[Tittel]

[Undertittel]

Gagandeep Kaur



[Oppgavetype ved institutt/fakultet]

UNIVERSITY OF OSLO

[Dato]

[Tittelblad]

(samme tekst som på forsiden, men studenten står friere i utformingen)

© Forfatter

År

Tittel

Forfatter

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

Trykk: Reprosentralen, Universitetet i Oslo

Abstract

Table of contents

[1 Introduction 1](#_Toc33534464)

[2 Context 2](#_Toc33534465)

[3 QR/Barcode 3](#_Toc33534466)

[3.1 Barcodes 3](#_Toc33534467)

[3.1.1 History 4](#_Toc33534468)

[3.1.2 Application areas 5](#_Toc33534469)

[3.1.3 Information storage 5](#_Toc33534470)

[3.2 QR-codes 7](#_Toc33534471)

[3.2.1 History 8](#_Toc33534472)

[3.2.2 Application areas 8](#_Toc33534473)

[3.2.3 Information storage 10](#_Toc33534474)

[3.3 Comparison 13](#_Toc33534475)

[3.4 Suitable code for this case 14](#_Toc33534476)

[4 Previous work & platform technology 16](#_Toc33534477)

[4.1 Paper-based system 16](#_Toc33534478)

[4.2 Baobab 18](#_Toc33534479)

[4.3 CommCare and others 19](#_Toc33534480)

[4.4 DHIS2 and open-source platform 20](#_Toc33534481)

[4.5 Mobile technology in health systems 22](#_Toc33534482)

[4.6 “*Okay, what next*”? 22](#_Toc33534483)

[5 Method 23](#_Toc33534484)

[5.1 Research Methodology 23](#_Toc33534485)

[5.2 Our Journey 24](#_Toc33534486)

[6 Patient identification 25](#_Toc33534487)

[6.1 The diversity of systems 25](#_Toc33534488)

[6.2 The potential solutions for unique ID 25](#_Toc33534489)

[6.2.1 Partially local 25](#_Toc33534490)

[6.2.2 National ID-based 27](#_Toc33534491)

[6.2.3 A combination of both 29](#_Toc33534492)

[6.3 Solution for this prototype 31](#_Toc33534493)

[7 Design & Implementation 34](#_Toc33534494)

[7.1 The prototype 34](#_Toc33534495)

[7.1.1 Functionality 34](#_Toc33534496)

[7.1.2 User Interface 34](#_Toc33534497)

[7.1.3 Use of the prototype 37](#_Toc33534498)

[7.1.4 Implementation details 38](#_Toc33534499)

[7.1.5 Contextual considerations 43](#_Toc33534500)

[7.2 Hardware 44](#_Toc33534501)

[7.2.1 Options 44](#_Toc33534502)

[7.2.2 The label printer 45](#_Toc33534503)

[7.2.3 The Printing part 49](#_Toc33534504)

[7.2.4 Other useful equipment 51](#_Toc33534505)

[8 Iterations 53](#_Toc33534506)

[8.1 First iteration 53](#_Toc33534507)

[8.2 Second iteration 55](#_Toc33534508)

[9 Discussion 59](#_Toc33534509)

[9.1 Comparison with Baobab 59](#_Toc33534510)

[10 Conclusion 60](#_Toc33534511)

[Literature 61](#_Toc33534512)

List of figures

Figure 1 - A platform architecture 21

Figure 2 - Action research 24

Figure 3 - Partially local 26

Figure 4 - National ID-based 28

Figure 5 - A combination of both 30

List of pictures

[Picture 1 3](#_Toc33534518)

[Picture 2 3](#_Toc33534519)

[Picture 3 - Various types of linear barcodes. A snippet of a table from Wikipedia. 4](#_Toc33534520)

[Picture 4 - Some types of barcodes and their storage ability 6](#_Toc33534521)

[Picture 5 - Decoding the information in a barcode 7](#_Toc33534522)

[Picture 6 - QR code for Google 9](#_Toc33534523)

[Picture 7: Malawian National ID card 9](#_Toc33534524)

[Picture 8: A MP3 QR code 10](#_Toc33534525)

[Picture 9 - The characteristics & properties of 2D barcodes 11](#_Toc33534526)

[Picture 10 - Positioning marks 12](#_Toc33534527)

[Picture 11 - Alignment mark 12](#_Toc33534528)

[Picture 12 - Timing pattern 12](#_Toc33534529)

[Picture 13 - Version information 12](#_Toc33534530)

[Picture 14 - Format information 12](#_Toc33534531)

[Picture 15 - Data & error correction keys 12](#_Toc33534532)

[Picture 16 - Queit zone 12](#_Toc33534533)

[Picture 17 - Data flow 17](#_Toc33534534)

[Picture 18 - Baobab workstation in one of the clinics 19](#_Toc33534535)

[Picture 19 - Patient information 32](#_Toc33534536)

[Picture 20 - Patient information in the TB program 32](#_Toc33534537)

[Picture 21 - Child Program in Tracker app 36](#_Toc33534538)

[Picture 22 - Child Program in prototype 36](#_Toc33534539)

[Picture 23 – Child Program screen in Tracker vs. the prototype for testing 36](#_Toc33534540)

[Picture 24 - DHIS2 Tracker 37](#_Toc33534541)

[Picture 25 - Prototype 37](#_Toc33534542)

[Picture 26 - App icon in both 37](#_Toc33534543)

[Picture 27 - Main screen in the app for testing 38](#_Toc33534544)

[Picture 28 - Fetching the data from the file 39](#_Toc33534545)

[Picture 29 - Data encoded from the QR is sent to the next activity, the DataHandlerActivity 39](#_Toc33534546)

[Picture 30 - Starting the TB activity 40](#_Toc33534547)

[Picture 31 - Add a new patient to the database 41](#_Toc33534548)

[Picture 32 - Checking for duplicates 42](#_Toc33534549)

[Picture 33 - Writing to the file 42](#_Toc33534550)

[Picture 34 - Existing printer at the campus in Malawi 45](#_Toc33534551)

[Picture 35 - Brother P-Touch Cube Plus label printer 46](#_Toc33534552)

[Picture 36 - Label from the Baobab system covering the front page 47](#_Toc33534553)

[Picture 37 - A worn health passport of an anonymous patient 48](#_Toc33534554)

[Picture 38 - Example of a label from the label printer 48](#_Toc33534555)

[Picture 39 - Print Code 49](#_Toc33534556)

[Picture 40 - Screen the user being redirected to for printing the code 50](#_Toc33534557)

[Picture 41 - The app has the ability to reprint the code 51](#_Toc33534558)

[Picture 42 - Community Health Worker on her regular village visit 52](#_Toc33534559)

[Picture 43 - The main screen in the prototype after the changes 56](#_Toc33534560)

[Picture 44 - A toast message "Missing input" at the bottom of the screen 58](#_Toc33534561)

1. Introduction
2. Context

To work on a problem and try to find some solutions for it, requires understating the whole context first. Therefore, *qualitative research* focuses on real-world situations trying to understand the context as much as possible so the solution can fit in the best way. We will write more about qualitative research in chapter X. However, the focus of this chapter will be, as the title says, on the context of the problem. We will be looking at some important information about the country and the current situation there, which includes the health situation, politics, weather, and economy, etc, and how these things have an impact on our case.

* 1. The country
  2. Challenges
  3. Health information situation
     1. Roles
  4. Mobile technology in health systems

*Write about other systems being used in the health sector. The focus will be on bar/QR-codes. However, I might also write about mobility work – using mobile devices in health systems and their effect (maybe).   
Look at some previous work done in the same field and how it went. Try to note some good/not so good advice from such studies that might be useful in my case.*

* 1. Impact on our case

1. QR/Barcode

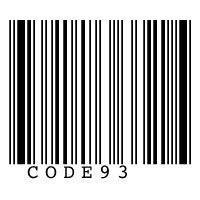
As mentioned in the previous chapter, this project focuses on the QR/barcode implementation in the DHIS2 Tracker app, which is described in chapter X, and its consequences in the health facilities. Before we go into the details of the implementation, we will go through these two types of codes and how they do work.

* 1. Barcodes

This is a technology that almost all of us have come across at least ones in our lives. The first example that comes up in our mind might be the retail industry. The product labeling with such barcodes can be seen everywhere. However, what are those lines in such codes and what kind of information do they give us?

A barcode can be defined as an image with straight vertical lines with some white spaces in between. These barcodes are also referred to as linear or 1D (1 dimensional) barcodes. Such barcodes can be read by an optical scanner or a barcode reader.[[1]](#endnote-1) We can find different patterns and “styles” in such barcodes – the picture below shows two types of barcodes.

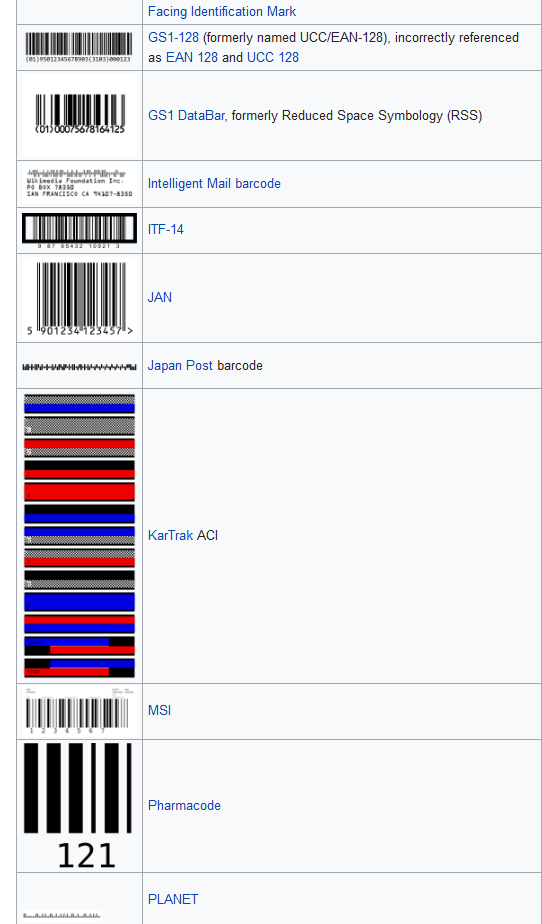
Picture



Picture



There is a huge variety of barcodes being used. The two pictures above show two of the “simplest” types of barcodes. The picture below shows a snippet from a bigger table from Wikipedia[[2]](#endnote-2) with a collection of linear barcodes.

[](https://en.wikipedia.org/wiki/Barcode#Use)

Picture 3 - Various types of linear barcodes. A snippet of a table from Wikipedia.

* + 1. History

Let us first have a brief look at the background of the barcodes and their usage. Like most of the successfully invented technologies, the journey of the barcodes was quite a long and tough one. It all started back in 1948, when a passionate undergraduate student, Bernard Silver overheard a conversation about searching on automation of reading the product information during checkout.[[3]](#endnote-3) [ref wiki]. Silver discussed with his friend Norman Joseph Woodland and they started working on that idea shortly.

In their very first version, they used ultraviolet ink for the barcodes. However, that does not seem to work very flourishing, since the ink faded too easily and was also too expensive. The work on this technology continued. The barcodes being used today have gone through several development phases. During this phase, others and Woodland and Silver worked on this idea as well. E.g. David Collins, who started working for a company where he worked on automatically identify cars. He developed a system called *KarTrak*, using other types of barcodes.[[4]](#endnote-4) [wiki ref].

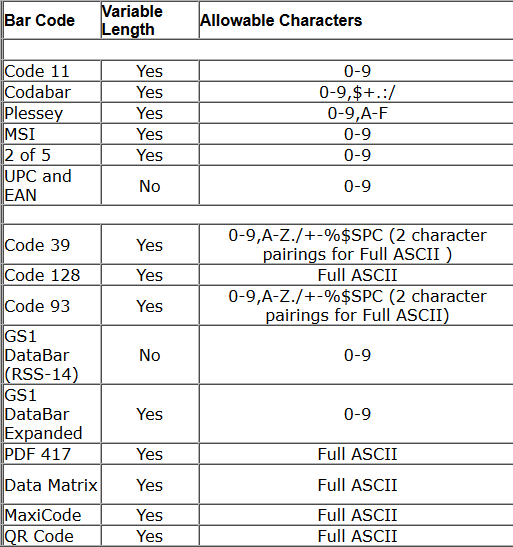
The list of all the phases of this part of the technology is long and is not that relevant to this project. However, our goal for mentioning some history about barcodes and their implementation is to gain some understanding that this technology has gone through many phases and is still under development. The ongoing research and development are making it even more relevant for this project.

* + 1. Application areas

The initial use of the barcodes was the automatic identification of products during checkout. In industrial use, the barcodes were used by the Association of America Railroad in the 1950s.[[5]](#endnote-5) [ref ]. Later, this technology was used in several other parts of the industry, such as in retail markets. Till today, the usage of barcodes has been widely used. Other than the retail market, the barcodes are being used in various identification areas, or even in the laboratories. The different domains use different barcodes that fit their requirements. E.g. barcodes used on products in the grocery shops might be of a different type than the ones used in the laboratories. We will write more about the use of these codes in the health sector in section X.X Moreover, the type of code used depends mostly on the area of usage of the product.

* + 1. Information storage

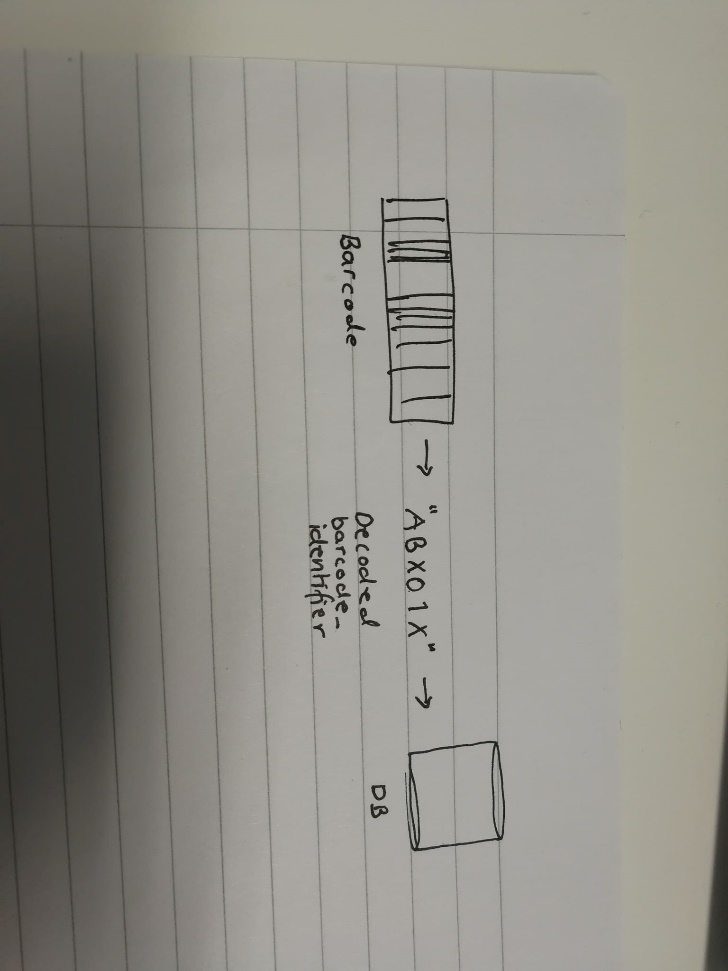
We’ve mentioned that linear barcodes are made up of vertical lines, called bars and spaces. The different combinations of these elements represent the various type of barcodes. These various barcodes have different ability to store the information according to their structure. The linear barcodes do normally store information in text format with characters that are allowed for that specific type of barcode. E.g. the Code 128 barcodes can store up to full ASCII characters[ref].[[6]](#endnote-6) A snippet from a table shows some types of barcodes and their ability to store the information.



Picture 4 - Some types of barcodes and their storage ability6

When decoding a barcode, a light source passes over the lines and the light source is absorbed by the dark bars and reflected by the white spaces. Then these reflected lines can be converted into electrical signals by a photocell detector.

These barcodes usually contain little information to identify some more information stored somewhere else, e.g. in the database. If we take the barcode on a product we buying int the store, it will contain a product code. When decoded, it will show the relevant product information by looking up for that product using the product code that has been decoded. Using that product code or an “identifier”, the user can perform relevant operations such as lookup for the price.



Picture 5 - Decoding the information in a barcode

Picture P.P illustrates one example of fetching the information from a given barcode. Let us say that the barcode is for a person. Decoding the barcode gives us the identifier for that person, which will then be used to lookup in the database where the information is stored.

* 1. QR-codes

As mentioned in the previous section, 2.1.1, the development of the barcodes is ongoing. This development of barcodes has led to a version to these barcodes that is, as the section title says, the QR-codes. The major difference between the barcodes mentioned in the previous section and these QR-codes is their so-called dimensions. The classic barcodes are 1-dimensional (1D), but the QR-codes are 2-dimensional (2D).[[7]](#endnote-7) [ref]. This difference in dimensions says something about their ability to store the data.

QR-codes stands for *Quick Response*, which is respectively a technology for quick access to the information.[[8]](#endnote-8) [book ref]. The QR-codes are quite like the barcodes in the sense that both are codes that are to some degree machine-readable and might contain information about the product or item it is being attached to. The QR-codes are in fact a type of barcodes.7 [ref]. The fact that separates these two is their ability to store the data. Section 2.2.3 explains more about this.

To read the information in a QR-code, one can simply use the camera of e.g. a smartphone. There are many applications available to read such codes – by downloading one of these QR-code reader apps you can easily retrieve the information by scanning the QR-code by simply capturing the code with the devices’ camera. The app will then automatically retrieve the data stored behind these codes and show in a more human-readable style depending on the app’s user interface.

* + 1. History

The QR-code system was ‘born’ in Japan in 1994.[[9]](#endnote-9) [ref]. They were originally developed to help in the manufacturing process in Denso Wave for tracking vehicles and parts of the cars. They were meant to make the production more effective, by automatically scanning the codes.[[10]](#endnote-10) [ref]. This technology started to get rapidly used due to its ability to decoding speed and information size.

The development of this technology was taken further with more data storage and better error correction. However, it came with some consciousness that the size of the QR-codes got increased. In some contexts, the system demanded that the codes stored more data, and less in others. This gives us QR-codes with different sizes, shapes, colors, etc. depending on what kind of information is encoded in it.

* + 1. Application areas

Like the barcodes, the QR-codes are being used for various purposes. The ability to store data in both dimensions is making this technology being used in even more domains than the traditional barcodes. We are in 2020 and you must have seen QR-codes being used in different contexts. Some typical use-areas of QR-codes can be; products, websites, email, advertisement, etc.

Let us take a typical example of a QR-code of a website. By scanning a QR-code you will automatically (depends on the app – it might ask you first before opening a browser) be redirected to a browser where it opens the encoded URL of that webpage. In the pictures below, there is a QR code for Google on the left, which by scanning with a regular app for QR codes can get you to the browser to open the link encoded in the code.

|  |  |
| --- | --- |
|  | Et bilde som inneholder skjermbilde  Automatisk generert beskrivelse |

Picture 6 - QR code for Google

The QR-codes are even being used on personal IDs. During the fieldwork in Malawi, we observed that the national ID card-system seemed to use the QR-codes. The card has a QR-code printed on it, which stores the information of the cardholder. The picture below shows an ID card for an anonymous Malawian citizen.





Picture 7: Malawian National ID card

* + 1. Information storage

We have written briefly about how and what type of information the barcodes can store in one of the sections above. Now, it’s the QR-codes’ turn. Since the QR-codes are 2D, they can store information both vertically and horizontally. This capability makes the QR-codes handle much more information than regular barcodes. Not only the amount of information but also the types of information. Compared to the barcodes, which mainly store text-based information, these QR-codes can deal with several types of information. The QR code has a section where it stores what type of information it stores. More about these sections is describes later in this section. The picture below shows an example of an MP3 QR code[[11]](#endnote-11) [ref].

Et bilde som inneholder objekt, tekst, kryssord, førstehjelpskrin

Automatisk generert beskrivelse

Picture 8: A MP3 QR code

QR-code can also store a much larger amount of information. A QR code like in the picture above can also store information about different entities. We can see an example of it in the DHIS2 tracker app. In the app, we can already find QR codes for various tracked entities and attributes. The QR codes in the tracker app currently store the information in text format [ref??].

Like 1D barcodes, the capacity of the 2D barcodes various upon the type of code being used[[12]](#endnote-12).[ref] Picture P.P shows a table of some of the 2D codes with their ability to store the information. Choosing the right type of code depends surly on the intended use and domain.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 9 - The characteristics & properties of 2D barcodes

How is the information stored?

The next question that comes up is how does a QR code store the information? To be able to answer this question, we need to look at the structure of the QR codes. As mentioned, there are various types of 2D codes, here we will try to explain using the most common type of QR codes.

A normal QR code consists of seven essential parts.[[13]](#endnote-13)[ref]. Each of the parts has its own function. Typical code will have three smaller squares in the corner which are placed there positioning marks that indicate the direction in which the code was printed, picture P.P. In picture P.P, we can see the alignment marks. These are especially useful in larger QR codes to help with the orientation. The next element is the timing pattern, picture P.P. These are dotted lines in the codes which are used by the scanner to determine how large the matrix is for that specific code. Then we have version information in picture P.P. These elements of the code indicate the version of the QR code type. We have already seen that there are many different types of QR codes, and thus those various code types have several versions. The next fundamental part of code is the format information, picture P.P. This part of the code contains important information about the error tolerance and the data mask pattern. This makes it easier to scan the code. This part is especially helpful when the code might be slightly damaged. The part of the code that covers a larger area of the code is the Data and error correction keys, picture P.P. This is the part that stores the actual information and data we want a code to carry. Last, but not least, we have the Quiet zone, picture P.P. This part consists of a sort of frame around the entire code. This frame is necessary for the scanner to distinguish the actual code from its surroundings. Using simple tools, a scanner can read the information from all these parts and decode the data stored in it, which can then be used as desired. In many cases, the code might store an “identifier”, which can then be used to fetch more information, like for the regular 1D barcodes.

|  |  |  |
| --- | --- | --- |
| Picture - Positioning marks | Picture - Alignment mark | Picture - Timing pattern |
| Picture - Version information | Picture - Format information | Picture 15 - Data & error correction keys |
| Picture - Queit zone |

* 1. Comparison

Both types of codes have their own characteristics and properties that distinguish these from each other. Both have some strong sides as well as some weaknesses on the other side. Choosing the right code might depend on several conditions such as efficiency, capacity, size, etc. There is a point that can be written about in such comparison, however, in this section, we will mainly discuss the points that can be relevant for this case.

Let us first look at the 1D barcodes. These barcodes are seen almost everywhere and are simple in both appearances and in storing information. The 1D barcodes have been in the “market” for a long time and have gone through numerous iterations of development and improvements. The fact of making the relevant hardware easily accessible. There are different types of software and a barcode scanner that can simply be purchased. However, the issue with this is that to perform the desired task, buying all the essential equipment can be costly. Take a grocery store as an example again. You might need a barcode scanner to scan the code, a monitor and/or a machine to handle de decoded code, along with the other complementary equipment. In addition, the software on the “backside” to scan and decode. Nevertheless, this software part is required anyway, regardless of the type of the barcode – the same implies to the 2D codes as well. But the hardware equipment could be less for the QR codes.   
One major difference between 1D and 2D codes is their ability to store the information. Even though (some types of) the 1D codes can have a variable length, most of them can store a limited amount and type of information. Moreover, these codes have also lower data error and damage tolerance. E.g. if some parts of the barcode have ripped off, the scanner may not be able to read the information properly, since the light source might not be able to reflect the light back in the correct manner – making hard or even impossible to decode the code.

On the other side, we have QR codes that can store a vast amount of information compared to 1D barcodes. What makes these QR codes even more popular is their ability to store various types of data. This ability has made these codes suitable for nearly most of the domains. These codes are also better at handling data errors with the help of elements described in the previous sub-chapter. Other than that, the QR codes do not have to rely on larger scanning gadgets. A normal QR code can be read by using the camera of a device, e.g. a smartphone.   
However, these codes can appear to be complex if they are not the right choice for that specific purpose.

The table below sums up the arguments presented above.

|  |  |  |
| --- | --- | --- |
|  | **1D Barcodes** | **QR codes** |
| Capacity and type of data |  | X |
| Hardware (equipment) |  | X |
| Error & damage tolerance |  | X |
| Iterations of improvements | X |  |
| Simplicity (appearance) | X |  |

Tabell 1 - 1D codes vs. QR codes

* 1. Suitable code for this case

There are several aspects that can be discussed on which of the codes fits this case best. There are mainly three reasons for choosing QR codes rather than 1D barcodes for this prototype. The first fact is that the QR codes have the capability to store much more information than barcodes. In this case, our focus is on patient identification and therefore we might have to store patient-related information that can differ in size. For example, the QR codes can store a patient object with all the relevant information, which might not be possible with a barcode because of the type of information. Therefore, saving a tracked entity with a QR code seems to be legitimate than with a 1D barcode.

Secondly, using the QR code scanner would not require an external scanner to scan the codes unlike for a barcode. The Tracker app on a device would be enough to perform the desired task. Even though there is a scanner for a barcode on mobile devices as well, the QR compared to those seems to be more effective or responsive. We tested this by downloading some barcode scanner from the Google Play store. Many of the barcode readers did not scan the barcode at all, and others were able to scan to some specific types of codes only. In other words, the barcode scanner on a mobile device does not work optimal compared to the scanner for QR codes on a mobile device. To make it work in a real context, like for our case, the barcode would require an optical scanner to perform the desired task, but not the QR codes.

The third consideration was the size of the barcodes. From the data gathered from the first field visit, we observed that barcodes being used for Baobab are quite wide, and thus cover much of the front page of the patient booklet. This is also commented in Chapter 4.2.2. Therefore, we wanted to try to generate such labels that would use less space compared to the ones being used there today. The compact QR codes seemed to be the right choice there.

1. Method

In this chapter, we will write about the methodology for the research and how we went to work on the problem. Moreover, the chapter will also describe our journey throughout the project along with the pre-work we had to do prior to the field-trips.

* 1. Research Methodology

For this topic, we are conducting *qualitative research*, which focuses on understanding the context and direct interactions with the “real world”[[14]](#endnote-14) [ref]. Qualitative research can have many different types of researches, such as *action research*, *case study*, *ethnography*, etc. For this research, the closest approach we used was action research.

Action Research

As mentioned above, action research is a research method in Information systems, mainly focusing on trying to solve current problems in the real world while working on gaining some knowledge that can contribute to the academic world. The action research can have several forms, from being more formal and systematic to more reflective and personal approaches[[15]](#endnote-15). [ref]. Action research has been represented in various models, some as a process of a single cycle and others with several cycles until a satisfactory outcome[[16]](#endnote-16) [ref]. For our research, we have tried to work in more than just one phase, which is described more in the next section.

Our approach

Our aim with this research is also to contribute to something that might fit in the context we have been looking at and help the real-world users. Therefore action research has been a path we have tried to walk on for this project. For our research, we have tried to work in a way that we can process in several phases. Therefore, our approach has been close to the one described in figure F.F24. After identifying the problem, we tried to learn more about the context and then planned what we could do. Our prototype was developed based on the previous phases, and then the next steps were somehow iterative. The prototype was tested in the context, then some analysis and evaluations were made – leading to modifications that had to be tested again. In such processes, these cycles continue until a satisfactory outcome. However, since this is a master thesis with limited time, continuing with the work until the desired outcome is not fully possible. Thus, considering the limited timeframe, our “exit point” became some useful data and a prototype that could assist in further research.

Et bilde som inneholder tekst, kart

Automatisk generert beskrivelse

Figure 2 - Action research

Another concept we would like to mention here is the Participatory Design (PD) that is relevant here. This is a concept of involving the end-users in the process to reduce the gap between the developers/designers and the end-uses by having interactions between them[[17]](#endnote-17) [ref]. We sitting here in Norway can not fully understand their requirements, thus, letting the users participate in the process would more likely end in a satisfactory outcome. Therefore, this participatory action research was our way to work on our goals. There are mainly four types of PD; *Singular PD*, *Serial PD*, *Parallel PD*, *and Community PD*, which represent the different scales of involving the end-users.

* 1. Our Journey

The journey of this master thesis started for about one and a half years ago, like all the other master students. For this thesis, we had a problem that we wanted to work on during this timeframe with a qualitative research method. The plan initially was to implement the solution for the problem area, which included doing the field trip to collect data and test our solution. The solution was to be implemented in the DHIS2 app that is under development.

First field-trip

At the beginning of the third semester, we went for our first field trip to Malawi. The goal was to understand the context, collect data, understand the potential requirements and available resources. During this trip, we got to visit 4 different clinics, from where we could get some fundamental information for the next phases for our research. We not only learned about the context and end-users, but we also got to see some nice places in the county – mainly near Zomba because that is where we stayed. Our assessment of the first tour was that is went better than expected.

After coming back again, the main goal was to implement the solution based on the data we got from this trip. However, we had to make some changes in the plan due to some challenges that are specified in section X.

Second field-trip

The next phase after the implementation was to try our solution. At the very beginning of the fourth semester, we packed our stuff again and landed in Malawi again to test what we had managed to create. This time, we were there for a longer period – four weeks. We tried to plan our visits systematically to utilize the time we had in the best possible way. We made plans and appointments to visit 4-5 clinics – both at the facilities and the outreach clinics.   
There was a little slow start for our project – we were not getting the desired data. However, it started getting better after some time. So at the end of our trip, we could say that had something useful despite the slow start and all the other challenges we faced.

One thing we would like to mention here is the visits were coordinated. We were two groups that were supposed to go the visits together to work on our own projects. Therefore, our testing sessions were right after one another. This, somehow, affects a little on the testing sessions and the data quality. One clear example is during one of the field visits. The other started with their session, demonstrated their work, explained things, and asked many questions, etc. We were the next to talk to the same participant. When it was our turn, we clearly say that the CHW seemed a little tired and a little overwhelmed with all the new information. So, when we asked the questions we say that he was not able to answer properly. He replied shortly and quickly – just to get over it. As we can see, this does not give us fully trustworthy data.

By challenges, we not only mean challenges like getting useful data from the end-users, and having little time with the CHWs, but also challenges like driving long distances to get to the clinics, getting dehydrated due the hot weather, getting stuck in the mud due to heavy rains, or even getting extremely food poisoned that delayed our process. We also had to cancel a few field trips because of the rainfalls.   
We have a little story from our trip when our car got stuck in the mud twice on our way to one of the clinics. We drove early, as usual, for an appointment we made with the clinic. This was one of the clinics furthest away and had to drive on a muddy road, like the other clinics. However, due to heavy rainfalls, we had to drive very slowly and carefully to ensure not getting stuck in the mud. We had to stop at several points to and confirm if we could keep driving. After more than two hours, we finally arrived at the clinic, but when we arrived almost no one was there. The one we had made an appointment with, along with all the other CHWs (except one nurse) had taken a day off because of the weather. Unfortunately, we had to drive back with disappointment. On our way back, the roads got worse and even muddier. Our car got stuck at a point. No matter how much the driver tried, he could not get it out. With some (paid) help from the locals, they managed to pull out the car on the road again. We kept driving until we came to a point where our car got stuck again. The process repeated itself; locals offered help, helped us out and asked for money for this help.   
Despite all these obstacles, we think this was one of the days to remember. On our way back, we also stopped at a place to say hello to some children and play a little in the rainwater. There are two pictures from that day below.

Et bilde som inneholder utendørs, himmel, vann, bil

Automatisk generert beskrivelseEt bilde som inneholder utendørs, himmel, tre, bakke

Automatisk generert beskrivelse

Picture 17 - Rainy days

Changes in the plan

As mentioned above, we had to make some changes to the plan due to some challenges. The original idea was to work on an idea and make the implementation in the Tracker app. However, we changed the plan to develop a prototype instead. The main reason is the limited time we had. After coming back in the second last week of September from the first trip, we had only a few months to work on the development. We had a little of September, October, and November for this before we had to start preparing for the next trip.

Secondly, we found it hard to get enough help and information from the DHIS development team. With full respect to the team, we know that they have a busy schedule, but we felt that there was too much waiting to get in touch with relevant participants, and the whole setup of the app took way too much time. Sometimes, we had to wait several days to get a reply to an email. To utilize the time-optimal, we started creating a separate app with only that functionality which we wanted to test while we were waiting for the assistance. We were already in mid-October and we still did not have something useable in the tracker app. Therefore, we had to make some modifications to the whole plan to be able to finish this thesis in time. We got an idea to work further on that separate app and develop a testable prototype. Thus, the rest of the semester was used to work on that idea – creating a prototype that represented the solution we wanted to test and looked somehow similar to the Tracker app.

Thirdly, we observed that there is a little lack of structure in the way the development is done. This affects especially the students who want to work on such topics. The way code is pushed to the branch seems to be messy; often the code that is not working is being pushed to branch the others are working on. This does not seem to be an appropriate way to work. In addition, the master branch is not working, so instead of fixing it, there is another brach that has been forked that everyone is supposed to work on. Again, not a very good practice as the master branch is always supposed to be a working version of the code.   
There is little to no proper documentation, which makes it even harder to work on such open-source platforms.

\**Project management?*\*

* 1. Data collection methods

A big part of the research is about collecting the data that can be used in a meaningful way. How do we collect the data? There are various methods for collecting the data – both for qualitative and quantitative research. The main methods for our data collection were *observations*, *interviews*, experiments that were supposed to imitate the real world – we call it *testing sessions* here.

Observation

Observation is to examine people and situations as they are. One of the definitions is as follows; “*Observation of people in situ, finding them where they are, staying with them in some role which, while acceptable for them, will allow both intimate observation of certain parts of their behavior, and reporting it in ways useful to social science but not harmful to those observed*” (Hughes 2005) [ref].  
During our visits, observations have been one of the methods that have given us most of the data. Especially when we were trying to learn about the context. Watching the CHWs doing their routine work, observing how they communicate, where they use time under certain tasks, seeing how they try to hold the mobile device, etc, are just a few of many good examples.

Interviews

The next essential method interviews. It can be described as a technique for “*conducting intensive individual interviews with a small number of respondents to explore their perspectives on a particular idea, program or situation*”[[18]](#endnote-18) [ref]. There are different types of interviews. Such as, *structured*, *unstructured*, and *semi-structured*.   
For this research, we used all three types mentioned above, but semi-structures was the one used most. We prepared an interview-guide prior to the field visits, which was updated throughout the visit to adapt to the changes, and to modify in a way that we could get more relevant and useful data. Often, we had to ask some probing questions to get the details, e.g. “*can you explain what you were thinking in the* beginning?” One of the first interview-guides is attached as an attachment.

Testing sessions

This part was especially relevant for the second visit. To be able to test the prototype, we had to prepare some “tests”, both systematically and unsystematically. One session consisted of four steps; *introduction* – where we presented ourself and the aim, *demonstration* – showing the app and its functionality and explain it works, *scenario* – telling them to use the app and perform on given scenarios, and *interview* – asking the participants questions about the prototype using mainly semi-structured interviews. Table T.T shows some of the scenarios presented to the participants.

|  |  |
| --- | --- |
|  | *Scenario* |
| 1 | The patient is already registered in the system. But the booklet is missing the label and no one knows their unique ID. |
| 2 | The patient buys a new booklet because the previous one is full. The patient has the old/full booklet available. |
| 3 | The patient has lost their booklet, and do not remember their unique ID. |
| 4 | The printer is not printing the label. No lights are blinking on the printer. |

Table 1 - Some of the scenarios presented to the users

There were some other little techniques that were used to get relevant data, such as the “*Think aloud*”-technique. The participant here is told to say aloud what they are seeing and thing throughout the session. There is another technique to track the eye-movement of the user[[19]](#endnote-19). [ref]. There are some advanced tools to catch the participants' eye-movement and how much time users spend on looking at a particular element, but in our testing sessions, we chose to do it in a simpler way. Just by observing the participant carefully, and paying attention to their eye movements.

* 1. Pre-work

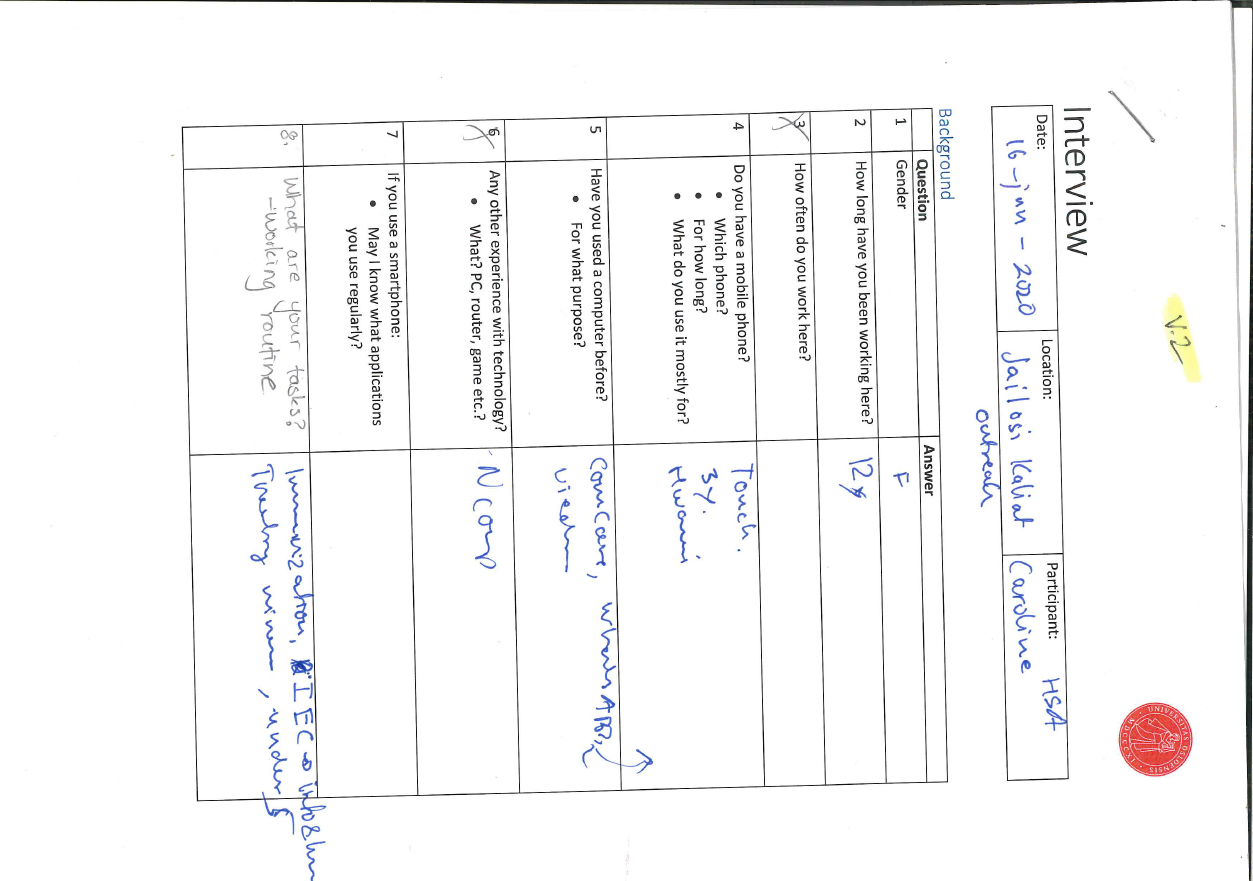
Besides all the research work, finding literature, writing the code, there is some other work that had to be done prior to the field trips. Our aim was to prepare in such a way that we could utilize the time optimally. Therefore, we tried our best to plan and prepare the material as well as we could.

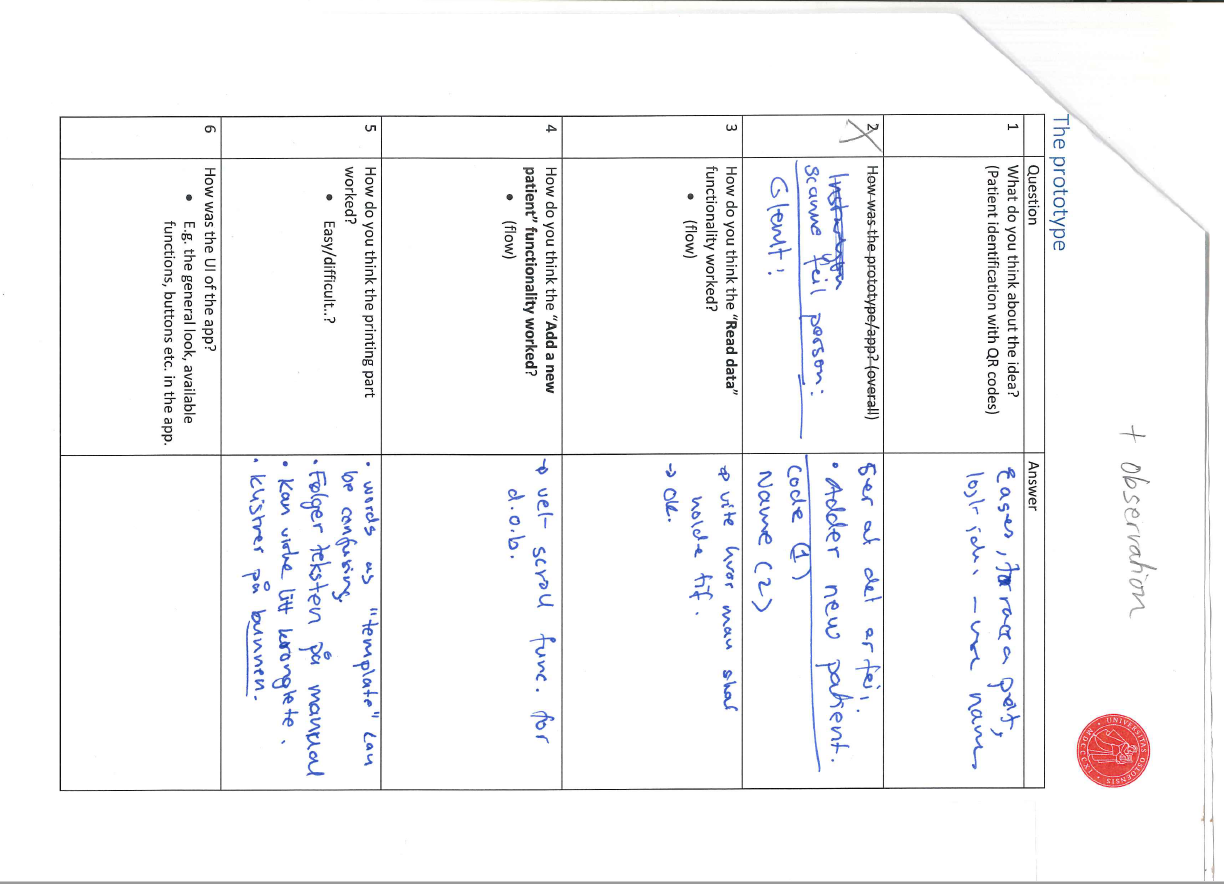
Goals

We think that to get the best out of the field trips, it is a good idea to have some goals. By goals, we do not mean large questions you would like to get answered, but rather smaller things you want to look at. To explain it in other words, we have an overall problem and the research question we want to work on. To achieve that goal, we think it can be vice to divide it into smaller sub-problems or questions that can be easier to work on. Therefore, we had smaller things we wanted to look into, to begin with, and then gradually going to the next ones.   
Prior to both of the trips, we wrote down some points we were especially interested in. Moreover, prior to the second visit, we made some other documents as well such as a timeline, weekly goals, a calendar for scheduling, and the desired data, etc.

Interview guides & scenarios

As mentioned above, we had some interview guides to assist us during the testing. These interview questions had to be modified throughout the testing phase. In addition, we had an extra guide with questions for those who had experience with Baobab. Pictures below show one of the early versions of these guides, and how they were being used. We created the document in a way that we could note down the information in a more structured way.



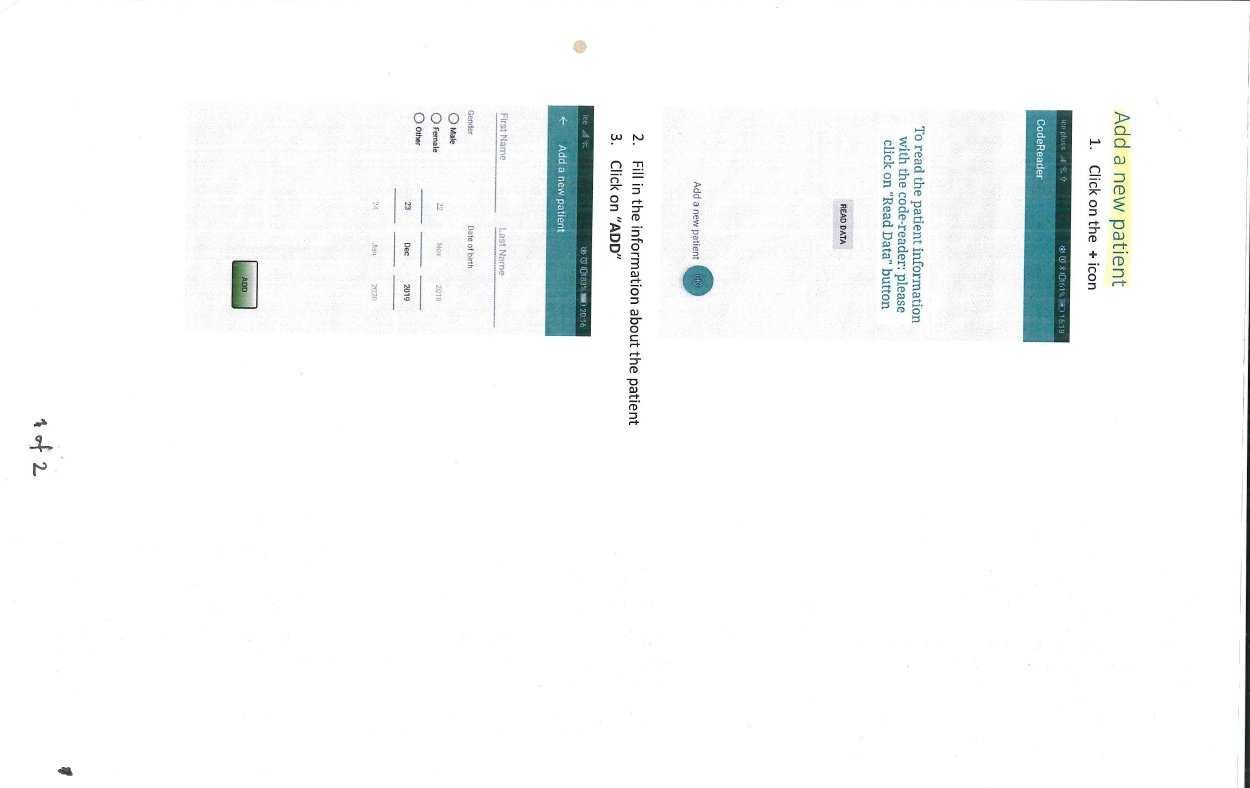




We also wrote down some situations that could occur in the real context. We had around 12 scenarios that we wanted the users to act on in the testing sessions. Some of the examples are already shown in table T.T above.

User manuals

Since this is a new thing for many of the users, we thought it might be hard for them to catch up with what we were explaining to them, and it might not be easy to remember everything after explaining it only ones. Therefore, using user manuals could be an ideal approach according to our opinion. Thus, our pre-work included making some user manuals that assist the users when they were testing the prototype. We made four manuals in total; explaining the basic functionality in the app, add a new patient, print the code, and print the code for an existing patient. To avoid using too much space here, we have added all the manuals as attachments to this thesis. Just to show one example; the pictures below show one of the manuals.

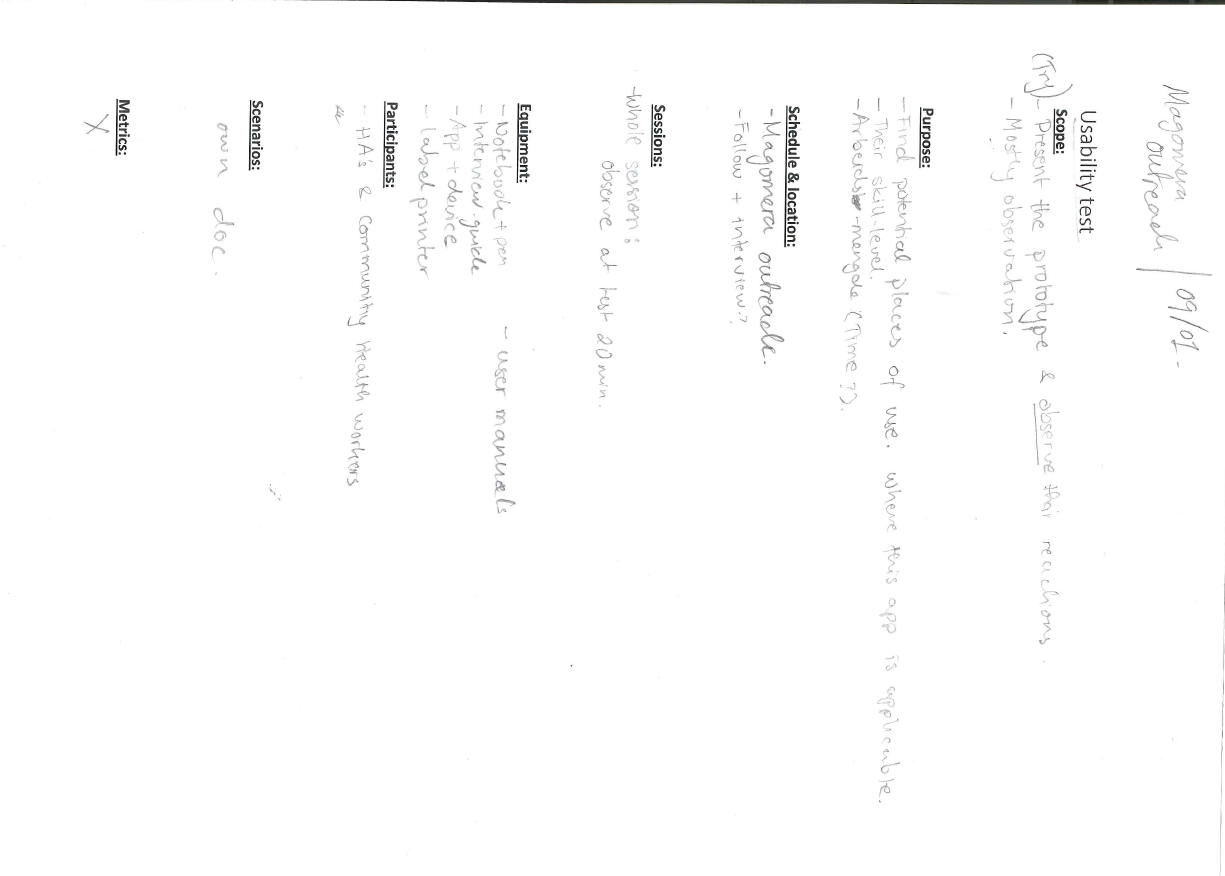




During many of the testing sessions, we not only observed but also got direct feedback from the participants that these user manuals were helpful.

Test-plans

Prior to each field visit, we had a *test-plan* to be better prepared and focus on what we wanted. On these plans, we often noted down some points on what we were going to focus on, what we needed for this testing session, etc. These were not followed very strictly, but we used these just to get ready for the sessions. Picture P.P shows one of the examples.



Dummy booklets

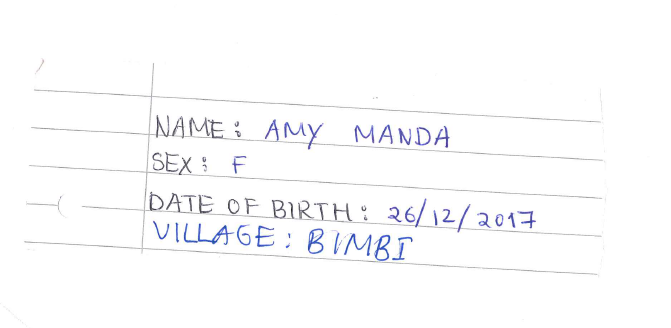
During this testing phase, our aim was to test the prototype with real end-users in the real context, which is one of the main points of such qualitative action research. Since we were trying to imitate the real-world, we wanted to make it easier for the participants and make it look more like actual scenarios. Therefore, we made plenty of fake patient booklets that could be used during the tests. This was to keep the flow. E.g., when a scenario, “*This is a new patient, who is not registered in the system*”. Here, we wanted the user to complete the whole action so that we could observe the overall workflow. Therefore, having some fake booklets that could be used to attach the labels on, seemed important and useful. E.g. we wanted to see where the users would most likely stick the code labels. We think asking them this directly would not give the same data, as they might think a little before answering and try to come up with a “suitable” answer. Therefore, the best way to understand this was to directly observe them. Picture P.P shows the booklets.



Picture 18 - Dummy booklets

Test-patients

To have a better flow during the sessions, we thought it would be easy to have some examples of patients ready, rather than telling them to assume a patient and come up with random names, etc. This could a little unnecessary use of time. Therefore, we prepared some small notes with some examples that we handed out to the users when needed, so they could focus on the actual task. Picture P.P shows one of the examples.



Picture 19 - Test patient

Obviously, we also added some test-patients in the system so that tasks such as finding a certain patient could be performed without any hassles.

QR-codes

We not only wanted to test what is good but also the things that do not work. Cases where unexpected situations occur. We printed some codes that had wrong patient information, codes that did not work, etc., and attached to some of the fake booklets that were given to the users. We got quite useful data during observation in these testing sessions. How they handle such things can have a huge impact on data quality in the system. Picture P.P shows a dummy booklet with a normal looking QR code, but this code has wrong patient information.



Picture 20 - Fake booklet with wrong QR code

\*recordings?\*

As we have seen, there is some pre-work that had to be done. All the planning and work takes time, which is one of the reasons to go for a prototype rather than using plenty of time for the implementation in the Tracker. What we wanted to focus on required such tasks, therefore, this was the best way we could utilize our time according to our opinion.

1. Existing Information Systems

Let us assume that we are standing on a point, from here we can look in different directions. This point is where we have the problem or case for this thesis. By looking back, we can see what is already there that might be relevant for this case. However, after having gathered the significant information and the data about the fundamental resources, we come to the point where we say; *okay, what next*? Thus, in this chapter, we will be looking “back” at what is there already, and then on what options it gives us with the available technology and resources. Then, in the end, try to look at the question we addressed here; *okay, what next?*

* 1. Paper-based system

The first common element we see in all the clinics we have been visiting is the paper-based system. All the clinics have manual *registers* that are being used on a regular base. A register is a large book with information to fill, and they are used to enter and store the patient data. Picture P.P shows one of such registers. These registers are mainly program-based registers – therefore, there is quite a variety of such registers. E.g. a register specifically for HIV patients. However, having various registers for different tasks seems to be a good practice even though it can appear to be messy. Having separate registers helps them to keep the order and maintain the data “easily”.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

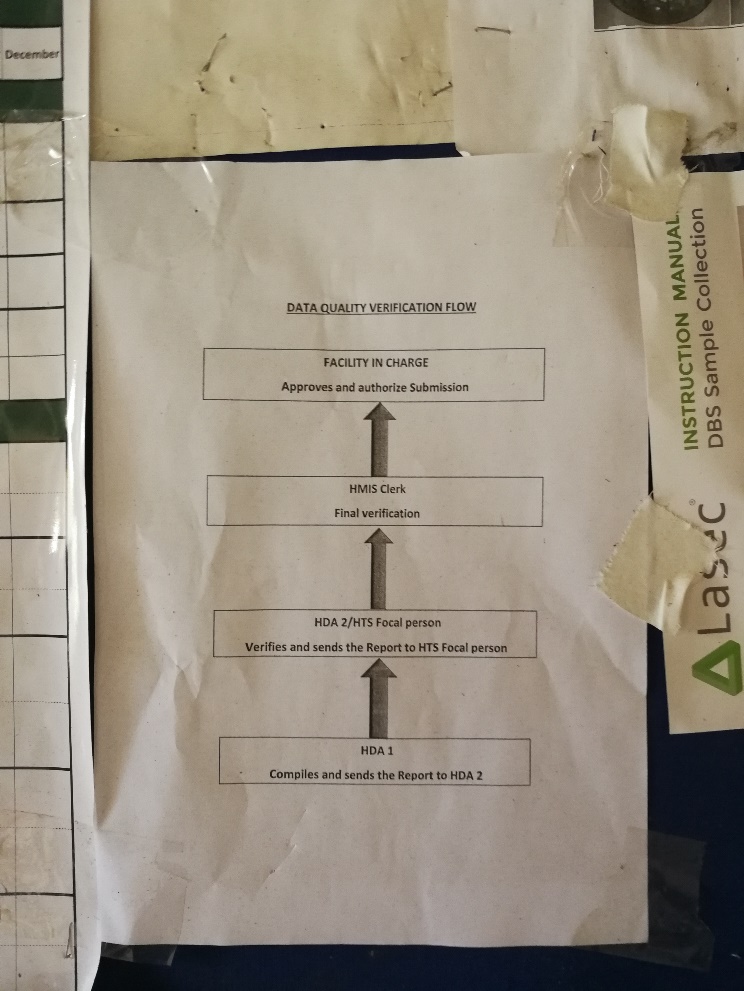
Picture 21 - A manual register

To have a closer look at this, we can look at ART, antiretroviral therapy. For having records about the patient and for following up the patient, there some *treatment cards* being used. A treatment card is a sheet for a specific program and is being used to store the information about the patient. Each card has information for one year. Each patient has their own treatment card that is being stored in their clinic[[20]](#endnote-20).[ref]. On the picture below P.P, we can see an example of a treatment card. For each visit, their representative card will be updated along with their *health passport* they carry. A health passport is a little booklet that is issued by the Ministry of Health, and it is to be always carried by the patient when visiting health facilities or other health-related gatherings[[21]](#endnote-21) [ref]. Besides the patient details about the patient on the front page, the booklet contains the health information about the patient that is written by the healthcare providers. E.g., by a CHW, community health worker. These booklets have also different colors; for male, female, child, etc. We can see one example of a yellow health passport for women in picture P.P.

This practice of storing the treatment cards may now vary depending on what systems the clinic is using. There are certainly several other details in this paper-based system, but we try to limit it by only writing about information that is relevant to this topic.

Another thing which we think is important to look at is the workflow in such systems. How the actual flow of working is will vary depending on the task or the aim of a task. Just to get to the actual point, we can look at one of the examples during the observation at the first field visit. The details might not be up to the mark. The patient comes to the first CHW who notes down the data entry in a register and booklet. Then the patient goes to the next room, where she waits for the treatment card. The CHWs there enter the data in a register again, finds the correct card and hands it out to the patient – who again goes to the next room for examination. During the examination, the nurse writes down on the treatment card and the patient booklet and sends the patient for further process. Depending on the consultation the patient will act accordingly. Let us say that the patient is advised to get tested for malaria. Then the patient will go to that queue for tests. Here the CHW will again note down the information in another register for malaria – for the data record. After the test, the patient may go to the pharmacist to collect the drugs, who will again write down the information – another register which is meant for drug stock. Before leaving the patient returns the treatment card that will be stored at the clinic for future use.

Now, what about the data that is being entered at several points? There are more steps before the data gets into the DHIS2 servers. The exact process may vary due to different practicalities in the clinics. We can see one example of some of the steps in picture P.P. This is taken in one of the clinics in Zomba for general information for the CHW working there. Looking briefly at the overall process of data, we can see that data entered in the registers is used to create regular reports that are sent further to be reviewed and altered before ending up in the DHIS2 servers.



Picture 22 - Data flow

Another thing to mention here is the reason for using this system. The obvious reason for continuing with this system is surely the budget. As we have seen in chapter 2X, there is little to spend on health information systems from the government. Implementing and maintaining other bigger systems alone without any donors could be a hard task for the Ministry of Health. Therefore, this choice seems reasonable. However, this initiates another issue that is data quality. This issue is discussed in several other pieces of researches too, talking about the reasons and ways to avoid it. E.g. Lippeveld in this paper discusses some of the reasons for low data quality[[22]](#endnote-22) [ref]. We will not go into the details of this issue, but we think it is important to mention it since it has relevance to our topic.

* 1. Baobab

As mentioned earlier, clinics have different systems and practices of their own. Some of the clinics we have visited are using a system named Baobab. It is a web-based application for managing and tracking patients during a point of care. It is also being used for reporting. For the database, they adopted OpenMRS, an open-source project[[23]](#endnote-23) [ref], database model, which is MySQL, which was explained by one of the ex-developers for Baobab. Baobab is mainly financed by different donors[[24]](#endnote-24) [ref]. From one of the developers who worked with development, we got informed that the donors have now pulled out the donations, so the organization of the will operate differently.

If we now look at the functionality, it may differ depending on clinics again. One of the clinics we went to had NEMR, HTS, and ANC – Antenatal Care. We can have an overall look at this system and the main area of focus. In Baobab you can add a new patient in the programs that are available. Adding a new patient requires essential information about the patient, such as name, gender, village, etc. After filling the required information, we get a label with a barcode and some patient information on it. This barcode, which will be attached to the patient health passport will be used to search for that patient in the system in the future. It also allows the user of the system to print labels with other patient-relevant information that is attached in the booklet, so the user does not have to write it by hand. It also allows to find and track a patient by their name and other patient information.

Now looking at the “appearance” of this system. For this system to work, it needs a machine or the touch screen computer, a printer to print the labels and a scanner to scan the barcodes, seen in picture P.P. The printer needs ink patrons and labels. Apart from these, it also requires internet connection (LAN), (local) server, and electricity obviously.

Et bilde som inneholder innendørs, bord, sitter, vegg

Automatisk generert beskrivelse

Picture 23 - Baobab workstation in one of the clinics

However, the clinics where we saw this system, it is not being used as the only system. The manual registers are still being used even though that certain task is possible in Baobab. During our interview with the CHWs working with the system, we asked them why they are using both systems. One of the answers we got was something like this: “*For cases like this*”, he said and pointed at the screen shown in picture P.P. In other words, in case the server is down, they will continue adding the data in the registers. Another answer we got was that it helps them to tackle the workload. In situations during many patients, they might e.g. make two ques for the same task, where one is using Baobab and the other one is writing in the registers. They will later add the data entered in the registers into Baobab and vice versa.

* 1. CommCare and others

As we have mentioned about the fragmentation in chapter X, we can see that there are other systems that are being used too. One of the technologies that we came across was the CommCare application. CommCare is an open-source platform that aims to provide customizable technology to low-resource communities[[25]](#endnote-25) [ref]. It has two main components; CommCare Mobile and CommCare HQ which the website used for reporting and management. The technology has been used for several functionalities such as data collection, job aid, decision support, etc.

The CommCare app is designed in a way so that it can easily assist CHW in their work. During our fieldwork, we got to meet a few workers who had some experience with this app. We even got to see the app on their mobile phone, and what they used it for. Unfortunately, due to some technical issues, the app was not being used as it was meant to be. E.g. one of the health workers explained that the app is not working for more than 3 weeks. He had reported this issue, but they did not get any response back. Talking to another health worker, we got information that she was using the app to enter the data afterward, even though the app is designed in such a way that the health worker enters the data as they are attending the patient. For example, writing down the pulse in the app and other information to get a diagnose that the health worker can use to give the patient drugs they might need.

Despite the user-friendliness and the easy user interface, the app has still some obstacles that are not fully able to cope with the context. In one of the studies done one using Electronic Data Collection methods, one of the issues mentioned in the paper is the connectivity issue for CommCare[[26]](#endnote-26) [ref].

Referring again to the previous chapter 2x, there is a diversity of systems being used in the country. The limitations in the resources have caused several organizations and donors to contribute to the health domain in the developing countries, resulting in a diversity of running systems and projects. Just to mention a few examples; Kunnika, WHO, EGPAF, Norad, etc.

1. Patient identification
   1. The diversity of systems

In the current situation of the health system in the Zomba district, we saw that the diversity of the systems in the health clinics is mostly caused by two main reasons. Firstly, the system and patient registration are program-based. E.g. the HIV program has its own registers, and the ANC has its own. Accordingly, each patient gets a “unique” number within that specific health program, which means that patients end up getting several “unique” numbers. The other issue is the number of different donors for these program-specific systems. In 2018, the total donor funding represented 66-70% of the health care costs[[27]](#endnote-27). To some degree, the sponsors create and organize the systems in their own way even though there is an “agreement” on some common principles between all the stakeholders[[28]](#endnote-28). Even though the article is published some years ago, but many of the points are still valid and relevant today. Which might end up creating a situation in a clinic where there are different types of systems being used at the same time. E.g. even though the clinics have the BaoBab system implemented in their clinics, many of them are still using the manual paper-based registers along with the CommCare app.

* 1. The potential solutions for unique ID

We have tried to look at the different possibilities for the solution to this issue. Certainly, there are many possible solutions, however, we have tried to look at three of them – *partially local*, *national-ID based*, and a *combination of both*.

* + 1. Partially local

The main idea is to have a unique ID that uniquely identifies the patient within the system. If we see it from a health worker's point of view, each patient will have one unique ID that can be used to track a patient. That one ID can be used to find a patient across the different programs. That is what makes it different from the systems being used there today. As mentioned above, the patients might get several IDs, depending on which programs they are enrolled in.

The DHIS2 database currently enables to track individuals across the programs.[[29]](#endnote-29) (ref). E.g. if you are in a specific program, you might be able to see the other programs that patient is connected to depending on the role you have. On the user level, the patient is given one unique ID which can be accessed across the system. However, this current approach has a similar behavior as the other existing programs to some degree.

Our idea here is to possibly use the same identifier on the user-level. In other words, use the same identifier for the codes that will be printed on the labels. Those codes can then be used by health workers to identify the patients – e.g. by scanning their QR codes. As mentioned above, the idea is to use the same identifier across the programs, such that it makes easier for the workers to track the patients. This can be achieved by linking the program-specific identifiers to the one unique identifier for each person. In other words, each (relevant) program-specific identifier will have access to the patient identifier. The figure below illustrates how this approach can be realized.

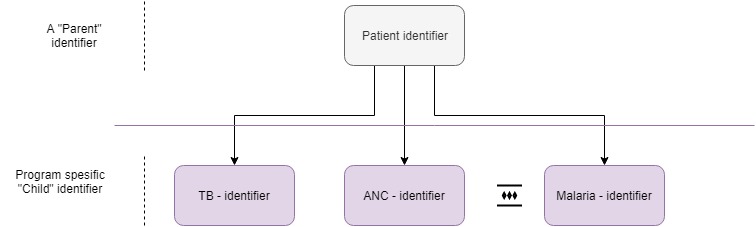


Figure 3 - Partially local

The *parent* identifier might give you information about the other programs for that specific patient – at least what other programs the patient is enrolled in. The program-specific identifiers are represented as *Child* identifiers here. The information you get access to might be limited depending on your role and which program you are in the app as mentioned above.

Criterion

The first motivation for this approach is the level of complication in the implementation – this solution might be “easy” to implement. Easy in that sense that this solution would require less time to implement compared to the others. Moreover, this solution can be solely implemented by e.g. DHIS2 without involving other parties. More about this choice is written in section 5.3, Solution for this prototype. Introducing a new identifier is not only a good theoretical concept but also a solution that is practically implementable.

Secondly, this solution is suited for the limited timeframe we have for this master’s project. The goal of this project was to implement a possible solution for QR code reader for patient identification. To achieve that goal, we had to limit the scope and consider the limitations while choosing the possible solution to implement.

Drawbacks

Every solution has its own challenges, and so does this one. As mentioned in **section** X many of the systems being used currently have a top-down structure. Looking at the overall structure of the various health systems and the program-based systems have a *silo-structure[[30]](#endnote-30)* which means that most of the systems are running as induvial silos, without even having any interoperability between the systems. Baobab is one example of such a silo. Introducing this approach could lead to having yet another silo in the various health system.

* + 1. National ID-based

The second approach is a little different. Here, we are trying to zoom out and looking at the system through *one* single glass. In other words – a national health system. The main idea is that the national ID of a patient is used as a unique identifier. The national ID would be the one that will be used by the health workers on a higher level, and the system at lower level might link the national IDs to the system Ids being used at the lower levels, in such manner that by searching a patient with their national ID would give all the desired information. The architecture of this approach appears like the one described in the previous section in the sense that one unique ID would give access to the underlying program-specific IDs.

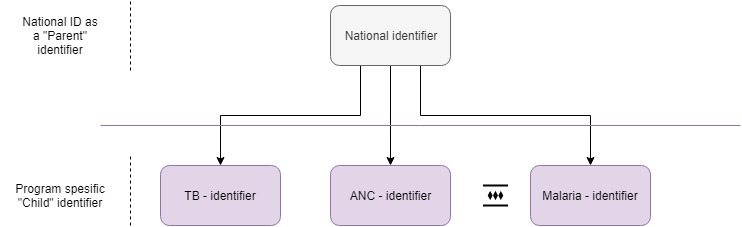


Figure 4 - National ID-based

In this approach, the national ID would be the parent identifier as we can see in figure F.F.

Criterion

This suggestion of using the national ID as the identifier is a tremendous thought, theoretically. Whether it is practically possible is still an unanswered question. Certainly, this solution would be the one many would desire due to its ability to identify individuals across the systems.

Drawbacks

However, this solution would not fit well for this master’s project due to the limited timeframe. This solution would be an enormous project that would require a longer time period and involvement of not only the government but also other entities such Ministry of Health and organizations running those various program-based systems in the health facilities.

Using a common identifier across systems would require the existing systems to be modified. Since there are different donors for those systems, it might be a hard task to convince the authorities to invest in this solution. They might not be as motivated to contribute to this modification.

Yet another challenge that we can see for this solution arise when we have patients who are not citizens of the country. They will not have the national ID, therefore is no unique identifier could be generated. This issue can probably be solved by having an option for foreign patients. The system might give the option to register that patient based on another ID shown by the patient – then it would generate an identifier for that patient. The identifier could somehow be able to show the workers that that specific patient is not a citizen in the country, e.g. by adding two letters in the identifier which stands for the country code of the country that the specific person belongs to.

If we try to look at another scenario with newly born babies or infants, they will most probably not have an ID in the very first time period. Therefore, they might have to face the same issue mentioned above if they needed treatment. This could again create a hurdle in the process of patient identification. Some of the issues mentioned here are similar to the case study done by Østmo in South Africa, where she also talks about the hybrid systems[[31]](#endnote-31).

* + 1. A combination of both

The third and last potential solution is trying to combine both solutions mentioned above. The concept here is that primarily the national ID will be used as the parent identifier, however, since not everyone owns a national ID it would arise a challenge like mentioned in the previous section. To deal with this problem, combining both solutions would give us a third option. Using a combination of both approaches means that the main identifier would be the national ID as described in section 2.2.2, however, there would be an additional opportunity here. If the patient does not have a national ID, there would be a possibility in the system to generate a new unique ID for that patient, like described in section 2.2.1. An important factor to notice is that the ID that would be generated for the patient would *not* be a new national ID – that identifier would be the one as in the first proposed solution. In other words, this architecture would give the possibility to have two types of identifiers. *Either* national ID *or* the one generated in the system at the lower levels.

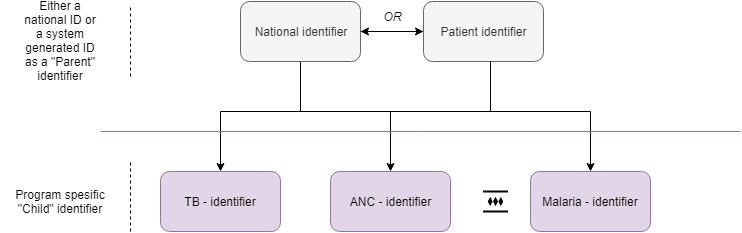


Figure 5 - A combination of both

The system will not allow both identifiers actively being used at the same time. E.g. the patient ID can be a “previous” ID for a patient who has recently obtained the national ID, which means that now the active identifier will be the national ID and not the one being used previously – the one enrolled by the system.

The major thought behind this approach is to, first, deal with the challenge that not everyone has a national ID card. The second is about developing an architecture that is open for future development. At the department of computer science at the University of Malawi, many are aware of this issue of national IDs and their usage. Therefore, if the government enforces everyone to get a national ID card to access the governmental services, this solution could adapt such modifications easier than the others.

Criterion

The major thought behind this approach is its ability to adapt the future development modifications. Since there have been discussions about implanting the national ID as the national identifier, this solution could readjust the system to fit these criteria. However, whether an optimal solution could be discussed.

Drawbacks

The first point that makes this solution challenging its complexity. The solution could be hard to implement at lower levels, making the overall structure and implementation of the whole app arduous. Moreover, this knotty implementation would not fit the limited timespan we have for this project.

Another issue that could arise at the user-level is the ability of health workers to identify patients using two different identifiers. It could be slightly confusing for them to use both identifiers in parallel. In our option, the issue can be resolved with accurate training and information about this solution.

* 1. Solution for this prototype

What solution to choose for this prototype depends mostly on the context. As mentioned in chapter X.X, we know that there are several points that affect the decision. The solution we thought fitted the most is the one described in section 5.2.1, Partially local. However, there is a difference between the implementation of this prototype and the DHIS2 app. This prototype app implies the first solution, Partially local, which is described in the next paragraph. However, the DHIS2 app does approximately the same, but not in exactly the same way. How the patient ID is stored and used is already described in Chapter 3.

As we know, the Partially local means that the patient ID is available locally in the system. However, what makes it different from e.g. the registers being used today is that the patient ID is a *parent identifier*, which gives access to the program-specific identifier. Therefore, with one unique patient ID, you can have access to the desired program.

We can look at figure 1 again to understand this better. The figure shows one *parent identifier* that is supposed to be the patient ID in this prototype. This ID is generated by the system when a new patient is added. This is will be the parent ID, and can then have other program-specific identifiers as *child identifiers*. Let us look at one example. We are adding a new patient in the system, and this patient will also be added to the TB program since that is the purpose of their visit. Picture 16 shows the patient details, along with the patient ID, which is *000058*. If we click further on the TB program, we can see that the patient ID is already filled and is the same. However, there is a field for another ID that is specific for that program, seen in Picture 17.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture - Patient information

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Child identifier

Parent identifier

Picture 25 - Patient information in the TB program

We have seen how this solution works in practice. However, what is the reason behind this choice? As we mentioned at the beginning of this section, the context has a lot to say. This solution is the that can be practically possible with the limited resources, compared to the two other suggested in this chapter. The National ID-based solution would require more involvement of other parties that might be hard to achieve because other parties might not have the same type of motivation to do the necessary work. The third option is a solution that would require more time for the implementation, and that is one of the challenges with this master topic – we only have a limited timeframe.

Another reason for not choosing the other two solutions is the limited technical knowledge of the CHW’s, and the fact that a large number of the patients in the health clinics do not own a national ID. Not having a national ID could be solved by solution 3, A combination of both, but then we meet the issue where CHW’s would find it difficult or confusing to use both types of identifiers. We can see from the interviews from the fieldwork, that most of the CHW have limited knowledge about the technology. This confusion could then lead to mistakes in the data.

This decision was made after the first field trip when we had talked to several CHW about their experience. This gave us some ideas. This thought got even stronger after the second field trip when we interviewed the CHW in detail. Let us take an example from the interviews. We asked the participating CHW to find a solution when a patient loses their booklet. Many of the participants got confused and suggested to just add that patient as a new patient. This might solve the problem for them, but we can clearly see that this would make duplicates in the data, which is one of the main problems this topic is trying to reduce.

1. Platform Technology, and Design & Implementation
   1. DHIS2 and open-source platform

The District Health Information System (DHIS), is an open-source platform that is being used globally. DHIS2 is developed by the Health Information Systems Programme (HISP), which is a research group at the University of Oslo (UiO). DHIS2 is now the world’s largest health management information systems platform that is being used in 67 different countries[[32]](#endnote-32) [ref]. There are ongoing development and adaptation to new technologies on this platform that is making it grow even more. This success is mainly caused by DHIS and HISP community and networks through different activities and projects being carried out. Ph.D. and masters have also made a valuable contribution to the expansion of this platform[[33]](#endnote-33) [ref].

DHIS like the many other platforms have the ability to let the developers customize the application to fit in the context. The platform core has the code and functionality that is limited to the main developers of the platform. However, other parties are free to use this platform and make modifications to the parts that are changeable. Figure F.F from a paper illustrates this concept of platform architecture for Participatory Design for the platform very clearly[[34]](#endnote-34) [ref]. This figure can also fit for this platform definition here. The Generic core is the part of the platform that is limited; however, the outer layers of the platform are more open to changes and modifications. By utilizing the core, and other bundled apps, the platform technology can be developed to some quite useful and powerful systems.

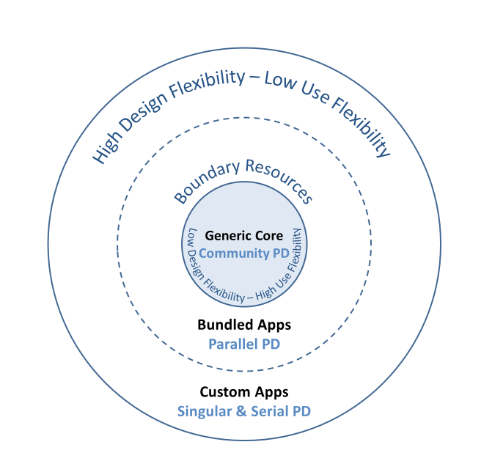


Figure 1 - A platform architecture

A concept we would like to mention is the Participatory Design that is explained in chapter X and described in detail in a paper by Rolland et al. 20 [ref]. The reason why we are bringing our attention to this concept here is to, again, enhance the success of platform development. Apart from the reasons mentioned above, the feedback from the user community has played a huge role. No matter what PD is used, direct interaction with the end-users has shown to be quite useful.

Tracker

The DHIS Tracker app is certainly the application developed by the DHIS platform. DHIS is being used at the district level for data records and management. However, the aim of this Tracker app is to bring this technology to the clinics-level. This is an Android app that can be downloaded from the Google Play store. The app allows offline data capture, and the data is automatically synchronized whenever there is internet connection29 [ref]. Through the API available for students and developers, we were able to contribute with our parts of the project. However, due to some limitations mentioned in chapter X, we ended up developing a prototype to test the concept rather than adding the functionality in the Tracker app.

“Okay, what next”?

Referring to the question we mentioned at the beginning of chapter X. Now we have seen what is there already and opportunities this open-source platform can give us. From this point, we will try to present the work we have been doing, the data we have collected, the decisions we had to make, the limitations we had to consider, and try to adapt the opportunities with the limited resources available. All this, by having the context and the ground rules information in mind.

* 1. The prototype

For the implementation of this project, the “obvious” choice of an editor was the Android Studio, with Java as the implementation language with android programming. The more technical details of the DHIS2 app is already mentioned in chapter X. Many of those details apply to this part of the implementation too.

The main goal, to begin with, was to implement this functionality directly into the DHIS2 code. However, due to some complications and limitations, we had to make some changes to the plan. More about those complications is written in chapter X. In this section, we will focus more on the implementation and the choices made for the prototype which was developed to be tested.

* + 1. Functionality

This prototype app aims to provide the functionality that fits the idea for this topic and only this. The app provides the user with the ability to retrieve the data about a patient by scanning the code for that patient. The user can see the information read from the code and be able to use that information to perform other desired tasks, such as registering this patient in the TB program.

The app gives also the user the option to register a new patient in the system and then print the label with the code, which can then be attached to the patient’s health passport. All this can be done from the same app, just within a few clicks. The only thing which requires the user to use other applications to finish the task is the printing of the label. Since there is no direct way to print the label from the same app, we had to use the app for the label-printer to do so. However, the user interface is in a way that makes the user quite easy to achieve that goal. How it can be done it explained in more detail in section X.

* + 1. User Interface

The first thing to note here is that the UI for the prototype app is quite simple. This is mainly due to two reasons. Firstly, the users of the app are mostly going to be the different health workers in the clinics in Malawi. After my first part of the fieldwork, I observed that a large number of health workers have limited knowledge about technology, more specifically electronic devices, and they also tend to have insufficient education. This is not only figured out during our fieldwork, but it is mentioned as a common issue in many other types of research. This paper about innovation in developing countries also writes about this issue[[35]](#endnote-35). [ref]. Therefore, we thought that we should try to avoid using a complex and advanced UI in this app. This UI was done by having usability in mind. When it comes to using such technology in health systems, the user-friendliness becomes even more important. Accordingly, the app’s simple UI makes it easy for the user to browse through it and use it for the desired task.   
Another phenomenon that can be brought up here the *minimalism*. This might be used more in a philosophical aspect. However, we think that is a phenomenon can be applied here as well. The main point of minimalism is about having little and focusing on that.[[36]](#endnote-36) [ref]. It can be easier for our brain to remember fewer things, and formerly utilize these things in a useful way. Therefore, having the most important functionality in the app makes it easy to remember and use.

Secondly, this app is just a prototype. This means that there will most probably be several elements to be modified, removed or added. According to our opinion, using time on designing a great UI on just a prototype is not the appropriate usage of the limited time we have for this master project. Spending a larger amount of time on just the appearance of the app would delay the overall schedule of this project.[[37]](#endnote-37) [ref]. However, we agree that the UI or the appearance of the app is important for the user experience. Still, there are some limits that need to be held. We decide to limit the design to a degree that could cope with the issue mentioned in the previous paragraph – user-friendliness.

While designing this prototype app, the original DHIS app was kept in mind. On most of the activities within this app, we tried to make these look as similar as possible to the original DHIS tracker app. E.g. the inputs values in forms for the different programs.

|  |  |
| --- | --- |
| Et bilde som inneholder skjermbilde, telefon, overvåke, mobil  Automatisk generert beskrivelse  Picture 26 - Child Program in Tracker app | Picture 27 - Child Program in prototype |

Picture 28 – Child Program screen in Tracker vs. the prototype for testing

The idea behind doing so is basically to avoid confusing the users. If there are some users who have been using the app, it will be easier for them to understand this part and can learn to use this quickly. Making the app alike something being used already could also reduce the mistakes or errors, mainly human errors. As we can see in the picture P.P, both forms have similar inputs and the position of the input fields. Another example is shown in picture P.P, where the app icons are similar.

|  |  |
| --- | --- |
| Et bilde som inneholder tekst  Automatisk generert beskrivelse  Picture - DHIS2 Tracker | Et bilde som inneholder svart, elektronikk  Automatisk generert beskrivelse  Picture - Prototype |

Picture 31 - App icon in both

* + 1. Use of the prototype

As we argue in some of the previous sections, the app can be easy to use. The prototype tries to give some basic information throughout the app to help the user to use it in the right way. E.g. the main screen, it gives the user two options – either read the data of a patient or register a new patient. This is shown in a screenshot of the main screen, P.P. Attachment A.A, which is a user manual, explains how to use this prototype.

Et bilde som inneholder skjermbilde

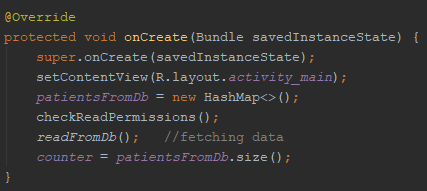
Automatisk generert beskrivelse

Picture 32 - Main screen in the app for testing

Since we know that the users might have limited knowledge, we have tried to make a sort of user manual that can help the users. It goes through all the steps in the prototype and tries to explain how it can be used. More about these user manuals are written in chapter X.

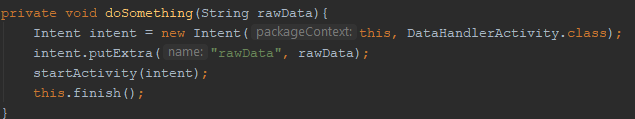
* + 1. Implementation details

When the app is launched and the *MainActivity* starts, the app will read the file stored on the device and then add all the data read from the file into a HashMap so that all the patient data is available for the app. Before this can be done, the app must ask for permission from the user to access the storage on the device, as shown in picture P.P. The app asks for permission only the first time after the app installation.



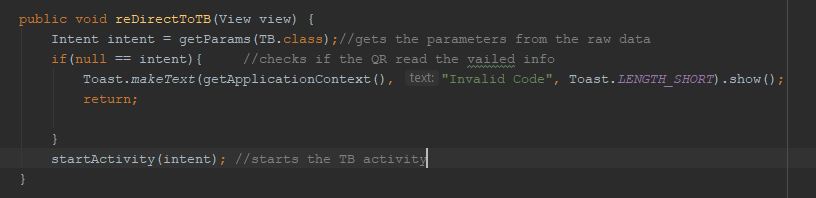
Picture 33 - Fetching the data from the file

When the user wants to read the patient data, the app starts a new activity with a *SurfaceView* where the user can scan the QR code by using the device camera. On the very first use of this functionality after downloading the app, it will ask for permission to use the camera. This choice will be saved for the future. By using *SurfaceView* and *CameraSource*, it will detect the code and then the information decoded from the code and used. The code snippet in P.P shows when the data has been encoded, it is sent to *DataHandlerActivity*, as well as it starts this activity.



Picture 34 - Data encoded from the QR is sent to the next activity, the DataHandlerActivity

The *DataHandlerActivity* does as the name says. It deals with the raw data that has been encoded. The data retrieved will be shown in a readable manner on the next activity and the user will be given some options to perform the desired task. By choosing the task, the user wants to perform, the app accordingly starts a new activity. E. g. the user chooses to click on TB, the app will start the *TB* activity. On this activity, the information read from that QR code will be auto-filled in the forms shown. The user can edit or add to the inputs field before he/she continues.

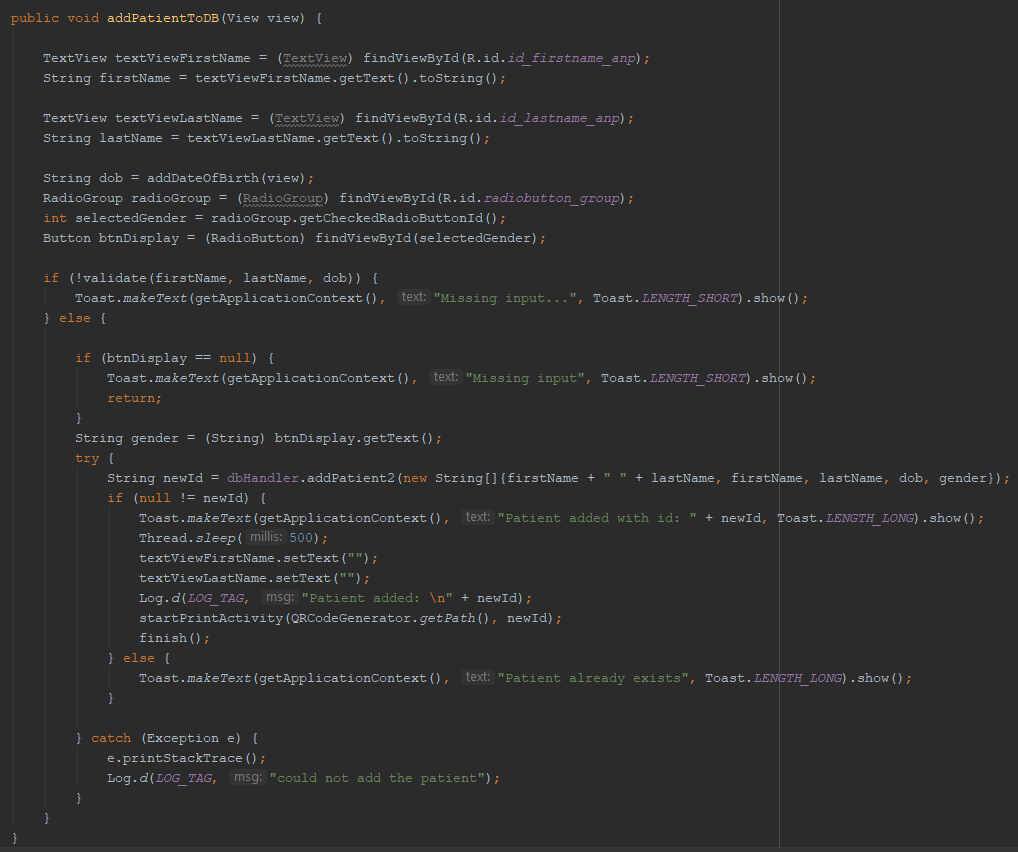


Picture 35 - Starting the TB activity

When the user wants to add a new patient, all the necessary fields must be filled accordingly. The AddNewPatient activity takes care of this part. It will first check the permissions to write to the device’s storage and proceed if permission granted. Then the input values will be evaluated before it is sent to the *DBHandler.java* class where it added the patient to the HashMap and then to the database file on the device. Picture P.P shows the method for adding a new patient.

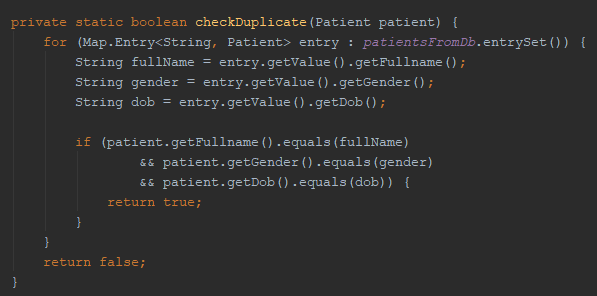
The DBHandler will generate a new unique ID for the patient which is stored along with the patient information added by the user. This is used as the key for that Patient object in the HashMap being used to handle the patients. If the patient was successfully added to that map, it will then be written to the file that stores all the patients. Each patient is added as one tuple in the file. An example of a tuple is shown below.

[000000, MyFirstName, MyLastName, Female, 10-06.1978]

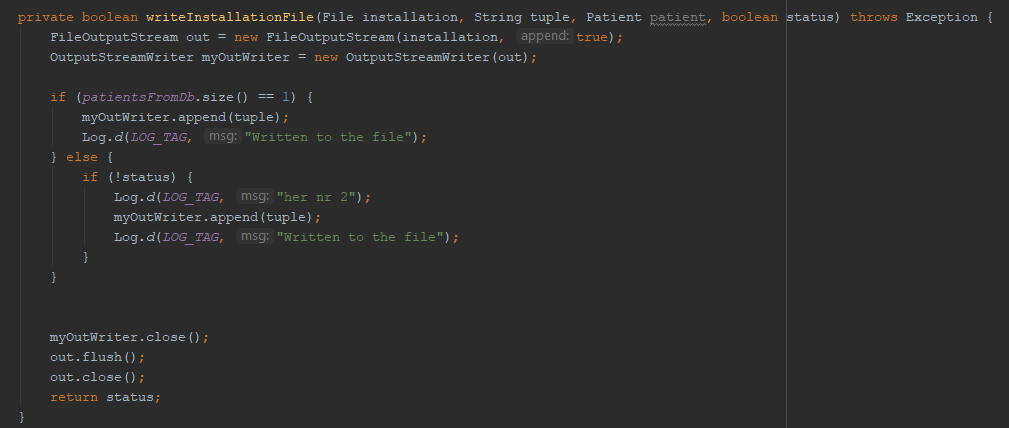


Picture 36 - Add a new patient to the database

The *DBHandler* will check if there is a patient already added with the same input. If so, the app will not add a new patient and alert the user about it. Picture P.P shows a code snippet of how the duplicates are checked. If everything is fine, then all the new patients will be added to the file on the device, picture P.P.



Picture - Checking for duplicates



Picture 38 - Writing to the file

Database

For this prototype, I chose to write all the data on a *.txt* file which will be stored locally on the device. The reason for this choice is again that this app is just a prototype to test the idea for this project. To create a dummy database for only this functionality would require time, which we did not have. Therefore, the easiest and quickest way to solve this was storing the data a file.

The DHIS has its database where the data is being stored. Therefore, if this prototype is to be implemented or merged in the existing app, this issue will be solved anyway as the data will then be stored in the original DHIS database. Thus, using time on creating own database would not be the right choice of using the time according to our opinion.

Storing the QR Codes

The *QRCodeGenereator* Activity does as the name says – it generates the QR codes with the information added for a patient and stores the newly generated code to the device. The codes are saved into the device’s memory in a *.png* format. These codes can be easily accessed as normal pictures. Each picture is saved with the unique id as the name of the picture making it simple to browse through all the codes.

However, there are some challenges associated with this way of storing the QR codes. The codes can start consuming more of the device’s memory in the longer term. If this solution of storing the codes was to be used in the real context, this could lead to this issue of load on memory. Nevertheless, this is a prototype to test the idea as mentioned in some previous sections, therefore, this choice of storing codes could work fine in this case. We will not be able to test in a fully real context, however, there will be a context that imitates the original context. More about how the testing environment will be is written in detail in section X.X(method?).

There are indeed several other ways of storing the codes, which can clearly work better in a broader context. Section x.x explains more about the possibilities.

* + 1. Contextual considerations

Even though this app is not the exact solution of the functionally that is to be implemented in the original app, it still takes some of the contextual limitations into consideration. After the first field visit, the very first concerns that come up are such as limitation in power supply, access to the internet connection, health workers with enough academic knowledge, and limited access to other resources such as the hardware are some of many other obstacles.

This prototype has tried to deal with issues mentioned above as far as it could be done in the limited timeframe we had. This prototype is implemented in such a way that it can be used fully offline – no internet connectivity could be required for using the app if it just saves the data locally on the device. The only occasion when the internet connection is required is when the app is being installed (& synchronization to the DHIS server in the original app). Thereafter, no connectivity would be required to use the functionality in the app.

The second issue that is tackled by this prototype is that it runs on a mobile device – an android smartphone or tablet. Thus, even if there are some cuts in the power supply, the app, and the functionality will still be accessible unlike the monitors in the Baobab system, which requires electricity all the time. Once the power is off – so will the system be. However, not the case for this app, it can still run if the devices’ battery is charged.   
Another benefit of using a mobile device is that power banks can easily be used to charge the device on occasions without electricity.

The app’s simple interface and design tackle the obstacle of having health workers with limited education and knowledge about technology.

* 1. Hardware

When we say hardware in this case, we mainly mean the essential equipment needed for this project, such as the android device. The main gadget we need is the android device with the operating system Android 4.4 (KitKat)[[38]](#endnote-38) or higher. To fully utilize this functionality, we will also need a label printer that can print the codes generated by the app, and thereafter attached to the health passport.

* + 1. Options

There are many such printers available in the market. The first printer we tried out was the one we found and browed at the university there. Brother QL-700, shown in the picture P.P. This printer was brought to Malawi along with a scanner by Jens Kaasbøll for some years ago.   
We downloaded the necessary drivers for that printer prior to the trip to Malawi. The label size was a little smaller than the label from the Baobab system, however, the printer required a computer/laptop to be able to use it. Since, we are talking about clinics with limited resources and budgets, buying a laptop for this purpose might not be motivating. Having that thought in mind, we decided that we should try a printer that is able to do the desired job with fewer resources and equipment.

We tried to find other label printers that could print labels without requiring a laptop or computer with printing drivers. Several printers were considered for this purpose such as *Brother QL700*, *VC500WZ1*, *TD-4100N,* etc. However, the reasons why we did not choose any of these is either due to the costs or the connectivity and user-friendliness. Some of them required Wi-Fi-connections and others were way too expensive – at least for a case where to are trying to deal with limited resources.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Picture 39 - Existing printer at the campus in Malawi

* + 1. The label printer

The label printer we have chosen for this prototype and project is *Brother P-Touch Cube Plus*, as the picture P.P shows. There are certainly many such gadgets available in the market. However, we had to take various points into consideration of choosing a suitable one. The very first point to check was that the printer was able to print labels such as barcodes and QR codes. Other than this, we also wanted a printer that would not require any complex software or drivers to be able to use it. Therefore, the printer we choose does not require any laptop or other drivers. To start using the printer, we only need to download the app for the printer.



Picture 40 - Brother P-Touch Cube Plus label printer

Another thing that makes it suitable for the context of the connectivity of the printer. This label printer does not require any wired connection as mentioned in the previous paragraph. Many other wireless printers would then require a Wi-Fi-connection instead. However, this *P-Touch Cube Plus* does not require that either. It only uses a simple Bluetooth connection to do the job.

The other benefit we have from this printer is that this printer has a reasonable price too – 1200kr, Inc. one label roll. The printer is portable, which means that the worker will have the ability to move around freely with that tiny little cube in their hands. This printer does not have a complex setup either. After a few simple steps, you will be able to start using the printer quickly. The printing app gives you the ability to save some templates so that you do not have to use time every time you print a label. A few clicks and you have your label ready. This makes it easy for workers with limited technical knowledge.

There was something we observed during the first field trip that we had to take a note of. The Baobab system has a printer that prints label with relevant patient information or diagnoses etc. which are then attached to the booklets. We can see on picture P.P than since the size of the label is quite a little vast, it covers some of the information on the front page. Even though the label has most of the information on it, we think it would be better if we could reduce the size of the labels attached to the front page. This would keep the readability and have a better appearance.

–



Picture 41 - Label from the Baobab system covering the front page

Another point to be noted was that the existing labels have a type of paper and ink, which gets easily worn just like the booklets and the ink may fade out. Therefore, the labels we can print with the printer we have chosen are of small size, printed on glossy paper tape, and are waterproof. P.P.





Picture 42 - A worn health passport of an anonymous patient



Picture 43 - Example of a label from the label printer

* + 1. The Printing part

The label printing part is not integrated into the prototype. To print the labels, you have to go to the label printer’s app to do so. However, this is made a little easy for the users in a way that after adding a patient, the users will get to a screen where he/she can click on the on “Print Code” as the picture P.P shows, and then the user will be automatically redirected to the printing app where they can execute the final task of printing. The full guide to how to print a label is shown in the user manual, attachment A.A.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 44 - Print Code

Picture P.P shows the main screen when the user has been redirected to the other app for printing. The user can choose to either use a previously stored template or create a new label, upload the saved code and then print the chosen QR code. Even though it says “Internet connection required” on the picture, it is not the case for the task we want to perform. In other words, we can print the code without an internet connection.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 45 - Screen the user being redirected to for printing the code

The user is also able to print the QR code for already existing users, picture P.P – the “PRINT CODE”- button on the top right. A use case where this can be useful is when the patient’s health booklet is all used and full and buys a new one. The health workers can then search for that patient in the database and print a new code for that patient.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 46 - The app has the ability to reprint the code

* + 1. Other useful equipment

Other than the required gadgets for this task, there might be useful with some other gears such as a power bank. In periods when the electricity is a problem, a charged power bank can support the work. It can even help community health workers, going around in villages to do their job.

During the first visit, I noted one thing that gave us an idea of using tiny backups for the electricity. For this project, we wanted to talk to a community health worker who was working in that specific area. It took us some time to find her, we had to stop at a couple of places to finally catch her in one of the villages. When we finally found her and asked her to show the current technology, she was using to perform her tasks, then she told us the following; “*I don’t have the mobile phone with me. Since I don’t have electricity at home, so I usually put the phone for charging in a local shop in the neighbor village*”. Then, we had to drive to that place to pick her mobile before she finally showed us the process. In this case, a power bank could help, at least to some degree. Instead of her taking notes with pen and paper and then transferring it all to the mobile phone when she comes back to sort of takes the idea of *mobile technology* away. Some of the tasks in the app required noting down some *live observation* in the app she was using, e.g. “*Tap on screen for each heartbeat of the patient in front of you*”. Therefore, adding the information later in the app based on her notes will surely give some erroneous data. If she had a couple of power-banks along with her on her visits in the villages, she would be able to use that technology in a more accurate manner.  
Picture P.P shows a health worker during her regular visits to the villages she is responsible for.



Picture 47 - Community Health Worker on her regular village visit

1. Iterations

With this topic and project, our aim was to try to make some practical implementations and changes – like it is in action research. However, since it is a masters' project with limited time and resources, it is not possible to conduct full action research – like described in chapter 4. Being able to work on some practically possible ideas have and have some data on that topic for further implementation has been a goal throughout the project.

Usability has been one of the main focuses of this topic. Therefore, being able to test the prototype with real users and get feedback is of the most essential parts of this thesis. Again, due to the limitation, mainly time, we were able to adapt the *singular PD* for this phase. Through interactions with a small number of end-users, we were able to test the prototype to make it better.   
Initially, the goal was to gather data from these testing phases and make the necessary modifications and then re-test again. However, due to the situation there during our second visit, these iterations became somewhere unclear or overlapping. In other words, we ended up making the changes throughout the whole testing phase, rather than sticking to the initial plan with weekly iterations. This means that we had planned to make iterations at the end of every week based on the data we had managed to gather so far. However, we had to make some changes in the plan due to the way we got the data. E.g. some of the days we would get very useful feedback, and the other days, we would come back from the field without any relevant data. Therefore, in our opinion the most practical way to solve this to make changes on the go, to be able to test it as soon as we could and utilize the time for fieldwork as best as possible. Thus we ended up having overlapping changes rather than clear framed iterations that were planned.

To make it a little systematic, we will write about those changes in two iterations. The first iteration will be about the changes that were made during the first two weeks of the field visit, and accordingly, the second iteration will explain the modifications from the last two weeks. The reason for dividing the iteration in exactly two is that we saw a little pattern in the way and type of information we got. During the first testing sessions, we mainly got feedback about the user interface, the appearance, of the app and other environmental issues. Therefore, the first iteration mainly explains such changes. After this phase, we started to dig more into some details. We focused more on the functionality of the app, and how the participants solved the challenges that were presented on purpose to them through the scenarios. Thus, the second iteration will describe more about changes around the important functionalities.

* 1. First iteration

The prototype before the testing phase was as described previously. Just to repeat - the main functionality was to scan a QR for a patient, find a patient with their unique ID, add a patient, and print the QR code for a patient. Coming to the main point here – during the early phases of testing of the prototype, we got some ideas on some basic things we had to fix first. Here we will call it “*environmental changes*”.

Environmental changes

During the very first testing sessions, we observed that the screen brightness has something to say in the user experience. Usually, the smartphones (and other smart devices) are normally on “auto-brightness” mode, which means the device will try to adapt to the light around and try to adjust the brightness of the screen accordingly. On mobile devices, this adjustment ability of the device can be a little slow if the devices are exposed to the light changes rapidly. It may take some time, a few seconds, for the device to adapt to the changes.   
However, this is a minor issue and can be resolved easily by disabling the default mode and adjust the brightness manually. That is how we solved it when we observed this little challenge. During the testing sessions, we started by increasing the screen brightness – usually to the maximum. Many of the testing session was held outdoor, therefore the screen had to be bright to make it easier for the testing participant to read.

The next issue we observed was when the screen of the device locked automatically. Firstly, this was something new to the participants, like it would be for anyone, so they used some time to read the information on the screen and the user manual to perform the scenarios given to them. Secondly, a number of the users were not very familiar with such technology and using a device like this, so they were being extra careful and using more time looking at the screen to find out what to do. It happened that the screen locked very often during the testing sessions and we had to unlock it frequently. Again, this is not a big concern, but just a tiny issue that we felt took the focus of the participant away for a little. Therefore, we solved this by increasing the auto-lock screen time to five minutes, giving the user enough time to think and look at the screen before doing what they wanted to.

UI & functional changes

During this first phase, the focus automatically went on the user interface changes. After the first few sessions, we understood that patient information shown after scanning the code could be a little bigger in size. We made this change of increasing the font based on mainly two reasons. Firstly, during the testing sessions, we observed where the users looked when they were testing, using the eye-movement technique explained in chapter X. We tried to pay attention to where the users looked carefully. Often it happens that the bigger things catch the eye first. During some sessions, we saw that the patient information was not the first thing they looked at. It seemed that they would read and look at the other buttons on the screen. We wanted the patient information to be the “main element” on that screen. Secondly, one of the participants said that something like this; “*My sight is not so good. This is small so it is a little hard to read*”. A takeaway point from this comment was that it could be the case for others as well. Thus, we decided to increase the font of the patient information on the screen to make it easier to read.

The next change we made was to increase the readability and make it easier to understand. Considering the lack of technological knowledge of the users, we made some changes to the main screen of the prototype. The first thing we changed was the text on the button on the screen. Initially, the button said “READ DATA” to scan the code and read the data. To make it even more clear on what it does, we changed it to “SCAN CODE” instead.   
Another change we made to make more intuitive was to modify the information-text on the main screen. It said “*To read the patient information with the code-reader: please click on the “READ DATA*” button. We understood that this seems to be hard to understand and should be more clear. Thus, to make it more accurate we changed it to “*Click on “SCAN CODE” to get the patient information*”, decreasing the amount of text and make it easier to understand.

Going further with the testing got us to a point where we say that we were getting comments on some of the same things. One of them was the information to be added when a new patient was being added. Even though we explained that this was just a prototype and not a real app, we felt that it was not so easy for them to think like that. Summing up their comment on this, when adding a patient we should also be able to add information like the village of the patient. Even though this was a prototype, we tried to make it easier for the participants. Making it closer to the real app and getting their feedback on other more important parts, we decided to add some other fields in the “Add patient” form. Such as the village of the patient and the physical address.   
Accordingly, we also added some of this information in the patient information shown on screen when a code was scanned. Now it also showed the village of the patient.

Summary

Let us just have a brief recap of this first phase. To begin with, our prototype looked something like shown in picture P.P, however, after some sessions of testing, we realized that we had to make some modifications in the user interface of the app to make it easier for the users, and increase the readability and to make the functionality more intuitive. Therefore, increasing the screen brightness, changing the font size, modifying the button names, etc. were some of the essential modifications we had to make.

Besides such minor yet still important changes, we figured out that there was one thing that we had to keep in mind during further testing, modifications, and development. The users' ability to understand our way of thinking. Referring again to the mental modal mentioned in chapter X – the gap between developers and end-users thinking. Even though the idea of testing an idea and the protoptye of it might seem quite clear to us, this might not work in practice. Therefore, one major outcome from this first phase was that we had to make things as close to the real world as possible. Unless we were extra good at explaining the whole concept in that limited time got for the testing sessions. Hence, some of our main take-away points for the next phase were; closer to the real-world, simple but still explaining, and intuitive functionality.

* 1. Second iteration

After two weeks of testing, we started to characterize some issues that would have an impact on the functionality. This leads us to the areas we were trying to work on in this thesis – the useability and effectiveness. The repetitive “mistakes” the participants did, and the questions we got from the confused users in certain scenarios gave us an idea to make yet another change. We added two more *buttons* on the main screen.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 48 - The main screen in the prototype after the changes

These buttons are *New Booklet* and *Lost Booklet* as seen in picture P.P. The reason behind this addition to the screen is what is mentioned above. When we presented some certain scenarios, many of the users got confused and were not able to act properly. One of the examples of these scenarios is when we asked the participant what they would do if a patient lost their health passport. Many of them got a little confused and found it hard to find an appropriate solution to this case. E.g. some of them added that patient as a new patient. As we can see this would lead to duplicates in the system, which is something we are aiming to reduce with this solution.  
This also makes the user use more time than e.g. finding an already existing patient in the system. This is also another point we are working on in our research.

In cases where we wanted the participants to perform a task when the patient’s booklet was full created also some confusion. We say that we had to guide the user in the right direction to do what was necessary for this case. Some users suggested that they would peel off the code sticker from the previous booklet and attach it to the new one. While some others suggested that they could add that patient again. Therefore, we had to do something here to make this a little clearer. The easy way for them was to just click on predefined buttons. We observed that having buttons as *tasks* was an approach to solve the cases in an appropriate way. Thus, we added yet another button – *New Booklet*.

After making these changes, at the end of the third week, we started to see that this reduced the confusion a lot. When we presented the same scenarios of a patient losing their booklet after these changes, the participants would immediately that they would click on “Lost Booklet”. This was a clear improvement.

However, the second button was apparently not as clear we thought to all of the participants. Many of the users, more than a half, understood its purpose and used it what it was meant to be used for. But a number of users were still confused. When we asked them what they think this “New Booklet” button does, some of them replied that it was meant for a patient who comes for the first time at this facility. For a new patient. This was quite surprising data we got here and shows again the importance of user involvement in a product. As developers or designers, we think we are understanding the requirements and creating the best possible solution, but this might not be the case. The developers' thinking-model is not the same as the end-users[[39]](#endnote-39). [ref]. Thus, what we thought was a good solution, in this case, was the clearest fix anyway.

Summary

At the end of the second phase, we had a prototype that looked like the one shown in picture P.P(beginning of this section). We had made some additions and modifications, by keeping the outcome from the first phase in mind. These changes helped us in a way that the user could focus more on what we were actually trying to test, rather than other issues that might be out of scope for our topic.

Throughout this testing phase, we made some other minor changes as well. Such as resizing the fonts, placement of some buttons, and increasing display time of “toast messages” – a message that pops up for a few seconds when the users try to add something with missing fields. Picture P.P shows an example. Then we came to the last few days of our testing phase. From here, we got some more data and feedback that gave us some ideas on other improvements in this prototype, but we did not have enough time to implement all the changes. We will write more about these potential improvements, and further work in chapter X.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Picture 49 - A toast message "Missing input" at the bottom of the screen

1. Discussion
   1. Comparison with Baobab

One of the last-mentioned hurdles above is the resources. If we compare this solution to the Baobab solution we have been talking about in this thesis, we can see some clear differences in resource usage. The first thing we can observe is that an android smartphone is cheaper than a touch screen monitor for doing the same job. If we zoom out a little and try to have an overall look at the hardware, the monitor and/or CPU, a label printer, a scanner, a power generator as backup vs, a cheap android device and a label printer. Certainly, there might be other contextual circumstances that may affect, nevertheless we can see that there is already an economic gain with this app to some degree.

1. Conclusion

Literature

1. Kenton, Will. “Barcode.” *Investopedia*, 2019, [www.investopedia.com/terms/b/barcode.asp](http://www.investopedia.com/terms/b/barcode.asp). [↑](#endnote-ref-1)
2. Barcode. (2020, February 10). Retrieved from https://en.wikipedia.org/wiki/Barcode#Use [↑](#endnote-ref-2)
3. Barcode. (2020, February 10). Retrieved from https://en.wikipedia.org/wiki/Barcode [↑](#endnote-ref-3)
4. Barcode. (2020, February 10). Retrieved from https://en.wikipedia.org/wiki/Barcode#History [↑](#endnote-ref-4)
5. A Short History Of Bar Code (For Bar Code News, visit the Bar Code News Page.). (2013). Retrieved February 10, 2020, from http://www.adams1.com/history.html [↑](#endnote-ref-5)
6. A Barcode Primer. (n.d.). Retrieved from https://www.barcodehq.com/primer.html [↑](#endnote-ref-6)
7. Ashish, Vaidyanathan, V., Daftardar, I., Peshin, A., Gupta, & Deshpande, M. (2019, October 11). QR Code vs Barcode: Differences Between QR Code and Barcode. Retrieved from https://www.scienceabc.com/innovation/whats-qr-code-how-its-different-from-barcode.html [↑](#endnote-ref-7)
8. «QR Codes for Dummies»

   ### [QR codes for dummies / by Joe Waters. By: Waters, Joe (Joe R.).](https://bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?vid=UIO&id=999920067876502204&inst=47BIBSYS_UBO&context=L" \t "_blank)

   #### Waters, Joe Type Publisher Hoboken, NJ : John Wiley & Sons, ©2012

   [↑](#endnote-ref-8)
9. QR code. (2020, February 7). Retrieved from https://en.wikipedia.org/wiki/QR\_code#History [↑](#endnote-ref-9)
10. Mobile-QR-Codes.org. (n.d.). Retrieved from http://www.mobile-qr-codes.org/history-of-qr-codes.html [↑](#endnote-ref-10)
11. One Code, many possibilities. (n.d.). Retrieved from https://www.qr-code-generator.com/qr-code-marketing/qr-codes-for-specific-applications/ [↑](#endnote-ref-11)
12. Saeed, K., & Snášel Václav. (2014). *Computer Information Systems and Industrial Management 13th Ifip Tc 8 International Conference, Cisim 2014, Ho Chi Minh City, Vietnam, November 5-7, 2014, Proceedings*. Berlin: Springer Berlin. Page 592-593. [↑](#endnote-ref-12)
13. QR Codes basics. (n.d.). Retrieved from https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/ [↑](#endnote-ref-13)
14. Becker, H., Berger, P., Luckmann, T., Burawoy, M., Gans, H., Gerson, K., ... & Pottieger, A. (2002). Observation and interviewing: Options and choices in qualitative research. *Qualitative research in action*, *6*, 200-224 [↑](#endnote-ref-14)
15. KristinBraaa. (2001, January 1). Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research. Retrieved from https://reader.elsevier.com/reader/sd/pii/S0959802298000186?token=11991C3ABABE4D05653F305B440A66FE79966EDB433BA08E2DD330083DFFDE0B790956DFBF91784C4CDF35F7B076A932 [↑](#endnote-ref-15)
16. McKay, J., & Marshall, P. (2001). The dual imperatives of action research. *Information Technology & People*. [↑](#endnote-ref-16)
17. Roland, L. K., Sanner, T. A., Sæbø, J. I., & Monteiro, E. (2017). P for Platform. Architectures of large-scale participatory design. [↑](#endnote-ref-17)
18. Boyce, C. & Neale, P. (2006) “Conducting in-depth Interviews: A Guide for Designing and Conducting In-Depth Interviews”, Pathfinder International Tool Series [↑](#endnote-ref-18)
19. Affairs, A. S. for P. (2014, December 2). Eye Tracking. Retrieved from <https://www.usability.gov/how-to-and-tools/methods/eye-tracking.html> [↑](#endnote-ref-19)
20. Douglas, G. P., Gadabu, O. J., Joukes, S., Mumba, S., McKay, M. V., Ben-Smith, A., Jahn, A., Schouten, E. J., Landis Lewis, Z., van Oosterhout, J. J., Allain, T. J., Zachariah, R., Berger, S. D., Harries, A. D., & Chimbwandira, F. (2010). Using touchscreen electronic medical record systems to support and monitor national scale-up of antiretroviral therapy in Malawi. *PLoS medicine*, *7*(8), e1000319. <https://doi.org/10.1371/journal.pmed.1000319> [↑](#endnote-ref-20)
21. Douglas, Gerald & Gadabu, Oliver & Joukes, Sabine & Mumba, Soyapi & McKay, Michael & Ben-Smith, Anne & Jahn, Andreas & Schouten, Erik & Landis-Lewis, Zach & Oosterhout, Joep & Allain, Theresa & Zachariah, Rony & Berger, Selma & Harries, Anthony & Chimbwandira, Frank. (2010). Text S6. [↑](#endnote-ref-21)
22. Lippeveld, T. “[Routine Health Information Systems: The Glue of a Unified Health System.](https://www.uio.no/studier/emner/matnat/ifi/INF5761/v18/pensumliste/resources/Lippeveld%20-%202001%20-%20Routine%20health%20information%20systems%20the%20glue%20of%20a%20.pdf)” In Keynote Address at the Workshop on Issues and Innovation in Routine Health Information in Developing Countries, Potomac, March, 14–16, 2001 [↑](#endnote-ref-22)
23. About OpenMRS. (n.d.). Retrieved from https://openmrs.org/about/ [↑](#endnote-ref-23)
24. Baobab Health Trust. (n.d.). Retrieved from https://healthmarketinnovations.org/program/baobab-health-trust [↑](#endnote-ref-24)
25. CommCare Fundamentals - CommCare Overview. (n.d.). Retrieved from https://confluence.dimagi.com/display/commcarepublic/CommCare Fundamentals - CommCare Overview [↑](#endnote-ref-25)
26. King, C. (2014, October 30). Electronic data capture in a rural African setting: evaluating experiences with different systems in Malawi. Retrieved February 19, 2020, from https://www.tandfonline.com/doi/full/10.3402/gha.v7.25878 [↑](#endnote-ref-26)
27. Josephine Borghi, Spy Munthali, Lameck B Million, Melisa Martinez-Alvarez, Health financing at district level in Malawi: an analysis of the distribution of funds at two points in time, Health Policy and Planning, Volume 33, Issue 1, January 2018, Pages 59–69, <https://doi.org/10.1093/heapol/czx130> [↑](#endnote-ref-27)
28. Chaulagai, Cn & Moyo, Chris & Pendame, Rb. (2020). Health Management Information System in Malawi: Issues and Innovations. [↑](#endnote-ref-28)
29. DHIS2 USER MANUAL. (2016). Retrieved February 10, 2020, from <https://docs.dhis2.org/2.24/en/user/html/ch27.html> [↑](#endnote-ref-29)
30. US20070271315A1 - Robust silo based system architecture. (n.d.). Retrieved from <https://patents.google.com/patent/US20070271315A1/en> [↑](#endnote-ref-30)
31. Elise Østmo, One Patient, one Record Keeping a patient record in HIV/AIDS care with Hybrid health information systems: An ethnographic case study from South Africa [↑](#endnote-ref-31)
32. About DHIS2. (n.d.). Retrieved February 19, 2020, from <https://www.dhis2.org/about> [↑](#endnote-ref-32)
33. Braa, Jørn & Sahay, Sundeep. (2017). The DHIS2 Open Source Software Platform: Evolution Over Time and Space. [↑](#endnote-ref-33)
34. Roland, L. K., Sanner, T. A., Sæbø, J. I., & Monteiro, E. (2017). P for Platform. Architectures of large-scale participatory design. [↑](#endnote-ref-34)
35. Braa, Kristin & Nielsen, Petter & Titlestad, Ola. (2014). Innovation for Health in Developing Countries. [↑](#endnote-ref-35)
36. Cappelen Damm. (2018). Minimalisme. In *Tenk enkelt*. [↑](#endnote-ref-36)
37. Affairs, A. S. for P. (2014, February 19). Prototyping. Retrieved February 17, 2020, from https://www.usability.gov/how-to-and-tools/methods/prototyping.html [↑](#endnote-ref-37)
38. Overview. (n.d.). Retrieved from https://www.dhis2.org/android [↑](#endnote-ref-38)
39. Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization science*, *3*(3), 398-427 [↑](#endnote-ref-39)