity***: We recognize a drawback in our interview process, which exclusively involved HISP experts. Capturing the diversity of end users' needs and preferences requires a broader range of interview subjects. While experts can provide valuable perspectives and insights, their views and experiences may be limited by their professional background and understanding. This limitation can lead to certain important user experience aspects not being adequately addressed or reflected in the requirements specification. A more diverse approach that includes direct feedback from a wide range of end users could have helped ensure that the product is more tailored to users' actual needs and preferences.

- ***Generalizations***: Testing of the product was limited to Sri Lanka. Testing against only one country can lead to limited perspectives and generalizations. Each country has unique cultural, socioeconomic, and technological contexts that can affect users' interactions and perceptions of the product. What functions well for users in one country may not be as effective or suitable for users in another. This one-sided approach can also lead to tunnel vision, where the focus becomes overly fixated on specific features or requirements identified during testing in a single country. Our goal is to develop a generic growth chart plugin that fulfill different health departments' needs. Thus, it is crucial to avoid becoming centered around country-specific requirements.

The limitations may have affected the project's results, potentially resulting in the product's requirements being centered around the needs of only a few countries. Acknowledging the limitations can help us see the product according to a broader range of design and functional solutions to meet the needs that must be in place for our product to be used in a globally expansive platform like DHIS2.

Chapter 4

Development Process

This chapter covers the development process and its essential elements. It aims to provide insight into our process of developing a product that can bridge the gap between manual methods of growth monitoring and DHIS2 by developing a generic and adaptable growth chart plugin. Before exploring technologies, planning aspects, and quality assurance, the foundation for the development process - requirement elicitation - must be inspected.

4.1 Role of Requirement Elicitation

Requirement elicitation is an essential stage of the development process. It establishes an understanding of requirements and stakeholders' demands. These requirements form the foundation for the entire development process, as every aspect will be based on the requirements gathered during this elicitation.

4.1.1 Understanding Stakeholder Needs

Requirements are gathered through *qualitative interviews* with stakeholders, introduced in *3.1.1 Requirements - Qualitative Interviews with HISP-experts* on page 11, highlighting the requirements for a generic growth chart that can accommodate a wide range of health systems. These interviews serve as the foundation of the requirement elicitation and ensure that the requirements address the core components of the problem statement: displaying children's growth over time, supporting different datasets, and supporting both *Z-scores* and *percentiles*.

4.1.2 Analyzing Core Requirements

When requirements are gathered, they need to be analyzed and prioritized. This involves assigning each requirement a priority level and an internal rank. By sorting these requirements based on level and rank, we can focus on addressing the critical issues first, laying the groundwork for development process activities. The priority level ranges from *A* to *C*, and the internal rank ranges from *0* to 999. Below is an explanation of the different priority levels:

- ***Priority A***: Grade A priority is the highest grade and refers to important features critical to the project's success.
- ***Priority B***: Grade B priority refers to important features that are not critical.
- ***Priority C***: Grade C priority features are less important and not a necessity for the final product.

4.1.3 Foundation for Development

The requirements identified and analyzed during the requirement elicitation serve as a foundation for the development process. By aligning our development with these requirements, we guarantee that the resulting product meets stakeholders' needs and expectations. However, this foundation is subject to change.

4.1.4 Iterative Requirement Elicitation

Even though we utilize principles from the linear Waterfall methodology in our requirement elicitation process, we found that the elicitation needed to be iterative, rather than a one-time event. This approach aligned with our use of principles from Scrum and Kanban in the design and development process. As the project progresses, new insights emerge, additional requirements may be identified, and priorities could change. Potential changes clarify that the requirement elicitation process should be iterative. Additionally, communicating with stakeholders via product demonstrations and meetings during the development process ensures that requirements and priority levels are continuously up to date. This iterative approach to requirement elicitation aligns with the principles of the agile development methodologies introduced in *3.1.2 Design and Development - Development Methodology* on page 11.

4.2 Technologies

Several key technologies have been used to facilitate the development process. Each technology plays a crucial role and contributes to various stages of the project. This section provides an overview of the key technologies, including their functionality, benefits, and contributions to the development process.

4.2.1 Jira Software

Jira Software is a solution that focuses on agile methodologies in software development (Li, 2018, p. 5-6). We primarily utilize Jira Software's backlog and issue board functionality, which enables effective project management and contributes to our agile methodologies. We organize the issues into four columns on the issue board, illustrated in Figure 4.1. This organization assisted us in monitoring the status of issues and project progress.

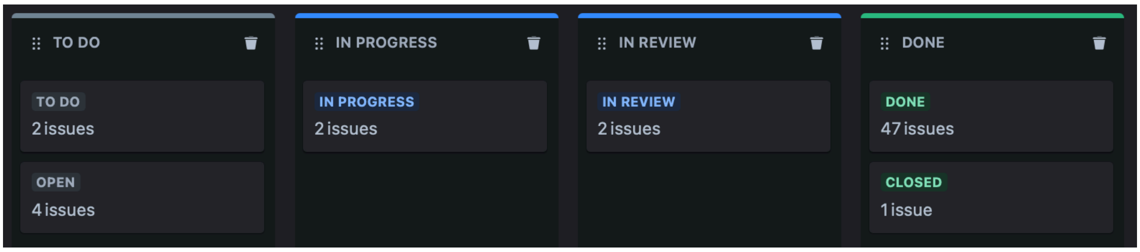


Figure 4.1: Jira board column configuration

In addition to using separate board columns, we also employed three distinct issue categories:

- ***Feature***: *Feature issues* refer to new functionalities in the software.
- ***Bug***: *Bug issues* refer to existing errors or problems within the program.
- ***Tech***: *Tech issues* are focused on technical tasks, such as code restructuring or performance improvements.

4.2.2 GitHub and GitHub Actions

We used *GitHub* repositories and *GitHub Actions* for collaboration and control of the development process. GitHub provides several features, such as branches, reviews, and pipelines, ensuring efficient project management. The project repository is open-source and is publicly accessible: *GitHub Repository*.

Branches

Branches let developers work together simultaneously, without interfering with each other (Zou et al., 2019, p. 306). A branch is created when an issue is moved from the ”TO DO” column to the ”IN PROGRESS” column on the *Jira board*, illustrated in Figure 4.1.

Branches are created from the *master branch*, which serves as the main code base used in production. Creating branches from master allows code for changes without affecting the production code. The branch is eventually merged back into the master branch with a pull request.

Pull Request

To facilitate collaboration, we use *pull request*s (PR) on GitHub. Each branch is associated with its own PR, providing an organized workflow for code integration. PRs provide control and overview of changes made in a branch, an essential requirement for reviews.

Reviews

The ability to *review* code changes and fix errors is provided by PRs. We used this feature to inspect changes made to the code, ensuring code quality. When the alterations made in a branch and PR are approved by another team member, the branch can be merged into master, and the PR can be closed.

Pipeline (CI/CD)

GitHub Actions is a popular tool facilitating the creation of *Continuous Integration/Continuous Delivery* CI/CD in *YAML* - the markup language for GitHub Action workflows - to test, build, and deploy code (Heller, 2021, p. 4-6). We actively use all of these features in the development process, which is further elaborated in *4.5.4 Pipeline (CI/CD)* on page 31.

4.2.3 React

React is a library used for web applications and user interfaces (UI) (Narayn, 2022, p. 1). The library allows the creation of discrete UI components that can be combined into complex systems. Components allow us to break down the application into smaller, reusable components and combine them seamlessly. Figure 4.2 on the following page illustrates how we split the growth chart plugin into smaller components, as well as the component render structure.

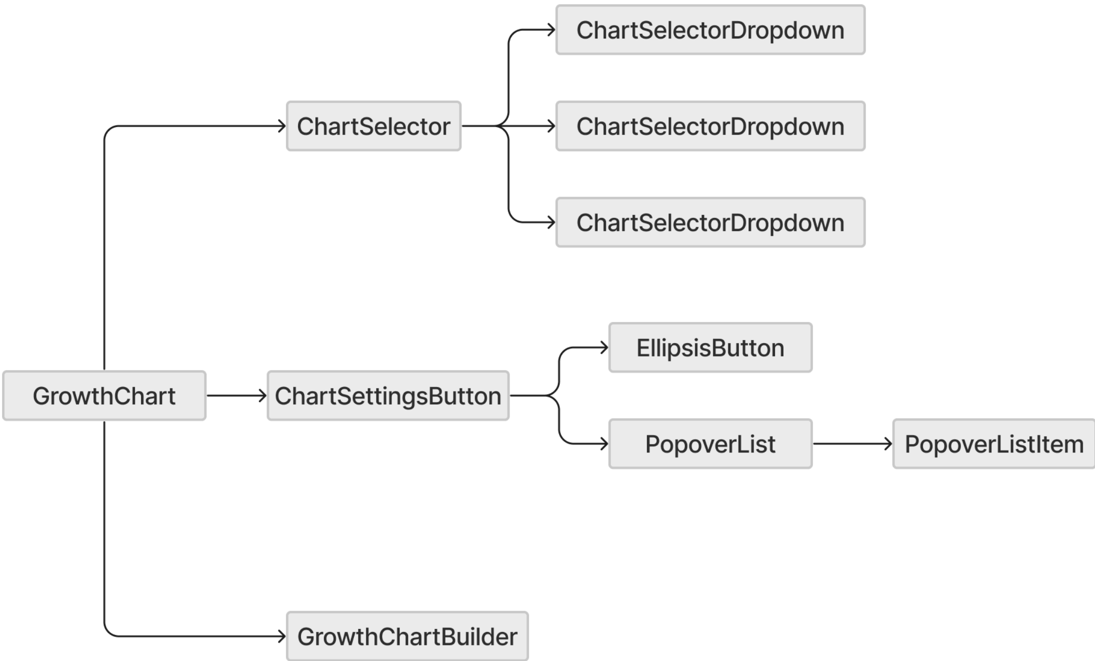


Figure 4.2: Component render structure

An example of reusable components is illustrated in Figure 4.2, as *ChartSelectorDropdown* component is reused three times in the *ChartSelector* component. This renders three drop-downs, as illustrated in Figure 4.2 and in the *ChartSelector* component's code, displayed in Listing 1.

Listing 1 ChartSelector reusing the ChartSelectorDropdown component, creating three dropdowns.

```
return (
    <div className='flex flex-wrap w-full gap-2 text-sm'>
        <ChartSelectorDropdown
            title={gender}
            items={Object.values(GenderCodes)}
            handleItemChange={setGender}
            isDisabled={isDisabled}
            dataTest='CGC-gender-dropdown'
        />
        <ChartSelectorDropdown
            title={CategoryToLabel[category]}
            items={Object.keys(chartData).map((key) => chartData[key].categoryMetadata.label
                )}
            handleItemChange={handleCategoryChange}
            dataTest='CGC-category-dropdown'
        />
        <ChartSelectorDropdown
            title={dataset}
            items={Object.keys(chartData[category].datasets)}
            handleItemChange={handleDatasetChange}
            dataTest='CGC-dataset-dropdown'
        />
    </div>
);
```

4.2.4 TypeScript

TypeScript is a programming language that builds on the JavaScript programming language by adding static typing (Jansen et al., 2016, p. 2-7). Static typing provides syntax checks at compilation, enabling the report of errors when type mismatches occur without adding overhead to the program execution. TypeScript enabled us to catch errors and potential issues during the development process and enhance our code's overall quality and reliability.

4.2.5 ESLint

ESLint is the most popular linter (static analysis tool) for JavaScript (Tomasdottir et al., 2020, p. 863-864). Linter's maintains code readability and detect possible errors. ESLint is designed to be flexible and easily customizable. We used ESLint actively during the development process, notably as part of our pipeline. By using ESLint, we ensure code readability and detect possible errors before pushing the code to the production instance.

4.2.6 Cypress

Cypress is a test automation framework designed for testing web applications (Mwaura, 2021, p. 3). We use this framework to perform component tests on the growth chart's components. Additionally, we integrated the component tests into the *pipeline* to ensure the application's reliability and prevent changes from compromising other functionality. The specific usage of this *test automation framework* will be further explored in *4.5.2 Component Tests* on page 31.

4.2.7 GitHub Copilot and ChatGPT

We also used *Artificial intelligence* (AI) in the development process by using *GitHub Copilot* and *ChatGPT*. These powerful AI tools assist us in generating code snippets, providing suggestions, and enhancing productivity in the development process.

4.3 Planning

This section explores the different aspects of planning, which is vital for ensuring a project's success.

4.3.1 Timeline

We initiated the project by establishing a timeline, allocating time for the different tasks, and setting due dates. Subsequently, we created a project plan, where we mapped out the project's key activities, illustrated in Figure 4.3. A more detailed description of each of the milestones is provided in the *Appendix*, under *A.10 Project Plan* on page 72.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Thesis kick-off	Milestone	All members	100 %	10.01.2024	21	31.01.2024
Requirement elicitation	Milestone	All members	89 %	01.02.2024	15	16.02.2024
Design and development	Milestone	All members	50 %	12.02.2024	32	15.03.2024
Implementation and testing	Milestone	All members	0 %	11.03.2024	28	08.04.2024
Mandatory assignments	Milestone	All members	75 %	23.01.2024	37	29.02.2024
Bachelor thesis	Milestone	All members	2 %	10.01.2024	132	21.05.2024

Figure 4.3: Milestones for the project

Development Process Timeline

The "Design and Development" milestone timeline indicated that this milestone would last 32 days, as illustrated in Figure 4.4. The milestones include designing a growth chart, planning the development process, developing the growth charts, creating a configuration file, and developing functionality for preterm babies.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Design and development	Milestone		80 %	12.02.2024	32	15.03.2024
Designing a Chart	On Track	All members	100 %	14.02.2024	7	21.02.2024
Planning the developement	On Track	All members	100 %	12.02.2024	7	19.02.2024
Developing the Charts (generic chart)	Med Risk	All members	100 %	14.02.2024	28	13.03.2024
Create configuration file	On Track	All members	100 %	04.03.2024	11	15.03.2024
Create functionality for preterm babies	High Risk	All members	0 %	07.03.2024	7	14.03.2024

Figure 4.4: Development process timeline

4.3.2 Risk Management

Risk management was conducted as a crucial part of planning the development process. It involves identifying, assessing, and mitigating potential threats. We adopt David Sutton's enhanced risk management matrix (Sutton, 2021, p. 76) to track these threats.

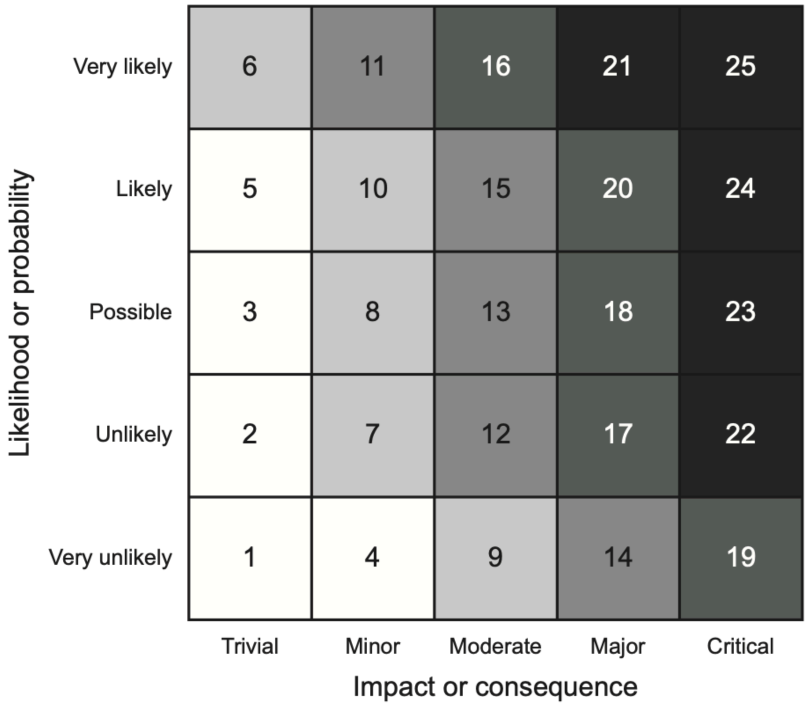


Figure 4.5: Enhanced risk management matrix (Sutton, 2021, Figure 6.2).

The risk management and assessment can be seen in Table 4.1, showing *threats*, *probabilities*, *consequences*, *risks*, and *mitigation techniques*. The risk is calculated using the risk management matrix Figure 4.5, and sorted from *high* to *low* risk.

Table 4.1: *Risk assessment table*

Threat	Probability	Consequence	Risk	Mitigation
Time shortage	Very likely	Critical	25	Prioritize tasks and follow project plan deadlines.
Timeline delays	Very likely	Major	21	Plan the project. Delegate more time for the task.
Dependencies shortage	Likely	Major	20	Contact the developer team of the Capture app and plugin functionality.
Project code loss	Very unlikely	Critical	19	Store code locally and on GitHub.
Documentation loss	Very Unlikely	Critical	19	Store documentation locally and in the cloud.
Lack of expertise	Likely	Moderate	15	Contact the task provider, Devotta, if any assistance is necessary. Contact experts from HISP.

Long Term absence	Very unlikely	Major	14	Eat healthy, exercise, and take breaks.
Conflict with task provider	Very unlikely	Major	14	Weekly meetings with task provider
Requirement alterations	Possible	Moderate	13	Thorough requirement solicitation and conversations with professionals regarding the requirements.
Stakeholder disagreement	Unlikely	Minor	7	Be clear in the communication. Regular updates with the task provider and stakeholders throughout the project.
Temporary absence	Very likely	Trivial	6	Share calendar within the team.

4.3.3 Requirement Gathering

Requirement gathering includes *identifying*, *analyzing*, and *documenting* the project requirements. Gathering the requirements is an essential part of the planning process, as it clarifies the project objectives and establishes a clear understanding of stakeholders' needs. We gathered, analyzed, and documented the requirements in a requirement specification, as illustrated in Table 4.2. After analyzing, we organized the requirements by their priority and internal rank, defined in *4.1.2 Analyzing Core Requirements* on page 19. This method was used to prioritize tasks when conducting sprint planning and creating issues.

Table 4.2: *Requirement specification*

Req ID	Requirement title	Priority	Internal rank
R.1	Create a chart with WHO's standard.	A	950
R.2	Chart shows WHO's reference lines.	A	900
R.3	Chart should be capable of plotting growth variables.	A	800
R.4	Create an implementation guide.	A	800
R.5	Change between different growth chart indicators.	A	750
R.6	Configuration file for mapping local variables with DHIS2 variable IDs.	A	700
R.7	Collect growth variables from the selected child in Capture	A	650
R.8	Possible to change age interval for the growth chart.	A	600
R.9	Tooltip with x and y values.	A	550
R.10	Possible to change between Z-scores and percentiles.	A	500
R.11	Weight support for both grams and kilograms.	B	800
R.12	Support for custom country-specific references.	B	750
R.13	Automatic sex selection based on the sex of the child.	B	600
R.14	Support for preterm babies.	C	800
R.15	Possible to convert chart to PDF	C	700
R.16	Possible to change time interval alternatives in chart configuration	C	650
R.17	Functionality to toggle annotations on or off.	C	550

4.3.4 Design

Design choices were primarily made during the development process. For the design, we used *WHO's growth* chart standard as a reference, illustrated in Figure 4.6. The widespread use of the WHO growth chart standard is why we decided to develop a growth chart with a design resembling WHO's growth charts. The design choices can be separated into four sections: the *growth chart*, the *Chart Selector Drop-downs*, the *Tooltip*, and the *Chart Settings button*.

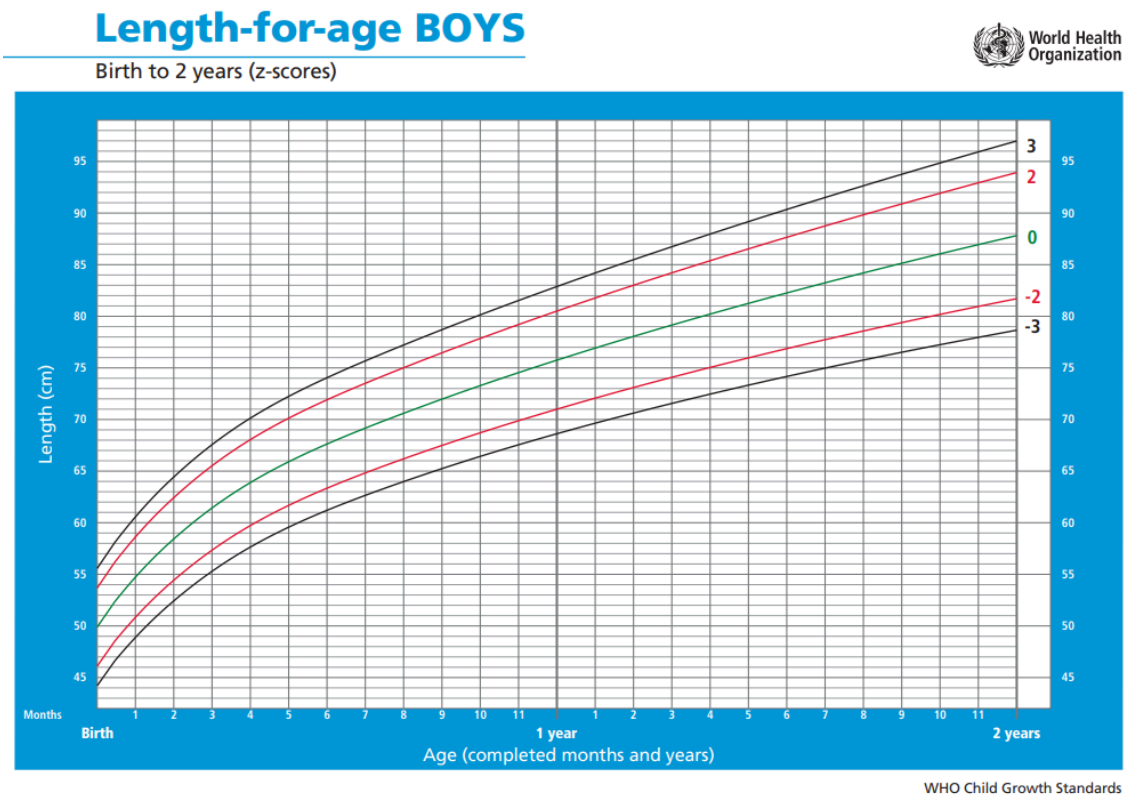


Figure 4.6: WHO's growth chart design (World Health Organization, n.d.)

Growth Graph Design

The growth graph design is shown in Figure 4.7. We use the WHO's *line coloring*, where green represents SD 0, yellow SD 1, red SD 2, and black SD 3. Additionally, we use similar line end annotations, displaying which SD the line represents.

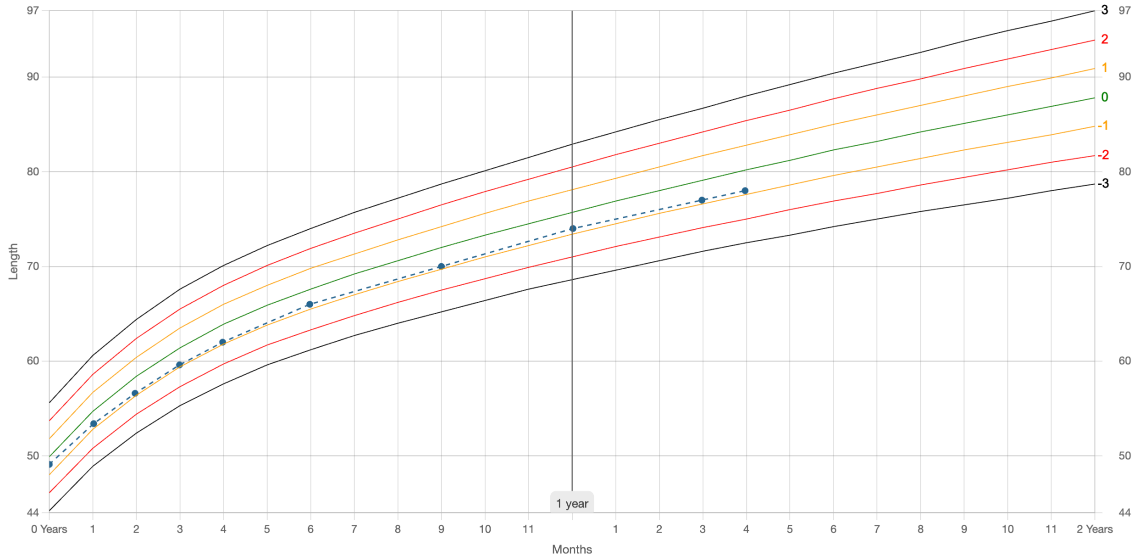


Figure 4.7: Growth graph design

Chart Selector Drop-down Design

The chart selector drop-downs provide the possibility to select what chart to render. There are three different drop-downs, one for *sex*, one for *indicator*, and one for time *interval*, as illustrated in Figure 4.8 below. *Indicator* refers to the type of growth chart, for example, length for age and weight for age, while the *interval* is the time interval. As illustrated, the sex is not a drop-down since we are looking at the growth graph in Capture with a boy selected. If there is no sex, this becomes a drop-down where you can select either "Boy" or "Girl". The drop-downs are designed to allow the user to navigate between the different growth charts easily.

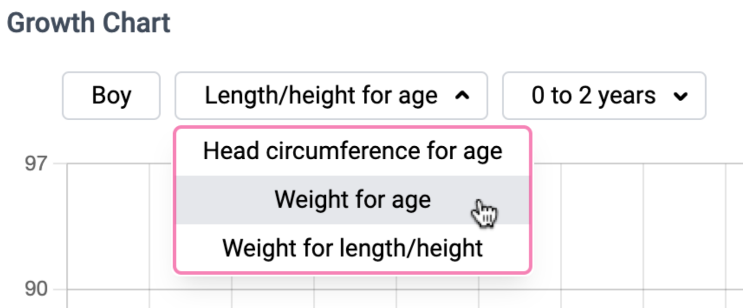


Figure 4.8: Chart selector drop-downs

Tooltip Design

The tooltip is designed to provide a more detailed view of a data point in the growth chart. It appears when hovering over a plotted data point in the graph. The design includes a date, value, and calculated age for the hovered data point, displayed in Figure 4.9. This gives healthcare workers a precise reference point with relevant information.

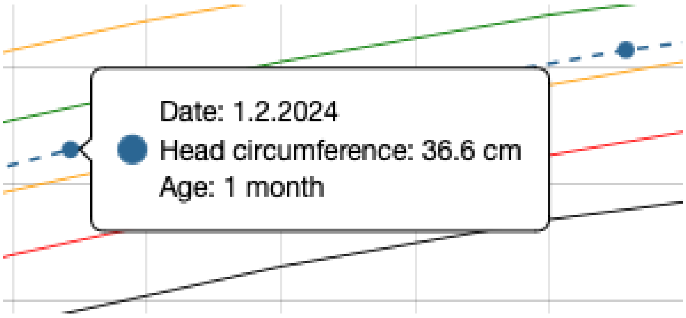


Figure 4.9: Tooltip design

Chart Setting Button Design

The design for chart settings is a drop-down menu accessible through a three horizontal dots button, as illustrated in Figure 4.10. Clicking this button displays available settings and additional features. The product at the project conclusion only had one additional feature, the "Convert to PDF" functionality.

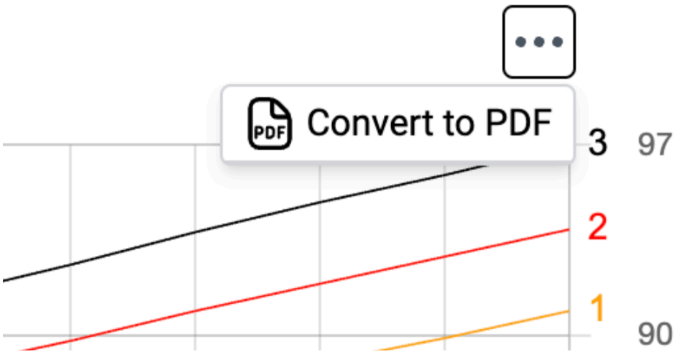


Figure 4.10: Growth chart settings design

4.4 Development Life Cycle

The development life cycle is an important part of the development process. It encompasses several stages, starting with creating an issue and ending with deployment in the production instance. This section provides detailed information about each stage in this life cycle.

4.4.1 Issue Creation

Issues were mainly created using *Jira Software* during sprint planning. However, some issues were created during a sprint if a severe error was detected or we required additional tasks. We focus on accurate and descriptive *issue descriptions*, ensuring a clear issue scope. Issue descriptions provide information about what should be done in that particular issue.

4.4.2 Issue Assigning

Once created, issues can be assigned to a team member ensuring sprint progression and accountability. Typically, we left the issues unassigned during the sprint planning and assigned them to team members as the sprint progressed. This approach provided task flexibility.

4.4.3 Development

When an issue is assigned, the team member creates a branch and pull request for the respective issue. The member then works on implementing, as well as making necessary adjustments to address the issue.

4.4.4 Review and Pipeline (CI/CD)

Following the development, the implemented changes must be approved through a review from another team member, and the pipeline (CI/CD) must pass. The review from another team member ensures that the changes made in the branch cover the issue description and maintain a high standard of code quality, as discussed in detail in *4.5 Testing and Quality Assurance*. Additionally, pipeline checks are conducted to ensure the integrity and stability of the growth chart plugin.

4.4.5 Merge and Deploy

Once changes are approved and component tests passed, the implemented changes are ready for merging and deployment. The designated team member assigned to the issue can then merge the branch with the implemented changes into master. This merge triggers the master branch to run through the pipeline with the new code. If the pipeline passes, the master branch with the integrated changes can be deployed to the production environment and the pull request closed. The respective issue is then moved to the ”Done” column, displayed in Figure 4.1, ensuring up-to-date project progress.

Throughout the life cycle of issues, several quality assurance measures are taken to guarantee the reliability, functionality, and overall quality of the software. In the subsequent sections, we explore a comprehensive discussion of these measures.

4.5 Testing and Quality Assurance

Several quality assurance measures are implemented into the issue life cycle. In this section, we provide a detailed examination of each one of these measures and how they were implemented in the development process.

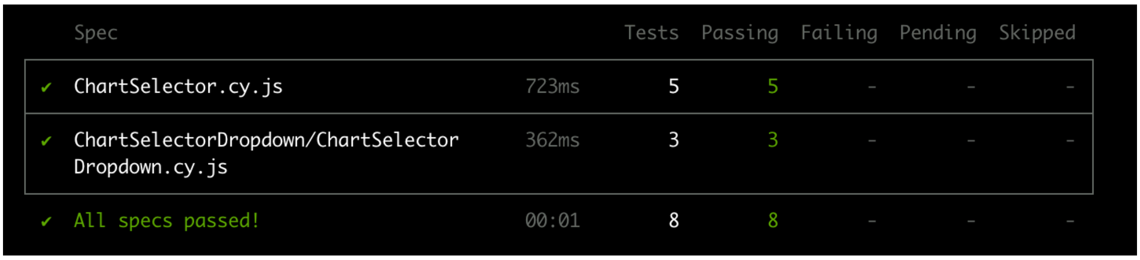
4.5.1 GitHub Review

GitHub reviews were essential to ensure code quality and identify bugs in implemented solutions. Additionally, this process will contribute to overall software quality by enabling feedback, suggestions, and collaboration within the team. We incorporated this feature by using GitHub reviews as part of our issue life cycle.

When the assigned team member had addressed the issue, they could request a review from the other team members. The other team members then reviewed the implemented changes, providing comments, suggestions, and corrections. This iterative process then continued until the reviewer was satisfied with code quality and approved the PR.

4.5.2 Component Tests

We created component tests for the chart selector drop-down component, described in 4.3.4. These component tests included five tests for the entire *ChartSelector* component, including all three drop-downs, and three tests for the *ChartSelectorDropdown* component, specifically testing the drop-down button itself, as illustrated in Figure 4.11.



	Spec		Tests	Passing	Failing	Pending	Skipped
✓	ChartSelector.cy.js	723ms	5	5	-	-	-
✓	ChartSelectorDropdown/ChartSelector Dropdown.cy.js	362ms	3	3	-	-	-
✓	All specs passed!	00:01	8	8	-	-	-

Figure 4.11: Cypress test spec

The specific test titles run for each of these test specifications can be found in Figures 4.12 and 4.13. Notably, the tests include rendering the *drop-down*, as well as changing the *drop-down value*.

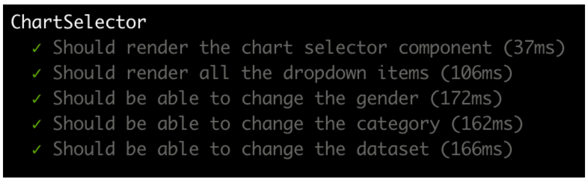


Figure 4.12: Cypress ChartSelector component test spec

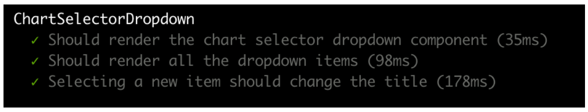


Figure 4.13: Cypress ChartSelectorDropdown component test spec

4.5.3 Absence of E2E tests

We determined that *End-to-End* (E2E) testing the growth chart plugin was unnecessary since the plugin is designed to work as a component in a more extensive system. E2E tests are designed to test all the components in a more extensive software application. Therefore, we concluded that the E2E tests should be run in the application rendering the plugin, not the plugin itself.

4.5.4 Pipeline (CI/CD)

We have actively used pipeline (CI/CD) through *GitHub Action workflows* in the *YAML markup language* throughout the development process. There are two different workflows, one for verifying the application and one for building and deploying the application, as illustrated in Figure 4.14.

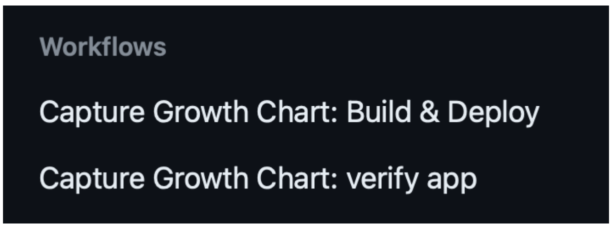


Figure 4.14: GitHub Action workflows

Verify App

The *Verify app* workflow enables quick identification of issues in the code, ensuring a reliable development process and application. Our Verify app workflow consists of four jobs: building the *application*, running *lint checks*, running *TypeScript* checks, and finally running the *component tests*. This workflow is triggered on push, meaning that these jobs are executed whenever changes are pushed to the repository. First, the workflow installs the necessary dependencies to run the application. Then, linting checks are run to identify possible syntax or code style issues using ESLint. Subsequently, a TypeScript check is run to ensure proper typing. Finally, the cypress component tests, previously described in Component Tests, are run. The workflow is illustrated in Figure 4.15.

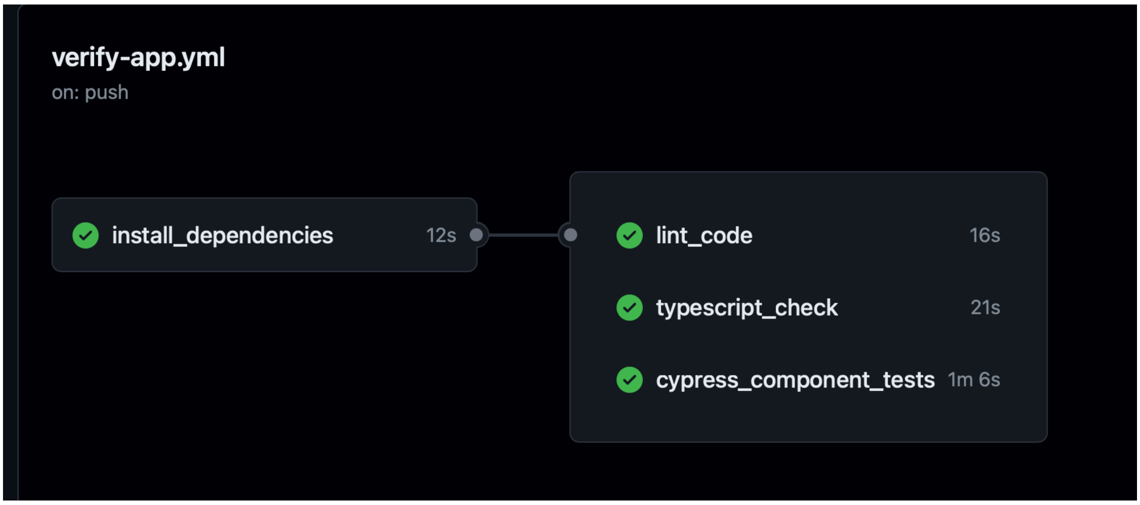


Figure 4.15: Verify application workflow test spec

Build & Deploy

The *Build & Deploy* workflow is triggered when the *Verify app* workflow completes in the master branch. This process ensures that the application is successfully verified before building and deploying the code to the production instance. The workflow consists of two jobs, *install* and *deploy*, as illustrated in Figure 4.16. Firstly, necessary dependencies are installed in the *install* job, same as in the *Verify app* workflow. Then, in the *deploy* job, the application is built and deployed to production based on the master branch.

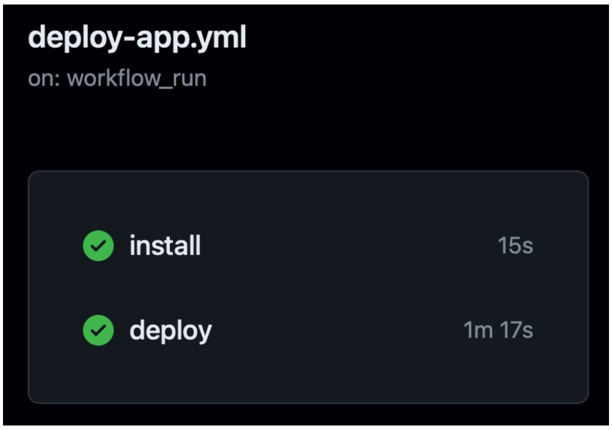


Figure 4.16: Build & Deploy workflow test spec

4.6 Product Status at Project Conclusion

In this section, we explore the product status at the end of the development process. Additionally, we look at the functionality offered in the growth chart plugin.

4.6.1 Product Status

Growth graph

Figure 4.17 presents the growth chart at the project conclusion. As mentioned in *4.3.4 Design* on page 27, the application's design is based on WHO's standard growth references. In the current solution, growth references have been added for *Length/height-for-age*, *Weight-for-age*, *Weight-for-length/height*, and *Head circumference-for-age*. Building the growth chart based on the WHO's standard references gives the application a reliable foundation.

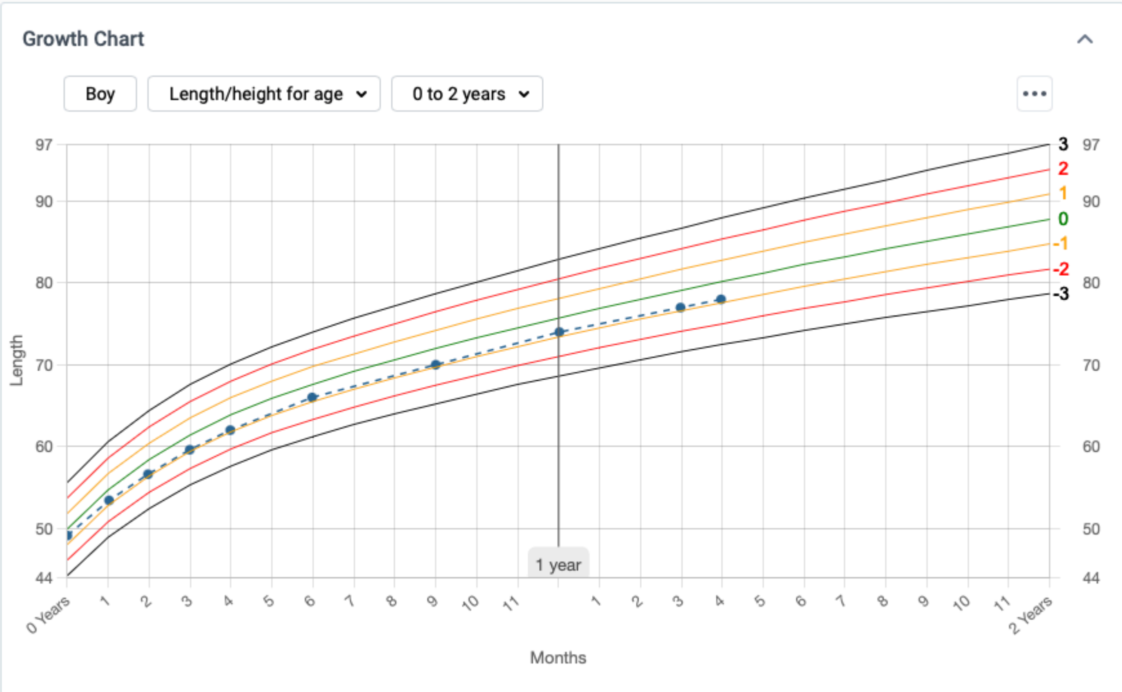


Figure 4.17: Growth chart plugin at project conclusion

Configuration file

The configuration file at the project conclusion includes two main sections: *metadata* and *settings*. For the growth chart plugin to work as intended on DHIS2, configuration needs to be added to the *Datastore Management application*, introduced in *2.3.4 Datastore Management* on page 9. An example of a configuration is illustrated in Figure 4.18 on the following page.

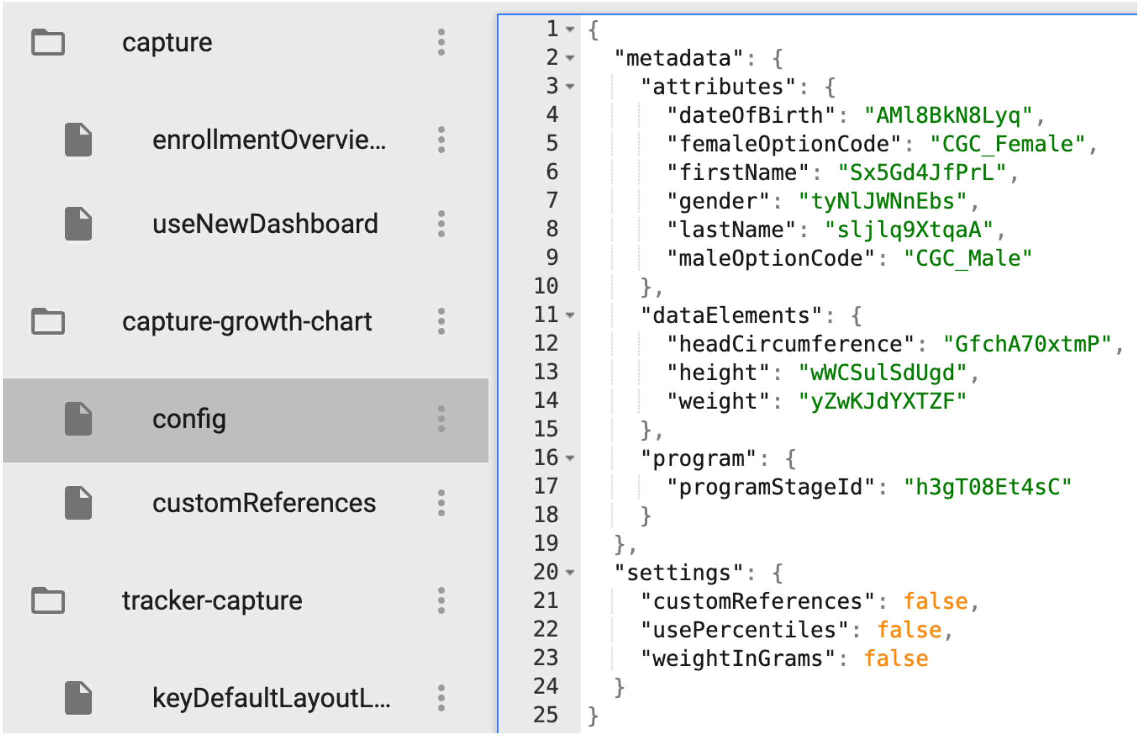


Figure 4.18: Growth chart configuration file at project conclusion

Explanation of the configuration file:

- ***Metadata***: This section provides the necessary attributes, data elements, and program IDs for the plugin. The attributes include information about the person, such as *date of birth*, *sex* (based on female and male Option Set), *first name*, and *last name*. Furthermore, data elements specify the IDs of *head circumference*, *height*, and *weight*. Finally, the program refers to the program stage ID for the program stage where the growth data is stored.

- ***Settings***: Here, the available settings for the growth chart plugin are specified. At the project conclusion, these settings include *customReferences*, *usePercentiles*, and *weightInGrams*.

  – ***customReferences***: This setting allows the user to use *custom reference values* instead of WHO's standard. The functionality is further presented in *4.6.2 Custom References* on page 35.

  – ***usePercentiles***: This setting determines whether the growth chart plugin should use *Z-scores* or *percentiles*. If this option is set to false, the chart uses Z-scores, if true, percentiles will be used. We go further into the functionality behind this in *4.6.2 Z-scores & Percentiles* on page 35.

  – ***weightInGrams***: This setting specifies if weight is provided in *grams* or *kilograms*. If false, weight is measured in kg. If true, weight will be measured in grams.

Implementation guide

We have created an implementation guide for the growth chart plugin, entailing necessary steps for deploying and using the growth chart plugin the DHIS2 platform. The implementation guide is available in the project's repository on GitHub: Implementation Guide.

4.6.2 Functionality

The growth chart plugin has several functionalities. Some features are displayed on the growth chart plugin itself, while others can be found in the configuration file.

Growth plotting

The primary functionality of the growth chart plugin is to visualize growth development for children. By plotting a child's growth data in the graph, health workers and professionals can compare the growth against the defined references displaying *standard growth patterns*. Plotting aids in the early detection of possible abnormalities and deviations, serving as a valuable tool for healthcare personnel.

Z-scores & Percentiles

Another functionality of the plugin is picking whether to use *Z-scores* or *percentiles*. This feature ensures the application's adaptability to different countries, with varying preferences, using DHIS2. A growth chart using Z-scores is illustrated in Figure 4.17, while a growth chart using percentiles is illustrated in Figure 4.19.

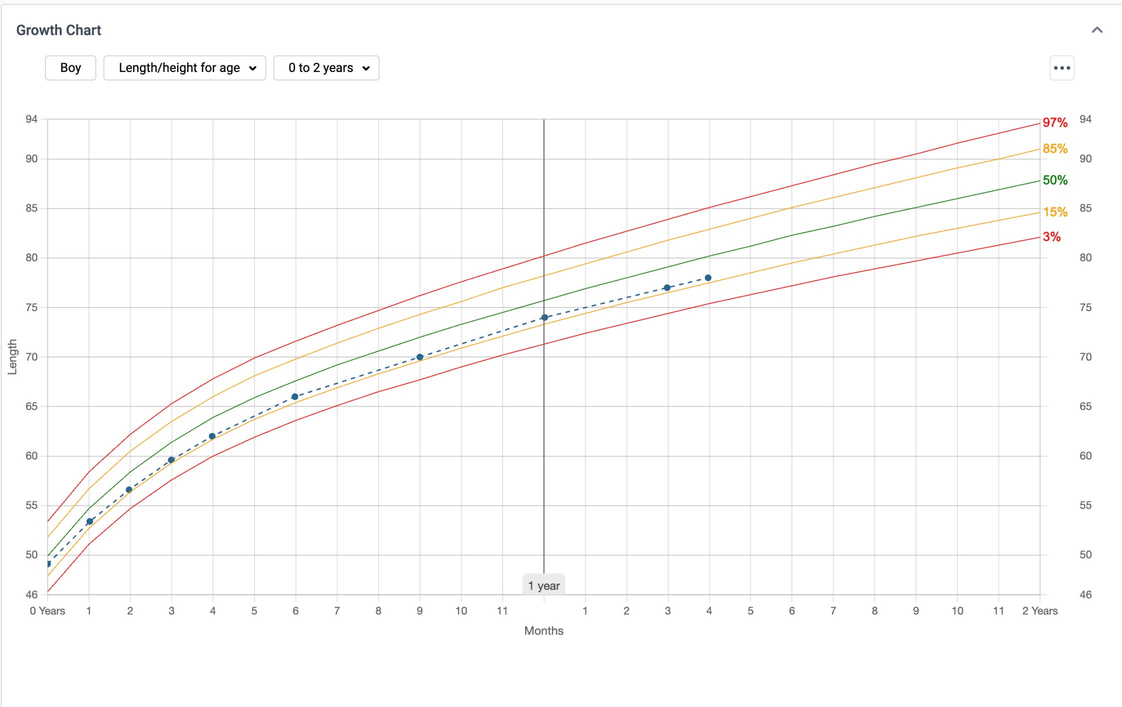


Figure 4.19: Growth chart plugin using percentiles

Custom References

The growth chart plugin supports the use of country-specific *custom references*. The functionality lets countries use their own references for standard growth patterns based on their population. The implementation guide includes the necessary steps to use custom references. An example is illustrated in Figure 4.20, where custom references from Egypt are used instead of WHO's standard references.

When comparing growth patterns based on the Egyptian population (Figure 4.20) to growth patterns based on the WHO standard (Figure 4.17) for the same person, indicator, and time interval, it is clear that the plotted growth curve varies. In the growth graph using WHO's standard, the growth follows the SD -1 line, but when using the custom references based on the Egyptian population, the child follows a growth curve between SD -1 and SD 0. This curve variation emphasizes the need for custom reference functionality in the growth chart plugin, where countries can insert their references based on their own growth data.

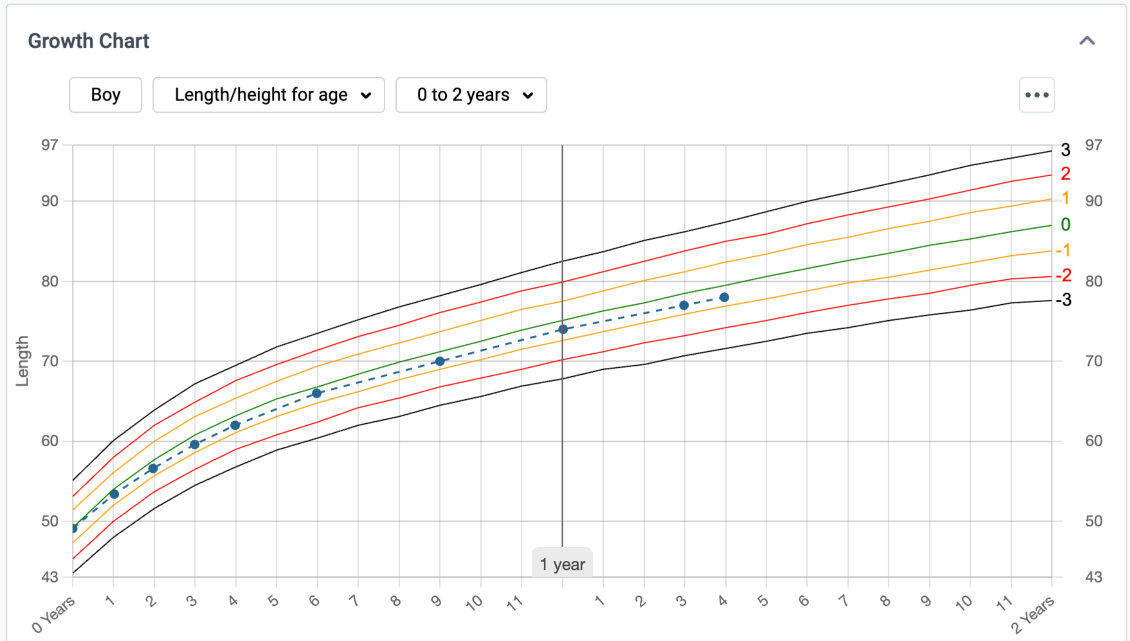


Figure 4.20: Growth graph using custom references based on Egyptian population.

PDF Generation

The growth chart plugin also has integrated *PDF conversion* functionality. This feature allows for easy conversion to PDF and printing out of a growth graph. The generated PDF includes the child's *name*, *sex*, chosen *indicator*, *time interval*, and a plot of the child's *growth*. An example PDF for *Michael Taylor* is illustrated in Figure 4.21.

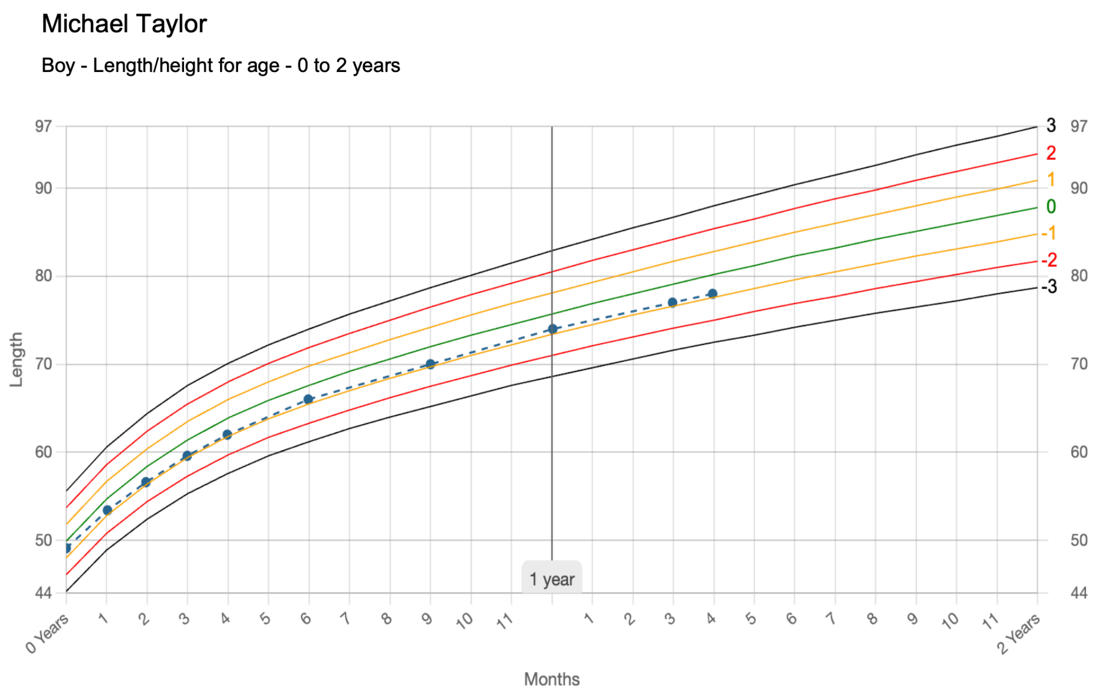


Figure 4.21: Printed growth chart.

Automatic Sex Selection

The growth chart plugin also has *automatic sex selection* functionality. This feature selects the growth chart *drop-down* for *sex* based on the selected *child's sex*. Information about the child is collected using the *tracked entity attributes* explained in *2.2.5 Tracked Entity Attribute* on page 5. However, if the sex of the child is not provided, the sex can be manually selected, as explained in detail in *4.3.4 Chart Selector Drop-down Design* on page 28.

4.6.3 Model of Product at Conclusion

Figure 4.22 on the next page, illustrates how the growth chart plugin's configuration file affects the growth chart itself, the *reference lines*, *data plotting*, and functionality.

Figure 4.22: Solution data model

This display can be split into two different main sections of the configuration guide, *metadata* and *settings*. Figure 4.23 refers to *settings*. This figure illustrates how the different settings affect the growth chart plugin, primarily how the reference lines displayed are created based on the configuration.

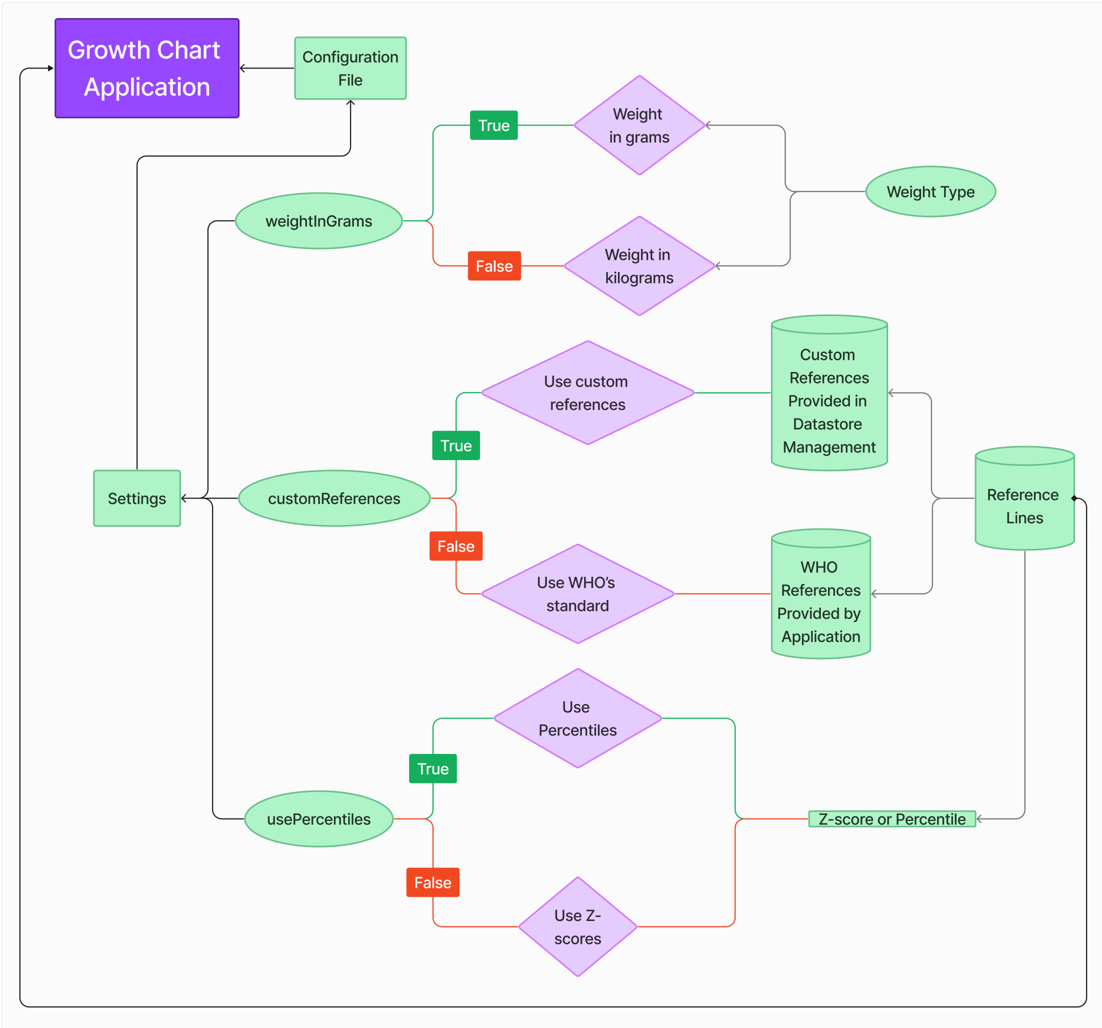


Figure 4.23: Solution data model settings

The *metadata* model, illustrated in Figure 4.24, illustrates how the metadata section of the configuration file affects the growth chart plugin. As seen in the figure, this mainly affects the plotted graph line and some growth chart functionality.

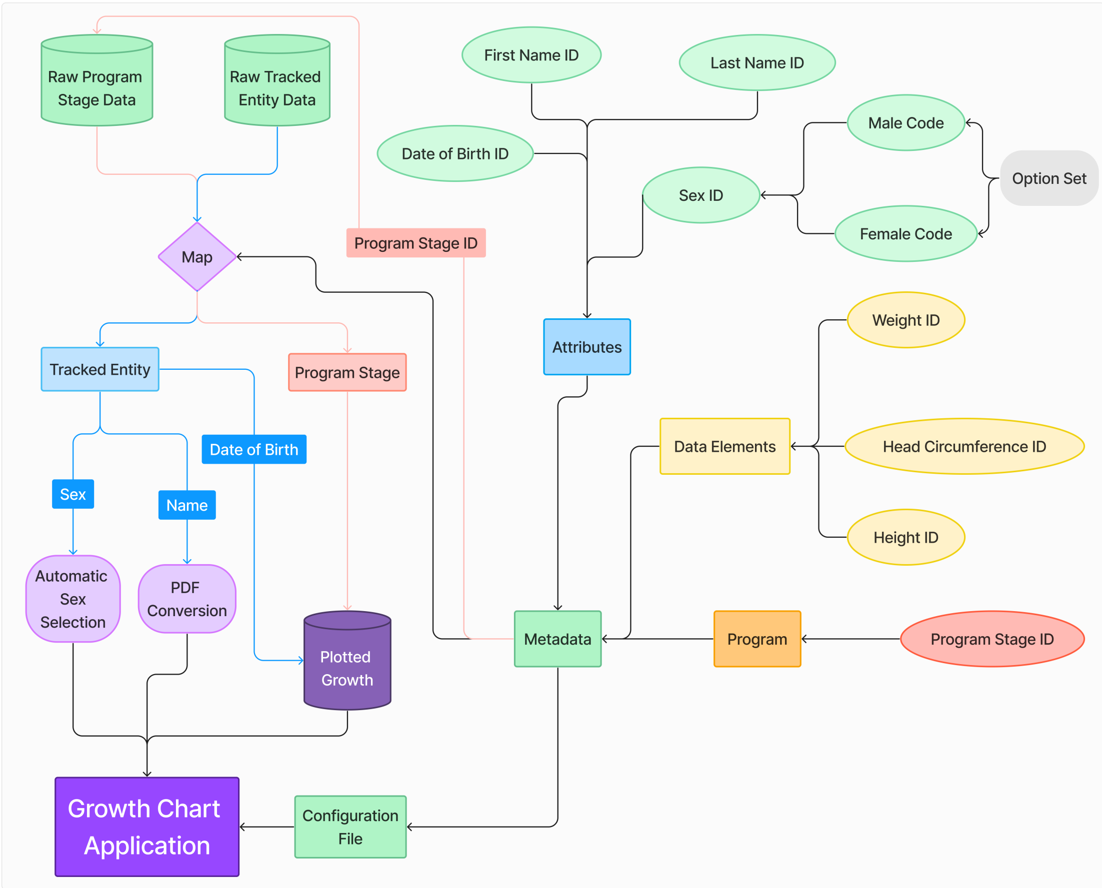


Figure 4.24: Solution data model metadata

Chapter 5
Results Evaluation

This chapter provides a comprehensive review of the project's results and an evaluation of the results. The chapter combines *qualitative* and *quantitative* data to provide a detailed insight into the functionality and user experience of the product we have developed. It begins by presenting and evaluating results from qualitative data, including *feedback* from task providers and stakeholders. Additionally, the qualitative data include *user testing* of the UI and the implementation process. The focus then shifts to quantitative results, where we have used the *SUS score* to measure the system's usability. The qualitative and quantitative data help us conclude whether the solution addresses the problem statement and research questions defined in *1.3 Problem Statement* on page 2. The chapter seeks to provide a holistic understanding of the project's results by giving a thorough overview of both qualitative and quantitative data as well as analyzing and evaluating these results.

5.1 Results from Qualitative Data

We explore the results of the qualitative data collection to provide insight into the user experience and usefulness of the growth chart plugin. The results are based on qualitative data obtained through semi-structured interviews and user testing.

5.1.1 Stakeholder Feedback

We have collected feedback from stakeholders of the project by conducting product demonstrations of the growth chart plugin. Throughout this project, close collaboration has been maintained with Devotta, HISP Sri Lanka, and HISP MENA. Devotta is the company that commissioned the project, while HISP Sri Lanka and HISP MENA are key stakeholders of the product, as well as possible users of the developed growth chart plugin. In this section, we review their feedback on the final product, described in *4.6 Product Status at Project Conclusion* on page 33.

Devotta

Devotta is the task provider of this project. Our goal has been to create a product that satisfies Devotta's requirements and allows for further development without the need for upheavals in code structure, design, or functionality. A product demonstration was held to collect feedback on the final product from Devotta, followed by a discussion of the result. The demonstration was held digitally, and all participants had the opportunity to test the product independently through a production environment available online.

In the product demonstration, we went through the solution's functionality, design, and configuration. In addition, we gave insight into the development choices made. The discussion with Devotta following the demonstration indicates that we have covered their wishes and requirements. At the same time, we received suggestions on how the product can be further improved. Feedback from Devotta highlights the following improvement points:

- ***Alternative to choose a preselected indicator in the configuration file*:** Devotta requested an option to configure which *growth indicator* should be displayed as default. Since a graph is displayed when the page is loaded, it is desirable that implementers can set this graph to show the indicator that users are most likely to use. For instance, if an implementation primarily uses the "weight for age" indicator, it can be configured to display by default.

- ***The growth chart plugin automatically chooses time interval according to the child's age*:** Instead of the user having to navigate to the graph with the desired *time interval*, the graph displayed as default can be set to an interval that corresponds to the child's age.

The two improvements proposed by Devotta were related to enhancing the functionality of the growth chart plugin to increase efficiency. When the indicator and interval are set to what the user is likely to use, it can greatly streamline usage, reducing the actions required to see the correct graph. Thus, improving user-friendliness and the users' impression of the product. Table 5.1 shows the suggested improvements by Devotta analyzed into possible future requirements.

Table 5.1: *Requirements received from Devotta*

Req ID	Requirement Title	Received From
P.R.1	Alternative to choose a preselected indicator in the configuration file.	Devotta
P.R.2	The plugin automatically chooses time intervals according to the child's age.	Devotta

Devotta Overall Assessment

"DHIS2 is the most prominent software for gathering, aggregating, and analyzing health data, and it forms the foundation of an incredible number of information management systems worldwide. DHIS2's customizable data model allows it to be configured to store information about any and all data variables, and this level of flexibility will always be one of the biggest challenges we face as core contributors. To the best of my knowledge, multiple attempts have been made in the past to create these growth chart modules, but none of them have ever been used in a real-world setting.

Generally speaking, many of these projects make the mistake of assuming that an implementation will set up their metadata to correspond with the developed software's specifications. Since most HMIS systems have been around for decades, changing their configuration will cost a huge amount of money and resources. The students have really taken a different approach to this, where they have held meetings and discussions to learn about how DHIS2 is used in the region, and how growth monitoring is managed around the world. They have shown great interest and professionalism in holding these discussions, and spent a lot of time on requirements and planning before starting the development process. The product is generic and configurable to such an extent that we feel confident that it can be used in any DHIS2 instance doing growth monitoring, and that the code quality upholds the standards of the DHIS2 community. The addition of a growth charts module in the Capture app will be a big step towards better data quality and more efficient data collection for the HISP network, and will also drive the development of more advanced tracker implementations in the future. This is a great achievement, and I am very impressed with the work they have done. They have been responsive to feedback and have shown *a great willingness to learn and improve. I am very happy to have had the opportunity to work with them, and I am looking forward to seeing the product in action. I feel very confident that it will be a great success."*

- Eirik Haugstulen
Project Lead
Devotta

HISP Middle East and North Africa (MENA)

Throughout the project, we maintained a good dialogue with HISP MENA and received feedback from them continuously. They helped establish the requirement specification and contributed to the design. As a conclusion to the project, we conducted a final product demonstration for HISP MENA. The final product was discussed, intending to uncover weaknesses that need to be improved before they can implement the plugin in the Middle East and North Africa. The feedback we received from the demonstration gave the impression that HISP MENA was satisfied with the growth chart. They particularly highlighted the function allowing to switch between both Z-score and percentile annotation for the growth references in the graph. They also expressed satisfaction with the result of the implemented design changes that they proposed at an earlier stage in the development process.

Feedback from HISP MENA also identifies a lack of functionality:

- ***Possibility of different configuration files for different uses of the growth chart*:** The solution does not allow different programs in Capture to use growth charts with individually adapted settings. For different programs in Capture to use growth charts with various settings, the solution must allow different *configuration files*, which the current solution does not support. To improve functionality, support for more than one configuration file for growth charts should be introduced.

The feedback shows important functionality we have overlooked in our assignment to create a generic plugin that can be adapted to the needs of different health departments. Despite this missing feature, HISP MENA still advises us to consider arranging a pilot project for our growth chart plugin. Table 5.2 displays feedback from HISP MENA as a possible future requirement.

Table 5.2: *Requirements received from HISP MENA*

Req ID	Requirement Title	Received From
P.R.3	Possibility of different configuration files using the growth chart in multiple applications.	HISP MENA

HISP MENA Overall Assessment

"The growth chart is an important step in the child's development milestones. Following international standards makes evaluating children's development easier and facilitates communication between healthcare providers. Developing growth charts in DHIS2 is a significant improvement in the software that directly impacts the quality of services and offers substantial benefits for end users. In addition, this project implements the WHO standards using the plugin feature in the Capture app which will pave the way for future

innovations and similar initiatives to enhance health data management and utilization. Moreover, the interfaces are easy to handle, and the fact that the system administrator can switch between more than one reference gives it an additional advantage. Growth charts are widely requested in countries that implement DHIS2 at the national level and this project contributed to addressing the gap in this area. Previously, children's development was monitored using indicators in DHIS2 without visuals. In this project, the system enables the real-time monitoring of children's growth patterns, facilitating the early detection of malnutrition and other growth-related issues. Finally, this will support policymakers in timely and informed decision-making at the individual or population level, ultimately contributing to improved health outcomes and more effective public health interventions."

- Hanin Saadah
DHIS2 Implementation Specialist
HISP MENA

HISP Sri Lanka

Feedback from HISP Sri Lanka has been collected through discussion and a product demonstration with HISP Sri Lanka employees, which is shown in Figure 5.1. With the growth chart's most critical functionality developed, we traveled with a representative from Devotta to Sri Lanka to evaluate the plugin with HISP Sri Lanka employees, and to test the product in a real environment. The field trip to Sri Lanka gave a unique insight into the product's ability to handle real-world needs and challenges.

Figure 5.1: Product demonstration with HISP Sri Lanka

The feedback we received gave the impression that the product satisfies most of their requirements and expectations. They particularly appreciated that we have a function to convert the graph to PDF. This is a function they did not ask for which we implemented on our initiative. In addition to positive feedback on the functionality and design, they provided feedback on how to improve the product further. The most important suggestions for improving the product are reiterated below:

- ***Customizable PDF header*:** A suggestion for improvement is to allow users to customize the header above the graph when converting to PDF. In the current solution, the header contains the *name*, *gender*, *indicator*, and *interval*. Sri Lanka requires functionality to enter custom attributes such as *National ID* on the PDF.

- ***Fill color between lines in the graph*:** We have implemented the design that WHO uses in its growth curves. In Sri Lanka, we experienced that the growth curves they use in their pen-and-paper solution do not have simple lines, but color zones. An improvement of the product would be to implement fill colors as an option in the configuration file.

- ***Automatic deviation detection and notification*:** A system that automatically detects when a child deviates from its growth pattern would be optimal to reduce human error. This is a more advanced improvement than the others mentioned previously, but it would significantly improve the product.

- ***Display intervention notes in the tooltip when deviations occur in the reference point*:** It was proposed to include several more functions in the tooltip of the graph. The current solution shows the date of measurement, the indicator, and the child's age when the measurement was taken. Additional functionality proposed by HISP Sri Lanka is that the tooltip should be able to display the actions a clinician has taken if deviance has been discovered. (e.g., supplements started, boost of vitamins, etc.).

- ***Include functionality for remote distribution*:** HISP Sri Lanka wants the ability to share the PDF with the child's parents. After a short discussion around this improvement potential, we find integration with WhatsApp to be a useful and realistic future solution to satisfy this feedback.

The feedback from Sri Lanka includes several improvements to the product. These improvements are important to consider in future development to strengthen the product's ability to replace current solutions. In Sri Lanka, we got the impression that the current pen-and-paper system works well. Therefore the benefits must be significant for them to consider transitioning to our solution to monitor growth. Further development of the improvements mentioned above would place the plugin in a stronger position to be able to replace the currently established systems. Table 5.3 display these improvements specified as possible future requirements.

Table 5.3: *Requirements received from HISP Sri Lanka*

Req ID	Requirement Title	Received From
P.R.4	Customizable PDF header.	HISP Sri Lanka
P.R.5	Fill color between lines in the graph.	HISP Sri Lanka
P.R.6	Automatic deviations detection and notification.	HISP Sri Lanka
P.R.7	Display intervention notes in the tooltip when deviations occur in the reference point.	HISP Sri Lanka
P.R.8	Include functionality for remote distribution.	HISP Sri Lanka

HISP Sri Lanka Overall Assessment

"Sri Lanka has been performing quite well over the last few decades in maternal and child health-related indicators compared to other countries in the region. One area Sri Lanka is struggling to excel at is nutrition. It has been identified that in order to enhance the nutrition status of children in the country, the collection of aggregate data will not suffice. The country demands individual-level nutrition monitoring conducted at the field level. In order to achieve this, the Ministry of Health has been trying to incorporate growth charts into dhis2 the digital information system currently deployed in the country. However, a significant challenge was growth charts was not incorporated as a core application in dhis2. This was a significant concern since sustaining custom developments has always been a challenge. Hence, the latest exercise by the UiO and the team of bachelors students is of utmost importance to the current dhis2 implementation in the country. Furthermore, their field visit helped them understand the context of the country as well as real-life concerns expressed by the end users. The ministry is awaiting deployment of the growth charts plugin into their current dhis2 implementation as soon as the development is completed. In addition, the field visit also benefitted the end users and the ministry stakeholders by creating an opportunity to interact with the development team contributing to dhis2."

- Pamod Amarakoon
Lead
HISP Sri Lanka

5.1.2 User Test of User Interface (UI)

The user tests conducted on the UI in Sri Lanka provided insight into its functionality and usability among healthcare professionals. The user tests aimed to map whether the end user finds the product intuitive and effective. The tests included healthcare staff at clinics in Sri Lanka with little knowledge of the setup and the DHIS2 platform, which provided us with a nuanced understanding of the UI's intuitiveness. A total of three user tests of the product were conducted. Below is a summary of the tests, and recommendations for improvement.

- ***UI Test 1***

  During the first user test, the participant encountered challenges related to language barriers and locating functions. Despite these obstacles, the participant was able to complete the tasks with minimal guidance. The execution of the test indicated the importance of comprehensible interface elements.

  We observed that the product was generally user-friendly and intuitive enough that even users with limited English skills and limited familiarity with DHIS2 could navigate through it.

  Identified weaknesses were that some functions, such as finding specific data in the graph and navigating to the correct chart, were difficult for the user. This may indicate that the interface could be improved to make it easier to find and interpret the data. Using the drop-down lists to switch between indicators and time intervals did not appear intuitive. Furthermore, we found that it was difficult for the test subject to find the print button.

  The test participant also tended to prefer using the "Child visits" program stage table over the graph to find information, which may indicate that the graphical presentation of the graph can be improved to be more informative and user-friendly. Another improvement to make the graph more visible is to place the plugin in a more strategic place in the Capture interface. In the configuration used in the user test, the graph itself is placed far down the page. The placement requires the user to scroll down to see the graph, which may be one of the reasons why they would rather use the "Child visits" program stage table to find data for specific data points.

- ***UI Test 2***

  Many of the same weaknesses revealed in the first user test resurfaced in the second. In this test, a critical error was also uncovered in the plugin's tooltip, which displayed the wrong age for the child. Despite this, user test two was completed successfully, with minimal guidance.

  A strength of the product was observed in the second user test. The test participant became comfortable with switching between indicators and time intervals. This shows that the second test participant quickly built an understanding of this functionality without training in advance.

  The identified weaknesses substantiate several of the findings from the first user test. The second test participant had difficulty locating the growth chart and needed assistance performing basic navigational tasks, such as scrolling down the page, due to limited familiarity with trackpads. This test also noted that the user often preferred using the "Child visits" program stage table rather than the graph to find growth data. This indicates that graphical presentations must be improved to be more informative and user-friendly.

  An additional weakness that was uncovered in this test was how the location of the data point was found. The test subject used the date of birth and the date of the measurements to calculate the position of the data point in the graph. This is not necessary because the date of the measurement is recorded in the tooltip. It can be identified faster by hovering the data points and checking the date on the tooltip instead of manually calculating it and subsequently checking the graph. This indicates that the participant did not expect there to be a tooltip when hovering over a data point. Thus, the tooltip feature should be more prominent.

- ***UI Test 3***

  Test subject three had experience with DHIS2 and was proficient in English. However, we noticed that the user encountered difficulties with navigation and struggled to understand certain functions. In addition, during the test, a critical error was discovered in the growth chart plugin, which caused an age miscalculation. Despite challenges, the test subject completed all the tasks in the test.

  A strength identified in the test was that the test participant effectively completed four out of ten tasks. She easily navigated to the right child, noted the child's weight easily, and quickly found the functions to convert the graph to PDF. She explained that she recognized the elements from earlier. This suggests that the user-friendliness of the interface was sufficient to support users with varied technical competence and experience with DHIS2.

  However, the test subject encountered some challenges related to tasks involving selecting indicator and time interval. In this test as well, we observed that navigating to the correct indicator and time interval is not intuitive for the user. She also discovered an error in the graph, showing the wrong age for the child, indicating flaws in the age calculation. This is a critical flaw that significantly compromised the product's reliability.

Possible Improvements

The improvement potential for our growth chart plugin in DHIS2 was revealed through UI user tests, examined in *5.1.2 User Test of User Interface (UI)* on page 46. The user tests identified several areas for improving usability. Below is an analysis containing possible improvements based on the challenges observed during the UI user tests:

- ***Interface improvements***: Improve the function to change the indicator and the time interval to improve usability. All UI user tests show that the current solution with *drop-down lists* weakens its intuitiveness.

- ***Error correction***: More careful testing to avoid critical errors, such as incorrect *age calculations*, to ensure the growth chart plugin's reliability and accuracy.

- ***Clarification of functions***: The function to convert graphs to PDF is located behind a *menu button*. The menu button is the button with three dots in Figure 5.2. Currently, a menu button is used to facilitate further functionality in the graph. Since this menu has only one function at the project conclusion, the solution would be more user-friendly if the button for PDF conversion was available directly from the graph without the user having to open a menu. See Figure 5.3 for a new and improved header suggestion.

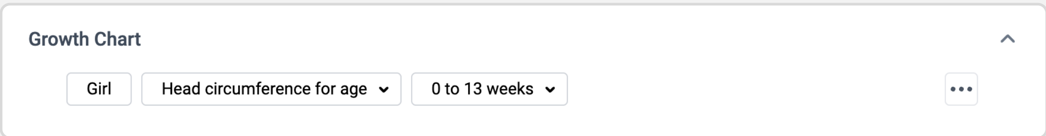


Figure 5.2: Growth chart header in current solution

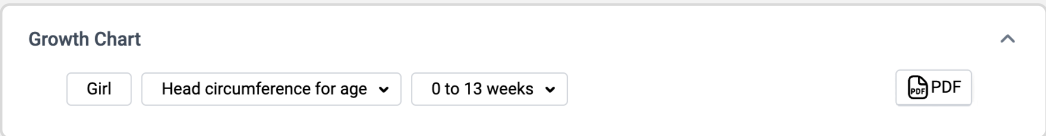


Figure 5.3: Improved growth chart header

Recommendations

Based on the analysis providing possible improvements, we have formed the requirements displayed in Table 5.4

Table 5.4: *Requirements received from UI user tests*

Req ID	Requirement Title	Received From
P.R.9	Implement a more intuitive interface for changing indicators and time intervals.	UI Tests
P.R.10	Add additional component tests to catch critical errors and ensure the growth chart plugin reliability.	UI Tests
P.R.11	Improve accessibility to the convert to PDF functionality.	UI Tests

5.1.3 User Test of Implementation

User testing of the implementation gave insight into how intuitive our implementation guide was for individuals with knowledge of the DHIS2 platform. The purpose of testing HISP employees with varying backgrounds and experiences was to build an understanding of how we can simplify and streamline the work of implementing our plugin in Capture on the DHIS2 platform. A total of three implementation tests were conducted on the implementation guide at project conclusion. The implementation guide is available in the project's repository on GitHub: Implementation Guide. Below is a summary of the tests and recommendations for improvements to simplify the implementation process of the growth chart plugin in the Capture application.

- ***Implementation Test 1***

  In the first user test of the implementation, the test subject encountered several challenges. The challenges pointed to areas in the implementation guide that weakened the effectiveness and user-friendliness of the implementation. The participant was able to implement the growth chart with some guidance.

  The first challenge encountered was linked to uncertainty about how the file tree should be structured. In the Datastore Management application, a predetermined folder structure is required for the implementation. The test participant struggled with establishing the correct file tree structure, requiring our guidance. In addition, the test subject struggled to find the ”Create new” button and understand the concepts of ”Namespace” and ”key”. This indicated the necessity for more understandable instructions or visual explanations to ensure correct implementation. Although he had some difficulties, he was able to complete the steps without any other problems.

  When uploading the growth chart plugin, the participant experienced problems with the build process because the command ”yarn” was not run before the ”yarn build” command. We received feedback that this was not specified in the implementation guide.

  In addition, the test subject experienced confusion about the setup and contents of the configuration file. At project conclusion, the template for the configuration file was attached beneath a detailed explanation of its contents. We recognize that the explanation can be overwhelming and that the file is hidden when it is placed after the explanation.

- ***Implementation Test 2***

  The second implementation test revealed significant points of improvement in order for the implementation guide to be explanatory and intuitive. The implementation guide's weaknesses meant that the test subject needed guidance to succeed with the implementation.

  The first weakness in the implementation guide was that yarn was not included as a prerequisite. The participant points out that knowledge of GitHub should also be a prerequisite for the implementation because the implementation requires the implementer to clone a repository from Git.

  Furthermore, we observe several ambiguities in the implementation guide. The guide uses several technical terms, both within the medical aspect and from the DHIS2 platform, without a further explanation of the terms. In addition, a lack of explanation about structure, punctuation, file creation, and storage of the files written in the Datastore Management application made the implementation difficult for implementers without an technological background.

Another weakness revealed in the test was that the implementation guide requires you to switch between applications in DHIS2 more than necessary. The test subject expressed a desire to finish all configurations in one app before switching to another app. This approach would make the implementation process more efficient.

The test participant was also confused by the guide's structure when creating the configuration file. We observed that the actual template for the configuration file was difficult to notice because a thorough description of the file precedes it. The test participant explains that a more natural structure would have been to expose the test subject to the file first and then explain the file's content.

- ***Implementation Test 3***

  The third implementation test identified several of the weaknesses found in previous implementation tests. The test also discovered other ambiguities in the implementation guide which caused trouble for the test participant. The weaknesses in the guide meant that the test subject needed guidance to complete the implementation despite her thorough technical understanding of the DHIS2 platform.

  The weaknesses identified during the test concern the intuitiveness of the implementation guide. As in previous tests, we found that the test participant was confused by the configuration file's structure. Another weakness was the lack of visual descriptions, which led to unnecessary doubts about her actions, despite them being in accordance with the implementation guide.

  Another identified weakness is ambiguity when entering the URL for the plugin in Datastore Management. During the implementation test, two errors related to the URL occurred. Firstly, using *http://* instead of *https://* resulted in the plugin not loading. The implantation guide contained no warning against this. Secondly, an error occurred because the tester misunderstood the attributes in the URL. As a result, the URL of the instance of the plugin was not completed.

Possible Improvements

Observations during the implementation user tests, introduced in *5.1.3 User Test of Implementation* on page 49, were evaluated to find improvements. The aim was to enhance the implementation of the growth chart plugin in the Capture app by making it more efficient and intuitive. Below is a summary of the most important recommendations:

> - ***Change prerequisites***: For implementation, there are more *prerequisites* than what we have listed. It is pointed out that since the implementation requires that yarn is installed, this must be included as a prerequisite. In addition, basic knowledge of GitHub may be a requirement.
>
> - ***Use of links***: Key concepts in the implementation guide should be explained. For concepts that include DHIS2, clarification can be achieved by actively using links to the documentation for concepts regarding DHIS2.

- ***Improve the structure***: The structure of the file should be changed to increase efficiency and user-friendliness. We observed during all tests that the test subject was confused by the order of the configuration file's template and the file's explanation. In addition, the implementation has the potential to be more effective if we reduce the number of times a user has to switch between applications on the DHIS2 platform during the implementation.
- ***Visualizations***: The implementation guide should contain illustrations as examples to better explain specific actions and particular file structures required from the user. *Visualizations* such as images or figures, can function as these illustrations. Images and figures are often straightforward yet effective in this situation.

Recommendations

Table 5.5: *Requirements received from implementation user tests*

Req ID	Requirement Title	Received From
P.R.12	Expand and enhance the prerequisites in the implementation guide.	Implementation Tests
P.R.13	Add visual illustrations to the implementation guide to clarify actions and structures.	Implementation Tests
P.R.14	Restructure and optimize the order of the implementation guide.	Implementation Tests
P.R.15	Integrate links to other DHIS2 documentation within the implementation guide.	Implementation Tests

5.2 Results from Quantitative Data

We have collected quantitative data using SUS as part of the testing process. The purpose of the SUS score is to quantify the usability of our product. Calculation of the SUS score of a product was presented in *3.1.4 Evaluation - Demonstrations for Stakeholders and SUS score* on page 15. The results of the SUS are presented for both testing the UI and the implementation guide.

5.2.1 System Usability Scale (SUS) for UI

Three tests of the UI were conducted. As part of the test, each participant filled out a SUS form after interacting with the UI. The results acquired in the SUS form are presented below:

- **UI Test 1:**
  The SUS form from the first UI test is in the Appendix *A.3 UI Test - Participant 1 - SUS Form* on page 65.
  - Odd-numbered points: $4+4+4+4+4=20$
  - Even-numbered points: $4+4+1+1+0=10$
  - Total points out of 100: $30 \cdot 2.5 = 75$

SUS score: Participant 1 had a user score of 75.

- **UI Test 2:**
  The SUS form from the second UI test is in the Appendix *A.4 UI Test - Participant 2 - SUS Form* on page 66.
  - Odd-numbered points: $4+4+3+4+3=18$
  - Even-numbered points: $4+4+3+4+3=18$
  - Total points out of 100: $36 \cdot 2.5 = 90$

  SUS score: Participant 2 had a user score of 90.

- **UI Test 3:**
  The SUS form from the third UI test is in the Appendix *A.5 UI Test - Participant 3 - SUS Form* on page 67.
  - Odd-numbered points: $3+3+3+3+4=16$
  - Even-numbered points: $3+3+3+3+4=16$
  - Total points out of 100: $32 \cdot 2.5 = 80$

  SUS score: Participant 3 had a user score of 80.

Mean SUS score - UI

The mean SUS score for the user tests of the UI is calculated as follows:

$$\text{Mean} = \frac{75+90+80}{3} = 81.67$$

5.2.2 System Usability Scale for Implementation

Three tests of the product's implementation were conducted. As part of the test, each participant completed a *SUS questionnaire* after completing the implementation. The results collected through the SUS form are reproduced below:

- **Implementation Test 1:**
  The SUS form from the first implementation test is in the Appendix *A.7 Implementation Test - Participant 1 - SUS Form* on page 69.
  - Odd-numbered points: $4+3+3+4+4=18$
  - Even-numbered points: $4+3+3+3+4=17$
  - Total points out of 100: $35 \cdot 2.5 = 87.5$

  SUS score: Participant 1 had a user score of 87.5.

- **Implementation Test 2:**
  The SUS form from the second implementation test is in the Appendix *A.8 Implementation Test - Participant 2 - SUS Form* on page 70.
  - Odd-numbered points: $3+4+4+4+4=19$
  - Even-numbered points: $4+3+4+4+3=18$

- Total points out of 100: $37 \cdot 2.5 = 92.5$

SUS score: Participant 2 had a user score of 92.5.

- **Implementation Test 3:**
The SUS form from the third implementation test is in the Appendix *A.9 Implementation Test - Participant 3 - SUS Form* on page 71.
  - Odd-numbered points: $4 + 4 + 4 + 4 + 4 = 20$
  - Even-numbered points: $4 + 3 + 4 + 4 + 4 = 19$
  - Total score out of 100: $39 \cdot 2.5 = 97.5$

SUS score: Participant 3 had a user score of 97.5.

Mean SUS Score - Implementation

The mean SUS score for implementation testing of the UI is calculated as follows:

$$\text{Mean} = \frac{87.5 + 92.5 + 97.5}{3} = 92.5$$

5.2.3 System Usability Scale Evaluation

We use Figure 5.4 to evaluate the mean SUS scores for the UI and implementation tests.

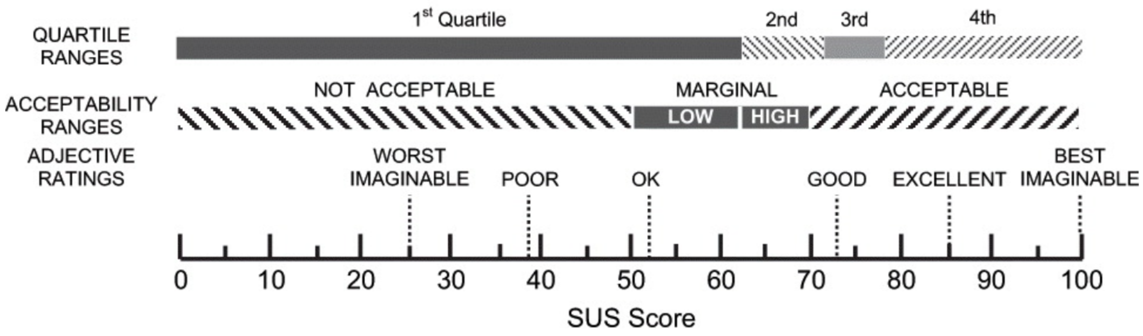


Figure 5.4: Interpretation of mean SUS score, (Bangor et al., 2008, Figure 13)

SUS - UI User Tests

The mean SUS score for the UI user tests is *81.67*, as calculated in *5.2.1 System Usability Scale (SUS) for UI* on page 51. By comparing this score to Figure 5.4, we see that the usability of the UI is in between ”Good” and ”Excellent” in the ”Adjective Rating” and the ”Acceptable” range according to the ”Acceptability Ranges”.

SUS - Implementation User Tests

The mean SUS score for the implementation user test is *92.5*, calculated in *5.2.2 System Usability Scale for Implementation* on page 52. This score indicates that the implementation process is between ”Excellent” and ”Best imaginable”, based on Figure 5.4.

5.2.4 Requirements Addressed

The completed requirements also serve as a quantitative result for this thesis. Table 5.6 displays the requirement status at the project conclusion.

Table 5.6: *Requirement completed at project conclusion*

Req Priority	Amount	Completed	Remaining	Remaining IDs	Percentage
A	10	10	0		100%
B	3	3	0		100%
C	4	1	3	R.14, R.16, R.17	25%
Total:	17	14	3	R.14, R.16, R.17	82.35%

We completed *82.35%* of requirements in the requirement specification, as Table 5.6 displays. All *A* and *B* priority requirements were completed, and just three *C* priority requirements remained. The remaining requirements from the requirement specification are displayed in Table 5.7.

Table 5.7: *Requirement specification containing remaining requirements.*

Req ID	Requirement title	Priority	Internal rank
R.14	Support for preterm babies.	C	800
R.16	Possible to change time interval alternatives in chart configuration	C	650
R.17	Functionality to toggle annotations on or off.	C	550

5.3 Evaluating Problem Statement

This section evaluates the results in light of the problem statement. First, the section evaluates the product at project conclusion in light of the research questions. Then, it responds to the research questions based on this evaluation.

5.3.1 Research Question Evaluation

We defined two research questions in *1.3 Problem Statement* on page 2. The two research questions defined were:

- **Research question 1:** How can we design a generic, user-friendly growth chart encouraging the target audience to adopt the solution over traditional pen-and-paper-based methods?
- **Research question 2:** How can a growth chart plugin be seamlessly implemented into existing instances in the Capture application on DHIS2?

Research Question 1

Evaluating the product at project conclusion in light of research question 1 requires the evaluation of relevant qualitative and quantitative results. Relevant qualitative results include stakeholder feedback and observations made during the UI user tests. The quantitative results include SUS scores from the UI user tests and an analysis of requirements addressed.

Designing and developing a generic and user-friendly growth chart requires a thorough understanding of the target audience's needs and requirements. The qualitative feedback obtained from stakeholders, including Devotta, HISP MENA, and HISP Sri Lanka, highlights the project's requirement elicitation process as fundamental. It explicitly highlights the gathering of key requirements and insights into the growth monitoring processes worldwide as well as learning about the use of DHIS2. Combining requirements gathered from various stakeholders gave us a requirement specification that provides the baseline for answering *research question 1*.

Since the requirements specification provides the baseline for answering *research question 1*, the amount of requirements fulfilled is a viable strategy for evaluating the product in light of the research question. We completed *82.35%* of requirements in the requirement specification, including all *A* and *B* priority requirements, as seen in Table 5.6, a part of *5.2.4 Requirements Addressed* on page 54. Completing all high-priority requirements, representing the baseline for answering *research question 1*, strengthens the product in addressing the problem statement.

Devotta additionally highlights the growth chart plugin's generic nature, allowing its adaptability to "any DHIS2 instance doing growth monitoring" as the feedback from Devotta stated in *5.1.1 Devotta Overall Assessment* on page 42. The generic nature of the plugin is crucial in ensuring a wide adoption of the plugin for countries utilizing DHIS2.

The qualitative observations from the UI user tests revealed some areas for improvement regarding the growth chart plugin's user-friendliness. These observations are examined in *5.1.2 User Test of User Interface (UI)* on page 46. We defined three possible requirements based on these observations, introduced in Table 5.4 on page 48.

Insights in the quantitative results obtained through SUS forms as a part of UI user tests reveal that the growth chart plugin's usability is between good and excellent with a SUS score of *81.67*, as examined in *5.2.3 SUS - UI User Tests* on page 53. Additionally, the score indicates that the plugin is within the acceptable range in the acceptability ranges defined in Figure 5.4. This outcome indicates that the current design is user-friendly and aligns with the user test participants' preferences.

The overall positive feedback from stakeholders and the mean SUS score resulting in the acceptable range indicate that the product accomplishes the elements of *research question 1*. It addresses this research question by being a generic, user-friendly plugin meeting the requirements for adopting the growth chart plugin. However, further encouraging the utilization of the growth chart plugin over traditional pen and paper use would benefit from the completion of additional requirements, introduced in *5.4 Next Steps* on page 56.

Research Question 2

Qualitative and quantitative results evaluate the plugin's possibility to be seamlessly implemented into existing DHIS2 instances. The qualitative results include stakeholder feedback and possible improvements based on observations of implementation tests, while the quantitative results are evaluated based on SUS score results of the implementation user tests.

Qualitative feedback from Devotta highlights the ability of the growth chart plugin to be implemented in any DHIS2 instance that features growth monitoring. Additionally, Devotta emphasizes the plugin's capability to be highly configurable while upholding the code quality standard of the DHIS2 community. Feedback from Devotta indicates that the growth chart plugin can seamlessly integrate into existing DHIS2 instances.

However, qualitative observations from implementation tests highlight possible improvements for the implementation guide. By evaluating these observations, we have defined possible future requirements in Table 5.5 on page 51. Addressing these possible requirements could further improve the implementation process of the growth chart plugin on DHIS2 instances with growth monitoring programs.

The quantitative SUS score result of *92.5* obtained from implementation user tests indicates an exceptionally high level of user-friendliness in the implementation process. This score falls between the "Excellent" and "Best imaginable" as discussed in *5.2.3 SUS - Implementation User Tests* on page 53. It thereby reflects that implementing the growth chart plugin is a seamless and user-friendly process.

The evaluation of *research question 2* demonstrates that the growth chart plugin is seamless to integrate into existing DHIS2 instances. Qualitative feedback from stakeholders highlights the adaptability and configurability of the plugin. Devotta emphasizes that the plugin's configurable nature allows it to be implemented to any DHIS2 instance with a growth monitoring program. While qualitative observations from implementation user tests suggest some areas for improvement, the quantitative SUS score of *92.5* indicates that the solution is user-friendly at project conclusion, allowing for seamless integration into existing DHIS2 instances. Overall, the evaluation suggests that the growth chart plugin is well established to be seamlessly integrated into existing DHIS2 instances.

5.3.2 Research Question Conclusion

The evaluation of the growth chart plugin in light of the problem statements indicates that the plugin accomplishes the elements of the research questions. This section further elaborates on the key aspects of our process and methods crucial for achieving these elements.

Research Question 1

Successfully answering *research question 1* requires a thorough requirement elicitation process, ensuring an understanding of stakeholders' needs and preferences. Closely collaborating with key stakeholders throughout the process has thereby played a crucial role in shaping the growth chart plugin's design and functionality. This thorough requirement elicitation process and collaboration with stakeholders has been essential for the accomplishment of *research question 1*.

Research Question 2

The accomplishment of *research question 2*, a growth chart that can be seamlessly integrated into existing DHIS2 instances, lies in the use of the Datastore Management application, introduced in *2.3.4 Datastore Management* on page 9. We utilized this application for storing the configuration file for the growth chart plugin, allowing us to map essential variables from the DHIS2 instance to local variables in the plugin. Our use of the Datastore Management application is further examined in *4.6.1 Configuration file* on page 33. Furthermore, *4.6.3 Model of Product at Conclusion* on page 37 illustrates how elements in the configuration guide in Datastore Management affects the growth chart plugin. Our utilization of the Datastore Management application was thereby crucial for the accomplishment of *research question 2*.

5.4 Next Steps

This section includes the next steps for the growth chart plugin. It provides a requirement specification suggestion that would further improve the plugin.

5.4.1 Possible Future Requirements

Table 5.8 include the uncompleted requirements, displayed in Table 5.7, and possible requirements gathered through stakeholder meetings and user test observations. Addressing these requirements could further improve the growth chart plugin and encourage target users to adopt the solution over traditional pen-and-paper-based methods.

Table 5.8: *Possible future requirements*

Req ID	Requirement Title	Received From
R.14	Support for preterm babies.	Uncompleted
R.16	Possible to change time interval alternatives in chart configuration	Uncompleted
R.17	Functionality to toggle annotations on or off.	Uncompleted
P.R.1	Alternative to choose a preselected indicator in the configuration file.	Devotta
P.R.2	The plugin automatically chooses time intervals according to the child's age.	Devotta
P.R.3	Possibility of different configuration files using the growth chart in multiple applications.	HISP MENA
P.R.4	Customizable PDF header.	HISP Sri Lanka
P.R.5	Fill color between lines in the graph.	HISP Sri Lanka
P.R.6	Automatic deviations detection and notification.	HISP Sri Lanka
P.R.7	Display intervention notes in the tooltip when deviations occur in the reference point.	HISP Sri Lanka
P.R.8	Include functionality for remote distribution.	HISP Sri Lanka
P.R.9	Implement a more intuitive interface for changing indicators and time intervals.	UI Tests
P.R.10	Add additional component tests to catch critical errors and ensure the plugin's reliability.	UI Tests
P.R.11	Improve accessibility to the convert to PDF functionality.	UI Tests
P.R.12	Expand and enhance the prerequisites in the implementation guide.	Implementation Tests
P.R.13	Add visual illustrations to the implementation guide to clarify actions and structures.	Implementation Tests
P.R.14	Restructure and optimize the order of the implementation guide.	Implementation Tests
P.R.15	Integrate links to other DHIS2 documentation within the implementation guide.	Implementation Tests

Chapter 6
Conclusion

In this bachelor's thesis, we have explored how to develop a product that can address the limitations of growth chart functionality on the District Health Information Software 2 (DHIS2) platform. Growth charts are crucial for detecting pediatric health problems and malnutrition early. The limitations of growth chart functionality prevent the transition from manual to digital systems for countries using DHIS2. To address these issues, the problem statement revolves around the need to develop growth charts into the Capture application on DHIS2 that are generic, user-friendly, and adaptable to various countries' health systems.

We have enriched DHIS2 with a plugin that successfully addressed the core requirements identified through a thorough requirement elicitation process. The plugin visualizes a child's growth over time, supports country-specific growth references, and references with either z-scores or percentiles. Qualitative results from stakeholders and user tests indicate that the plugin is user-friendly and adaptable to existing DHIS2 instances. The quantitative results, including calculated percent of addressed requirements and System Usability Scale (SUS) scores for both the user interface (UI) and implementation process, highlight the plugin's usability and seamless implementation process.

In our search to answer the problem statement, we evaluated the qualitative and quantitative results in light of these questions. We found that the accomplishment in answering the problem statement lies in our method with a thorough requirement elicitation process. This elicitation process was conducted with close collaboration with Health Information Systems Program (HISP) Middle East and North Africa (MENA) and HISP Sri Lanka. We found that the product developed answers the problem statement by evaluating the plugin and results from a holistic perspective.

The results indicate that the growth chart plugin has the potential to transform the growth monitoring practice for countries using DHIS2. However, the analysis of qualitative and quantitative results highlights possible improvements. Future work could focus on addressing these possible improvements, including the uncompleted requirements and possible future requirements gained by analyzing the results. Future research could explore how the growth chart plugin impact Health Management Information Systems (HMIS) and if the addition of the plugin aids in the digitalization of growth monitoring.

References

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, *24*(6), 574–594.

de Onis, M., Wijnhoven, T. M., & Onyango, A. W. (2004). Worldwide practices in child growth monitoring. *The Journal of Pediatrics*, *144*(4), 461–465. https://doi.org/https://doi.org/10.1016/j.jpeds.2003.12.034

Dehnavieh, R., Haghdoost, A., Khosravi, A., Hoseinabadi, F., Rahimi, H., Poursheikhali, A., Khajehpour, N., Khajeh, Z., Mirshekari, N., Hasani, M., et al. (2019). The district health information system (dhis2): A literature review and meta-synthesis of its strengths and operational challenges based on the experiences of 11 countries. *Health Information Management Journal*, *48*(2), 62–75.

de Onis, M., Garza, C., Onyango, A., Rolland-Cachera, M., et al. (2008). Who growth standards for infants and young children. *Archives de pediatrie: organe officiel de la Societe francaise de pediatrie*, *16*(1), 47–53.

DHIS2. (n.d.). *Dhis2 in action*. DHIS2. Retrieved May 14, 2024, from https://dhis2.org/in-action/

Heller, P. (2021). Automating workflows with github actions.

Jansen, R. H., Vane, V., & Wolff, I. G. d. (2016). *Typescript* (1st ed.). Packt Publishing, Limited.

Khadilkar, V., & Khadilkar, A. (2011). Growth charts: A diagnostic tool. *Indian journal of endocrinology and metabolism*, (Suppl3).

Lewis, J. R., & Sauro, J. (2009). The factor structure of the system usability scale. *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International*, *5619*, 94–103. https://doi.org/10.1007/978-3-642-02806-9_12

Li, P. (2018). Jira software essentials : Plan, track, and release great applications with jira software (Second edition.).

Maulana, F. I., Susanto, V., Shilo, P., Gunawan, J., Pangestu, G., & Budi Raharja, D. R. (2021). Design and development of website dr.changkitchen diet catering using sdlc waterfall model. *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, 75–79. https://doi.org/10.1145/3479645.3479652

Mwaura, W. (2021). End-to-end web testing with cypress : Explore techniques for automated frontend web testing with cypress and javascript.

Narayn, H. (2022). *Just react! learn react the react way* (First). Apress L. P.

Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas*, 117–134.

Sutton, D. (2021, September). *Information risk management* (2nd ed.). BCS, The Chartered Institute for IT.

Tomasdottir, K. F., Aniche, M., & van Deursen, A. (2020). The adoption of javascript linters in practice: A case study on eslint. *IEEE transactions on software engineering*, *46*(8), 863–891.

UIO. (n.d.). *About hisp*. University of Oslo. Retrieved April 17, 2024, from https://www.mn.uio.no/hisp/english/about/

World Health Organization. (n.d.). *Length-for-age boys*. World Health Organization. Retrieved May 12, 2024, from https://cdn.who.int/media/docs/default-source/child-growth/child-growth-standards/indicators/length-height-for-age/cht-lfa-boys-z-0-2.pdf?sfvrsn=d26cbe91_10

Yang, Z., Duan, Y., Ma, G., Yang, X., & Yin, S. (2015). Comparison of the china growth charts with the who growth standards in assessing malnutrition of children. *BMJ open*, *5*(2), e006107.

Zou, W., Zhang, W., Xia, X., Holmes, R., & Chen, Z. (2019). Branch use in practice: A large-scale empirical study of 2,923 projects on github. *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*, 306–317.

A Appendix

A.1 System Usability Scale (SUS)

System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

A.2 UI Test Document

You are a healthcare worker working at “Rukumo MCHP” clinic located in “Zuwara”, “Makonde Region”. Today, a parent named Fred brings his child, Michael Taylor, to your clinic to record Michael’s growth measurements. Michael is a 15-months old boy who needs regular check-ups to ensure he is growing healthily according to WHO standard growth charts. All the necessary measurements, including weight, height, and head circumference, have been taken.

Now, it's time to register Michael Taylor's growth data using the “Growth Monitoring” program.

Here's how you can proceed:

1. Choose the “Growth Monitoring” program.
2. Assign your clinic as “Registering unit”. Your clinic is called “Rukumo MCHP” which is in the city “Zuwara” in the “Makonde Region”.
3. Find and choose Michael Taylor as “person” to show his growth charts.
4. Record Michael Taylor's latest measurements that you measured today, 2025-05-01 (01. May 2025). Michael Taylor had the following measurements:
    - head circumference: 45.8 cm
    - weight: 9600 g
    - length: 78 cm
5. Check Michael Taylor's length when he was about 6 months old and note it down.
6. Find the length that you recorded for Michael Taylor today and note it down.
7. Find a deviation in Michael's growth data and write down the date the deviation occurred.
8. Print a PDF with the graph showing Michael's head circumference in the interval "0 to 13 weeks".

9. Note how much Michael weighed when he was on the date 2025-10-01 (01. October 2025).

10. Note how much Michael weighed when he was about 70 cm tall/long.

A.3 UI Test - Participant 1 - SUS Form

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently					✓
2. I found the system unnecessarily complex	✓				
3. I thought the system was easy to use					✓
4. I think that I would need the support of a technical person to be able to use this system	✓				
5. I found the various functions in this system were well integrated					✓
6. I thought there was too much inconsistency in this system				✓	
7. I would imagine that most people would learn to use this system very quickly					✓
8. I found the system very cumbersome to use				✓	
9. I felt very confident using the system					✓
10. I needed to learn a lot of things before I could get going with this system					✓

24.03.27

A.4 UI Test - Participant 2 - SUS Form

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently					✓
2. I found the system unnecessarily complex	✓				
3. I thought the system was easy to use					✓
4. I think that I would need the support of a technical person to be able to use this system		✓			
5. I found the various functions in this system were well integrated				✓	
6. I thought there was too much inconsistency in this system	✓				
7. I would imagine that most people would learn to use this system very quickly					✓
8. I found the system very cumbersome to use	✓				
9. I felt very confident using the system				✓	
10. I needed to learn a lot of things before I could get going with this system		✓			

2024/03/27

A.5 UI Test - Participant 3 - SUS Form

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently				✓	
2. I found the system unnecessarily complex		✓			
3. I thought the system was easy to use				✓	
4. I think that I would need the support of a technical person to be able to use this system		✓			
5. I found the various functions in this system were well integrated				✓	
6. I thought there was too much inconsistency in this system		✓			
7. I would imagine that most people would learn to use this system very quickly				✓	
8. I found the system very cumbersome to use		✓			
9. I felt very confident using the system					✓
10. I needed to learn a lot of things before I could get going with this system	✓				

Dr. L.S.T. Madduwithana
MO/IC - Agalawatte
27/03/2024

A.6 Implementation Test Document

User Test of implementation documentation of The Growth Chart Plug-in

User Story:

As a member of the HISP Tanzania organization, you've found documentation for a growth-chart plug-in in DHIS2. Now, your task is to implement it on your platform. You need a solution that seamlessly integrates with DHIS2, enabling efficient visualization and analysis of growth data to support healthcare interventions for children in Tanzania. Through the user test, you aim to evaluate compatibility and performance, ensuring a successful implementation that aligns with your organization's technical specifications and mission of improving child health outcomes.

Steps:

1. Follow the documentation of the implementation guide until “Growth References”.
2. Verify that the Growth Chart is working properly by using the program called “Growth Monitoring”. Pick **Registration Unit** called “Jamala MCHP” which can be found under:
   - HMIS
     - Gisamba Region
       - Nyala
         - Jamala MCHP

   Then pick with “Michael Taylor” as **Person**.

A.7 Implementation Test - Participant 1 - SUS Form

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently					✓
2. I found the system unnecessarily complex	✓				
3. I thought the system was easy to use				✓	
4. I think that I would need the support of a technical person to be able to use this system		✓			
5. I found the various functions in this system were well integrated				✓	
6. I thought there was too much inconsistency in this system		✓			
7. I would imagine that most people would learn to use this system very quickly					✓
8. I found the system very cumbersome to use		✓			
9. I felt very confident using the system					✓
10. I needed to learn a lot of things before I could get going with this system	✓				

03/04/2024.

A.8 Implementation Test - Participant 2 - SUS Form

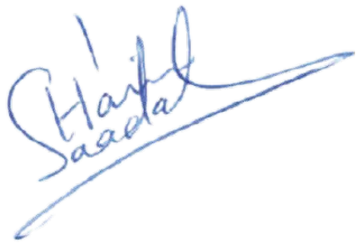
System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently				✓	
2. I found the system unnecessarily complex	✓				
3. I thought the system was easy to use					✓
4. I think that I would need the support of a technical person to be able to use this system		✓			
5. I found the various functions in this system were well integrated					✓
6. I thought there was too much inconsistency in this system	✓				
7. I would imagine that most people would learn to use this system very quickly					✓
8. I found the system very cumbersome to use	✓				
9. I felt very confident using the system					✓
10. I needed to learn a lot of things before I could get going with this system		✓			

A.9 Implementation Test - Participant 3 - SUS Form

System Usability Scale

	1 (Strongly disagree)	2	3	4	5 (Strongly agree)
1. I think that I would like to use this system frequently					✓
2. I found the system unnecessarily complex	✓				
3. I thought the system was easy to use					✓
4. I think that I would need the support of a technical person to be able to use this system		✓			
5. I found the various functions in this system were well integrated					✓
6. I thought there was too much inconsistency in this system	✓				
7. I would imagine that most people would learn to use this system very quickly					✓
8. I found the system very cumbersome to use	✓				
9. I felt very confident using the system					✓
10. I needed to learn a lot of things before I could get going with this system	✓				



A.10 Project Plan

A.10.1 Necessary resources

There are few necessary resources we need access to for this thesis, as we are working with open source software. We however rely on community support, both from our task provider but also the HISP groups we are cooperating with. Additionally, we will rely on online documentation, regarding both growth charts and developing a plugin on the DHIS2 platform.

A.10.2 Central activities and distribute responsibility

We have split this bachelor thesis into 6 central activities, each with their respected start and estimated end date as visualized in the figure below. Each of these activities has been marked under the milestone category and been assigned to all members on the team. We will go more in depth on each milestone, showing each of their respected tasks, under Milestones. All tasks in this thesis are assigned to all members of the team, as we are a small team and cooperate on everything.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Thesis kick-off	Milestone	All members	**100 %**	10.01.2024	21	31.01.2024
Requirement elicitation	Milestone	All members	**89 %**	01.02.2024	15	16.02.2024
Design and development	Milestone	All members	**50 %**	12.02.2024	32	15.03.2024
Implementation and testing	Milestone	All members	**0 %**	11.03.2024	28	08.04.2024
Mandatory assignments	Milestone	All members	**75 %**	23.01.2024	37	29.02.2024
Bachelor thesis	Milestone	All members	**2 %**	10.01.2024	132	21.05.2024

Figure A.1: Milestones for the project

A.10.3 Milestones

Here are the different tasks included in each of the Milestones. Category is empty for tasks that we haven't started working on yet. The figures in this sections are taken from our Gantt chart that can be found as an attachment.

Thesis kick-off

The thesis kick-off marks the start of our bachelor. We started with the kick off at NTNU on the 10. of January and planned to be finished by the end of the month.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Thesis kick-off	Milestone		100 %	10.01.2024	21	31.01.2024
Standard agreement	Goal	All members	100 %	10.01.2024	21	31.01.2024
Research and analysis	On Track	All members	100 %	10.01.2024	21	31.01.2024
Define project goals	High Risk	All members	100 %	23.01.2024	7	30.01.2024
10 Thousand feet Plan	On Track	All members	100 %	15.01.2024	8	23.01.2024
Create project plan	Goal	All members	100 %	23.01.2024	8	31.01.2024

Figure A.2: Thesis kick-off section of the projects Gantt chart

Requirement elicitation

The requirement elicitation included interviews with HISP groups and creating a requirement specification to document the requirements of our solution. We planned that we would start on this part in the start of February and be done on the 16. of February.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Requirement elicitation	Milestone		89 %	01.02.2024	15	16.02.2024
Interview planning	On Track	All members	100 %	25.01.2024	7	01.02.2024
Interview with HISP groups	High Risk	All members	66 %	01.02.2024	14	15.02.2024
Requirement specification	On Track	All members	100 %	08.02.2024	8	16.02.2024

Figure A.3: Requirement elicitation section of the projects Gantt chart

Design and development

The design and development milestone of this bachelor mainly includes creating the growth chart as a plugin in the capture app on the DHIS2 platform. This also included designing and creating a configuration guide for the solution.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Design and development	Milestone		50 %	12.02.2024	32	15.03.2024
Designing a Chart	On Track	All members	100 %	14.02.2024	7	21.02.2024
Planning the developement	On Track	All members	100 %	12.02.2024	7	19.02.2024
Developing the Charts (generic chart)	On Track	All members	50 %	14.02.2024	28	13.03.2024
Create configuration file		All members	0 %	04.03.2024	11	15.03.2024
Create functionality for preterm babies		All members	0 %	07.03.2024	7	14.03.2024

Figure A.4: Design and development section of the projects Gantt chart

Implementation and testing

The implementation and testing milestone include everything regarding the implementation and testing our solution, ensuring quality and that it meets the required specifications. This will also grant us valuable feedback that we can review and reflect over to create a better end product.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Implementation and testing	Milestone		0 %	11.03.2024	28	08.04.2024
Plan the implementation face		All members	0 %	11.03.2024	14	25.03.2024
Create a implementation guide		All members	0 %	11.03.2024	14	25.03.2024
Testing implementation guide		All members	0 %	18.03.2024	7	25.03.2024
Implement with HISP groups		All members	0 %	25.03.2024	14	08.04.2024

Figure A.5: Implementation and testing section of the projects Gantt chart

Mandatory assignments

The mandatory assignments includes project plans part 1 and 2 which both has a delivery date on the 29. of February.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Mandatory assignments	Milestone		75 %	23.01.2024	37	29.02.2024
Projectplan part 1	Goal	All members	70 %	01.02.2024	28	29.02.2024
Projectplan part 2	Goal	All members	80 %	23.01.2024	37	29.02.2024

Figure A.6: Mandatory assignments section of the projects Gantt chart

Bachelor thesis

The final milestone we have added to our Gantt plan is the bachelor thesis itself. This includes tasks linked to the thesis as well as each section in the report structure.

Milestone description	Category	Assigned to	Progress	Start	Days	End
Bachelor thesis	Milestone		2 %	10.01.2024	132	21.05.2024
Introduction		All members	0 %	01.03.2024	14	15.03.2024
Planning and Methodology		All members	0 %	08.03.2024	7	15.03.2024
Technology		All members	20 %	15.03.2024	7	22.03.2024
Development Process		All members	0 %	01.03.2024	30	31.03.2024
Role of Requirements Specification		All members	0 %	21.03.2024	14	04.04.2024
Results		All members	0 %	01.04.2024	14	15.04.2024
Evaluation		All members	0 %	01.04.2024	25	26.04.2024
Conclusion		All members	0 %	14.04.2024	30	14.05.2024
Closing Section		All members	0 %	14.04.2024	30	14.05.2024
First bachelor thesis draft		All members	0 %	01.03.2024	60	30.04.2024

Figure A.7: Bachelor thesis section of the projects Gantt chart

NTNU
Norwegian University of
Science and Technology

devotta