Implementing a sensor-based cold-chain monitoring application in resource constrained settings

A case from Mozambique

Filip Christoffer Larsen

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

Implementing a sensor-based cold-chain monitoring application in resource constrained settings

A case from Mozambique

Filip Christoffer Larsen

Implementing a sensor-based cold-chain monitoring application in resource constrained settings

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Vaccines are among the most significant advances in global health and development. Immunization is widely acknowledged as one of the most successful and cost-effective public health interventions, saving two to three million lives every year [24]. However, immunization programs still struggle to close the immunization gap.

Related research on Cold Chain Monitoring (CCM) found a need to raise knowledge, as countries in resource-constrained settings often are unaware of their current CCM situation. In addition, has the recent Covid-19 pandemic addressed the importance of conducting correct CCM on vaccines.

This thesis and its study are the initial examinations of how DHIS2 can support real-time temperature CCM. It examines the process of developing, implementing, and piloting a custom Android application, called the CCM application in the Zambezia province of Mozambique.

Today, the current CCM of vaccines in Mozambique is conducted at the health facility level. The CCM routine consists of manually reading Cold Chain Equipment (CCE) temperatures twice a day captured in a paper-based form. Once a year, the form gets reported to respectively immunization programs. DHIS2 is the leading Health Management Information System (HMIS) in Mozambique. Through good support from the local Health Information Systems Program (HISP) group and other programs, DHIS2 is well established supporting program for Malaria and HIV, in addition to educational and logistics programs. As HMISs are dependent on quality data to support decision-making, supporting a program for CCM could strengthen the potential for aggregates combining temperature data with related health data. In addition, unlocking DHIS2s potential to support logistics as a Logistic Management Information System (LMIS).

The CCM application utilizes sensor-based technology to report temperature data to the DHIS2 platform ecosystem to support the current CCM and routines at health facilities in Mozambique. The application is developed through Design science research (DSR) iterations based on evaluation and feedback through close collaboration with the HISP Oslo LMIS team and piloting for the health facility level end-users.

Feedback through demonstrations, implementation, and piloting of the application indicates that the CCM application is a helpful tool to include in CCM work processes. Though the application showed promise, the pilot proved that the application needs to be more generic before being able to scale. In addition, there is a need to improve both current work processes and local knowledge to improve CCM in resource-constrained settings.

Acknowledgements

Firstly I want to thank HISP Saudigitus for facilitating my piloting of the app, arranging meetings, accessing health facilities, taking care of me, and giving me a fantastic educational experience. Mainly thanks to Alfredo, David, Emilio, Fernando, Inacio, and Zefernio. I also want to thank Marcos Campos from the DHIS2 Android team for always helping troubleshoot when I faced application errors.

I thank my supervisor Johan Sæbø for arranging my trip to Mozambique. Furthermore, for excellent guidance and discussion of the content of this thesis. Thanks to the HISP Oslo LMIS team and George McGuire for believing in me, helping with DHIS2 configuration, and providing lots of domain knowledge regarding CCM. A big thanks to Breno Horsth for including me in the LMIS team, always believing in me, and being a great travel companion the last week in Mozambique. Lastly, I also want to thank my friends and family for supporting and motivating me throughout the time of this thesis.

Contents

List of Tables

List of Figures

Chapter One

1 Introduction

The cost of vaccines in the developing world has increased from less than US$1 per child in 2001 to about $28 in 2014[56]. This growth is due to an increased number of vaccines in the immunization programs. In 1974 the Expanded Programme on Immunization (EPI) was launched, their goal was to deliver routine vaccinations to children. By 1991 the EPI had increased the vaccine coverage of children younger than one year from an initial 5% to 80%, saving millions of lives[56].

Introducing new vaccines to immunization programs increases the vaccine unit cost and the program's complexity. Therefore, poor vaccine handling can have significant financial consequences. One impact of introducing new vaccines is increased vaccine storage capacity, transport, and temperature tracking. Cold chain and logistics management has always been part of the EPI blueprint, but as programs grow, there is a need to strengthen critical components like; cold chain, logistics management, and a well-trained health workforce, to mention a few [56].

The global open source project District Health Information System 2 (DHIS2) has existed for two decades and is now the world's most extensive Health information system and the primary Health information system in around 70 countries. The platform provides data warehousing, visualization, and analysis features of real-time data.[26]. Some countries use DHIS2 to manually record cold chain temperatures as part of their immunization programs. In cases where DHIS2 is used for CCM today, it is still far from fulfilling the goal of the global vaccine alliance GAVI's goal of real-time CCM. This thesis explores how DHIS2 can be applied to improve this by piloting a custom CCM application in an actual use case in Zambezia province in Mozambique.

This thesis will implement a sensor-based cold-chain monitoring application developed with Android technology integrated with DHIS2 tracker capture. The application runs on Android using BlueMaestro Bluetooth temperature sensors to capture temperature data. The data gets stored in DHIS2, where one can utilize the platform functionality. The application's use case is to provide real-time temperature data from CCE to monitor vaccines, decrease waste, and ensure the desired effect of the vaccine. In order to achieve these requirements, one key application feature is providing instant alerts if a temperature surpasses a threshold so that one can immediately take action to remedy the situation of the vaccines. Furthermore, the application gain analytics on temperature data enabling other decisions such as the condition and performance of CCE at a health facility or if there is a need for improvement in Cold Chain work processes. These features align with the GAVI Target Software Standards for Vaccine Supply Chain and answer real-world needs from Immunization programs. One of the thesis objectives is to confirm the application's usability and value to incorporate it as a native feature in future software releases of the DHIS2 platform core. [9]

1.1 Research Question

This thesis examines the different challenges and opportunities and how CCM can be supported by the DHIS2 platform ecosystem since there is no support for CCM in the DHIS2 core functionality. And no use of sensor-based technology in the platform ecosystem as of today.

The research question for this thesis is:

> ”How can real-time cold chain temperature monitoring be supported in DHIS2.”

DSR has been applied to design, develop and pilot an artifact to address the research question. The artifact is an Android application aiming to support real-time CCM in the DHIS2 platform. The application uses sensor-based technology to support temperature CCM, which acts as a gateway providing temperature data to decision-makers through DHIS2 tracker capture.

This thesis will examine the DSR of developing and designing the application's functionality. Discuss the design choices made during the DSR iterations with an agile development approach. Lastly, address the research question by studying the need for CCM support in DHIS2 through a pilot, implementing the application in resource-constrained settings at health facilities in Zambezia in Mozambique.

1.2 Thesis Structure

This thesis is structured as follows:

Chapter 2 – Background
This chapter introduces the background of this thesis, providing some context by presenting the DHIS2 platform ecosystem and HISP.
Chapter 3 – Related literature
Through related literature, this chapter presents the concept of understanding information systems and health information systems and their common challenges in developing countries. Furthermore, the chapter presents related literature regarding cold chain monitoring, logistic management information systems, and digital platforms to provide a context to the application developed in this thesis.
Chapter 4 – Methodology
This chapter presents the methodology used in this thesis – design science research. Furthermore, the chapter presents the data collection methods used before and during the piloting and agile software development methods used to develop the DSR artifact - the CCM application.
Chapter 5 – App development.
This chapter presents the technical environment and architecture of the CCM application. Further, it describes the application's dependencies like the DHIS2 Android SDK, BlueMaestro sensors, and the tools used during the development.
Chapter 6 – Development iterations.
This chapter presents a timeline of the development process. It then presents each development iteration of the DSR and agile software development. Each iteration describes critical features and evaluation feedback.
Chapter 7 – Piloting in Mozambique.
This chapter presents the pilot period in Mozambique chronologically from start to finish. It presents the work conducted, challenges, and opportunities during the piloting.
Chapter 8 – Main application functionality
This chapter presents the current and final product of the CCM app. Furthermore, it describes the application's different activities, functionality, and design choice.
Chapter 9 – Findings.
This chapter presents technical and socio-technical findings made during the application's development, implementation, and piloting.
Chapter 10 – Discussion
This chapter reviews the CCM application, Pilot, and Methods used in this thesis. Furthermore, each topic is discussed, and their challenges and opportunities are presented - based on findings and related literature.
Chapter 11 – Conclusion and future work
This chapter provides a summary and conclusion of the thesis, utilizing the findings and discussion to answer the research question. Lastly, presenting some thoughts and suggestions for future work on the CCM application.

Chapter Two

2 Background

This chapter's primary purpose is to present the background of this thesis by providing a context to the pilot, application implementation, and an understanding of the use case within its domain of logistics and health. In addition, this chapter introduces some of the organizations and stakeholders involved in this thesis.

2.1 DHIS2 and HISP

This thesis takes place in the context of the Health Information Systems Program (HISP), based at the University of Oslo. "In 1994, HISP was initiated in post-apartheid South Africa as a part of the decentralization of the health system to give more local control at the district level. It has become a global movement, with HISP groups in 17 countries in Asia, Africa, and the Americas. HISP is a sustainable and scalable research project enabling and supporting health information systems implementation of their software District Health Information System Two (DHIS2) in more than 100 countries [15]. The collaboration of different HISP projects has developed the DHIS2 software - a generic open-source software system for collecting, validating, analyzing, and presenting health information management activities. The program's primary goal is to "enable and support countries to strengthen their health systems and their capacity to govern their Health Information Systems in a sustainable way to improve the management and delivery of health services [1].

DHIS2 is the world's largest HMIS platform, deployed in more than 100 countries covering approximately 2.4 billion people [1]. DHIS2 as an HMIS provides data warehousing, visualization features, and the possibility for data users and policymakers to generate analysis from live data in real-time. DHIS2 is primarily designed to support integrated data warehouses for all relevant health programs in a country. Sæbø et al. discuss that the data warehouse approach, when applied successfully, can facilitate practical information integration across various organizational structures if rooted at the institutional and organizational political levels. Moreover, act as an essential mediator in negotiations between the different organizational actors [52, p. 58].

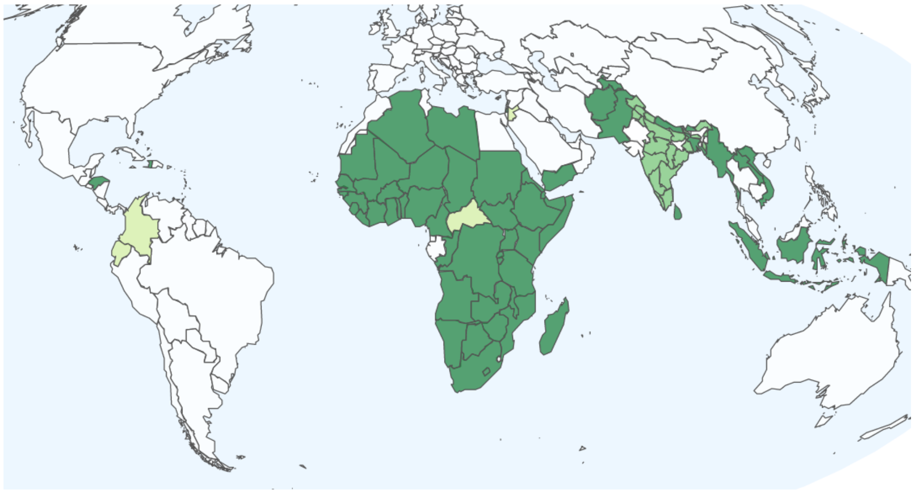

Figure 1: DHIS2 map of implementation

DHIS2 provides various data management, analysis, and visualization functions through built-in applications such as maps, dashboards, and pivot tables. This network of integrated HISs typically collects data entries via mobile applications and web-based solutions. Data are then accessible to relevant decision-makers at national levels, health clinics, and international donors to facilitate evidence-based decision-making.

The development of DHIS2 is a collaboration between HISP groups and the DHIS2 ecosystem. Furthermore, HISP UiO coordinates the development of DHIS2 and is now a professionalized software development organization [3, p.75]. HISP and DHIS2 activities have been funded through various arrangements by stakeholders like; PEPFAR, the Global Fund, UNICEF, GAVI, CDC, USAID, WHO, and Bill and Melinda Gates Foundation. DHIS2 depends on continuous support from core funders to balance the tension between serving those who can pay for functionality and those who cannot [3, p. 76]. While the primary goal of HISP is to support the health system, DHIS2s being a largely generic platform, can meet needs within other domains such as; education, logistics, and e-government. This generic approach brings up a tension between developing generic software for a global need and providing functionality to answer the needs of a specific user or user group. Furthermore, the platform's development must prioritize whether to focus on developing features to maintain quality and a stable core or innovation to meet the different needs that arise.

> ”To successfully build platform ecosystems, the focus of the platform owner must shift from developing applications to providing resources that support third-party developers in their development work” [16, p. 174].

To cope with the tensions between stability and innovation, DHIS2 is free to use and open-source, with its code available to be used, modified, and redistributed freely. This approach allows the HISP groups and other interests to innovate within the platform ecosystem. At the same time, the DHIS2 development team and HISP Oslo can focus on the quality and development of a stable core. In addition to providing platform boundary resources (BR) that enable third-party development through tools like; APIs and SDKs. Providing such BR, the platform owners facilitate cultivation by developing custom apps to meet particular user needs. In addition to standards and regulations through documentation.

Furthermore, by keeping custom applications within the ecosystem, the platform core can adopt the functionality from the custom application if there is an increased need. Such effort helps prevent useful functionality developed as custom apps by local developers from being unknown beyond the local implementation. In order to facilitate stability and growth, DHIS2 interoperates with other relevant applications such as OpenLMIS, iHRIS, OpenMRS, and the World Health Organization (WHO)[3, p. 75]. Lastly, an innovation platform architecture needs support from its surrounding ecosystem to succeed. DHIS2 facilitates such support by providing additional BR like a community of practice, Jira backlog for developers, GitHub for sharing source code, and academy training to facilitate collaboration and innovation [51, p. 28] [3, p. 76].

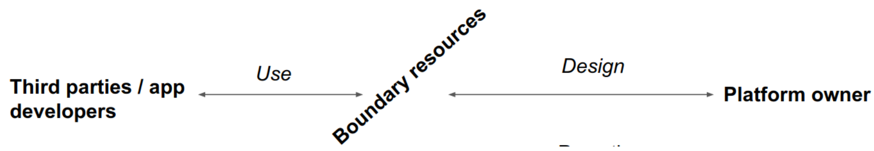

Figure 2: Innovation using platform BRs [27, p. 24]

Chapter Three

3 Related literature

As this thesis examines information systems (IS) in the health sector, it is essential to first look at how we understand IS and specific challenges and trends for health information systems (HIS). In addition, we will review related literature to uncover the need and potential for CCM. Lastly, working within the DHIS2 platform ecosystem, I will present digital platform theory and how it can support innovation and reduce complexity.

3.1 Understanding Information systems

Allan S. Lee defines information systems (IS) as

> ”... not the information technology alone, but the system that emerges from the mutual transformations interactions between the information technology and the organization.” [31, page.11].

In the context of this thesis, I work with information systems from a socio-technical perspective. The socio part represents humans, routines and practices, rules, culture, norms, politics, and motives. The technical part represents physical structures and artifacts, hardware, software, and paper-based tools [50, p.127]. Only looking at the technical aspect will limit one's ability to understand how the system works and its role in work processes, which can affect the design of the IS. A useful concept to understand the socio-technical nature of IS is information infrastructure (II) [21], which has been defined as

> ...as a shared, open (and unbounded), heterogeneous, and evolving socio-technical system of Information Technology (IT) capabilities [21, p. 4].

Central to an II understanding is the notion of the installed base (IB), which represents everything that is present and affects the II, both existing IT systems, organizational routines, practices, and tools. Due to the nature of IIs, as outlined in the definition above, they are often complex. Hanseth and Lyytinen define complexity:

> ”...as the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions.” [21].

The concept of the II helps to understand IS and their relationship, as it points out the importance of the IB and that evolving it could be a source of complexity. Nevertheless, there are various ways to reduce complexity by following design principles and rules.

3.2 Health Information systems

There are many different IS; this thesis has primarily focused on one specific kind of IS; Health Information Systems (HIS). HIS is a collective term

for IS related to health.
HIS primary role is to collect and report health data so that decision-makers can act on behalf of the data. Typically data collected is quantitative so that one can make evidence-based decisions. Examples of usage of health information would be the number of vaccinated populations over time compared with health outcomes like; disease outbreaks and mortality. Such health data can contribute to decisions like encouraging people to vaccinate or a need for quality assurance like CCM in the different parts of a vaccine supply chain.
Just as there are different types of information needs within health, such as human resources, patients, diseases, and vaccines, different types of sub – information systems of HIS are built to manage these information needs. Because of these different systems, HIS quickly grows in complexity. An effect is increased fragmentation as distinct HISs get tailored to meet a concrete demand. Fragmentation of different HISs can make it challenging to share and aggregate data for decision-makers.
An IS that manages health data from integrated HIS is called a Health Information Management System (HMIS), which is an approach to counter HIS fragmentation. HMIS prevents fragmentation by collecting the data from the relevant HIS via multiple integrations into a management system so one can access different health data from the same system. Integration can be achieved either by having the HMIS serve the different information needs directly so that it is the source of the data generated or by interoperability that allows sharing of data between different data sources. This way, decision-makers gain access to needed data for making evidence-based decisions. One such example is logistics management information systems (LMIS). The Systems for Improved Access to Pharmaceuticals and Services (SIAPS) Program defines logistical management information systems as;

> "an information system that is used to collect, organize, and present logistics data gathered from all levels of the health system." [65, p. 3].

A well-functioning LMIS is essential for improving global health; better integration between HMIS and LMIS could contribute to that improvement. As mentioned in The Global Fund's In-country Supply Chain Processes: "Reliable health and supply systems are identified by the World Health Organization as one of the four key elements for improving access to medicines." [60, p. 3].

3.3 Common Challenges with HIS in developing countries

There are various challenges related to health information systems; this section describes some typical challenges with HIS in developing countries. As mentioned in the sections above, HIS should constitute a network of interconnected systems. Implementing HISs correctly and keeping them interconnected is a complex task, especially in developing countries where one faces infrastructural challenges and other barriers. An example could be the lack of adequate infrastructure, like reliable electricity and low-quality and expensive Internet access [36, p.4]. In addition, one must consider that developing countries face more basic needs, like food, equality in healthcare, and education.

Many developing countries rely on donors to expand their HISs and increase national health. After a study in Malawi, Sanner and Sæbø pointed out some challenges regarding external funding. At a health facility, only one of four solar panels installed by different donors functioned due to facility staff not knowing whom to contact for maintenance [53, p. 41]. Being dependent on vendors when implementing HISs can lead to vendors prioritizing their platforms. They might need quantitative data such as dollars spent and several people trained to keep funding. This can challenge the success of the implementation if the focus is not on staff training and maintenance. In addition, can be scenarios where the implementation is affected by participants trying to maintain the status quo.

Furthermore, HISs implemented at a national level can scale to a position where they are in-replaceable. The platform gets strengthened because integration with other systems becomes complex. Nevertheless, such vendor lock-in can have some upsides if provided with sufficient funding to support long-term training of human resources and practice as long as there is a need for additional knowledge [28, p. 18]. In addition, could a vendor lock-in lead to competition, where new specializing HIS are implemented without thinking about long-term investment.

Another challenge when working with HISs is that decision-makers need routine high-quality coverage data. Many developing countries face the challenge of poor health information data quality. Decision-makers need correct data to map the progress towards their health goals, not only complete data [7]. Shamba et al. provide key findings from their EN-BIRTH study regarding such challenges. After visiting five public hospitals, they reported that data collecting was time-consuming and delayed because of low staff numbers. Furthermore, they discovered that data elements were not always standardized and were affected by staff having variable training, limited supervision, and availability of logistical resources [7]. In addition, they experienced cases where complete data was higher valued than correct data by the collectors. Such an approach could be due to health workers not wanting to report data that is not desirable.

Braa et al. also point out that lack of shared standards can affect the data quality resulting in reporting duplicated data in different formats, coverage data not getting reported, or collecting in-relevant data [10, p. 382]. These challenges can be rooted in the complexity of coping with the fragmentation of HIS, as data collectors report using different HIS with different standards and formats [10, p. 399].

In addition to shared standards for improving data quality and only capturing necessary data elements, feedback from decision-makers at an HMIS level to the data collectors is critical to improving [7]. Feedback from the top level could provide a return on investment to the data collectors who register and report data and their other duties. Therefore, knowledge of these challenges is central to the success of HIS implementation, especially in developing countries where these challenges grow with the gap between Urban and rural areas[36].

3.4 Logistics management information system and Cold Chain Monitoring

A Logistics management information system (LMIS) is an information system of records and reports – whether paper-based or electronic – used to aggregate, analyze, validate, and display data from the different levels of the logistics system [12, p. 3]. The purpose of collecting this data is to make logistics decisions and manage the supply chain. A well-functioning LMIS provides decision-makers with the supply chain status like data regarding stock on hand, losses, adjustments, consumption, demand, issues, shipment status, and information about the cost [12].

This thesis examines a IS called Cold Chain Monitoring (CCM), a phenomenon within the scope of logistics and HIS. One key element in research regarding optimizing supply chains and LMIS is the need for better CCM, which is defined as:

> "A temperature-constrained supply chain. An ideal cold chain is a continuous series of refrigerated production, stocking, and supplying activities with the help of associated equipment and logistics, which maintains the desired low-temperature limits. It helps preserve and increase the shelf life of commodities such as frozen foods, pharmaceutical drugs, chemicals, seafood, chemicals, and fresh agricultural products." [20, p. 141].

In the food industry, CCM is widely used, but we focus on pharmaceutical products and, more specifically, vaccines in this thesis. Vaccines are among the most significant advances in global health and development. Immunization is widely acknowledged. However, immunization programs still struggle to close the immunization gap [24]. Temperature damage to vaccines is a known problem in many countries and is often due to failures in Cold Chain Equipment (CCE). One challenge is the wide usage of "domestic fridges" as CCE. These refrigerators are cheap, available, and easily purchased. However, they are unsafe for vaccine storage, as they do not reliably maintain optimal temperature ranges [66]. In addition, poor temperature control is also due to old CCE. Equipment older than ten years is not recommended, as they are more likely to break down. Despite this, some countries meet 45 percent of their CCE needs with such fridges [25, p. 2220]. Improving cold chain systems can reduce spoiled vaccines due to poor temperature control, nonfunctional cold chain equipment, poor stock delivery, and vaccine stock-outs. In addition, improvements would support national immunization programs to reach their goal of equity in immunization.

CCM of vaccines is about protecting patient safety. Because one cannot see or smell if a vaccine has been frozen and has lost its potency (or might even be harmful), CCM could reduce the challenge of proving the vaccine's quality and potency [4]. It can be applied to multiple use cases, from monitoring temperatures throughout the whole supply chain of distributing vaccines to this thesis use case; CCM within the last link of the supply chain - storing vaccines at a health facility. This thesis focuses on CCM as an ongoing process regarding the storage of vaccines. By monitoring vaccine temperatures, one can trust the quality of the product and reduce the usage of spoiled products. CCM can also prevent spoiled products by providing alerts of intermittent Fridge failure.

The importance of a well-structured, functioning CCM gets highlighted by Privett and Gonsalves in their article "The top ten global health supply chain issues." The article addresses cold chain issues like insufficient temperature readings in storage units or cold rooms. Their study found that temperature readings "are often measured, but not always well monitored and frequently lack temperature history charts" [49]. In addition, they address the issue of periodic monitoring in subsequent supply chain stages. They report work processes "where temperatures are recorded two times per day. Such periodic monitoring is insufficient and makes cold chain procedures difficult to control." [49]. For CCM to improve Global Health, it needs to fit the existing II with a socio-technical perspective. It includes demands on infrastructure like power supply, internet connectivity, and sufficient integration with the other IS. Existing CCM solutions are often expensive and can include significant changes and large installations to the installed base.

In another study of LMIS architectures for public health supply chain management in developing countries, Sæbø et al. [6] describes the current landscape as "patchworks." They discovered fragmented and heterogeneous systems covering LMIS functionality through a field study in two countries. They address the need for relevant information during all steps of the supply chain and the importance of data standards and integration between systems. They state that the fragmented and loosely coupled architecture was expected but also, to a certain degree, beneficial. Efficient integration between the systems would benefit the various levels, and the flexibility of the LMIS [6].

Often countries are unaware of the risks of an insufficient cold chain until studies prove the degree of exposure occurring during the transport and storage of vaccines. Ashok et al. found that up to two-thirds of vaccines were damaged by freeze exposure in the cold chain between vaccination storage and administration sites across ten States in India. In addition, a study in eastern Nigeria found that 42% of yellow fever vaccines available in the private sector had been compromised by freeze damage [25]. They conclude their study with evidence showing that many countries struggle to maintain required storage temperatures. They found that malfunctions were expected at the facility level and that CCE was not functioning correctly. In addition, they found a mean of 29% of the facilities they visited to have CCE exposing vaccines to freeze, and that 9–20% of CCE at the facilities maintain sub-zero conditions for longer than 24 hours and cases where the temperature was above the threshold of 2 - 8 degrees for more than five days. Lastly, this shows that facility levels are struggling to detect and resolve temperature excursions [25].

3.5 Digital platform theory

In the previous section on understanding IS, the importance of taking the socio-technical perspective and the IB into consideration were discussed as this could reduce complexity. In addition, introducing digital platform architecture is relevant as it is a way of reducing complexity. Furthermore, since this thesis examine how we can extend the open-source platform DHIS2 to support a HIS like CCM, it is helpful to understand how digital platforms can support such innovation to understand why some IS fail and some succeed during the development.

According to Bonina et al., "Digital platforms share three basic characteristics: they are technologically mediated, enable interaction between user groups and allow those user groups to carry out defined tasks" [14].

Digital platforms are divided into two main platform categories; innovation and transaction platforms. Transaction platforms are characterized by facilitating transactions between platform users. An example of such a transaction platform is Uber, where the platform connects the driver with the passengers.

On the other hand, innovation platforms act as foundations upon which others can build complimentary products, services, or technologies [14, p. 54]. In this way, the technical architecture of an innovation platform contains modules, or building blocks, as accessible, innovative capabilities [14]. Introducing a new IS will always have a socio-technical impact on the installed base, leading to fragmentation and increased complexity. Due to its layered architecture, platforms minimize this fragmentation and reduce the complexity of introducing new IS to the platform.

A platform consists of a generic stable "core," providing functionality and interfaces through modules allowing third parties to develop complementary components. These components, often called third-party apps [61], are loosely coupled to the platform core, making it integrated but not a part of the core. This makes the platform resilient and maintainable by keeping dependencies to a minimum so that one third-party defect app does not affect others.

Android is an example of an innovation platform. The Android core provides functionality to its end-user; it allows for downloading standalone applications made by others while not affecting the core functionality. Through the principle of loose coupling to the core and boundary resources (BR) like Android studio, Google Play Store, and knowledge BR as API documentation [16]. The Android platform supports innovation by allowing experimentation and developers to build on top of the core's stable set of resources.

Figure from [14] Overview of innovation platform functional architecture

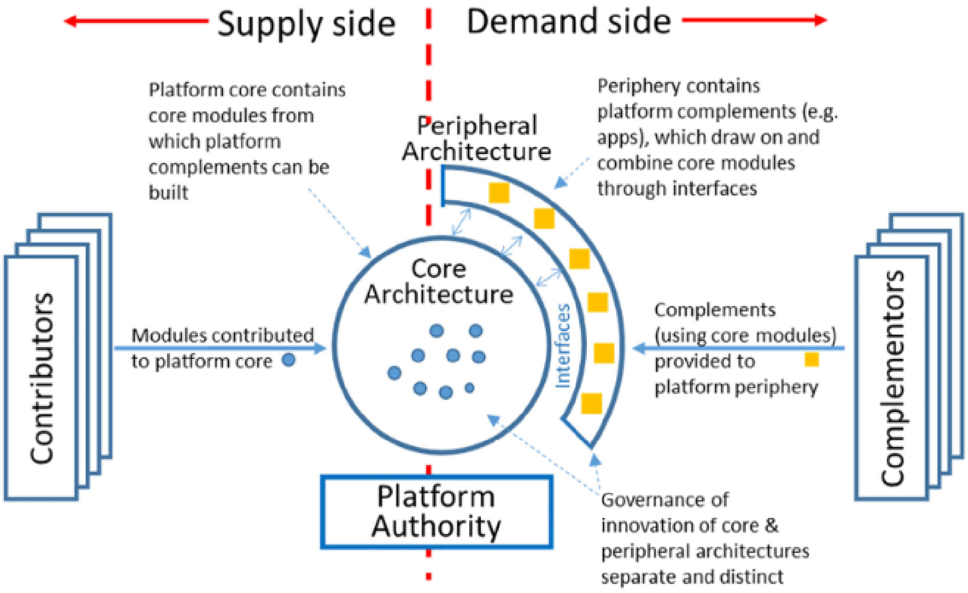

Figure 3: Platform Structure

Although digital platforms reduce complexity and fragmentation, there are some challenges with platforms. Developing third-party applications for Innovation platforms requires a set of skills to contribute. There are also some network effects involved, meaning that every additional user of a platform or app makes it more valuable to every other existing user. One challenge is that developers do not want to contribute to platforms with a small user group, and end-users do not want to use a platform with a good set of applications [61, p. 33]. Another challenge that will always be present in innovation platforms is how it will maintain applications developed for the platform. As chapter two mentions, the platform core development team should focus on developing a stable core and provide BR to facilitate innovation from third parties [16, p.177]. Moreover, sometimes, consuming third-party applications can bring different dependencies to the platform, challenging who is responsible for maintenance. In addition, ”the challenge of managing the delicate balance between app developers' autonomy to innovate freely and ensuring that apps seamlessly interoperate with the platform” is referred to as the ”Seesaw problem” [61, p. 42].

Platforms are also challenged by their ecosystem, the socio-technical setting, different stakeholders, and politics. Bonina et al. present the ”dark side” of platforms concerning three different areas of interest; ”These concern the surveillance of citizens with a resultant loss of freedoms and discrimination, the concentration of power and the concomitant imposition of practices and standards, and the negative impact of platforms on the labor force” [14]. One concern they address is the power of accumulating large datasets affecting the democratic process, as demonstrated by the Cambridge Analytica scandal [14].

Chapter four

4 Methodology

This chapter presents the research methodology used in this thesis; design science research. Furthermore, the chapter presents the data collection methods used during the field study and the software development process. Lastly, I reflect on the use of these methods and how they affect the development of the artifact.

4.1 Choice of research methodology

My study aims to examine how CCM can be implemented in DHIS2. Therefore, choosing an appropriate methodology to design the application and the pilot is essential for the development and implementation of the study. Choosing the correct methods when implementing the pilots is essential, as pilots quickly can end after the piloting period if nobody is responsible for iterating the artifact.

In choosing the proper methodology for this thesis, the different methods of Engaged Scholarship were of interest; as stated, "In Engaged Scholarship, researchers collaborate with practitioners in addressing problems relevant to a real-world problem situation and scientific knowledge" [33]. Initially, Action Design Research (ADR) was intended as it "aims at building innovative IT artifacts in an organizational context and learning from the intervention while addressing a problematic situation" [55, p.38]. ADR requires an introduction of artifacts into organizational practice and emphasizes iterative design on the artifact. The iterative design includes evaluating and redesigning the artifact according to the organizational context. Including the HISP group in the different iterations by multiple visits could give them ownership and increase the possibility of further developing the artifact.

However, the pilot was postponed several times due to many uncertainties regarding the covid-19 pandemic. Because of the delays, I kept developing the artifact in Oslo, resulting in the custom cold chain application. Because of the uncertainties regarding my pilot, I changed my methodology to design science research (DSR). Peffers et al. argue that DSR can be done without introducing the artifact for use by the organization and that evaluations could be done with stakeholders familiar with the organizational context [45]. Focusing on the artifact itself helped to continue the software development. Knowing I could develop an artifact that could be evaluated by stakeholders familiar with the organizational context worked as a backup for my thesis if the pilot never were to be conducted.

4.2 Design science research

DSR focuses on developing and designing artifacts [22]. In the iterative process of designing the artifact, the evaluation must be grounded in theory, and empirical data collections [18, p. 340]. Hevner et al. also point out that ”Design science, as the other side of the IS research cycle, creates and evaluates IT artifacts intended to solve identified organizational problems” [22, p. 77]. Figure 3 illustrates the artifact iteration during the DSR process.

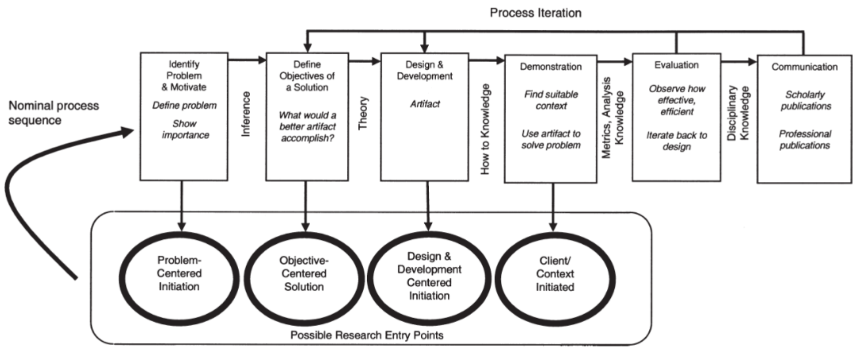

Figure 4: Design Science Research Model [44, p. 58]

In addition to designing and evaluating the artifact over iterations, there are two different strategies for conducting DSR [23]. The first strategy is that the researcher develops a general IT artifact addressing a class of problems. After the development, the researcher can introduce the artifact to an organization.

The second strategy, as Iivari states;

> “a researcher attempts to solve a client’s specific problem by building a concrete IT artifact in that specific context and distills from that experience prescriptive knowledge to be packaged into a general solution concept to address a class of problem”[23].

The first strategy was fitting due to the possibility of working on the artifact without including stakeholders. Nevertheless, following strategy one - the artifact was tested in a different context than the developing context in Norway. Even if one aims to create a generic artifact, it had to be tested in a context and with an infrastructure that you typically see in countries that use DHIS2. Therefore the second strategy emerged, partly due to the uncertain future of the pilot. The second strategy was fitting to meet the goal of this thesis because the pilot is specific to its context and could prove the need for generalization with findings addressing a class of problems within the DHIS2 platform ecosystem.

4.3 Data collection methods

This section presents the different data collection methods used in this thesis before and during my field study in Mozambique. I attended regular meetings in person and online before my trip with the HIPS Oslo LMIS team. The collaboration provided knowledge about the context I was contributing. It also helped when creating an effective plan before conducting the field study. Lethbridge et al. suggest a set of field study techniques divided into three main categories to choose the most suited data collection methods. I have taken inspiration from the first degree, where I am in direct contact with the participants, and the second degree, where I observe the work without needing to communicate directly to the participants [32]. Hence I am studying events and activities in a natural context. I have also taken inspiration from ”The art of case study” to define the scope of the data collection. Stake suggests observation and interviews as data collection methods since data from a case study should be qualitative [58]. Table 1 gives an overview of the data collection methods used throughout this master thesis.

Method	**Description**	**Outcome**
Participation in meetings	Online video calls and psychical meetings with HISP Oslo LMIS team and HISP Mozambique	Domain knowledge, insight regarding CCM, application evaluation, and pilot planning.
Interview	Individual semi-structured interviews during field study	Qualitative data, insight into the current state of CCM, work processes, and challenges.
Observation	Multiple observations of health facility workers during field study	observation of current work processes, and comparing data from the interviews.
Demo	Multiple demonstrations of the application to different stakeholders before and during the pilot	Feedback on usability, observing how users navigate the application, and training facility workers.

Table 1: Data collection methods.

4.3.1 Interviews

Interviews can be considered a ”conversation with a purpose” [47, p. 233], conducting interviews allows for collecting qualitative data by asking a set of questions regarding a topic. The interviewees get the opportunity to give a detailed response and go in-depth about the topic. In general, there are three different kinds of interviews; Unstructured-, semi-structured-, and structured-interviews.[30, p. 179][47, p. 233-235]

Type	**Description**
Unstructured interviews	Exploratory and open questions act as an in-depth conversation about a topic.
Structured interview	Asking predetermined standardized questions to all participants, often short and closed.
Semi-structured-interview	Combining features from the interview types above, the interview guide can contain both open and closed questions with the possibility of probing questions.

Table 2: Main interview types.

One of the pros of interviews is their ability to go into detail, they can be exploratory, and although one has a set of questions, the interview can be flexible. Compared to other data collecting methods, one con of conducting interviews is that they are time-consuming. Interviews require the preparation of questions in advance and the work related to the collected data, as it has to be transcribed, coded, and analyzed.

The interview was a natural choice when choosing data collection methods for my research, as interviews efficiently provide contextual insight when done right. During my field study, semi-structured interviews were performed when collecting data from health facility workers. The first round of interviews was conducted during the pilot by visiting three facilities. The interview type was semi-structured as this offered more flexibility and less formality regarding the interview guide. The interviews lasted up to an hour, I had some fundamental questions, but the opportunity to improvise relevant interview questions from the surroundings was an advantage. The degree of informality also led to health facility workers being more relaxed and not only providing the answers we wanted to hear. The informal approach also led to findings we could not foresee because of the health facilities' work processes and challenges. After the initial interviews, we performed another round of semi-structured interviews at all the facilities to examine the status quo after using the application. These interviews lasted approximately a half hour. Lastly, at the end of the pilot, we conducted another round of semi-structured interviews lasting approximately a half-hour to gain feedback on the application and the pilot from the health facility workers. During all interviews, I took notes by hand as this was more flexible than recording the interview, and I used the time between facilities visits to complete and analyze the notes.

4.3.2 Meetings

A collaboration between HISP-group Oslo and Mozambique identified the necessity of implementing CCM to DHIS2. Due to the covid-19 outbreak, digital meetings have been a big part of gathering data for my research. I tried to understand the global need for better CCM by attending regular meetings of varying lengths over six months. These meetings acted as an unstructured form of interviews. The meetings were open conversations with domain experts where I asked domain-specific questions and noted relevant data. The qualitative data from the meetings was sufficient enough so that I did not need to conduct a formal semi-structured interview, as I could instead ask during a meeting. During the application development in Oslo, I meet with the Mozambique HISP group three times before the pilot, planning our collaboration. Furthermore, during the pilot, we had additional meetings, planning the pilot's execution and the application's future.

During the pilot, I had two meetings with The Provincial Directorates of Health (DPS) Quelimane for a half-hour each, explaining and demoing the solution to get permission to pilot the application at the health facilities. Digital and physical meetings have been an essential part of data collection. Nevertheless, attending meetings not being an official qualitative data collection methodology, the close follow-up and enrollment in the HISP-Oslo LMIS team have provided crucial contextual insight. Besides their understanding of CCM and the different use-cases, attending regular meetings provided a feedback loop on the application. Being alone in the software development process, feedback, and evaluation from the LMIS group was essential for meeting the requirements. In addition, evaluation during meetings also helped develop an application closer to the global application requirements from vendors like WHO and Gavi, making the pilot more likely to succeed - and be adopted by the DHIS2 platform core.

4.3.3 Observation

Observations were another widely used data collection methodology during this thesis and the field study. Since I did not speak the language and was dependent on an interpreter, observation helped to see the correspondence in the data from the interviews and what the practice was.

"Observation is a useful data-gathering technique during product development" [47, p. 252]. During this thesis, observation has been used to collect qualitative data in evaluation with the HISP-Oslo LMIS team during application development in Oslo. Furthermore, I conducted direct observation in the field during facility visits. These observations acted as a form of usability testing of the application and training of health workers with different degrees of participation. The observations at the facility visits were often conducted simultaneously or after an interview. Lastly, we observed during a polio campaign where the application was used to collect temperature data from vaccine carriers. The table below shows the different observations done during the pilot.

Type	**Observation variation**	**Description**
Naturalistic observations	Observing excising work processes	Observing the excising work processes regarding CCM and vaccination.
Participant observations	Observing application usage	Observing health facility workers capturing and uploading data using the CCM application.
Naturalistic observations	Observing campaign work and data reporting	Observing how they performed a campaign and reported temperature data with the application to DHIS2.

Table 3: Pilot observations.

The naturalistic observations of health facility workers conducting their work tasks helped map their actual needs and the challenges that indirectly influence their work processes. Challenges like poor infrastructure, overcapacity, challenges with vaccination freezer, and internet connectivity were identified.

The second type of observation conducted was participant observations, which allowed for interaction with the health facility workers. This was used to participate with guidance, control, and training of health facility workers in using the custom application. Such observation was practical when unexpected incidents accrued - like technical bugs or data not being reported. Observing facility workers' application usage made it easier to identify the root problem and conclude if the application design was poor or human error. Participating also provided an opportunity to correct the incident and prevent similar cases, for example, by fixing a code or providing additional training.

Figure 5: Fixing technical bug and giving additional training.

Lastly, I conducted participant observations by monitoring data uploaded to the DHIS2 platform. Observing this helped identify technical bugs, infrastructural challenges, challenges of implementing a digital tool in excising work processes, and the need for additional training in the application usage. As the pilot covered multiple health facilities, monitoring their upload rate was a way of controlling the implementation. This observation is categorized as participatory because we contacted the health facility and asked why they did not upload data.

4.3.4 Demo

Demonstrating the artifact in this thesis has been an essential part of getting feedback on the artifact. During the thesis, I held demons of the artifact digitally and physically for various stakeholders as this was a unique opportunity to demonstrate the product's value. I demoed the application during this thesis at regular meetings with the LMIS team. In addition, to HISP Mozambique before the pilot and to the DPS and the health facilities during the pilot. The different demos have provided the thesis with qualitative data regarding usability, application features, and future work.

In the early stages, getting feedback by demonstrating the application to the LMIS team was beneficial. Because of their domain knowledge about CCM, requirement overview from stakeholders such as WHO, and references from other similar solutions. As part of the LMIS team, the frequent DSR iterations were a way to continuously demo the artifact's progress. The demos provided feedback and ideas on design and functionality.

After the artifact represented a minimum viable product for a pilot, I made a video demonstration of the application. The video was valuable for promoting the application's functionality and design to stakeholders. Distributing the video was also helpful in getting feedback on functionality, design, and new features. The video was also helpful in closing a pilot as its stakeholders could see the artifact features.

During the pilot in Mozambique, the video and live demos of the application were helpful as proof of concept. Proving that the application functioned properly by demoing to DPS was also crucial to accessing the different health facilities. The feedback from the demos held in Mozambique provided qualitative data from stakeholders understanding the environment of the application and the end-users. The demoing acted as a form of evaluation and usability testing. Although I did not collect metrics, I received verbal feedback for changes. I observed the health facility worker use the application during implementation and training in the field. Observing led to metrics like how long and many clicks it took before a user completed the flow. Nevertheless, the metrics were not noted as they would create an uncomfortable situation for the test subject.

Lastly, demonstrating the application was used for training health facility workers in the application usage. There was a need to train both Mozambique HISP implementers and DPS members so that they could demo the application to health facility workers because of the language barrier. In addition, training DPS and HISP implementors is a long-term benefit as they provide maintenance and DHIS2 training.

4.4 Agile Software Development

There are many different processes and frameworks in software development; strict linear sequential frameworks like the waterfall model, typically used in big companies, focus on time and cost optimization. The opposite approach of such linear development processes is agile software development methods.

In 2001 The Manifesto for Agile Software Development presented values and twelve principles adding value to software development frameworks

Figure 6: HISP employee demonstrating the app to health facility workers.

like Scrum, Kanban, and agile hybrids. These frameworks fit in small and big projects with multiple developers, practicing autonomous teams merging increments of a product in ongoing iterations.

Being alone in the development process, following an agile approach would make it easier to welcome changing requirements, even late in development. Principles number ten in the Agile manifesto state.

> “simplicity—the art of maximizing the amount of work not done—is essential ”[48].

Not having other developers collaborate with over-engineering and perfecting the artifact is a common pitfall. By following the principle of simplicity, rapid prototyping made prioritizing parts of the application efficient. It became clear what was essential and which parts could be downgraded.

Rapid prototyping is an iterative approach to quickly improving design and functionality by iterating on a prototype. In the article Rapid prototyping: lessons learned Gordon and Bieman discuss the pros and cons of rapid prototyping; they state that.

> “selecting an appropriate development approach is crucial to building a successful software system ”[17].

Rapid prototyping became the iterations of this thesis DSR by continuously designing different low- and high-fidelity prototypes of the software functionality and design. Low-fidelity prototypes typically represent non-functional prototypes on paper and wire-frames. These prototypes are then

Minimum viable product (MVP)
A prototype as a proof of concept
A low-risk investment
Short development time

Table 4: Characteristics of an MVP

evaluated and make up the starting point for high-fidelity prototypes, which are functional prototypes that often solve tasks.

The main goal of rapid prototyping in the iterations was to have a minimum viable product (MVP) eventually. An MVP is a version of an end-product with just enough functionality and features to give a perspective of the end-product. The concept of an MVP comes from the Lean Startup method, which combines agile and lean methodology, with lean empathizing producing just the right amount [59]. Table 3 displays the main benefits of an MVP.

Hence the MVP should be usable by early users, who can then test and provide feedback for future development. As the agile manifesto states, "Working software is the primary measure of progress" [48]. The combination of rapid prototyping and MVP kept productivity and motivation during the development. The MVP became the measure of progress, and it prevented me from spending time fixing bugs and adding unnecessary functions.

Figure 7: The iterations of building an MVP.

4.5 Reflections

Because of the uncertainty of the pilot being conducted, the choice of methodology was not textbook. Nevertheless, DSR was an appropriate research method because it focused on the artifact's design. Agile software development method rapid prototyping was ideal with the iterations to determine which functionality should be continued. Focusing on the development resulted in a minimal viable product (MVP) being sufficient for piloting. There were also some challenges conducting agile software development alone, as I did not have anybody following up during the iterations. In addition, it was challenging to be strict enough not to be stuck perfecting tasks - as keeping momentum was essential to finish in time for the pilot.

Attending regular meetings with the LMIS team provided me with evaluations of the application. The evaluation was based on the LMIS team's knowledge and empirical data when iterating on the artifact. Combined with the rapid prototyping, a continuous feedback loop on the artifact was established. One downside of regularly presenting the work to the LMIS team was the impression that development time was faster than it was. Therefore, expectation management became important when presenting the work to clarify that some feature was "work in progress." Moreover, I did not perform in-depth semi- or structured interviews before the pilot. As more extended interviews could have provided me with more in-depth data, I chose that the data collected during meetings was sufficient based on plenty of domain-specific information. Nevertheless, one challenge has been taking notes during meetings and keeping the data collected from meetings organized, resulting in additional time to review the data.

The data collection methods during the pilot created a relaxed environment for the interviewing objects, resulting in a good relationship between the HISP group, DPS, researcher, and facility workers. The semi-structured interviews made it easy to make follow-up questions, for instance, regarding what we discovered in the room. One challenge in conducting an interview was taking notes without giving the impression that we were supervising. Since I depended on an interpreter to conduct interviews and demos, giving the HISP group training and more responsibility for the solution was natural, which resulted in efficient interviewing and training of health workers. Having a relaxed environment and being exploratory lead to opportunities like being invited to participate in a polio campaign.

Data analysis

Because of the uncertainty of the pilot, the data gathering was conducted in the last stages of the thesis. This led to a lack of time to transcribe and code the data gathered during the pilot. Therefore, I have not performed a formal form of data analysis like a grounded theory. Nevertheless, I argue that some analysis was conducted during the pilot. Because of the interviews, I was analyzing trough-out the process of collecting data by comparing the answers from the different facilities. In addition, traveling by car between facilities, I used the time to group the data from my interview notes and comparisons data between the facilities. In addition, I kept a diary of all findings done during different visits to identify needs at different levels.

Chapter Six

5 App Development

This chapter presents the development process of the CCM app and the resources and tools used during the software development.

5.1 Developing with DHIS2

The leading actor in this thesis is DHIS2, and the application is developed intended for use in the DHIS2 platform ecosystem. As chapter two introduces, DHIS2 is an open-source innovation platform, meaning third parties can innovation platform using core DHIS2 features. DHIS2 is a Java-based web application that runs on multiple platforms and is compatible with all major web browsers. Through their Application Programming Interface (API), DHIS2 conduct representational state transfer (REST) by providing endpoints that expose data stored in the systems databases, running on PostgreSQL, MySQL, and H2 DHIS2 [3, p.75]. DHIS2 offers various web-based REST-APIs, allowing users to upload and download data from their servers. They also provide an Android SDK as a native solution, exposing the same features as the web API but within an Android environment. The Android SDK is also designed to support DHIS2 core features without an internet connection. In addition, DHIS2 provides documentation on how to use the core functionalities through their REST API's and Android SDK.

Furthermore, they have regular academy training regarding these development tools - to facilitate innovation and sharing of boundary resources within the ecosystem. The CCM application uses two main features from the DHIS2 platform ecosystem facilitating the development, which shall be explained in detail. The first feature is the DHIS2 Android Software Development Kit (SDK), and the second is the DHIS2 Tracker Capture Web Application.

5.1.1 DHIS2 Android SDK

The DHIS2 Android SDK facilitates the development of custom Android apps using DHIS2 as a platform. As a platform BR, it aims to help independent app developers build custom ad-hoc apps by providing them with standardized tools, and functionalities [5]. The SDK contains a local database representing the DHIS2 data model; this allows synchronizing with the server fetching Java objects instead of calling multiple APIs. The SDK gives a significant advantage as the application can synchronize with the servers, capture data without internet access, then upload the data when connected [5].

	API	SDK
Purpose	Connect and integrate software	Contains a verity of development tools
Characteristics	Lightweight, fast and contextual specific	More robust, usually including many utilities
Use case	Used to add specific features to an application	Used to make a new application, adding many features using one package

Table 5: Difference between API and SDK [54]

5.1.2 Tracker Capture

The CCM application connects to the DHIS2 core and the Tracker Capture application by utilizing the Android SDK. Tracker is a generic web application focusing on individual-level (or case-based) transactional data. It supports data collection, case monitoring, and follow-up, as well as analyzing and automatic aggregation for HMIS dashboards and reporting [63]. The cold chain application captures temperatures and stores them as events on a tracked entity instance (TEI) - representing a sensor. The tracker app can then use the temperature data to aggregate graphs and reporting, for instance, regarding deice outbreaks and vaccine quality.

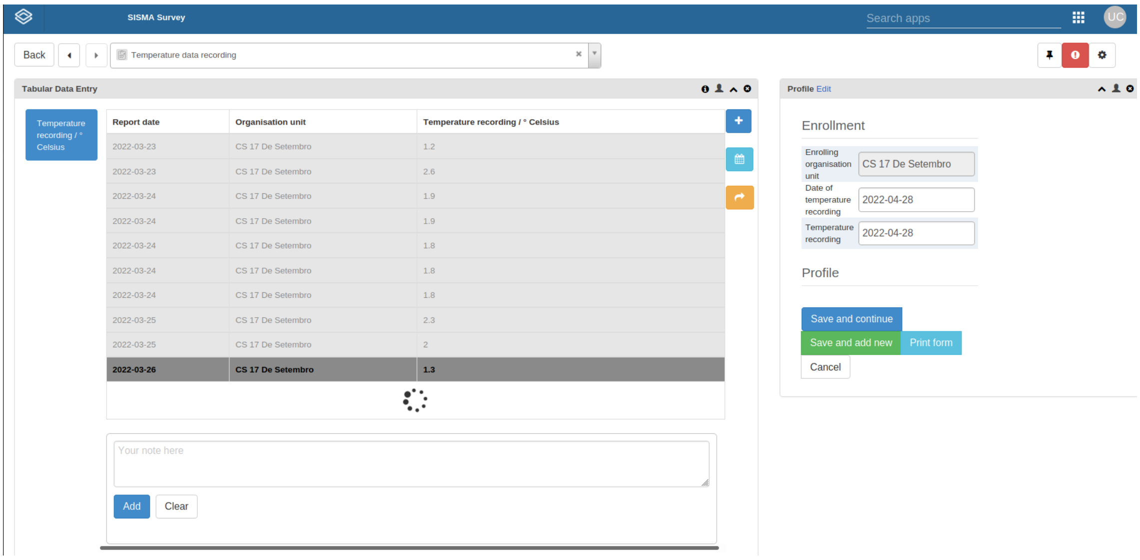

Figure 8: Tracker capture with temperature data

5.2 Blue Maestro temperature sensors

Blue Maestro is a tech company providing hardware in the form of different Bluetooth temperature sensors and software to manage the sensors. This thesis utilizes Blue Maestros Tempo Disc™ Temperature Sensor Logger, as these Temperature loggers fulfill the requirements regarding User-programmable temperature data loggers from the World Health Organization [46].

The BlueMaestro sensors monitors and log data values like; temperature, humidity, dew point, and pressure. They continuously transmit the current value and store 20,000 data points for downloading. The sensors are small and have a replaceable battery with a lifetime of approximately one year in regular use. Most of the devices are also water-resistant, and some are waterproof.

BlueMaestro was the sensor of choice in this thesis because they fulfill WHO performance standards within the operating temperature range of +55°C and -30°C. Having an accuracy of ±0.5°C or better within the range -20°C to +20°C for a minimum of 12 months. One sensor is relatively cheap compared to other CCE, and the only maintenance required is changing the battery.

These sensors are also used by the company mSupply and are approved by WHO for providing temperature monitoring devices [11]. To get such approval, organizations need to fulfill the requirements of a working cold chain application in addition to sensor technology. By connecting the DHIS2 core with the Blue Maestro sensor, the application developed in this thesis is a possible solution for getting such WHO Approval.

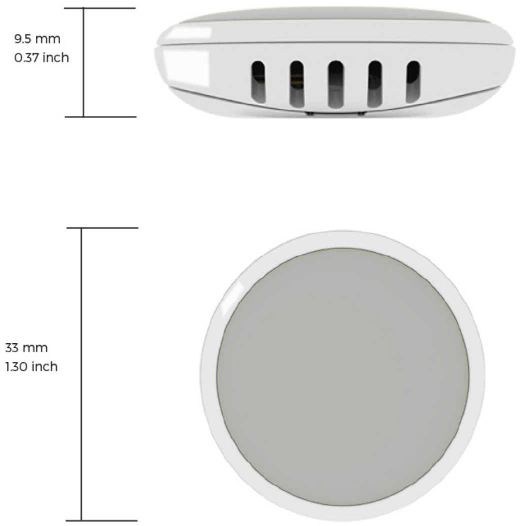

Figure 9: Blue Maestro Tempo Disc™ Temperature Sensor Logger

5.3 Architecture

In this section, I will introduce the system's architecture and application, introducing the different models and the design pattern used in the development process. The system architecture represents all software and hardware involved in the entire solution of the HIS, and the application architecture represents the architectural choice in developing the application.

5.3.1 System Architecture

The system architecture of the custom cold chain monitoring application is represented in figure 6, presenting the system components, behavior, and flow of data.

The application system architecture consists of several components, both software, and hardware. The first component and a critical part of the system architecture is the BlueMaestro temperature sensors. These hardware components capture temperature data at intervals and advertise it to the Android application through Bluetooth Low Energy (BLE). The component that acts as the link between the sensor and the DHIS2 platform is the CCM Android application, making it possible to upload temperature data from the sensors to DHIS2 servers. The CCM application is the component that provides a user interface for the data capture end-user, showing temperatures and providing additional functionality.

The application's main functionality is capturing and uploading temperature data to DHIS2 servers. Through the login functionality, it is compatible with connecting to different DHIS2 instances. After data reaches the

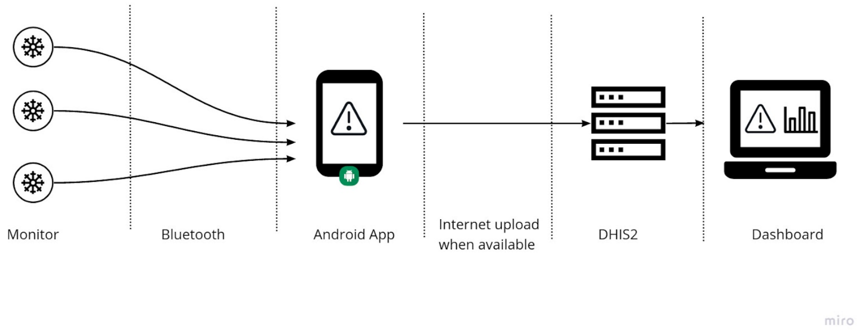

Figure 10: System Architecture

DHIS2 server, the last component in the system architecture is the tracker capture application. The web solution provides a user interface and makes it possible to visualize, aggregate, and administrate captured temperature data for decision-makers. All these components combined are what I have called the system architecture, as they depend on each outer for the solution to function. This system architecture digitalis a manual work CCM process being conducted in real-life context today. In addition, it provides the DHIS2 platform with a new data element, temperature data.

5.3.2 Application Architecture

The Android application developed acts as a link-layer between the temperature sensors and DHIS2 and is an essential part of the system architecture. The application consists of different DHIS2 modules making integration possible through their Android SDK. The other parts of the application are modules explicitly developed for Android Bluetooth communication and databases for storing temperature data with and without an internet connection. Figure 7 illustrates the relationship between the different modules in the application. Since the application is a custom standalone Android

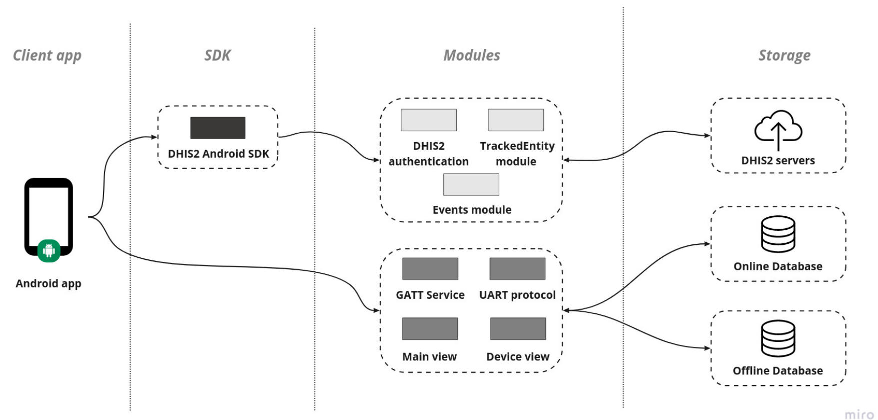

Figure 11: Application Architecture

application, its architecture has taken a monolithic structure. It is built as a single unit, consisting of all the modules the applications need to function. The opposite would be a microservice architecture, allowing an extensive application to be separated into smaller independent modules. These modules have their responsibility and are fully functional without the other modules. Moreover, in this project, microservices would not be sufficient as the application relies on all its modules and services to run in order to function. In addition, the application size is manageable. Nevertheless, if the application scales to a more generic system, the monolithic architecture should still be the fitting architectural choice. Table 4 presents the main characteristics of monolithic application architecture and its usage in our case.

Characteristics	Our case
A database	Consisting of two separate databases with tables structures in a relational manner.
A client-side user interface	An Android Application with multiple dynamic activities.
A server-side interface	Consisting of the DHIS2 Android SDK and Bluetooth communication developed in Java. Populating the client-side user interface by retrieve and updating data from the database.

Table 6: Monolithic application architecture.

In addition to making a monolithic architecture choice, efforts have been made to follow a software design pattern during development. Software design patterns are structuring methods that can help build a well-structured application. Since the application is developed with the object-oriented language Java, the development process strived to follow the Model-View-Controller (MVC) software design pattern, which is widely used in Java development. The MVC design pattern structures the software into three main objects, the first object in the pattern being the Model. The Model contains application data; its primary function is storing and exposing the application data. In order to provide this main functionality, the structure of the Model is typically a database with helper functions to get, add, and delete data.

The next object of the MVC pattern is the View. The View's primary function is to present the data provided by the Model to the end-user. The View can only read the Model data, not manipulate it. This is often done by UI components that take the data from the Model and dynamically render it as a component. An example is the text component displaying temperature values in the CCM application. This component shows empty values before the application gets temperature data from the sensor. When the application gets data from the Model, the View components update and dynamically display the temperature data.

Lastly, the final object in the MVP pattern is the Controller object. The Controller object's primary function is to conduct logic operations. The data from the Model object goes through the Controller object. The Controller acts as the link-layer between the View and the Model. The Controller listens to events triggered by the View and executes the appropriate action to these events. An example of such events can be a button click in the View; the button click then triggers an event in the Controller calling a function in the Model. Lastly, the Controller returns the Model result to the View. Since the View and the Model are connected through the Controller, the result of the button click is then automatically reflected in the View. Figure 12 illustrates the relationship between the Model-View-Controller objects of the MVC design pattern.

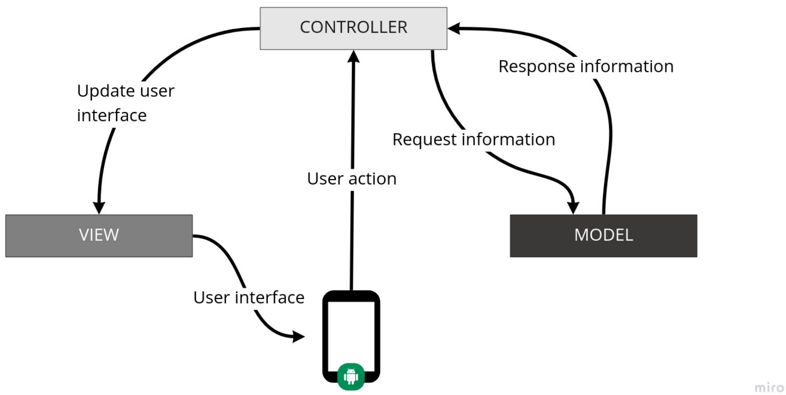

Figure 12: Model-View-Controller design pattern

Because of the application size and its custom standalone solution, combining a monolithic architecture with the MVC pattern helped reduce complexity by developing in a controlled environment. The focus was on making things work within the app, not connecting multiple back-ends and different endpoints. Nevertheless, the choice of a monolithic architecture approach could be a disadvantage as the application scales. A monolithic architecture struggles to facilitate integrations with other systems. However, because the application is developed as a third-party application based on the DHIS2 core, I argued that it is being monolithic is an advantage. Merging it to the platform core is less complex than an architecture based on a network of multiple microservices.

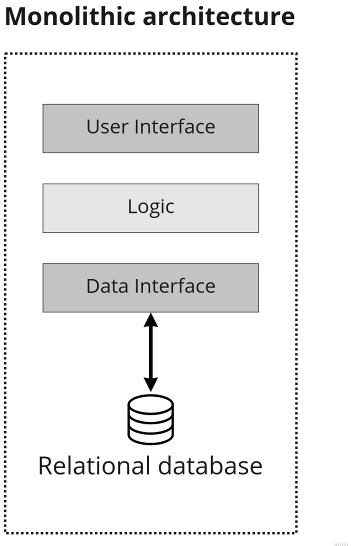

Figure 13: Monolithic application architecture

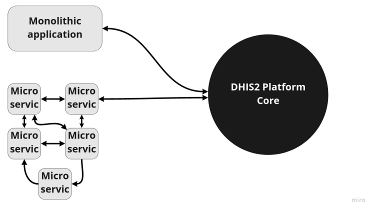

Figure 14: Platform consummation

5.4 Bluetooth communication

As mentioned, is Bluetooth communication an essential part of the system architecture. This section introduces the Bluetooth technology used in this thesis and the essential parts of developing an application with Bluetooth Low Energy (BLE). Lastly, it goes in-depth on how the BLE communication in the application is built.

5.4.1 BLE devices

The BlueMaestro sensors are known as BLE devices, and the applications use the Bluetooth technology known as BLE to communicate with them. BLE is not confused with the "classic" Bluetooth radio.

The Bluetooth Classic radio (BCR) is a low-power radio that streams over 79 channels in the 2.4GHz unlicensed band. Supporting point-to-point device communication, BCR is mainly used to enable wireless audio streaming and has become the standard radio protocol behind wireless speakers and headphones. The BCR also allows data transfer applications such as mobile printing and much more [8].

The BLE radio uses low-power operations. Transmitting data over 40 channels in the 2.4GHz unlicensed ISM frequency band, the Bluetooth LE radio provides developers more flexibility in building products. Bluetooth LE supports multiple communication nodes, expanding from point-to-point to broadcast and, most recently, mesh, enabling Bluetooth technology to support the creation of reliable, large-scale BLE device networks. While initially known for its device communications capabilities, Bluetooth LE is also widely used as a device positioning technology to address the increasing demand for high-accuracy indoor location services. Initially supporting simple presence and proximity capabilities, Bluetooth LE now supports Bluetooth® Direction Finding and, soon, high-accuracy distance measurement [8].

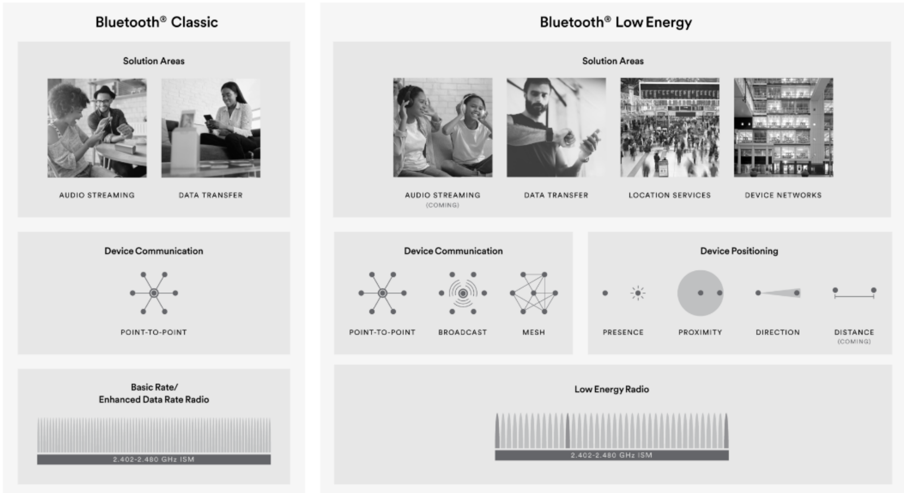

Figure 15: Bluetooth protocols

In addition, after the Bluetooth 4.0 specification, devices can implement either or both BLE and BCR systems. The Bluetooth BLE protocol intends to provide considerably reduced power consumption and cost while maintaining a similar communication range as the BCR. Due to these advantages, mobile and computer operating systems such as IOS, Android, Mac, Linux, and Windows10 support BLE. Several chip producers, including Nordic Semiconductor and Silicon Labs, introduced BLE-optimized chipsets. A study by beacon software company Aislelabs reported that peripherals such as proximity beacons usually function for 1–2 years powered by a 1,000mAh coin cell battery [39]. This long-lasting battery life is possible because of the power efficiency of the BLE protocol, which only transmits small packets compared to BCR [39].

Figure 16: BlueMaestro BLE device configuration

5.4.2 BLE GATT service

GATT is an acronym for the Generic Attribute Profile. It defines how two Bluetooth Low Energy devices transfer data back and forth using services and Characteristics. Moreover, it uses a generic data protocol called the Attribute Protocol (ATT), which stores Services, Characteristics, and related data in a simple lookup table using 16-bit UUIDs for each entry in the table. Characteristics distinguish themselves via a pre-defined 16-bit UUID. For example, identifying the BlueMaestro temperature sensors characteristic is mandatory for the GATT Service and uses a UUID of 0x2A6E [64].

The GATT protocol comes into play once a dedicated connection is established between two devices, meaning that the device has already gone through the advertising and scanning process by identifying the different BLE devices. An essential point to remember with GATT and connections is that connections are exclusive. Only one BLE device can be connected to one central device, in this case, an Android device at a time.

The reason for this functionality is to provide asynchronous BLE communication. As soon as a BLE device connects to an Android device, it will stop advertising itself. In addition, other devices will no longer be able to see or subscribe to the BLE device before the existing connection is broken. Moreover, as mentioned, BLE technology support mesh networks of BLE devices. Nevertheless, establishing a connection through GATT is the only way to allow two-way communication, where the Android device can send meaningful data to the BLE device and vice versa. Furthermore, communication can be achieved through characteristics, as they can write data through characteristics to the BLE device.

5.4.3 Nordic Semiconductor UART Service

To achieve the read-write communication between the android- and BLE device from the GATT characteristics, one must implement a Universal asynchronous receiver-transmitter (UART) interface to send commands back and forth. In this thesis, the UART protocol needed two characteristics to function, one for the TX channel and one for the RX channel.

The TX channel (0x0002) is used to send data back to the sensor. By allowing read access, the TX channel can write commands to the BLE device by the connected Android device. The RX (0x0003) characteristic sends data out to the connected Android device. In this case, the BLE device can send data to the Android device every time a command is sent through the TX channel.

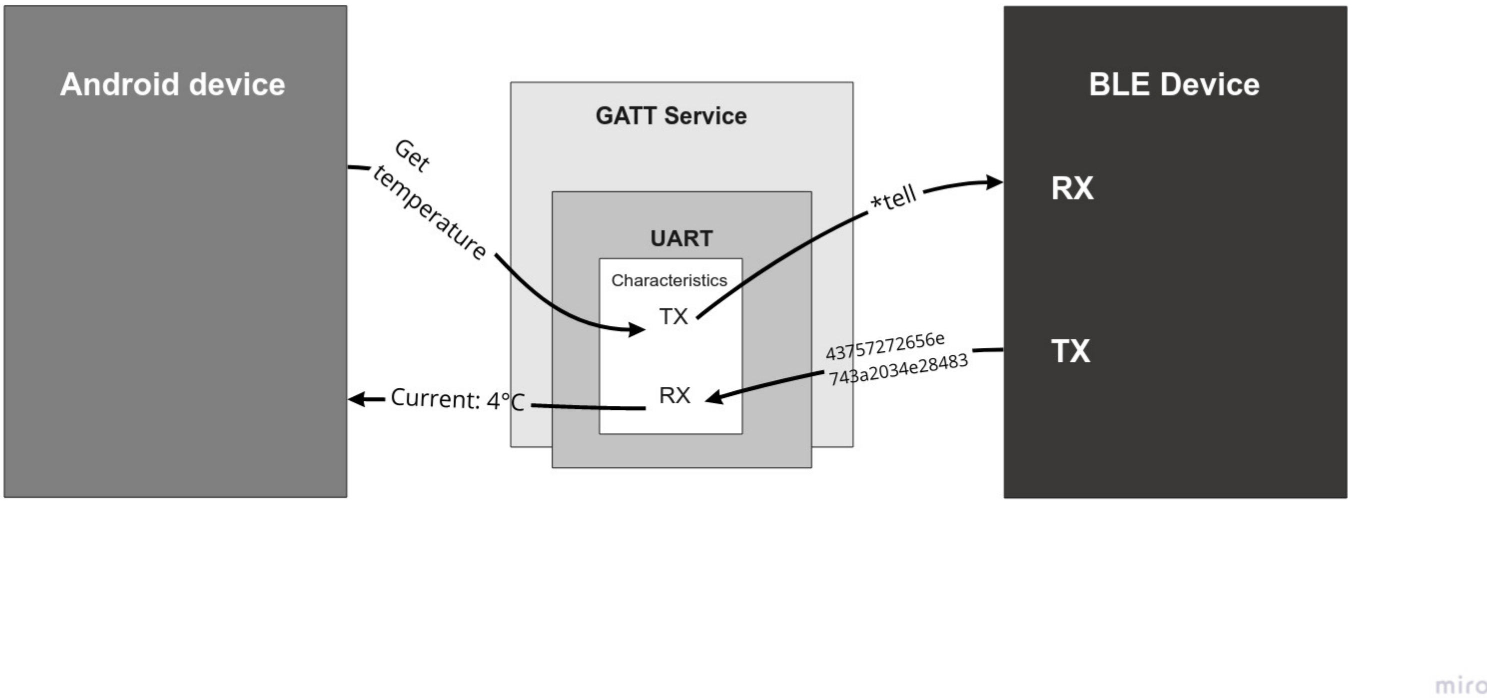

Figure 17: GATT and UART interaction

5.5 Development tools

This section introduces some of the essential tools that have been used during the development process of the application.

Tools	Description
Git	A version control system. Characterized by being distributed and not folder hierarchy-based.
GitHub	Web application utilizing Git to deliver distributed version control, source code management, hosting, and more.
Documentation	BlueMaestro and DHIS2 SDK documentation
Nordic Semiconductor's nRF toolbox	Applications demonstrating different BLE technology usage.
Android studio and DHIS2 Skeleton App	Integrated development environment for android, and DHIS2 application template.

Table 7: Development tools.

5.5.1 Git

The development process of the application consisted of implementing pieces of code that worked separately to keep up the productivity. Furthermore, to keep track of these different pieces of code, there was a need for version control and a tool merging the different pieces of code into a working application. Using Git as a version control system was natural, both based on experiences from earlier projects and because DHIS2 has its open-source code on GitHub.

Git is a version control tool suitable for large and smaller code bases. Git makes sharing, editing, merging, and re-basing code to a functional version possible. These operations can be done through a terminal interface or through one of Git's web solutions, GitHub, where others can inspect, contribute, and download the code if it is open-source.

During the piloting of the application, Git was an essential tool as it functions locally without an internet connection. Using Git locally made it possible to keep track of different versions of the application, as different branches and changes were made to meet the health facility's needs. If the application is re-iterated, the code can easily be shared with the DHIS2 development team and other interests in the DHIS2 ecosystem through Git.

5.5.2 Documentation

During the development process, additional documentation has been essential for the progress. The BlueMaestro documentation was a vital knowledge boundary resource in the application development, providing access to all the different commands essential for communicating with the sensors over the UART protocol. In addition, it explains the sensor technology and exposes the UUID of the Nordic UART Service used in the sensors.

In addition, the BlueMaestro SDK also functioned as a form for documentation. Despite being deprecated and non-functional, the code was written in Java - the same language as the CCM application. Reading and examining the code gave an idea of how one could connect to their sensors. The deprecated code acted as a guide to options in the official Android documentation, which was developed to replace the deprecated code.

Lastly, the DHIS2 Android SDK documentation has been a good knowledge boundary resource for developing in the DHIS2 ecosystem. The documentation provides an overview of extracting data and sending queries to their SDK. In addition, it describes how to work with the tracker capture app and configure a tracker program within a DHIS2 instance.

5.5.3 Nordic Semiconductor's nRF applications

The sensor technology and Bluetooth communication have been the most crucial part of the application to fulfill the CCM requirements. As mentioned, the chipset inside the blue maestro sensor is a Nordic Semiconductor's nRF52832. The nRF52832 is a general-purpose multiprotocol System on Chip (SoC). It provides advanced BLE features, protocol concurrency, a set of peripherals, and features like the UART protocol. The chip also offers both Flash and RAM [43].

During the development, I discovered the Nordic semiconductor's different free-to-use applications to test their products. During the development the Nordic semiconductors' nRF-logger, -Connect, and -Toolbox applications were used to communicate through the UART protocol [42] [41]. These tools provided the opportunity to get familiar with the response from the sensors at given commands.

5.5.4 Android studio and DHIS2 Skeleton App

Hence, the application development is in the Android environment; the IDE Android Studio has been an essential tool in the development process. It has been helpful for programming the application, debugging, and sharing APK-files. The Android studio emulators let developers compile the code on a computer partition to emulate an Android device. Since the application utilizes Bluetooth testing, the application had to be done with an external cable on an Android device, as one cannot use the computer Bluetooth radio on the emulator.

An additional boundary resource in this development process was the DHIS2 Skeleton App. The Skeleton App is an Android template provided on DHIS2 Android team GitHub. Forking the Skeleton App gives access to a functioning application and pre-built features like login functionality and synchronizing with a tracker program. Utilizing this template as a start point for the application saved a lot of configuration time to set up an Android environment after the DHIS2 documentation. The Skeleton App also provided some reusable components and design standards like colors and fonts that could be reused to give the feeling that this was a DHIS2 application.

6 Development iterations

This chapter presents the development iterations during the DSR and application development. There are five iterations utilizing a combination of DSR and agile software development - rapid prototyping. The different iterations and their goals are based on requirements from the HISP LMIS team; this section describes the primary goal, changes, and how the iterations were conducted. The figure below represents the application development timeline, each iteration, and its progress.

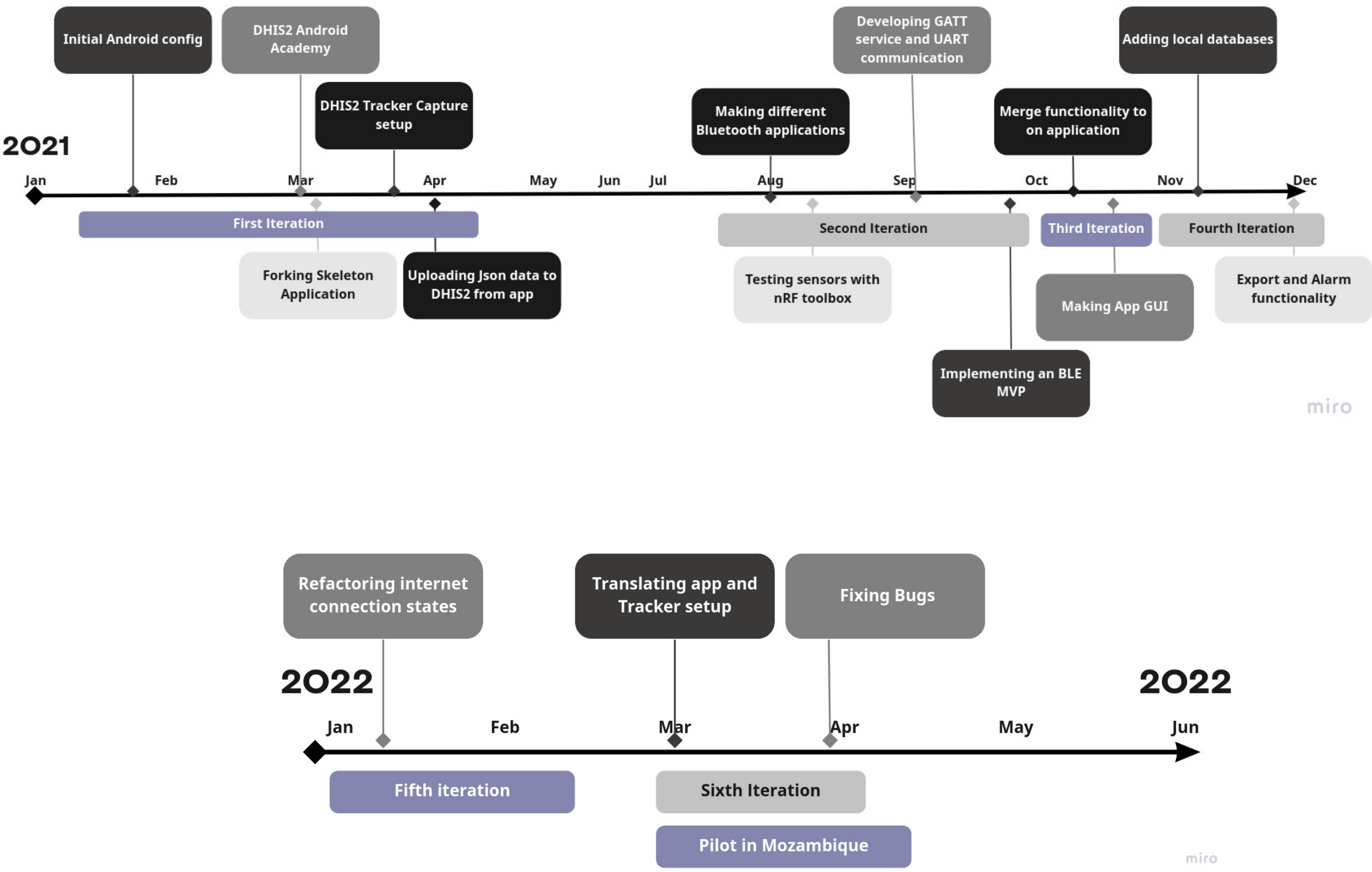

Figure 18: Development iteration timeline.

6.1 First iteration

Introduction
The artifact from the first iteration was developed in Oslo, starting in early 2020. The development was conducted in iterations with the DHIS2 Android Skeleton application as the starting point. The development was based on the requirements provided by the HISP Oslo LMIS team based on the WHO and GAVI requirements on CCM. Since the application had to be developed within the DHIS2 and Android environment, it led to a more high-fidelity prototype in the first iteration. The main goal for the first iteration was to get access to the DHIS2 tracker capture tool through the Android application. Lastly, the iteration was finished by evaluating with HISP Oslo and based the next iteration on the evaluation feedback. The list below presents the essential development tasks done during this iteration.

Iteration development

- Attending DHIS2 Android Academy.
- Forking Android Skeleton App.
- Configuring Tracker program on LMIS test instance.
- Uploading real temperature data from JSON.

Development
To reach the primary goal of the iteration, I attended the DHIS2 Android academy. This taught me how the API was structured and how to develop applications in the DHIS2 platform ecosystem. At the academy, we used the Android Skeleton application, which became the application's starting point. After forking the application, the LMIS team provided a DHIS2 test instance. The next step in the iteration was configuring a tracker capture program. This was so that I could programmatically upload temperature events to tracked entity instances (TEI).

Furthermore, after the first iteration, it was possible to log into the Android application with the test instance and synchronize it with the tracker program. The LMIS team provided a CSV file containing actual temperature readings. The application then iterated the CSV file and uploaded temperature events to the tracker program. At the end of the iteration, I demoed the application to the LMIS team and gathered feedback and system demands. The first iteration's primary goal was to complete the connectivity between the android application and the DHIS2 platform was met. However, the feedback from the LMIS team stresses that the next intuitive step was making a Bluetooth connection with the Blue Maestro sensors.

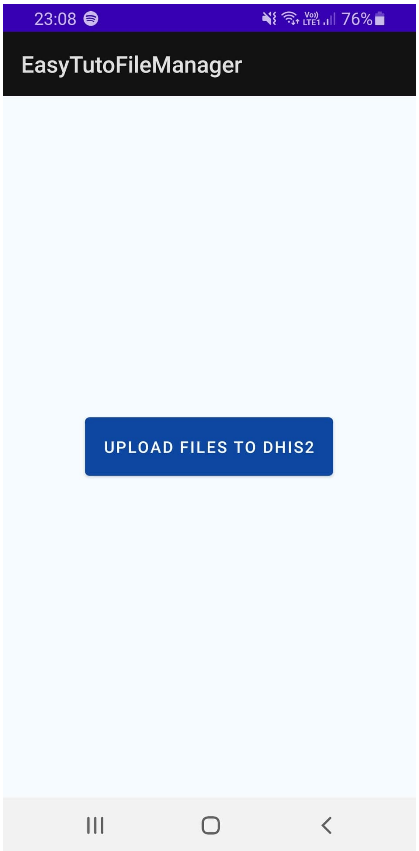

Figure 19: DHIS2 upload-button

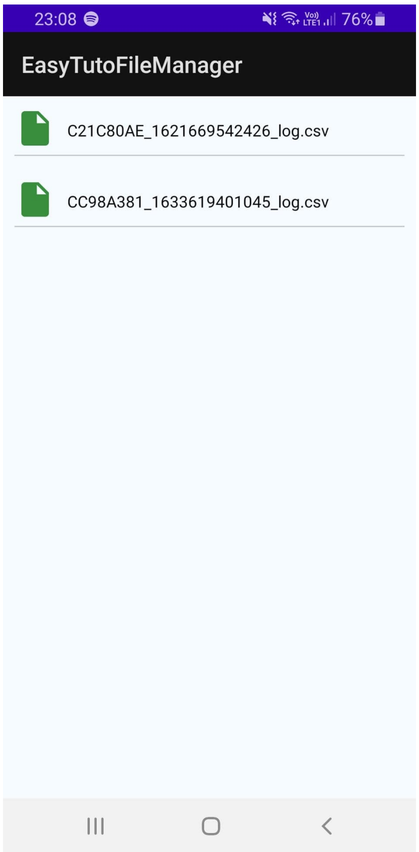

Figure 20: CSV-files with temperature data

6.2 Second iteration

Introduction
After the first iteration's evaluation, the second iteration's goal was clear - making a Bluetooth connection. Furthermore, despite the precise goal, the second iteration was the most extended and demanding of the development iterations. Since the LMIS team already chose the Bluetooth sensors, the scope was narrowed in choosing technologies. During the iteration, we met with the BlueMaestro founders regarding sensor specifications, API, and their open-source SDK. The list below presents the essential development tasks done during this iteration.

Iteration development

- Developing multiple applications with different Bluetooth technology.
- Testing with nRF toolbox application.
- Developing an GATT service and UART communication layer.
- Making an MVP of the application.

Development
This iteration was all about making a Bluetooth connection. Before purchasing sensors for this project, we meet with the BlueMaestro team. After the meeting, they informed me about their open-source SDK written in the same technology I used. Nevertheless, the Blue Maestro SDK was deprecated and had not been maintained since 2017.

After researching alternative solutions to the BlueMaestro SDK, I developed a scanner application that allowed a mobile device to scan and pair with devices. Working with Bluetooth requires testing on an actual android device to enable Bluetooth. There were multiple resources on Bluetooth development. However, I struggled to communicate with the BlueMaestro sensors. After some weeks of trying different solutions to solve the problem, looking at examples from other vendors. I learned about the difference between Bluetooth Classic and Bluetooth Low Energy (BLE) and that the UART Protocol was specific for the chipset.

I then discovered the nRF toolbox application from Nordic semiconductors, making it possible to see the sensor details. The last part of the iteration was then used to refactor the application to use the GATT-service model and send commands with the UART protocol to the BlueMaestro sensors - resulting in a callback with temperature data.

After one month of intensive development, I finally met the iteration goal. After evaluation with the LMIS team, the application was now an MVP, and the opportunities for piloting were addressed. The feedback from the evaluation was that the applications had to have an alarm system and an automatic connection that retrieved data from several sensors and sent it to DHIS2.

6.3 The third iteration

Introduction

Hence the most critical functionality was developed in the last iteration; this iteration's primary goal was to merge the code from the first and second iterations into one functioning application. As this did not include as much development, it is a very time-consuming job, and inaccuracies create bugs and make the whole application crash. In addition, there was a focus on making a fitting graphical user interface (GUI) for the application end-users. The list below presents the essential development tasks done during this iteration.

Iteration development

- Merging first and second iteration into one application.
- Multiple applications with different Bluetooth technology.
- Sending data from BlueMaestro sensor to DHIS2 tracker.
- Making a GUI.

Development

The application made in the first iteration communicated with the DHIS2 tracker. The second iteration communicated with the BlueMaestro sensors to capture temperature data. After a week of development, these two iterations were merged into one functioning application, sending data from the sensor to the DHIS2 tracker. The rest of the iteration period was spent on developing a GUI. Since I forked the DHIS2 skeleton application, I tried to re-use as many components from the DHIS2 standard as; buttons, list items, icons, and colors.

This iteration resulted in an MVP closer to the final product after evaluation with the LMIS team. This iteration did not meet their requirements regarding alerts and automatically fetching temperatures. Therefore, these requirements for additional functionality had to be given higher priority in the next iteration.

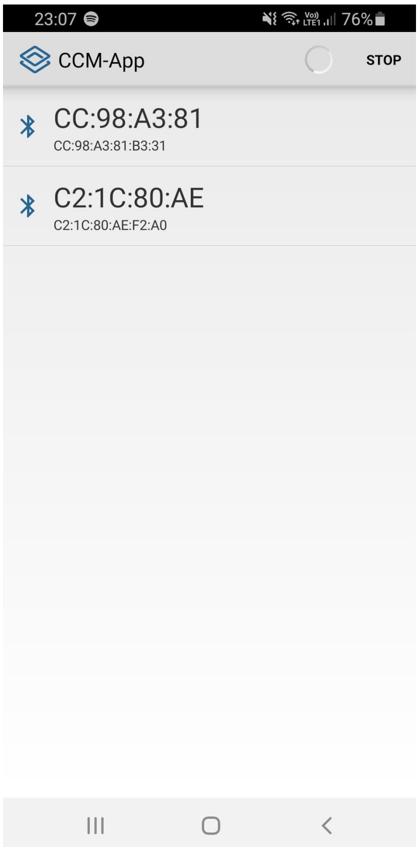

Figure 21: Sensor device list

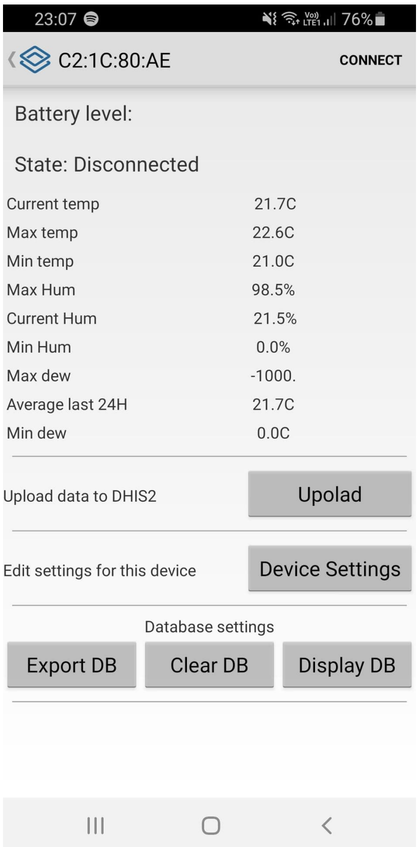

Figure 22: Sensor details

6.4 The fourth iteration

Introduction

The opportunity to pilot the applications was more significant than before, and we were now looking for possible partners. Due to this, the main goal of these iterations was to implement the alarm, optimally an SMS that would notify various stakeholders about critical temperature levels at a facility. The evaluation from the two previous iterations became the starting point for what should be prioritized in this iterations. The list below presents the essential development tasks done during this iteration.

Iteration development

- Storing data in a relational database.
- Making a local notification.
- Providing export of backup data.

Development

The first thing I focused on was implementing a local relational database. To create the alarm system, I had to store data to check if there were cases of critical temperature levels.

By storing temperature readings in a database, the application could store data even if the application does not have an internet-connected connection. After researching alarm API-based SMS solutions like Twilio, I concluded that SMS notification was out of the scope of an MVP being piloted, as this was a paid service. Therefore, the alarm system was downgraded to proof of concept in the form of a local notification triggering when the temperature reached a specified temperature threshold.

Since I had more time in this iteration due to a smaller implementation of the alert notifications, I developed additional functionality for aggregating a graph of temperature readings stored in the database. Implementing a local database also made it possible to implement a function exporting the database as a .csv file. The last part of the iteration was making GUI enhancements. The iteration ended with the LMIS team concluding that the development could end as the MVP was ready to be piloted.

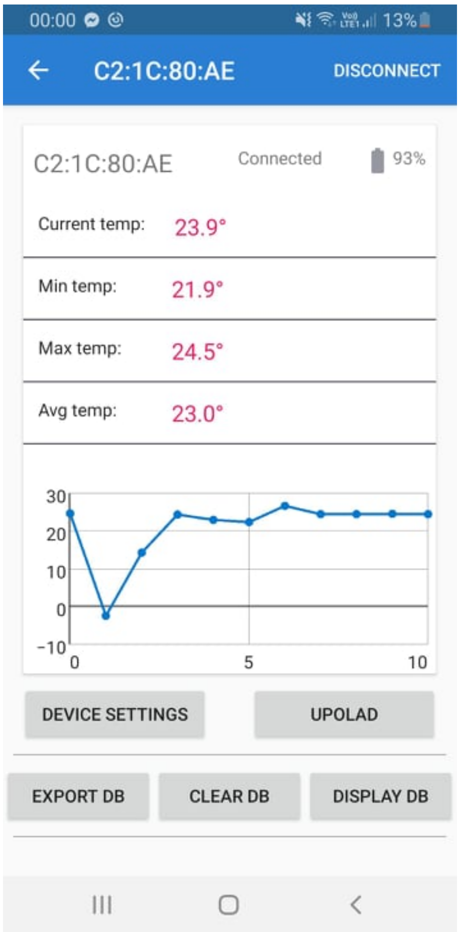

Figure 23: Finalized MVP

6.5 The fifth iteration:

Introduction

This iteration was not a planned iteration but a product of discovery when I made a video demo of the application. After additional unit testing, I realized the application did not work as intended without an internet connection. At the time, the LMIS team communicated with HISP Mozambique regarding piloting the application. The error was detected when I logged out for the first time and tried to log in again without an internet connection. The DHIS2 Android SDK needs internet for logging in and synchronizing with DHIS2. Therefore, I had to refactor the app so one could use the main functionality, login, and synchronize when connected to the internet. The list below presents the essential development tasks done during this iteration.

Iteration development

- Implementing a second database.
- Refactoring DHIS2 login state.
- Refactoring to check internet connectivity in all stages of using DHIS2 SDK.

Development

Therefore I had to do significant refactoring concerning checking for internet connectivity at every step using the DHIS Android SDK. In addition, I had to make two separate login states where the application worked as intended with and without internet connectivity. To handle the data collected in offline mode, I stored this in a separate database. The reason for this was that if the internet access were unstable, data would be downloaded into separate databases when switching between connected and disconnected sate - which would prevent data from being mixed or lost. Having separate databases was necessary because the data from the offline database was then used to create events when one had internet access automatically. After the events are uploaded to DHIS2, the offline database sanitized itself, so that old data is not uploaded several times. After solving these issues, I finally had an MVP ready to be piloted.

Chapter seven

7 Piloting in Mozambique

This chapter describes the pilot in Mozambique. The first part of the pilot was spent testing the application's stability and use-ability and conducting another development iteration. The second part was a field study, examining existing workflow regarding CCM and piloting the application at health facilities in the Zambezia province.

7.1 Context

Mozambique is located in the south-east of Africa with a population of 32.90 million people. The Portuguese colonized the country of Mozambique in 1505. In 1975 the first president of Mozambique, Samora Machel, liberated the country. One of the country's highest incomes is agriculture. Regarding health the estimated lifetime from birth in Mozambique is 60.8 years compared to 82.9 in Norway [34] [35]. In addition the median age of people in Mozambique are 17.6 years compared to 39.8 years in Norway [38] [40]. Mozambique is the second most country with HIV/AIDS cases in Africa. Another big threat to the country's health is malaria [38], despite medication Malaria is fatal or can cause long-term health complications. In addition, polio is still a disease that one struggles to eradicate, even though it has been eradicated in large parts of the world.

Mozambique offers public and private health options, but most cannot afford private health care. Public health is divided into the national, province, district, and health facilities levels. In this thesis, the health facilities level is where the pilot will be conducted and the data gathered. A health facility acts as a small hospital in more rural locations. The facility conducts urgent healthcare, vaccination, weighing children, registration and treatment of malaria, and more. The facility-level are the once collects most of the health data, both digital and in paper format.

DHIS2 is the major HMIS in Mozambique, and this benefits decision-making since the collected data is stored in one place. One example of usage of DHIS2 in Mozambique is the malaria program with the newly integrated iMISS. They use tablets, and facility workers input malaria cases directly into the DHIS2 using digital forms. In the case of CCM, the forms are paper-based, and the data never get reported to DHIS2.

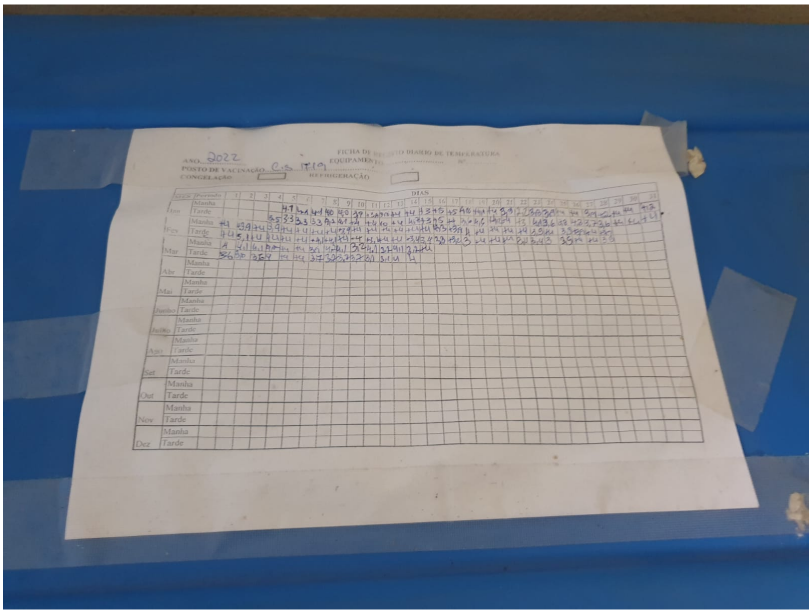

Figure 24: Current CCM in Mozambique

7.2 First week in Maputo

On the first day of the pilot, I met with the Mozambique HISP group at their office, and they introduced me to the development-, implementation-team, and the administration. The next day I presented a live demo of the application for the HISP group. After the demo, we discussed the pilot. The HISP group wanted the application to be more generic and made some application demands. There was also a request to make a custom web-based dashboard so that one can share data with users that did not have access to DHIS2. These demands lead to an additional development iteration, with the goal of a more generic application running on the Mozambique government server, translated to Portuguese.

On the third day, we visited the ministry of health to update their DHIS2 servers. There was no routine for updating the government servers, but the HISP group wanted to make this a yearly routine. After the server update, we meet with a representative from DPS Quelimane to discuss the pilot in Zambezia. We discussed that the pilot would run on four facilities with various infrastructures, providing four Android devices.

The following days were spent in meetings planning the pilot. I was assigned a team of two HISP implementors to travel with me to Quelimane. After an additional demo for the team and management, we discussed the future of the application. They want to be ahead of the process to easier adapt after the pilot and my time in Mozambique. In addition, the pilot needed approval from The Ministry of Health, and having a vision of the future could help get the approval.

The solution includes both hardware and software to function. In order to implement and conduct the pilot at a facility, there is a need for temperature sensors, android devices, and SIM-Cards with mobile data for the application to function. Because of these requirements, the only thing missing for it to be production-ready is a stable and generic application version.

When conducting the pilot, in addition to introducing new technologies, will the pilot impact the facility workers' daily work process. Therefore, it is essential to maintain the standard procedure and the new work process the technology introduces so that the reporting to the immunization program does not stop during the pilot.

If this research and pilot discover the need for such technology, there is a possibility that the application could be a way for DHIS2 to support temperature CCM.

7.2.1 The sixth development iteration:

Introduction
This additional development iteration accrued based on the needs presented by the HISP group. There was a need to use the server of the Mozambique government. In addition, there was a need to refactor the app so that a health facility could have more than one TEIs to upload data to - something I had not tested during development in Oslo. Moreover, there was a need to translate the entire application into Portuguese.

Iteration development

- Refactored to Mozambique government DHIS2 server.
- Implementing the possibility of uploading to multiple TEI in an Org unit.
- Refactoring the entire application to Portuguese.

Development
Although this iteration was a bit unexpected, it makes sense to refactor the applications to adapt to the environment in which they will be piloted. The first week we already had updated the DHIS2 government servers at the ministry of health. When logging in with the new server credentials, the applications stopped working, and access was not granted. After some troubleshooting, I contacted a representative from the Android SDK DHIS2 team. After some guidance, it turned out that I was using an old DHIS2 SDK version. The application worked as intended after updating the SDK to the latest version.

The next step in the development iteration was to change the applications so that it could upload temperature data from different sensors to multiple TEIs at an organization unit. The final solution was tailored to the pilot, but due to little time for improvement, it was impossible to create this generic functionality. The problem was solved by using the mac address of the sensors so that one sensor represented one TEI. Lastly, the application was translated using Google translate and help from the DHIS implementer team.

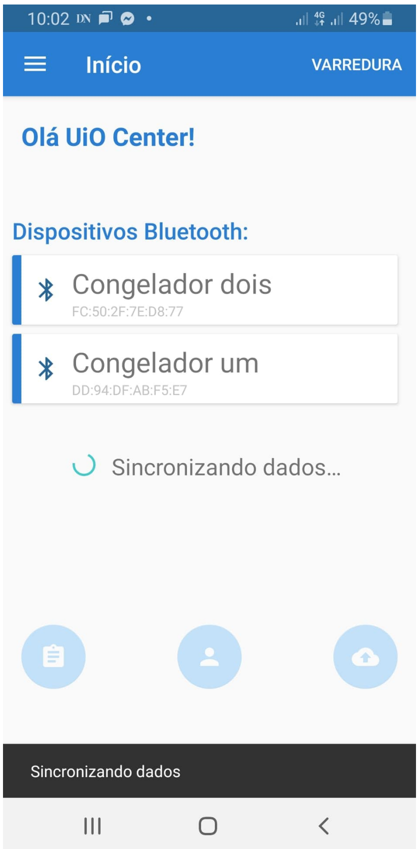

Figure 25: Translated device list

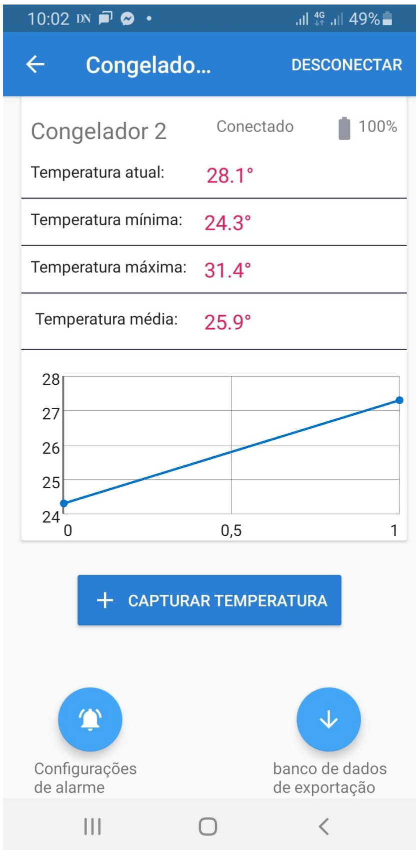

Figure 26: Translated and modified device view

7.3 Week two - piloting in Quelimane

After the first week of planning and conducting an additional development iteration, the application was ready to be piloted. The pilot took place in Quelimane city in the Zambezia province for over 14 days. The goal was to implement the application at four health facilities, gathering temperature data and qualitative data regarding the existing CCM processes and their need for CCM. The CCM Application would run on a Lenovo tablet with a sim-card providing mobile data so that health facility workers could upload temperature readings directly to DHIS2. To prevent temperature data from ending in the wrong TEI, each health facility had a different application version to match their organization unit in DHIS2.

In order to implement the application and access the health facility, we had to get permission from DPS Quelimane. DPS is the statistics department in charge of the district level. In collaboration with representatives from the Saudigitus team, we held a live demo of the application to the head of DPS and the head of statistics. They raised some concerns about android devices and sensors being damaged or theft. Nevertheless, the visit was a success, and we were granted access to the pilot at the facilities we requested. In addition, they wanted to implement the solution right away at 174 of the 276 health facilities in the province of Zambezia.

Visiting facility A

The next day, two representatives from DPS, Saudigitus and I, visited three of the facilities. The first facility, let us call it facility A, was the largest and most central of the health facilities in the pilot. The facility manager was informed about our visit, and we connected with the health facility worker responsible for the CCM. Facility A had two CCE fridges, one of the type TCW80 and the other a "domestic fridge."

After inspecting their CCE, we demoed the application to the health facility workers. We explained the use case and how the application could prevent giving spoiled vaccines to patients and demonstrated that temperature data from the sensors was sent directly to DHIS2. The facility worker in charge of the vaccines immediately stated that they did not need the application because they already had a digital thermostat built into the fridge.

After the demo at facility A, we had a semi-structured interview with the health facility worker responsible for the CCM regarding their current work procedures. They report the temperature on a piece of paper twice a day, the first reading in the morning and the second reading in the evening before going home. The paper repented for a year which was reported to the immunization program at the end of the year. After reading temperatures in the morning, the vaccines estimated to be used that day were taken outside the vaccination station in a vaccine carrier. Most of the time, the vaccine carrier had an ice pack to keep vaccines at the correct temperature, but they did not have a thermostat, and if there were any vaccines left after vaccination, they were put back into the CCE fridge.

In this process, there is no way of telling the current temperature in the vaccine carrier, and the readings in the morning and evening did not reflect the temperature of the vaccines that had been taken out. In addition, there was no routine to track which vaccines had been out of the CCE multiple times or which vaccines were getting expired.

Visiting facility C
The next facility we visited that day was the rural facility farthest from the city. Because of the ongoing cyclone, which had ravaged since February, we could not reach the facility by car - so we had to walk the last part. Walking to the facility, we saw the outcome of the cyclone. The houses were empty, people had evacuated, and the streets filled with water.

Facility C was smaller than facility A and mainly used paper forms to capture data, as they were unfamiliar with digital devices. After demonstrating the application, the interest was more optimistic than at facility A.

We healed the same interview to identify existing work procedures at this facility. They reported the same procedure as facility A regarding temperature CCM, and they had one fridge of the type TCW80. One difference in the CCM from facility A was that facility C vaccination station was inside, right next to the fridge. Nevertheless, they followed the same process of taking out the estimated daily need into a vaccine carrier instead of taking out bulks of vaccines based on demand from the fridge.

Visiting facility B
On the way back to the city, we visited facility B. It was a medium-sized facility compared with the others. They had two fridges, one of the type TCW80-AC and the main fridge a Dometic MK304. They also conducted the same work process as the other facilities. After the demo, they had a positive attitude towards the pilot and saw an opportunity of using the application in their work during vaccination at rural locations.

When they travel with vaccines to rural locations, they also bring vaccine carriers with ice packs to keep the vaccines at the correct temperature. Nevertheless, they do not have a routine of bringing a thermostat to the vaccine carriers.

The ice for the vaccine carriers is produced in their Dometic fridge. As the intention to produce ice is to keep vaccines at a low temperature when traveling, the result is that all the vaccines stored in that fridge are stored below zero degrees - destroying all the vaccines.

Such as other facilities, facility B reported that they never had any cases of spoiled vaccines or had thrown vaccines because of power outages or other issues.

Figure 27: Pilot health facilities

Challenges during the second week
Before leaving Maputo, there were some issues with the Lenovo tablets provided. Since we could not find new devices in a week, I tried to fix the problem during the pilot. The Lenovo tablet ran on Android version 9, with Bluetooth version 4.2. The recommended Android version from the DHIS2 SDK team is Android 10. In addition, BlueMaestro sensors support the newest Bluetooth version, 5.0, which is twice as fast as 4.2.

Another challenge during the second week was the ongoing cyclone. It had destroyed the bridge to the neighboring city where the last health facility was located. We, therefore, had to adapt the pilot by excluding the last health facility.

7.4 Last week piloting in Quelimane

In the last week of the pilot, we took the same route when visiting the facilities. Because of the issues with the Lenovo tablets, we had to adjust the pilot to using health workers' devices and our own. More data were gathered regarding existing work processes, fixing technical bugs, and asking about their experience in using the application.

Facility A
Facility A got a Lenovo tablet with a purple Bluemaestro sensor, which worked to some extent. The purple sensor transmitted temperature data faster than the red BlueMaestro sensors. The application only captured the current temperature with this setup, and all other functionality worked as intended.

On the second reporting day, facility A had some problems with the application. After fixing the bug and reinstalling the application on Lenovo, we asked if they would download it on their personal devices. The facility workers agreed and preferred it. Utilizing the facility workers' device eliminated all technical issues regarding the application.

During the troubleshooting, we conducted a test by placing a sensor in the vaccine carrier during vaccination. The temperature showed 1.3 degrees Celsius inside the vaccine carrier, and the temperature outside in the shadow was 36 degrees Celsius. After one hour, the temperature increased by 0.6 degrees Celsius to 1.9 degrees.

We conducted the test in the morning, two hours after the vaccines were taken out of the fridge. This tells us that the environment in the vaccine carrier initially was too cold, as the ideal threshold is between 2-8 degrees Celsius.

Facility B

Implementing the application at facility B, we provided health facility workers with my personal Android device, as this facility was most engaged in participating in the pilot. The Android device ran on android 10 with Bluetooth 5.0 and was the device the CCM application was tested on during development in Oslo. When providing the health facility worker with the device, he had to use another device as they already had four different Android devices.

After some days, they faced the same issues with the application. Facility workers reported that the application sometimes shuts down when trying to upload to the DHIS2 server. This was due to the DHIS2 servers timing out because of the unstable internet connection at the facility. After fixing the bug, we stressed the importance of not stopping to capture the temperature so they could upload temperatures when connected. In addition, could the readings sent to DHIS2 be compared with the CVS export from the local database to see if some temperature readings got lost.

Figure 28: One of the pilot health facilities

Facility C
Revisiting health facility C, we ran out of working devices and depended on health workers downloading the application on their personal devices. As facility B reported, health workers ta facility C also preferred to use their personal devices. The facility worker at facility C ran the application on an android ten without any problems. We then trained the facility workers once more to assure that they knew the application well enough. Before leaving, they requested more sensors for vaccination at rural locations, just like facilities A and B requested.

Figure 29: Common vaccination station

Polio campaign
A polio campaign took place in the last week of our time in Quelimane. The campaign goal was to vaccinate children in a particular age group against polio, and vaccination was done by going door to door and visiting rural locations.

Facility A invited us to participate in the campaign using the application to conduct CCM. As we went from house to house, after vaccination, the children got their pinkie fingers dipped in ink as a confirmation. In addition, the health facility workers wrote a code on the wall representing the data and how many children were vaccinated in that household.

The vaccination team we observed worked within a radius of 2 kilometers from health facility A. They explained that the initiative for the campaign was due to the problem that people did not take the time to go to the nearest facility to get their children vaccinated. However, there were also vaccination teams covering rural areas way further from a health facility that relays on the campaigns and vaccination visits to get vaccinated.

During the campaign, health facility workers used the application to capture temperature from vaccine carriers. After the campaign, we got feedback from the health workers that they saw the potential of the application and wanted more sensors to be able to use the application when vaccinating in rural locations.

Figure 30: Attending a Polio campaign.

Challenges
During the last week of the pilot period, we had the opportunity to research the application in use at health facilities and attend a polio campaign. During this period, we had to adapt several times, as we faced some challenges and discovered different findings that will be discussed in the findings chapter.

One major challenge we faced during the piloting was the internet connectivity, as this was hard to test in advance of the pilot. There were different telecom vendors, mainly Movitel and Vodacom, and the stability and connection strength varied. This made some issues regarding the uploading of temperature data.

Another infrastructure challenge at the health facilities was that all the different thermostats showed different temperatures. The old non-digital thermostats were inaccurate, and the digital thermostats were often out of battery. In addition, the built-in digital thermostats on the TCW 80 AC always show a different temperature than the BlueMaestro sensor, creating confusion about which temperature measurement one should rely upon. Furthermore, a challenge when informing health facilities workers that the temperature was non-optimal as all thermostats were showing different temperatures. In addition, this led to another challenge from a research point of view where it was difficult to observe during the pilot without giving the impression that we were supervising.

7.5 Last week - finishing the pilot

During the last time in Quelimane, we collected additional data regarding the health facility workers' experience being a part of the pilot and using the application. In addition, we collected my Android device from facility B and uploaded the application to the health facility worker responsible for CCM.

After traveling back to Maputo, the finding and work from the pilot were presented to the Saudigitus team. We then further discussed the future for the application, if Saudigitus should take over the maintenance, and further development of making the application generic enough to scale. Alternatively, if the application should be implemented as a part of the DHIS2 core, developed and maintained by the Android DHIS development team. To be rolled out in other parts of Mozambique and neighboring countries.

To prevent multiple DHIS2 stakeholders from working on the same solution, we decided that the DHIS Android development team should take the solution to make it more generic - as they were looking for a dedicated LMIS developer to take over the project.

Chapter eight

8 Main Application functionality

The following chapter presents the artifact, the custom CCM applications, and its functionality. Furthermore, the application's different components, how they function, and their design choices are discussed.

8.0.1 Login

When opening the application, the first activity the user is granted is the login activity. This activity lets the end-user login into a DHIS2 instance. The activity greets the user with a DHIS2 splash page before presenting a login form. In order to authenticate, the DHIS2 instance has to be active. The user then authenticates by passing in the URL of the DHIS2 instance, username, and password. The authentication functionality then checks user credentials against the DHIS2 instance server. This authentication is necessary for the DHIS2 SDK to access the different programs related to the instance so that the SDK can get access and synchronize the data and metadata related to the program. Accessing the right program is also necessary for the application to be able to upload temperature data as events in a tracked entity instance.

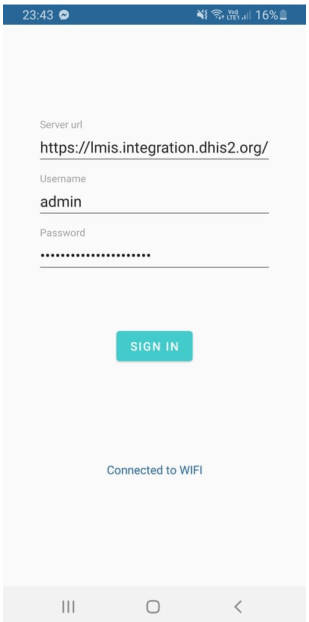

Figure 31: Login view

8.0.2 Main view

After being granted access to the application by login into a DHIS2 instance, the user gets presented with the application's main view. The main view functionality is to give an overview of the different sensors available in the range of the Android device by scanning for BLE devices. The first time using the application, the main view asks for application permissions, like location and Bluetooth. Providing the location is essential for scanning after BLE devices. Furthermore, the android device must have Bluetooth turned on to scan and communicate with BLE devices.

After the user grants permissions to location and Bluetooth, the application scans for BLE devices and continuously displays them in a list view as they are discovered. The items in the list view represent a BLE device and display the device name and MAC address for each blue maestro sensor. This design choice was taken so that the application users can differentiate between multiple sensors by matching the list item and the MAC address on the backside of the sensors. Furthermore, if the user is out of range and no devices are found, a scan button in the top right corner makes the application re-scan for BLE devices. In addition, in the top left corner, there is a menu with the functionality of clearing the locally stored program data and logging out of the DHIS2 instance.

In addition to displaying the different BLE devices as items in a list view, the application makes a temporary connection with this sensor using the GATT service by selecting one of the items. It then automatically sends commands through the UART protocol to get relevant temperature data from the sensor. At the same time, the main view sends the user to an activity that displays the response from the selected BLE device to the user.

The other important part of the main view is downloading and synchronizing data from the DHIS2 instance through the SDK. This step must be done before collecting data from the sensors for the application to access a program, organization unit, and TEI. Downloading and synchronizing data does not have to be done every time, as the data gets stored locally on the device. After the applications are synchronized and temperature data captured, An event gets stored on a TEI and locally stored in the database. Lastly, after storing the collected temperature data, the main view allows uploading the changes to the DHIS2 instance. This is done by uploading the TEI stored locally in the device with its respectively new temperature events to the correct organization unit in the tracker program.

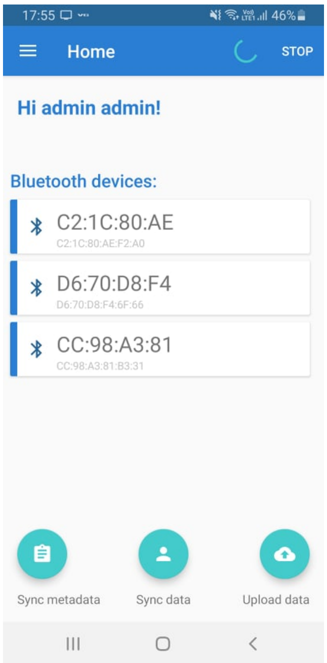

Figure 32: Main application view

Figure 33: Bluetooth device list

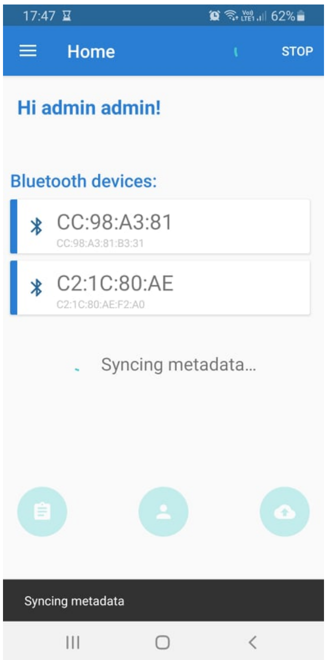

Figure 34: Synchronizing metadata

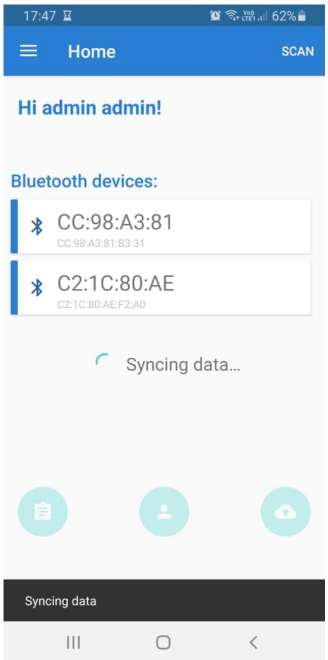

Figure 35: Synchronizing data

8.0.3 BLE device view

When a user selects a BLE device from the scanner list in the main view, the android device makes a temporary connection to the selected BLE device. As mentioned, the application subscribes to the selected sensor via the GATT service, making communication possible through the UART protocol. Furthermore, the application sends the user to an activity for that specific sensor - the BLE device view.

The first function in the BLE device view sends UART commands to the sensor. The command sent from the application to the sensors works as a callback function. The application asks the sensor for data; if it gets the command, it returns the relevant data. If the command is not recognized or valid, the sensor returns an error status code. The following function in the BLE device view takes the sensor response and pares it from a 16-bit array to string values. Lastly, the UI gets populated with the sensor responses like; various temperature readings and other relevant information, including the sensor's battery level and name. If the sensor returns an error status code, the command function will keep looping for some time, asking for temperature data before it stops.

The BLE device view also provides additional functionality in displaying the values from the BlueMaestro sensors. The activity's most important button is the upload button. If the sensor has returned temperature data, pressing the upload button captures temperature data readings and stores them in a local SQL-lite database. Such a temperature reading consists of the sensor's name, timestamp, the current temperature, last 24-hour average, last 24-hour minimum, and 24-hour maximum temperature.

The local database plays an essential role in the application's functionality. The temperature readings stored in the local database are used to make events that get stored on a TEI. These TEI gets stored locally in the application and can be uploaded to DHIS2 tracker capture through the main view when the Android device has an internet connection. If the Android device has an internet connection, these events are automatically made and ready to be uploaded to DHIS2 tracker capture through the upload button in the main view.

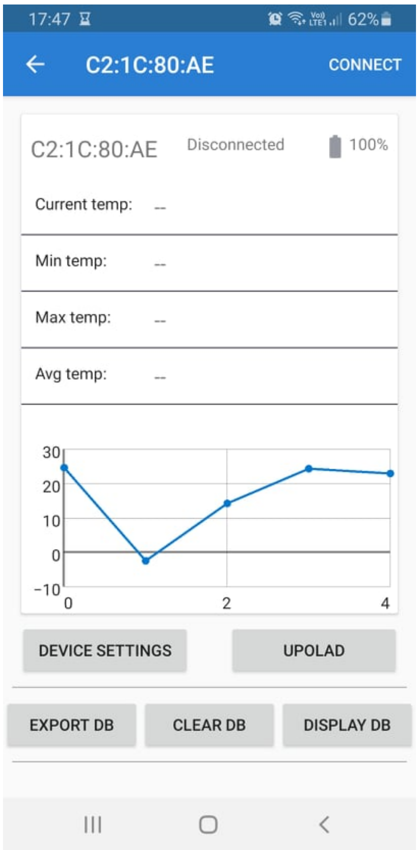

Figure 36: Before connection

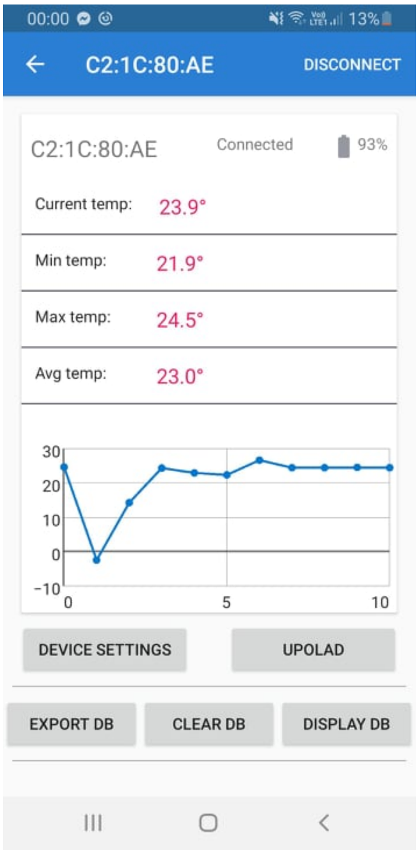

Figure 37: After connection

8.0.4 Additional functionality

The data stored in the local database are in addition to making events used for providing different types of helpful additional functionality to the application.

One such additional functionality is the line graph displayed in the BLE device view, representing the historical temperature captured from the application. The graph displays the number of measurements taken and the current temperatures when the measurement was uploaded, making a visual representation of the database content. In addition, it efficiently presents the CCM status over time to those who manage and collect temperature readings locally on the Android device. This is helpful as CCM workers do not need access to DHIS2, and managers do not need to check DHIS2's tracker capture to see aggregated data readings.

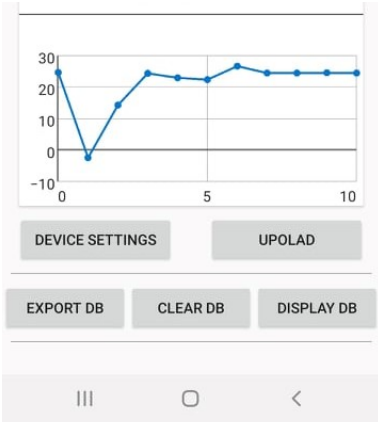

Figure 38: Additional functionality

In addition, do the BLE device view provide an application requirement in the necessary alarm notification functionality. The application users can set a temperature alarm threshold for the selected sensor by pressing the device settings button. The alarm notification then uses the provided maximum and minimum temperature to check against the current temperature. Suppose the temperature data collected from the sensor rises above the maximum temperature or below the minimum temperature. In that case, a local notification is displayed on the Android device that the temperature is too high or low. The alarm functionality works individually on each device. From this thesis scope - the alarm functionality is a work in progress as it is much work to fulfill the application alarm requirements. The alarm functionality intends to alert health facility workers when they need to correct the CCM situations. Providing a minimum and maximum threshold gives application users flexibility when the alarm should be set off so that one can act before temperature levels are critical.

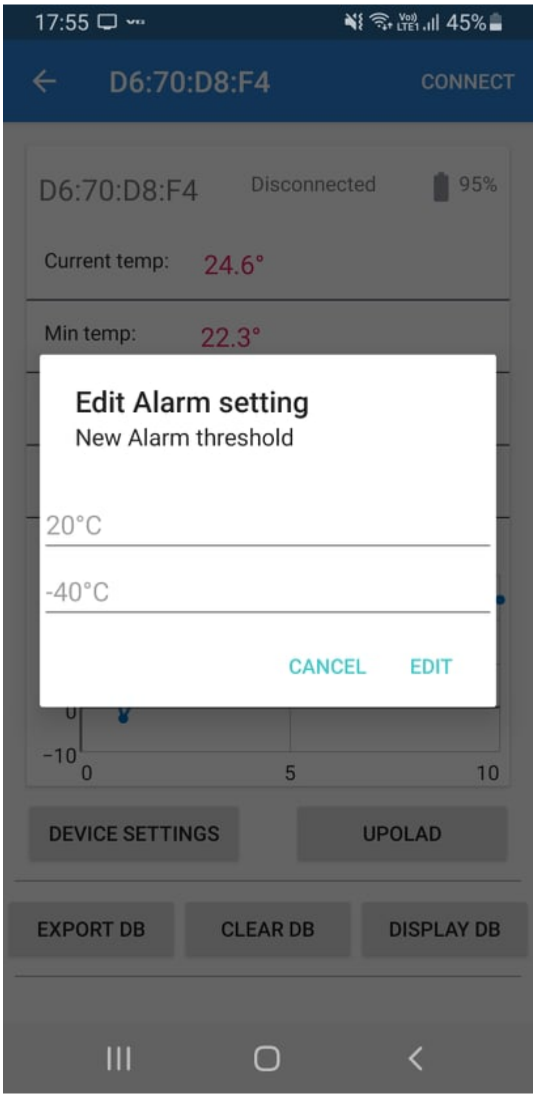

Figure 39: Alert notification

During the application development, there was a need to expose the database to have control of its content while developing - this need resulted in the display button. Pressing the display button, the entire database contents is displayed inside a modal. It displays all the information stored in the database table. This was useful for testing purposes, for instance, to check the database readings with the line graph.

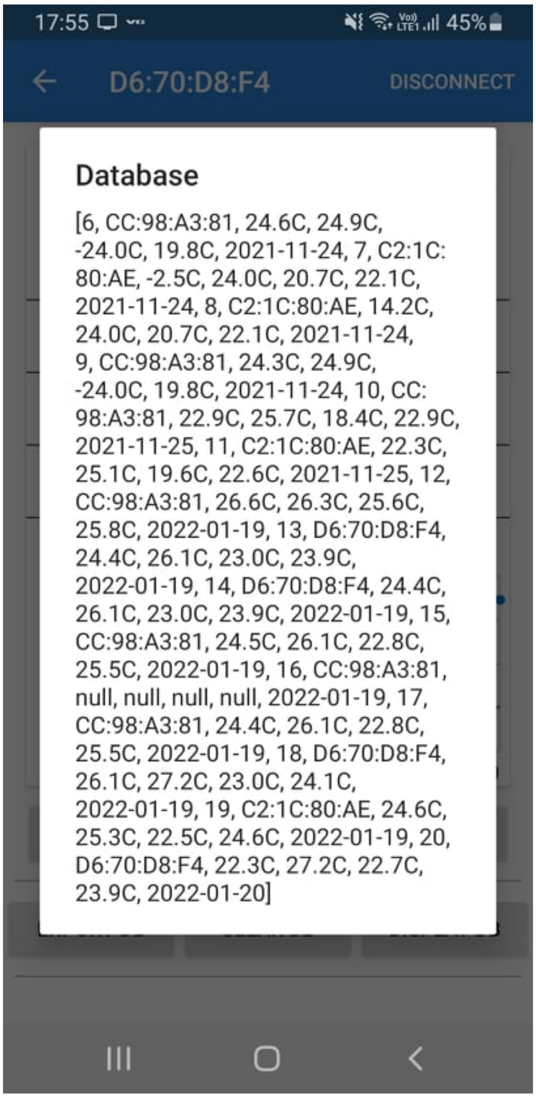

Figure 40: Display DB

In addition, to display the entire database, there was a need for having the possibility to clear the entire database. From a test point of view, this was useful to further test the application after applying changes. Initially, I thought that the functionality of displaying and deleting the local database was useful for users. Clearing the database functionality could be helpful if routine workers only want to work with relevant data for a specific period. In addition, routine workers can adopt clearing the database into their work process to prevent the local database from overflowing. For instance, clearing the database after a batch of temperatures is uploaded to DHIS2.

Later in the development process, I decided to remove the options for end-users to display and delete the local database to constrain the end-users and hopefully remove the possibility of other user errors. The modal view displaying the database became challenging to read as the database grew. Therefore, exporting the local database as a CSV file was developed. When the user presses export, the database table gets formatted into a CSV file. When the formatting is done, the file gets downloaded locally on the android device. After the download, a success notification is displayed on the device. By pressing the notification, the user gets redirected to the file destination of the Android device where the CSV file is stored.

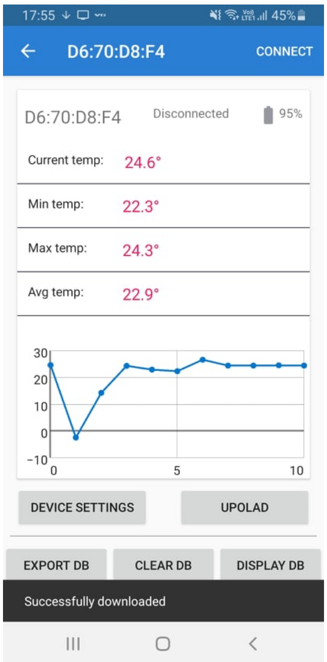

Figure 41: Download

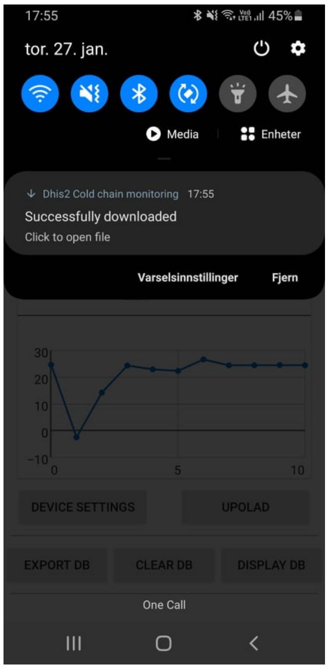

Figure 42: Download notification

The export functionality is intended to make the database more readable for the end-user. However, it can also act as a backup of collected temperature data. Suppose any problems arise while uploading data to DHIS2 servers, for example, due to an unstable internet connection. In that case, the application user can export all the data as a CSV file and upload it manually to DHIS2.

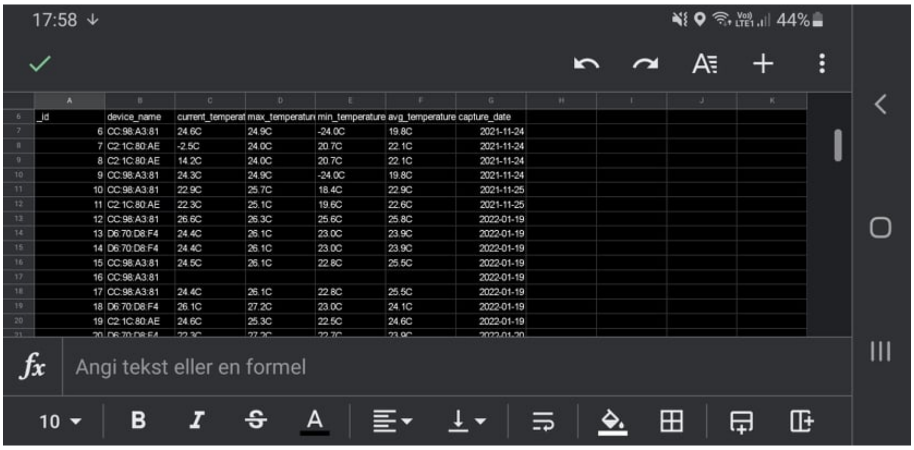

Figure 43: CSV file from local database

8.0.5 Internet connectivity

The DHIS2 Android SDK is a toolbox with various tools and functionality to develop Android applications in the DHIS2 platform ecosystem. In the CCM application, the SDK enables the uploading of temperature data to DHIS2's servers. To conduct the uploading to DHIS2, the application, at some point, needs an internet connection. Because the application use case is being implemented in resource-constrained settings, it must function without an internet connection.

The happy day scenario of the application starts with the Android device having an internet connection. Application users can then log in, synchronize with the DHIS2 instance, and collect temperature data immediately. When sufficient data is collected, the application user can then upload the collected data to the DHIS2 server.

As for the piloting of the application, the happy day scenario is often not the case. Therefore the application is designed to handle the different internet connectivity issues.

Suppose the application is downloaded to an Android device, and the user opens the application for the first time without an internet connection. In that case, it will skip the login activity and go straight to the main view. The main view has a button group for synchronizing and uploading to DHIS2 gets disabled. Nevertheless, the initial functionality remains. The user can still scan for devices, select a specific device, and collect temperature data from the device view, storing it in the local database. When the application connects to the internet, the user gets prompted with the login view. By logging in, the button group enables and the user gets the ability to synchronize with DHIS2. After synchronizing, the upload button gets enabled, and the user can upload the collected data from the local database into DHIS2.

Since the application utilizes DHIS2s Android SDK, application users can log in and synchronize while connected to the internet after downloading the application for the first time to their Android device. The DHIS2 TEI will then be stored locally on the device, and the application will function as excepted when brought to a place without an internet connection. An example of such a scenario could be that the user has internet access while at a facility but loses connectivity when capturing temperature data in rural areas. After collecting temperatures in offline mode, when the application is reconnected to the internet, users get access to upload the collected temperature data to DHIS2. The screenshot below displays the login and main view when the Android device has no internet connection.

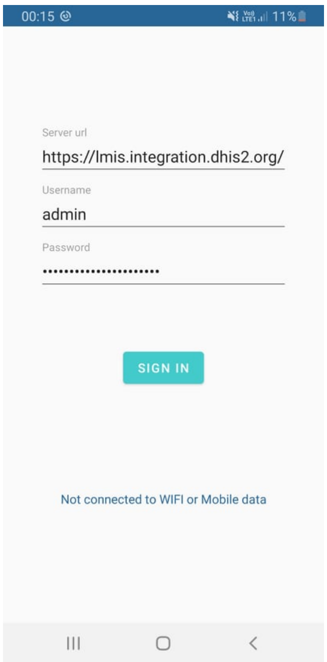

Figure 44: Login view offline

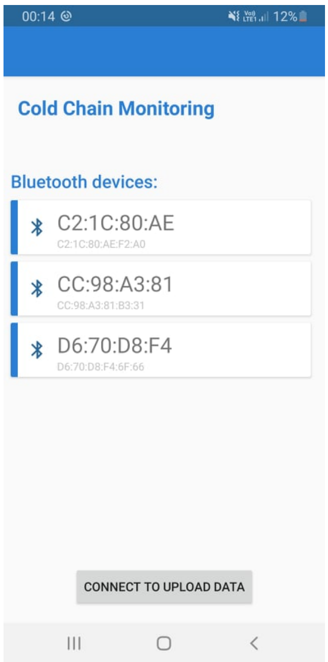

Figure 45: Main view offline

Chapter nine

9 Findings

Multiple findings were addressed during the CCM application's development, implementation, and piloting. This chapter presents these technical findings related to the development and implementation of the application. Furthermore, social findings related to implementing and piloting the applications in Mozambique.

9.1 Findings from the development process

During the development iterations in Oslo, several technical challenges were encountered. These challenges often increased the length of the development iterations or led to a downgrading of desired functionality.

The first technical challenge was discovered in the second development iteration, where the main goal was to communicate with the Bluetooth sensors. Understanding the difference between Bluetooth Classic and Bluetooth Low Energy was vital in developing the application. The finding was all the different BLE communication protocols, hence the specific chipset of the BlueMaestro sensor. I had to use the UART Protocol with a GATT service using a specific UUID for the BlueMaestro sensors. I classify these as technical findings because there was limited information on these topics and how to work with BlueMaestro sensors. This information gathering led to the finding that the BlueMaestro SDK was deprecated and not functioning. In addition, the technical finding was that not all the UART commands provided by the BlueMaestro API documentation were functioning, resulting in an increased length of the development iteration.

For instance, the *logger commands were supposed to return the relevant log for temperature, humidity, and dew point. This log would contain temperature values stored in the sensor's memory, captured at a given interval. Since this command did not function with the application, I had to retrieve current temperature readings and store them in a local database.

Using the *logger command is essential for further development; the sensor's logging interval is, by default, ten minutes. By utilizing this feature, one can access temperature readings from the BlueMaestro sensors memory. The application could retrieve all temperatures stored in the sensor's memory instead of manually storing the current temperature. Furthermore, be closer to supporting the requirement of real-time CCM.

As this finding could meet real-time CCMs support in DHIS2, it leads to another technical finding regarding storage. After meeting with representatives from the DHIS2 core team, the storage issues using tracker capture were addressed. If the application was to be implemented on a large scale, storage could quickly become an issue. By storing every temperature reading as an event on a tracked entity instance, the DHIS2 instance would quickly run out of storage. One example would be if every temperature reading were captured every ten minutes - this would result in a total of 144 temperatures stored as an event on the tracker capture every day. After one year, one TEI would have 51 840 events stored. Moreover, if the solution were to be scaled to the 174 facilities in Zambia, this would result in 9 020 160 stored events a year, given that all the facilities have one refrigerator reporting every 10 minutes.

The storage problem can be solved in different ways, like passing aggregated values like the Mean kinetic temperature (MKT), which will be discussed in the chapter presenting future work. Furthermore, utilizing cloud-based solutions could deal with a future storage issue but would be an expensive solution.

Another technical finding identified during the pilot was due to a technical bug. During the pilot period, the login functionality suddenly stopped working. It was due to the application running a one-year-old DHIS2 SDK version. After upgrading the SDK versions, the applications worked as intended. This finding shows the importance of minimizing dependencies and keeping the application updated. Nevertheless, the Android SDK was a reliable development tool, and I faced minimal challenges developing with the SDK not functioning as promised. The documentation, the Skeleton application, and the support from the Android development community were sufficient to implement the DHIS2 part of the application.

In addition, interesting technical findings were discovered during facility visits. A sudden discovery was that the application ran as intended with Android version 8 on one of the facility workers' personal devices. Hence, having problems with implementing the application on devices below Android version 10. The previous assumption that the applications required Bluetooth 5.0 and Android version 10 or greater was disproved. Furthermore, the reason the application functions differently on a mobile device and the tablets provided is still unknown. Nevertheless, this finding most likely lowers the application's Bluetooth and Android version requirements, which could potentially result in more end users.

Lastly, the finding of the Nordic toolbox was critical to understanding how to interact with sensor technology and providing an overview of how to communicate over the UART protocol. Working with BLE has been a big challenge, and understanding the different levels of the GATT service and UART protocol has been the key to doing the application work. As for the BlueMaestro sensors, they have functioned reliably during the entire development- and piloting period.

9.2 Findings from piloting

In this section, I present the findings made during the pilot work. These findings have an impact beyond the purely technical implementation and cover work processes, infrastructural challenges, and stories told regarding challenges the facilities face. Through semi-structured interviews, we asked about routines for handling spoiled vaccines. All facilities reported that they never had cases where vaccines had to be discarded due to not maintaining the correct temperature or expired. During the pilot, we had findings disproving this claim.

Findings at health facilities

One such story was from the facilities we visited. Previously they had their

CCE in the maternity ward due to a lack of space. The facility reported cases where the socket for their "domestic fridge" was pulled out by patients to charge their phones. Because they lacked electrical sockets, this became a problem, with an outcome that could damage all vaccines in the refrigerator. The facility had to move that fridge to the delivery room, which was not an ideal space. Instead, maybe an extension cord could solve the problem. Given the facility's current work process on CCM, they checked the fridge in the morning and end of the day. The fridge could have been unplugged during the day and night without any knowing, leading to vaccines being damaged in the fridge.

Another finding of non-optimal vaccine storage was after implementation at one of the facilities. The temperature reported through the application showed a temperature as low as below zero degrees. At our next visit, the facility worker explained that this was because they produced ice packs for the vaccine carriers they used at their vaccination station. Most vaccines stored at health facilities have a one-month storage period if stored between 2-8 degrees. In cases where vaccines freeze and defrost, the effect reduces drastically. For instance Hepatitis B vaccines are immediately destroyed if they freeze [67, p.6-10].

Our observations contradict what had been told us about correct temperature maintenance. One can imagine that this is due to the Hawthorn effect. The interviewees change their behavior because of their awareness of evaluation as participants in the pilot [37]. The balance between not giving the impression that we inspected but observing was subtle. The reason for believing that the hawthorn effect was present was, to some extent, based on other stories we heard during facility visits. Cases where Vaccine Vial Monitor(VVM) markings beyond the discard point were used or put back to the CCE to prevent it from being marked as spoiled. In addition to stories about patients being referred to private pharmacies nearby, which bought vaccine stock from the public facilities.

Nevertheless, based on these findings, more transparency and automatic CCM reporting could prevent temperature data from being manipulated. We also experienced some difficulties regarding surveillance that supports the suspicion of the Hawthorn effect. During the polio campaign, health workers avoided reporting temperatures in cases where the temperature was unacceptable. The reason was that the facility did not have enough ice bags for all the cold boxes. By being aware of our access to the data they upload, facility workers avoided reporting. Therefore, we did not collect data proving that vaccines had maintained the wrong temperature. This data is the most important to collect to prevent vaccination with spoiled vaccines and to argue the need for more cooling elements for their campaign work.

In addition, one of the concerns was mentioned by the head of DPS Quelimane before the pilot arose. One sensor disappeared after the equipment was returned to the health facility at the end of the campaign. As the sensors are small, they can easily fall out of the vaccine carrier when stacked for storage - or disappear as they are moved between CCE.

During the facility visit, we also experienced different degrees of internet connectivity, which was one of the biggest challenges to the application's performance. Multiple Telecom towers were only a hundred meters across the road at the facility with the most unstable internet connection. Nevertheless, the connection was almost non-existing. We were told that the towers had a different provider than the sim card we used on the devices. Such a finding reflects the problem of infrastructure being developed but not maintained. This can also be compared with the effort of this master thesis. The CCM application's technology and implementation will not improve the CCM if not the health facilities are being trained, followed up, and gets additional knowledge about the importance of CCM.

CCM experiment

Lastly, additional findings regarding the vaccination workflow were discovered during one of the facility visits after conducting a small experiment. Chapter six presents health facility workers packing vaccine carriers in the morning before taking them to the vaccination station. When we arrived in the morning, health facility workers had packed the vaccine carriers and had them at the vaccination station for two hours. When we arrived, the vaccine carrier had an internal temperature of 0.6 degrees Celsius. After one hour, the internal temperature was one degree Celsius, increasing to 0.4 degrees. At this point, the experiment had to end because we had to visit the next facility. Nevertheless, this was an exciting finding.

The experiment shows that the vaccine carrier was packed with ice packs resulting in a non-optimal storage environment, and the internal temperature of the vaccine carrier increased the risk of damaging the vaccines. In order to prevent this, the vaccine carrier should have been packed with water packs, which can be achieved by letting the ice packs melt to the point where liquid water surrounds the ice when shaking the pack [19, p.1121].

The graph below is a tentative representation of the temperature increase. This shows that trained facility workers, CCM knowledge, and changes in the work process could prevent such cases where the initial temperature increases the risk of damaging the vaccines [67]. Hence, the data set is far from complete, not considering additional variables like the degrees in the air and melting of the ice pack.

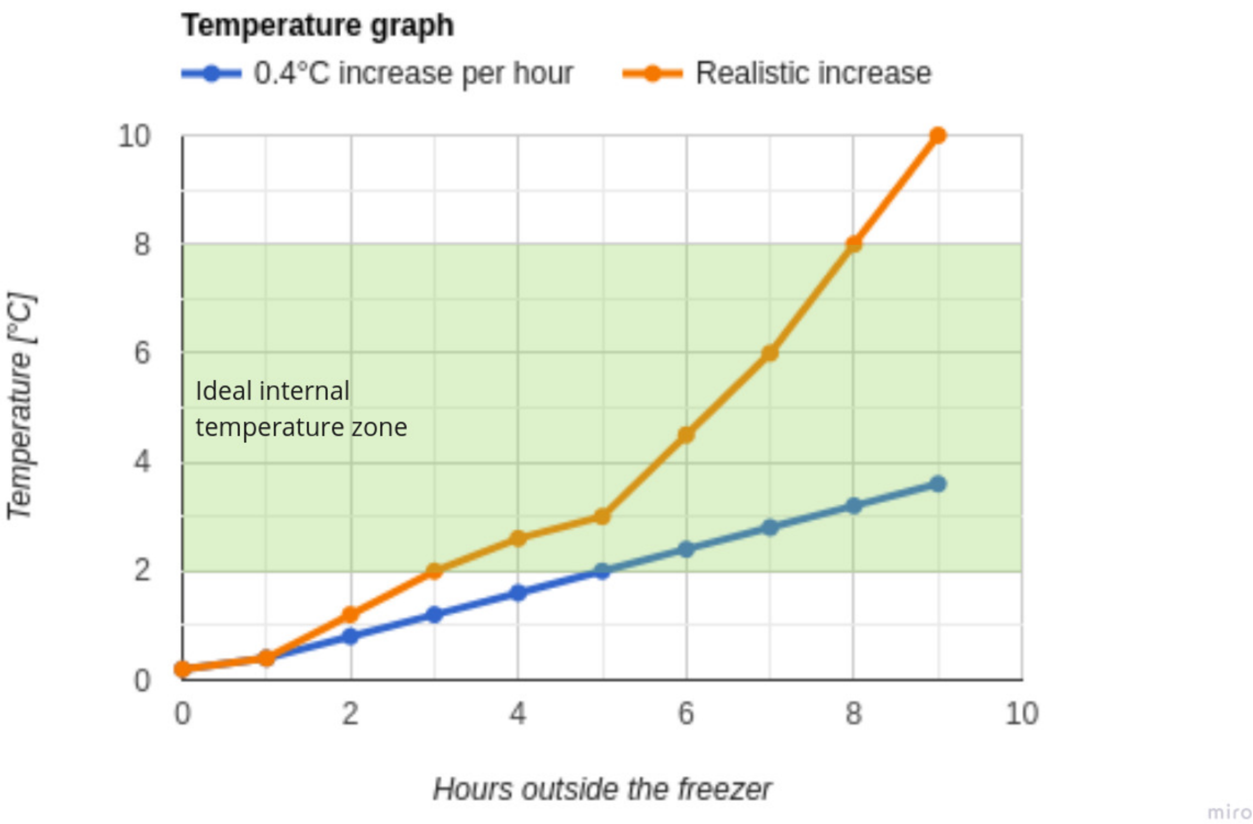

Figure 46: Tentative temperature increase graph

Chapter ten

10 Discussion

This chapter discusses the challenges and opportunities related to the CCM app's pilot, development, and implementation. Furthermore, it discusses the approaches and methodologies used in this thesis - with the research question in mind.

> " How can real-time cold chain temperature monitoring be supported in DHIS2."

10.1 CCM Application review

The application of this thesis was customized to fit the needs and conditions for running a pilot in Mozambique. DHIS2's innovation platforms architecture facilitates both stability and flexibility at the same time [51, p.27]. The Platform Core provides stable functionality by offering BR-like APIs, SDK, Academy training, and documentation. This BR facilitates cultivating platform ecosystems through third-party development. When developers use these resources in their application, the BRs extend the scope and diversity of the platform [16, p.187].

Because of DHIS2s platform architecture, CCM can be supported through developing applications in the outer layer-based and loosely coupled to the platform core [51, p.24]. This approach will still facilitate a resilient platform due to custom applications not affecting the platform core [62, p.94].

Utilizing DHIS2 BRs to develop standalone applications had different challenges and opportunities. The platform needs to balance the tensions between supporting third-party development through BR and controlling the platform development [16, p.187].

As a third-party developer, it was challenging to understand the DHIS2 universe, read the documentation and get an overview of the API structure. On the other side, without the BRs, the opportunity to develop the application would be challenging. Utilizing the BRs by attending the Android academy, forking the Skeleton app, using the Android SDK, and getting support to set up a test instance with a CCM program was crucial for the success of the app development.

The result of the development is a specific standalone application supporting CCM to DHIS2. Findings from the pilot show that changing minor aspects of the application would make it generic enough to scale at many health facilities. Also, the Bluetooth communication part of the code is generic and can be implemented as a stable part of the DHIS2 core.

10.1.1 Potential

Cold chain monitoring's primary purpose is to prevent people from being incorrectly vaccinated and vaccines being spoiled [24]. After testing the application and based on finding from the implementation, there is a potential for health facility workers to have better CCM at their facility with the opportunity to act and prevent incorrect vaccination immediately - by using the CCM application as a supplement to facility workers' current CCM routines. Since the application is on an android device and the sensors are small and durable, the solution can be implemented in different parts of the Cold Chain, for instance, to capture and report vaccine temperatures during transport [4].

The application can also improve CCM at a temporary vaccination station established in a rural area by health facility workers. The cheap sensor technology has little maintenance and changeable batteries. This is an advantage over other non-sensor-based expensive CCE with high maintenance [25, p.2221].

During the pilot, the potential for the application was discussed. The statistics department at DPS highlighted the reporting of temperature data to decision-makers at the regional and national levels. They saw the potential in aggregating data across different relational use cases: an example, spoiled vaccines at a facility and their vaccine stock.

Based on evaluation feedback from the LMIS team and feedback during the piloting, the CCM application also has potential for improvement. One key feature to improve in the application is the alerts. By further developing the alerts to send SMS notifications to different stakeholders, health facilities can get improved CCM when the facility is not staffed, for example, during the weekend. Reporting alerts from the health facilities to DHIS2 would provide valuable data to decision-makers. The number of alerts can be used to notify facilities that they have spoiled vaccines in stock. In addition, the data can be used in combination with facility stock to distribute vaccine shipments by prioritizing facilities with the greatest need.

After the pilot period, some potential for the applications was revealed. For the application to work stably and make a more significant impact at multiple health facilities, there is a need for the applications to be more generic. The DPS wanted to scale the application to all of the health facilities throughout the province. In order to support such a scale-up, the application would need to provide a configuration so that each facility could link their CCM application with their organization unit in the DHIS2 tracker application.

The evaluation meetings with the LMIS team also addressed the potential of a fully automated CCM process by improving the application to capture temperature data automatically when in range of sensors. Automation can potentially support new use cases, such as collecting data from multiple sensors more efficiently. For instance, a fully automated application at a vaccine department store could provide functionality allowing storage workers to collect data as they pass through the storage units. By achieving such a state of automation, it has the potential to remove human error and report efficient, correct, and quality data.

Lastly, from an innovation point of view, this thesis hopefully brings the potential to introduce sensor technology and the Internet of things (IoT) to the DHIS2 platform. As this is the first application using BLE sensor technologies within the DHIS2 platform ecosystem, there is potential in some of the code being a part of the core. The Bluetooth communication with the GATT service and UART protocol is a feature that can be a generic core application. The code finds BLE devices and sets up communication between them by collecting different IoT UUIDs. Having BLE communication as a stable core application could facilitate innovation in utilizing other temperature sensors beyond BlueMaestro. The core application would then act as a BR, facilitating the development of applications using different sensor technology for other use cases than CCM. IoT within healthcare has great potential and could enhance access to care, strengthen the quality and reduce the cost of care [29].

10.1.2 Challenges

During the CCM application's development process, there were technical challenges regarding the code, testing, and hardware requirements. In addition to some challenges related to the implementation and piloting of this thesis.

In the application development process, the LMIS team and I met with representatives from the DHIS2 core team. After presenting the solution, they addressed a challenge regarding server space. Because the application architecture uses tracker capture, each temperature reading gets represented as one event on a TEI. This design can lead to many stored events, which over time or by scaling the solution to multiple facilities can exceed server capacity. The DHIS2 technical lead saw two solutions: increasing server capacity and paying for a CCM server instance. The other solution, and maybe a more convenient way, was making a function sanitizing old data. Historical temperature data in CCM is not that helpful because if there has been a temperature excursion three months ago, the vaccines are either replaced or used.

In advance of the pilot, the application had only been tested with a stable internet connection on a Samsung device. In retrospect, the testing environment was insufficient before conducting the pilot. Because of the testing environment, it was not possible to test the application's functionality with an unstable internet connection. Testing this before the pilot could have saved some time spent fixing bugs related to the poor connection.

Due to this, one major technical bug that occurred irregularly was when health facility workers tried to upload captured temperature data with a connection of H+ (High-Speed Packet Access) or EDGE (E). Because of the unstable connection, the application was online, but the packet transmission speed was slow. When health facility workers attempted to upload captured temperature data from the application to DHIS2's servers, the DHIS2 servers timed out after waiting some minutes due to the slow connection. Because the application design did not consider a context of low bandwidth and poor internet coverage, the result was inconsistent temperature data uploads to DHIS2 and, sometimes, the application closing.

Furthermore, only testing the application on one specific android device led to little knowledge regarding Android specifications for different devices. This led to some challenges when choosing devices for the pilot, as we did not know for sure which android requirements and versions were needed for running the CCM application. Some of the requirements were related; the device having an internal GPS - as BLE technology needs location activated to function [2].

In addition, there were some challenges regarding the devices provided for the pilot. The application did not manage to run as intended, struggling to collect all temperature data from the sensor before the UART communication was terminated. The devices provided were Android tablets running OS version 9 (Pie) with 1GB RAM and a Quad-core 1.3 GHz CPU, running an older Bluetooth version of 4.2. The BlueMaestro sensors support both version 4.2 and 5.0. The difference is in data transmission speed as Bluetooth 4.2 transmits data at 1 Megabit per second (Mbps). If an android device has Bluetooth 5.0, data is transmitted from the BlueMaestro sensor at 2 Mbps. Since it was impossible to get new android devices at short notice, we coped with this challenge by lending my phone to a health facility. In addition, download the application on health workers' private phones - as they usually have OS Android 10 or higher.

10.2 Pilot review

During the pilot in Mozambique, there were several challenges of both technical and contextual nature concerning excising work processes. During the pilot, there were also times when things did not go according to plan. These cases brought challenges, but adapting them gave the pilot potential opportunities.

10.2.1 Potential

During the piloting of the application, we faced some challenges related to different understanding from the facility workers on why they should use the CCM applications. After demoing and initial training, the users quickly understood the graphical user interface and the application's use case. In some cases, there was a need for additional training, but after the initial setup, all health facilities reported temperature data to the tracker capture app.

In addition, were the BlueMaestro sensors stable and reliable during the pilot period. Utilizing sensor technology was a new experience for the health facilities. Despite this, they were optimistic about the application and saw the potential in using the sensors for rural vaccination.

Furthermore, the CCM application digitizes the current CCM process reported by a paper form and introduces a new use scenario of DHIS2. In addition, the application and temperature sensors remove manual reporting steps and improve the data quality captured into DHIS2. Automation makings data harder to manipulate, in addition to reporting a standardized data set capturing only necessary data elements through the CCM application [7, p.2]. Reporting quality data is important for the CCM application in order to succeed. LMIS and HMIS rely on quality data for evidence-based decision-making. [7, p.2].

In addition to more collaboration between HISP groups, the pilot gives the potential for CCM to get on the agenda. Implementing the applications in Mozambique and performing an in-the-wild study, as its feedback provides an actual use aspect [47, p.490], increasing the opportunity to develop an artifact that is more suitable for end-users needs, and more likely to succeed. The pilot has also opened up the potential for better collaboration between HISP-Mozambique and HISP-Oslo regarding LMIS.

10.2.2 Challenges

There were various challenges in piloting the applications in the field. Chapter three presented the socio-technical perspective and highlighted that the socio part of the perspective represents humans, routines and practices, rules, culture, norms, politics, and motives [57, p.71]. During the implementation and pilot period, we faced some socio-technical challenges as we introduced a new HIS into an organization such as a health facility.

One of the first socio-technical challenges was the information flow before the pilot. Traveling to Mozambique, there was no common concrete plan for how the pilot should be performed. I had made an individual plan, but we had not communicated a common strategy. Therefore, the pilot planning was carried out during the first week of the pilot.

At the initial visit to a health facility, we faced additional challenges. During the first minute of demonstrating the CCM application, a health worker immediately said:

> "What am I going to do with this? I have a digital thermometer on the outside of the freezer."

With the socio-technical perspective in mind, implementing the application meant only more work for the health facility workers. An implementation would mean changes in their CCM routines and work practices. To deal with this challenge, we had to express the importance of the application and inform about the potential and importance of CCM and how the facility level could better national immunization through improved CCM.

Creating engagement around the pilot was essential to ensure that health facility workers used the application. At the facilities we visited, health facility workers had different competence in using digital IS as part of their work process, and most data capturing was paper-based. Capturing data in paper forms gives control to the data collector of which data to report. Implementing a digital solution that eliminates the facility workers' control stage was challenging due to a different set of norms and work culture - the control of what to report acted as a safety net protecting the one reporting the data.

During our participation in the polio campaign, we experienced temperature data not being reported. Because the campaign did not have enough ice packs for all the vaccine carriers, temperature data was not reported to DHIS2 using the application. Health workers did not want to report temperature data stored in a non-optimal environment during the pilot.

When demonstrating the application to the DPS at the province level, they addressed some challenges that could arise. The first concern they had was how much the sensor could withstand. If it endures being moved between vaccine carriers and other CCE, and to which degree it can withstand water - as water can accumulate at the bottom of CCE. The second concern DPS addressed was the possible sensor theft because of its appearance, the battery inside, or the sensors being lost. DPSs second concern happened during the polio campaign, as one sensor was lost after the campaign.

Furthermore, the pilot faced some challenges regarding Android devices provided for the pilot. These challenges led to findings regarding introducing devices to health facilities. One health facility worker responsible for the CCM at one of the facilities we visited already had four phones and did not want to take care of another reporting device. Most facility workers preferred to have the CCM application installed on their personal devices. One can argue that utilizing health facility workers' devices is more sustainable and cost-effective.

We also faced some challenges regarding payment of mobile data, finding the balance providing health workers with enough mobile data to use the application. During the pilot, we discovered that health facility workers used the mobile data provided for the pilot for other non-work-related activities. If facility workers run out of mobile data, it will challenge the integrity and reporting capability of the application. Providing too much mobile data will increase the expenses of the implementation.

Lastly, another core challenge during the pilot was the lack of information on the importance of CCM. As Ashok, Brison, and LeTallec state in their study on improving cold chain;

> "Countries are often unaware of the risks of an insufficient cold chain" [25].

One of the questions asked during the pilot interviews was;

> "What procedures do you have for spoiled or outdated vaccines?"

All facilities replied that they never had any cases of spoiled or discarded vaccines because they had expired. These answers show a need for better information at health facilities regarding CCM.

During interviews, interviewees also reported that they had regular power outages. Nevertheless, There was a slight perception of how this could affect the vaccines. Temperature data collected during the pilot showcases temperatures from CCE below two degrees over time. Therefore, improving information and training for health facilities workers regarding the importance of proper vaccine temperature could improve the facilities' CCM, as this is important to incorporate before introducing any CCM automation solution [13, p.5].

10.3 Pilot methods review

This thesis consists of different parts like development, implementing, and piloting. These parts result in the usage of multiple methods with a different purposes. In the first part of the thesis, agile software development methods were conducted in combination with the thesis's primary methodology, DSR. These methods' purpose was to make an artifact answering whom to support real-time CCM in DHIS2. Furthermore, in the last part of the thesis, different qualitative research methods were used to collect data during the pilot of the application.

10.3.1 Potential

The qualitative research methods conducted during the pilot provided the potential to examine the root problems in this use case. The data regarding exciting work processes gathered through semi-structured interviews and observation gave an in-depth insight into what influenced the existing CCM.

During data collection, we were told stories about vaccines being transported in cold boxes without ice packs due to the facilities not having enough CCE. In addition, stories about polio vaccines marked as spoiled were put back into the freezer to turn the mark back to their valid state. Hence, general enthusiasm regarding the application - information and focus on CCM was lacking.

Through the semi-structured interviews, we discovered that all the health facilities reported the same regarding routines of spoiled and expired vaccines. The fact that they never had trow spoiled vaccines is a shocking finding and the reality today - but also a step in the right direction to improve the knowledge. Furthermore, the potential of this qualitative insight tells that the CCM application only can do so much. These stories show potential in improving CCE, knowledge, and training of facility workers and raising attention regarding the importance of CCM.

10.3.2 Challenges

As a participant in a participatory case study piloting in a natural context, there were some communication challenges regarding the ambition level. The application was an MVP tailored to fit only one TEI at each facility. Hence, the MVP being close to production-ready, making it more generic became the level of ambition. The issue is understandable, as piloting semi-finished solutions is a fact. Nevertheless, if the ambition level had been more precisely communicated, the application could have been further improved before leaving Oslo.

Another obvious challenge when piloting was that I did not speak the native language. The language barrier was one of my main concerns in advance of the pilot. Nevertheless, it was not a big issue as HISP implementors translated everything. In addition, giving the HISP implementors more belonging to the application led to an excellent dynamic. Both in data collection and when presenting the solution at health facilities.

Furthermore, there were some challenges regarding data collection methods, as only qualitative data was collected during the pilot. Because time is scarce, during the health facility visits and the pilot period, there was no time to collect all the initially intended data. Another reason for only collecting qualitative data was the time it took before the facilities started to report data into DHIS2, which led to a smaller amount of quantitative data than desired. The intention was to collect a temperature data set to co-relate captured application data with the data health workers collected on paper. The data set could compare temperatures collected from the sensor BlueMaestro and the digital CCE thermostat. In addition, an extensive quantitative data set would make it possible to test the accuracy of the sensors and other potential findings like data visualization with other parameters in DHIS2.

10.3.3 Summary

The table below summarize the key challenges and opportunities discussed in this chapter:

	Challenges	**Opportunities**
CCM application	Fitting current work processes , Infrastructure, Storage problem as scaling	Supporting CCM, introducing IoT and BLE communication to DHIS2. Automation of current CCM.
Piloting	Pilot planning, Lack of CCM knowledge	HISP collaboration, Evaluating in natural context
Methods	Collecting sufficient data, Quantitative data, Ambition level	In-depth insight in a natural context

Table 8: Key challenges and opportunities

Chapter twelve

11 Conclusion and future work

This thesis has explored how real-time CCM can be supported with DHIS2 by implementing a custom android application in a generic software platform ecosystem. The development, implementation and piloting effort of the thesis aims to answer the specific need for improving CCM and answer the research question:

> *"How can real-time cold chain temperature monitoring be supported in DHIS2."*

Through piloting an application for this purpose, knowledge has been gained regarding Mozambique health facilities and their existing work processes of CCM. In addition, examined the possibility of implementing technology to improve CCM, and discovered its challenges. An artifact was developed through design science research and agile software development iterations, collaborating with HISP Mozambique, resulting in a pilot in the province of Zambezia. The pilot was conducted by implementing a custom Android application at the health facility level. The application uses BLE sensor technology to introduce real-time CCM in the DHIS2 platform ecosystem utilizing the DHIS2 tracker capture application.

The primary purpose of cold chain monitoring is to improve the handling of vaccines so that they stay effective and safe. There are various existing and recognized solutions for CCM today. This thesis incorporates CCM into DHIS2 by building a sensor-based application within the DHIS2 ecosystem, allowing health facility workers to have better CCM at their disposal. By decentralizing the data collection, HIS fragmentation is prevented by reporting to DHIS2 so decision-makers can access data in one HMIS.

From a technical point of view, the application combination with the sensor technology and its connection to DHIS2 is the core contribution to introducing real-time CCM to DHIS2. The solution has minimal dependence, the primary one being the BlueMaestro sensors. Nevertheless, the sensors have a low unit cost without a subscription like other CCM solutions. Beyond the application itself, introducing the GATT service and UART protocol to the platform core facilitates support for a more generic real-time CCM and other temperature sensors and potential IoT technologies.

Different potential and challenges were discovered during the pilot and implementation of the CCM application in resource-constrained settings. The fact that DHIS2 was the main HMIS in Mozambique brought potential, as the HISP group was an established actor with expert knowledge of DHIS2. The HISP group provided developers and implementors supplying innovation, training and support. In addition, DHIS2 being well established provided the pilot access to health facilities.

Throughout the pilot, health facility workers emphasized that the CCM application and the concept of sensor technology sending data to DHIS2 was something they could implement into their existing work processes and routines. The fact that the existing CCM work process today is paper-based and only reported to the immunization program once a year and not to DHIS highlights the potential of the implementation. The Provincial Directorates of Health (DPS) in Quelimane emphasized the applications potential. It would digitize manual statistics work and improve control of the CCM process through transparency to decision-makers. In addition, DPS reported that such an implementation could improve challenges related to data quality in DHIS2, improving their data analysis and statistics. Moreover, based on findings at facility visits, the application could contribute to the global need for better CCM and increase the vaccination rates by proving that vaccines in storage are reliable and stored at correct temperatures.

Achieving the potential mentioned above is more than implementing a generic version of the needed applications. Based on the findings in chapter nine and the challenges introduced in the discussion, one of the main challenges in DHIS2 supporting real-time CCM is the lack of knowledge regarding the importance of CCM. To accomplish a state of CCM being a successful core functionality, there is a need for sufficient training of health facilities workers', as this would drastically improve the situation and their knowledge. This shows that the socio-technical arrangements locally are as necessary as supporting CCM in DHIS2 through the application developed.

Additionally, being aware of infrastructural challenges, such as unstable internet connectivity and power outage, is essential in making a solution that can support real-time CCM to DHIS2. Lastly, strengthening measures concerning CCE like functioning and approved CCM fridges, ice packs and thermostats could improve health facility workers' possibility to conduct efficient CCM at the facilities in resource-constrained settings.

11.1 Future work

The future work of this thesis would mainly evolve the development of the CCM application. For it to be a stable implementation or a core DHIS2 application, it would benefit from being more thoroughly tested to ensure its robustness. Both technically improve the reliability of the application during unstable internet connectivity and ensure that the application is robust so that end-users do not experience cases where the applications stop working. Such measures would also minimize the amount of support needed from HISP implementors.

Furthermore, the CCM application needs to be improved by generalizing the tracker capture configuration in a way that the CCM application supports multiple TEIs in an organization unit. Making the application more generic would allow it to scale faster at an efficient matter. There is also a need to develop a design in the application to organize which BlueMaestro sensors belong to their respective CCE. Such improvement can be made by, for example, uploading the sensor's MAC address as a value in tracker capture and allowing end-users to choose which TEI to upload temperature data. The application or decision-makers can monitor facility data and report when one or more sensors upload to the same TEI. Additional improvement to the application would be to further research the utilization of the built-in BlueMaestro commands, as some of the commands not used in the MVP potentially can make the application more automatic. Utilizing the temperature logs from the sensor's memory will minimize manual work in the application as it can capture temperature data at a given interval. In addition, it will increase the number of temperature readings and the user experience for health workers as they do not have to report every reading manually through the applications. One specific feature to increase automation could be that the application captures the log at a given interval if one is within range of the sensor. This feature could be done by looping the GATT service, listening to a subscription, sending a UART command for the subscribed logs, and storing the result in a local database.

In addition, could decision-makers benefit from the application uploading, time and date to DHIS2. Implementing a timestamp will display when the temperature data was captured that day, making it possible to observe if, for instance, a facility tries to report the temperature twice in the morning. Such functionality will provide more control to decision-makers and minimize the opportunity of reporting low-quality data.

Future work on this application would also include addressing the problem regarding storage. There are multiple solutions to such a problem, and it depends on the future of the application. One way to reduce the storage issue is to store aggregated values to minimize the number of events on a TEI. Aggregated values like the MKT can represent temperature fluctuations in a single value. Utilizing the MKT could contribute to solving a future storage problem for the application [4].

Historical temperature data more or less lose relevance after a month as events that trigger an alarm are most relevant. Suppose aggregated values are not an option and the issue of storage becomes complex and expensive. In that case, one measure could be to refactor the application to use the data entry app, as this can be a measure to decrease the memory stored. In addition, researching possibilities to sanitize old temperature data, either in the application or in the tracker app, would also be a measure to reduce the storage problem.

Lastly, the alarm function is an essential feature of the application, especially concerning requirements from WHO and GAVI. Alarms would provide the opportunity to aggregate an alarm value representing the number of alarms in DHIS2. The number of alerts can then be visualized to decision-makers, making it possible to identify areas where additional CCM support is needed and improve the global CCM. Future work on the alarm functionality should involve research on efficiently sending alarms to different stakeholders and health facility workers, both with and without internet connectivity.

The work conducted during this thesis has aimed to raise awareness of the importance of CCM. The CCM application created in this thesis is an MVP implemented and tested at health facilities in Mozambique. The application and data collection through the pilot aims to answer how real-time CCM can be supported in DHIS. Hopefully, this can be the initial effort of a broader CCM support in DHIS2.

References

[1] *About DHIS2*. en-US. URL: https://dhis2.org/about/ (visited on 05/04/2022).

[2] accton. *BLE Beacons and Location-Based Services*. en-US. URL: https://www.accton.com/Technology-Brief/ble-beacons-and-location-based-services/ (visited on 06/13/2022).

[3] Eric Adu-Gyamfi and Petter Nielsen. "The Dynamics of a Global Health Information Systems Research and Implementation Project". en. In: (2019), p. 8.

[4] ELPRO-BUCHS AG. *Cold Chain Monitoring - everything you need to know ELPRO*. en. URL: https://www.elpro.com/en/cold-chain-monitoring (visited on 05/05/2022).

[5] *Android SDK*. en-US. URL: https://dhis2.org/android/sdk/ (visited on 04/28/2022).

[6] Bjørn-Ingar Bergum, Petter Nielsen, and Johan Ivar Sæbø. "Patchworks of Logistics Management Information Systems: Challenges or Solutions for Developing Countries?" en. In: *Information and Communication Technologies for Development*. Ed. by Jyoti Choudrie et al. Vol. 504. Series Title: IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 2017, pp. 47–58. ISBN: 978-3-319-59110-0 978-3-319-59111-7. DOI: 10.1007/978-3-319-59111-7_5. URL: https://link.springer.com/10.1007/978-3-319-59111-7_5 (visited on 05/30/2022).

[7] EN-BIRTH Study Group et al. "Barriers and enablers to routine register data collection for newborns and mothers: EN-BIRTH multi-country validation study". en. In: *BMC Pregnancy and Childbirth* 21.S1 (Mar. 2021), p. 233. ISSN: 1471-2393. DOI: 10.1186/s12884-020-03517-3. URL: https://bmcpregnancychildbirth.biomedcentral.com/articles/10.1186/s12884-020-03517-3 (visited on 05/31/2022).

[8] *Bluetooth Technology Overview*. en-US. URL: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/ (visited on 03/09/2022).

[9] Carla Bonina et al. "Digital platforms for development: Foundations and research agenda". In: *Information Systems Journal* 31.6 (Nov. 2021). Publisher: John Wiley & Sons, Ltd, pp. 869–902. ISSN: 1350-1917. DOI: 10.1111/isj.12326. URL: https://pericles.pericles-prod.literatumonline.com/doi/10.1111/isj.12326 (visited on 05/25/2022).

[10] Braa et al. "Developing Health Information Systems in Developing Countries: The Flexible Standards Strategy". en. In: *MIS Quarterly* 31.2 (2007), p. 381. ISSN: 02767783. DOI: 10.2307/25148796. URL: https://www.jstor.org/stable/10.2307/25148796 (visited on 05/25/2022).

[11] *Cold Chain (for Vaccines and other Cold chain items) [mSupply Foundation Documentation]*. URL: https://wiki.msupply.foundation/en:mobile:user_guide:vaccine (visited on 05/04/2022).

[12] "Computerizing Logistics Management Information Systems. A Program Manager's Guide, October 2012". en. In: (), p. 72.

[13] "Computerizing Logistics Management Information Systems. A Program Manager's Guide, October 2012". en. In: (), p. 72.

[14] *Digital platforms for development: Foundations and research agenda*. en. DOI: 10.1111/isj.12326. URL: https://pericles.pericles-prod.literatumonline.com/doi/epdf/10.1111/isj.12326 (visited on 02/28/2022).

[15] *Facts and figures - HISP Centre*. en. URL: https://www.mn.uio.no/hisp/english/about/facts/index.html (visited on 05/04/2022).

[16] Ahmad Ghazawneh and Ola Henfridsson. "Balancing platform control and external contribution in third-party development: the boundary resources model: Control and contribution in third-party development". en. In: *Information Systems Journal* 23.2 (Mar. 2013), pp. 173–192. ISSN: 13501917. DOI: 10.1111/j.1365-2575.2012.00406.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2575.2012.00406.x (visited on 06/13/2022).

[17] V.S. Gordon and J.M. Bieman. "Rapid prototyping: lessons learned". In: *IEEE Software* 12.1 (Jan. 1995). Conference Name: IEEE Software, pp. 85–95. ISSN: 1937-4194. DOI: 10.1109/52.363162.

[18] Shirley Gregor and Alan R. Hevner. "Positioning and Presenting Design Science Research for Maximum Impact". In: *MIS Quarterly* 37.2 (2013). Publisher: Management Information Systems Research Center, University of Minnesota, pp. 337–355. ISSN: 0276-7783. URL: https://www.jstor.org/stable/43825912 (visited on 04/25/2022).

[19] George Mc Guire. *Handbook of Humanitarian Health Care Logistics: Designing the Supply Network and Managing the Flows of Information and Health Care Goods in Humanitarian Assistance during Complex Political Emergencies in low-resource settings*. en. Google-Books-ID: FRREP4qhdDoC. George Mc Guire, Oct. 2015.

[20] C K Nagendra Guptha. "Design and Implementation of Monitoring System for Retail Cold Chain - Case Study". en. In: (2018), p. 6.

[21] Ole Hanseth and Kalle Lyytinen. "Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet". en. In: *Enacting Research Methods in Information Systems*. Ed. by Leslie P. Willcocks, Chris Sauer, and Mary C. Lacity. Cham: Springer International Publishing, 2016, pp. 104–142. ISBN: 978-3-319-29271-7 978-3-319-29272-4. DOI: 10.1007/978-3-319-29272-4_4. URL: http://link.springer.com/10.1007/978-3-319-29272-4_4 (visited on 05/30/2022).

[22] Alan Hevner et al. "Design Science in Information Systems Research". In: *Management Information Systems Quarterly* 28 (Mar. 2004), p. 75.

[23] Juhani Iivari. “Distinguishing and contrasting two strategies for design science research”. en. In: *European Journal of Information Systems* 24.1 (Jan. 2015), pp. 107–115. ISSN: 0960-085X, 1476-9344. DOI: 10.1057/ejis.2013.35. URL: https://www.tandfonline.com/doi/full/10.1057/ejis.2013.35 (visited on 04/25/2022).

[24] *Immunization*. en. URL: https://www.unicef.org/immunization (visited on 05/05/2022).

[25] *Improving cold chain systems: Challenges and solutions — Elsevier Enhanced Reader*. en. DOI: 10.1016/j.vaccine.2016.08.045. URL: https://reader.elsevier.com/reader/sd/pii/S0264410X16307307?token=A152359DEC6550A3FBEF8CD5409CC7EE979708008EB25161E846E15FA533331BF1D6 originRegion=eu-west-1&originCreation=20220531123658 (visited on 05/31/2022).

[26] *In Action*. en-US. URL: https://dhis2.org/in-action/ (visited on 06/02/2022).

[27] *in5320-7-platform-ecosystems-fundamental-concepts.pdf*. URL: https://www.uio.no/studier/emner/matnat/ifi/IN5320/h18/slides/in5320-7-platform-ecosystems-fundamental-concepts.pdf (visited on 06/07/2022).

[28] Honest C. Kimaro. “Strategies for Developing Human Resource Capacity to Support Sustainability of ICT Based Health Information Systems: A Case Study from Tanzania”. en. In: *The Electronic Journal of Information Systems in Developing Countries* 26.1 (Aug. 2006), pp. 1–23. ISSN: 16814835. DOI: 10.1002/j.1681-4835.2006.tb00171.x. URL: https://onlinelibrary.wiley.com/doi/10.1002/j.1681-4835.2006.tb00171.x (visited on 05/25/2022).

[29] Ravi Kishore Kodali, Govinda Swamy, and Boppana Lakshmi. “An implementation of IoT for healthcare”. en. In: *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. Trivandrum, Kerala, India: IEEE, Dec. 2015, pp. 411–416. ISBN: 978-1-4673-6670-0. DOI: 10.1109/RAICS.2015.7488451. URL: http://ieeexplore.ieee.org/document/7488451/ (visited on 06/10/2022).

[30] Lazar, Jonathan, Feng, Jinjuan Heidi, and Harry Hochheiser. *Research Methods in human-computer interaction*. URL: https://books.google.no/books?hl=no&lr=&id=hbkxDQAAQBAJ&oi=fnd&pg=PP1&dq=Re-+search+Methods+in+human-computer+interaction&ots=Sq644a16bV&sig=KhH6-vJzVLUrPpqe8c__7s9VF-E&redir_esc=y#v=onepage&q=Re-%20search%20Methods%20in%20human-computer%20interaction&f=false.

[31] Allen S Lee. “Thinking about Social Theory and Philosophy for Information Systems”. en. In: (), p. 26. URL: https://www.researchgate.net/publication/251376923_1_Thinking_about_Social_Theory_and_Philosophy_for_Information_Systems.

[32] Timothy C. Lethbridge, Susan Elliott Sim, and Janice Singer. "Studying Software Engineers: Data Collection Techniques for Software Field Studies". en. In: *Empirical Software Engineering* 10.3 (July 2005), pp. 311–341. ISSN: 1382-3256, 1573-7616. DOI: 10.1007/s10664-005-1290-x. URL: http://link.springer.com/10.1007/s10664-005-1290-x (visited on 04/24/2022).

[33] Magnus Li. "Entering and Organizing Engaged Information Systems Research". In: (Dec. 2021). URL: https://www.researchgate.net/publication/356843971_Entering_and_Organizing_Engaged_Information_Systems_Research.

[34] *Life expectancy at birth, total (years) - Mozambique — Data*. URL: https://data.worldbank.org/indicator/SP.DYN.LE00.IN?locations=MZ (visited on 06/14/2022).

[35] *Life expectancy at birth, total (years) - Norway — Data*. URL: https://data.worldbank.org/indicator/SP.DYN.LE00.IN?locations=NO (visited on 06/14/2022).

[36] Daniel Luna et al. "Health Informatics in Developing Countries: Going beyond Pilot Practices to Sustainable Implementations: A Review of the Current Challenges". In: *Healthcare Informatics Research* 20.1 (Jan. 2014). Publisher: Korean Society of Medical Informatics, pp. 3–10. DOI: 10.4258/hir.2014.20.1.3. URL: https://synapse.koreamed.org/articles/1075717 (visited on 05/05/2022).

[37] Frank Merrett. "Reflections on the Hawthorne Effect". en. In: *Educational Psychology* 26.1 (Jan. 2006), pp. 143–146. ISSN: 0144-3410, 1469-5820. DOI: 10.1080/01443410500341080. URL: http://www.tandfonline.com/doi/abs/10.1080/01443410500341080 (visited on 05/23/2022).

[38] *Mozambique: main causes of deaths 2019*. en. URL: https://www.statista.com/statistics/1268021/main-causes-of-deaths-in-mozambique/ (visited on 06/14/2022).

[39] Nick. *The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs (2015)*. en-US. URL: https://www.aislelabs.com/reports/beacon-guide/ (visited on 03/09/2022).

[40] *Norway Population (2022) - Worldometer*. en. URL: https://www.worldometers.info/world-population/norway-population/ (visited on 06/14/2022).

[41] *nRF Connect for Desktop*. en. URL: https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-desktop (visited on 04/19/2022).

[42] *nRF Toolbox App*. en. URL: https://www.nordicsemi.com/Products/Development-tools/nRF-Toolbox (visited on 04/19/2022).

[43] *nRF52832 - Nordic Semiconductor*. en. URL: https://www.nordicsemi.com/Products/nRF52832 (visited on 04/19/2022).

[44] Ken Peffers et al. "A design science research methodology for information systems research". In: *Journal of Management Information Systems* 24 (Jan. 2007), pp. 45–77.

[45] Ken Peffers et al. "Design Science Research Evaluation". In: vol. 7286. May 2012, pp. 398–410. ISBN: 978-3-642-29862-2. DOI: 10.1007/978-3-642-29863-9_29.

[46] *PQS_E006_TR05.1.pdf*. URL: https://extranet.who.int/pqweb/sites/default/files/documents/PQS_E006_TR05.1.pdf (visited on 05/04/2022).

[47] Preece, Jenifer, Sharp, Helen, and Rogers , Yvonne. *Interaction design - beoynd human-computer interaction 4th edition*.

[48] *Principles behind the Agile Manifesto*. URL: https://agilemanifesto.org/principles.html (visited on 04/25/2022).

[49] Natalie Privett and David Gonsalvez. "The top ten global health supply chain issues: Perspectives from the field". en. In: *Operations Research for Health Care* 3.4 (Dec. 2014), pp. 226–230. ISSN: 2211-6923. DOI: 10.1016/j.orhc.2014.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2211692314200002 (visited on 03/09/2022).

[50] Mark de Reuver, Carsten Sørensen, and Rahul C. Basole. "The Digital Platform: A Research Agenda". en. In: *Journal of Information Technology* 33.2 (June 2018), pp. 124–135. ISSN: 0268-3962, 1466-4437. DOI: 10.1057/s41265-016-0033-3. URL: http://journals.sagepub.com/doi/10.1057/s41265-016-0033-3 (visited on 06/13/2022).

[51] Lars Kristian Roland, Terje Aksel Sanner, and Eric Monteiro. "Architectures of large-scale participatory design". en. In: (), p. 33.

[52] Johan Ivar Sæbø et al. "Comparing strategies to integrate health information systems following a data warehouse approach in four countries". en. In: *Information Technology for Development* 17.1 (Jan. 2011), pp. 42–60. ISSN: 0268-1102, 1554-0170. DOI: 10.1080/02681102.2010.511702. URL: http://www.tandfonline.com/doi/full/10.1080/02681102.2010.511702 (visited on 05/06/2022).

[53] Terje Aksel Sanner. "Paying Per Diems for ICT4D Project Participation: A Sustainability Challenge". en. In: *Information Technologies* 10.2 (2014), p. 16.

[54] *SDK vs. API: What's the Difference?* en-us. July 2021. URL: https://www.ibm.com/cloud/blog/sdk-vs-api (visited on 06/14/2022).

[55] Maung Sein et al. "Action Design Research". In: *MIS Quarterly* 35 (Mar. 2011), pp. 37–56. DOI: 10.2307/23043488.

[56] Angela K. Shen, Rebecca Fields, and Mike McQuestion. "The future of routine immunization in the developing world: challenges and opportunities". en. In: *Global Health: Science and Practice* 2.4 (Dec. 2014). Publisher: Global Health: Science and Practice Section: COMMENTARY, pp. 381–394. ISSN: 2169-575X. DOI: 10.9745/GHSP-D-14-00137. URL: https://www.ghspjournal.org/content/2/4/381 (visited on 06/02/2022).

[57] Ian Sommerville et al. "Large-scale complex IT systems". en. In: *Communications of the ACM* 55.7 (July 2012), pp. 71–77. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/2209249.2209268. URL: https://dl.acm.org/doi/10.1145/2209249.2209268 (visited on 06/13/2022).

[58] Robert E. Stake. *The Art of Case Study Research*. en. Google-Books-ID: ApGdBx76b9kC. SAGE, Apr. 1995. ISBN: 978-0-8039-5767-1.

[59] *teachLeanStrategy-with-cover-page-v2.pdf*. URL: https://d1wqtxts1xzle7.cloudfront.net/38971773/teachLeanStrategy-with-cover-page-v2.pdf?Expires=1650906615&Signature=UQbl8CUysl8IyFOYECgqyQvU6EGicqon9
53NHQ80f3E9kgz0JUCo1uRb65jgRZ9mXEPpeZTKnZLkq7BypovD4o4fsFMJum~KfR0EXVFv9lt
J6pUx8cLiEnpi3HDGRV-3R5vi6Vebjirwp-jhe5fBm8sSvdlyL8h7NAHaOkVmN7-
Ob2xZ99Qwbihzhx8cwh8IZ3YdVntaqi1~DpOydjRxLJY8wNmRD93026wjTDfmu98NmgF7x48hX
evWTLFEKWsfgyH59l40hSdVDu7aJwZmsANhS33R-g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (visited on 04/25/2022).

[60] *The Global Fund's In-country Supply Chain Processes*. URL: https://www.theglobalfund.org/media/6363/oig_gf-oig-17-008_report_en.pdf (visited on 05/31/2022).

[61] Amrit Tiwana. "Core Concepts and Principles". en. In: *Platform Ecosystems*. Elsevier, 2014, pp. 23–48. ISBN: 978-0-12-408066-9. DOI: 10.1016/B978-0-12-408066-9.00002-3. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780124080669000023 (visited on 06/07/2022).

[62] Amrit Tiwana. "Platform Architecture". en. In: *Platform Ecosystems*. Elsevier, 2014, pp. 73–116. ISBN: 978-0-12-408066-9. DOI: 10.1016/B978-0-12-408066-9.00005-9. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780124080669000059 (visited on 06/13/2022).

[63] *Tracker Overview*. en-US. URL: https://dhis2.org/tracker/ (visited on 04/28/2022).

[64] Pera Tran. "16-bit UUID Numbers Document". en. In: (), p. 29.

[65] Aliya Walji. "Considerations for the Integration of HMIS and LMIS". en. In: (), p. 15.

[66] *WHO — World Health Organization*. Publisher: World Health Organization. URL: https://apps.who.int/immunization_standards/vaccine_quality/pqs_catalogue/categorypage.aspx?id_cat=17 (visited on 05/05/2022).

[67] *WHO_EPI_LHIS_98.02.pdf*. URL: https://apps.who.int/iris/bitstream/handle/10665/64776/WHO_EPI_LHIS_98.02.pdf?sequence=1&isAllowed=y (visited on 05/23/2022).

Abbreviations and acronyms

CC - Cold Chain.

CCM - Cold Chain Monitoring.

CCE - Cold Chain Equipment.

IT - Information Technology.

IS - Information Systems.

II - Information Infrastructure.

IB - Installed Base.

HIS - Health Information System.

HMIS - Health Management Information System.

LMIS - Logistic Management Information System.

EPI - Expanded Programme on Immunization.

DHIS2 - District Health Information System 2.

HISP - Health Information Systems Program.

DPS - The Provincial Directorates of Health.

DSR - Design Science Research.

ADR - Action Design Research.

WHO - World Health Organization.

GAVI - The Global Alliance for Vaccines and Immunization.

SIAPS - The Systems for Improved Access to Pharmaceuticals and Services .

BR - Boundary resources.

API - Application Programming Interface.

REST - Representational State Transfer.

SDK - Software Development Tool.

TEI - Tracked Entity Instance.

MVP - Minimum viable product.

BLE - Bluetooth Low Energy.

BCR - Bluetooth Classic.

MVC - Model-View-Controller.

UART - Universal asynchronous receiver-transmitter.

ATT - Attribute Protocol.

GATT - Generic Attribute Protocol.

SoC - System on Chip.

GUI - Graphical User Interface.

MKT - Mean kinetic temperature.

VVM - Vaccine Vial Monitor.

IoT - Internet of Things.

Appendix

- Link to GitHub repository: `https://github.com/Filipcl/CCM-App`
- Link to video demo of the CCM app: `https://player.vimeo.com/video/649687929?h=ad4e689fd3`
- DHIS2 community post: `https://community.dhis2.org/t/developing-and-piloting-a-` `47087?u=filipcl`