



Uio • University of Oslo

# Implementing videos for health promotion and community health worker training in DHIS2 Android Capture app

*Integrating and field-testing video library functionality for mobile devices*

Ariel Dahlberg Karlsen & Arne Røen

Master's Thesis

Informatics: Programming and System Architecture

60 credits

Department of Informatics

Faculty of Mathematics and Natural Sciences

Spring 2020



## Abstract

Community health workers (CHW) serve a large proportion of the population in low- and middle-income countries and have a variety of duties, including conducting health promotion in the communities. CHWs, as well as the community they work with, need ways of sustaining the knowledge they receive. Video is a medium that is well-suited for acquiring knowledge. Embedding videos in an electronic patient information system could therefore contribute to increased sustainability.

This thesis explores the usefulness of videos for CHWs and how a video feature could be implemented in the District Health Information Software 2 (DHIS2) Android Capture app. Since the current version of the Android app does not support videos, this thesis saw the implementation of a video library. The design of our implemented interfaces was largely based on the interfaces of existing applications and the usage of standard Android UI elements. The video library was tested among a selection of CHWs in Malawi in order to identify common thoughts, opinions, and ideas for the video functionality.

Through interviews, feedback indicates that videos are a coveted feature among the CHWs. Two main areas of video usage were explored in this thesis: Videos for health promotion and videos for training and re-training of the CHWs. Health promotion and re-training proved to be areas well suited for videos, but videos should not completely replace traditional training. Benefits such as higher knowledge retention, less memorization for the CHWs, the physical aspects of an electronic device, and increased interest among the clients were mentioned. Additionally, using videos in health promotion would make it easier to appeal to parts of the community that are difficult to reach.

While we had based elements of our video library's interface on standard design patterns, these patterns turned out to not necessarily be standard to all the CHWs in Malawi. Feedback from the CHWs suggests that parts of the interface were easy to understand and use, such as controlling the video player and navigating the listing of videos. However, elements like the search bar and navigation from the video player back to the video library proved to be difficult for the CHWs to use. To overcome this, frequent user testing or discovering and documenting universally understood design patterns, as well as standards among the CHWs in Malawi, is needed.

We also highlight several important factors pertinent to a successful adoption of this video-based intervention. Among these are the screen size of devices, the support for multi-language videos, and the resolution and encoding of the video files.



## Acknowledgement

First of all, we would both like to express our deepest appreciation to our supervisor Jens Johan Kaasbøll for his invaluable help, guidance, and feedback, and, together with the Information Systems group at UiO, for giving us the opportunity to travel to Malawi.

We are also highly indebted to Chipso Kanjo and Tiwonge Manda, from Chancellor College in Zomba, Malawi, for their suggestions, making our fieldwork possible, ensuring we had the required official permissions, keeping us safe during our stays, and feeding us. Many thanks to Chancellor College in Zomba, Malawi, for hosting us in Malawi. Furthermore, we want to thank Isaac Mwakabira for his help with the Android app.

We'd like to give a special thanks to our friend Joshua Rowan Hara, not only for his role as our translator and guide in Malawi, but also his dedication to making us feel welcome, even outside of our field trips. Our research would have been difficult without his help, and our stay much less fun without his company. Thank you for showing us your beautiful country!

We are grateful to the CHWs and other clinic staff for making time for us in their busy schedules, for teaching us about their work, and for their invaluable cooperation.

We also want to thank our families for all their support. Ariel would also like to thank Sigrid and Christin for all their encouragement throughout our work with the thesis. Lastly, Arne would like to thank his parents and Rebekka for their continued support and love.



# Table of contents

|       |  |    |
|-------|--|----|
| 1     | Introduction.....                          | 1  |
| 1.1   | Background.....                            | 1  |
| 1.2   | Personal motivation .....                  | 2  |
| 1.3   | Research questions.....                    | 2  |
| 2     | Background.....                            | 3  |
| 2.1   | Health Information Systems.....            | 3  |
| 2.2   | DHIS2 and HISP.....                        | 4  |
| 2.3   | Malawi.....                                | 5  |
| 2.4   | Current health situation in Malawi .....   | 7  |
| 2.5   | Video learning .....                       | 9  |
| 3     | Methodology .....                          | 11 |
| 3.1   | Research methodology.....                  | 11 |
| 3.2   | Data collections methods.....              | 13 |
| 3.2.1 | Interview.....                             | 13 |
| 3.2.2 | Observation .....                          | 13 |
| 3.3   | Research approach .....                    | 14 |
| 3.3.1 | First trip – Exploring .....               | 15 |
| 3.3.2 | Second trip – Testing in the field.....    | 16 |
| 4     | Video as a tool .....                      | 22 |
| 4.1   | Requirements gathering.....                | 22 |
| 4.1.1 | Health promotion .....                     | 23 |
| 4.1.2 | (Re)training.....                          | 27 |
| 4.2   | Using video as a tool.....                 | 29 |
| 4.2.1 | Previous knowledge/use and experience..... | 30 |
| 4.2.2 | Videos for health training.....            | 33 |
| 4.2.3 | Videos for health promotion .....          | 34 |
| 4.2.4 | Desired video topics and use.....          | 36 |
| 5     | Software .....                             | 38 |
| 5.1   | Remote video repository.....               | 38 |
| 5.1.1 | Remote video repository endpoints.....     | 39 |
| 5.1.2 | Video meta-data.....                       | 40 |
| 5.1.3 | Video repository management.....           | 41 |
| 5.1.4 | Video format.....                          | 42 |
| 5.1.5 | Video propagation/downloading .....        | 44 |

|        |  |     |
|--------|--|-----|
| 5.2    | Introducing videos to the DHIS2 Android capture app .....  | 45  |
| 5.2.1  | Internal database and database communication.....  | 46  |
| 5.2.2  | Video downloading and storage.....   | 48  |
| 5.2.3  | Video download Worker .....  | 49  |
| 6      | Iteration 1.....   | 53  |
| 6.1    | Software implementation .....  | 53  |
| 6.2    | Data collection.....   | 59  |
| 6.3    | Action items for Iteration 2 .....   | 62  |
| 7      | Iteration 2.....   | 65  |
| 7.1    | Software implementation .....  | 65  |
| 7.1.1  | Clicking the video title to open the video.....  | 65  |
| 7.1.2  | Issues finding video controls .....  | 66  |
| 7.1.3  | Confusion about main menu (hamburger menu) .....   | 67  |
| 7.1.4  | Confusion related to the video library VS the main view.....   | 67  |
| 7.1.5  | Trouble using the (Android) back-button.....   | 67  |
| 7.1.6  | Missing thumbnails.....  | 68  |
| 7.1.7  | New users might need training .....  | 69  |
| 7.1.8  | Screen size vs crowd size.....   | 69  |
| 7.1.9  | Language select .....  | 70  |
| 7.1.10 | Search functionality.....  | 72  |
| 7.2    | Data collection.....   | 74  |
| 7.3    | Challenges and action items.....   | 79  |
| 8      | Discussion .....   | 81  |
| 8.1    | Compared to similar application .....  | 81  |
| 8.2    | Illiteracy .....   | 83  |
| 8.3    | Training and refreshers .....  | 84  |
| 8.4    | Health promotion .....   | 85  |
| 8.5    | Language .....   | 88  |
| 8.6    | Implementation.....  | 91  |
| 8.6.1  | Frontend.....  | 91  |
| 8.6.2  | Backend .....  | 100 |
| 8.7    | Device .....   | 102 |
| 9      | Conclusion .....   | 103 |
| 9.1    | How can videos be useful for community health workers?.....  | 103 |
| 9.2    | How should the interaction and user interface of the DHIS2 Android Capture app be extended to provide the videos in a usable way?..... | 104 |



|       |   |     |
|-------|---|-----|
| 9.3   | How should the backend solution be designed to provide videos on cheap devices in rural settings? ..... | 106 |
| 9.4   | Limitations .....   | 106 |
| 9.5   | Further work.....   | 107 |
| 9.5.1 | Development .....   | 107 |
| 9.5.2 | Design and use.....   | 107 |
|       | References.....   | 109 |
|       | Appendix 1 – Interview guide iteration 1 .....  | 113 |
|       | Appendix 2 – Interview guide iteration 2 .....  | 114 |
|       | Appendix 3 – Code.....  | 115 |

# Acronyms

**API** Application programming interface

**AR** Action Research

**CHAM** Christian Health Association of Malawi

**CHW** Community health worker

**DHIS2** District Health Information Software 2

**DNS** Domain Name System

**HDI** Human Development Index

**HISP** Health Information Systems Programme

**HSA** Health Surveillance Assistant (Title of CHWs in Malawi)

**ISO** International Organization for Standardization

**LMIC** Low- and middle-income countries

**MoH** Ministry of Health (Malawi)

**SDK** Software development kit

**UI** User interface

**UID** Universally Unique Identifier

**UiO** University of Oslo

**UN** United Nations

**UUID** Universally unique identifier

**WHO** World Health Organization

## List of figures

|  |    |
|--|----|
| Figure 2.1: A detailed view of a health information cycle (DHIS 2 User Guide, n.d.).....   | 4  |
| Figure 2.2: Map of Malawi (UN Malawi, 2004) .....  | 6  |
| Figure 3.1: Illustration of our action research cycle.....   | 12 |
| Figure 3.2: Observing an CHW testing the video library .....   | 18 |
| Figure 3.3: The road to Nwgelero Clinic after a heavy rainfall.....  | 19 |
| Figure 3.4: Interview session at Bimbi clinic .....  | 20 |
| Figure 4.1: Flipchart facing the client.....   | 23 |
| Figure 4.2: Flipchart facing the CHW.....  | 24 |
| Figure 4.3: A CHW using a flipchart during counseling with a woman .....   | 25 |
| Figure 4.4: A stack of flipcharts stored at a clinic.....  | 26 |
| Figure 4.5: Health manuals found at a clinic.....  | 28 |
| Figure 4.6: Diagram of how many CHWs have used videos in health promotion situations.....  | 31 |
| Figure 4.7: Diagram of how many CHWs have used videos in training situations.....  | 31 |
| Figure 4.8: Diagram of the desired topics for videos .....   | 37 |
| Figure 5.1: Video meta-data.....   | 40 |
| Figure 5.2: Class diagram of 'VideoDatabaseInstance' .....   | 47 |
| Figure 5.3: Sequence diagram showing database communication.....   | 47 |
| Figure 5.4: Code of 'VideoDatabaseInstance' .....  | 48 |
| Figure 5.5: Sequence diagram showing the video downloading process .....   | 51 |
| Figure 6.1: Screenshot of the main screen of the app with the hamburger-menu expanded .....  | 54 |
| Figure 6.2: Screenshot of video library .....  | 56 |
| Figure 6.3: Example of Android back button highlighted.....  | 58 |
| Figure 6.4: Back arrow in the app bar highlighted .....  | 63 |
| Figure 7.1: A row in the video library .....   | 66 |
| Figure 7.2: Screenshot showing the added instructions about displaying video controls.....   | 66 |
| Figure 7.3: App bar with a back arrow in video player highlighted .....  | 68 |
| Figure 7.4: The video library with thumbnails.....   | 68 |
| Figure 7.5: Showing hideable elements (top), hiding hideable elements (bottom) .....   | 70 |
| Figure 7.6: Implemented language selection dialog .....  | 71 |
| Figure 7.7: Current language selection dialog with a dropdown menu.....  | 75 |
| Figure 7.8: Alternative language selection dialog with radio buttons .....   | 76 |
| Figure 7.9: Alternative language selection dialog with a list of buttons.....  | 76 |
| Figure 7.10: Sketch of variation of the third option.....  | 77 |
| Figure 7.11: Sketch of video categorization .....  | 78 |
| Figure 8.1: Niconico.com (top), iQIYI.com (bottom) .....   | 93 |
| Figure 8.2: Aparat.com (top), bilibili.com (bottom) .....  | 94 |
| Figure 8.3: YouTube.com (top), dailymotion.com (bottom) .....  | 94 |
| Figure 8.4: vimeo.com (top), Google video search (bottom).....   | 95 |
| Figure 8.5: YouTube Android app.....   | 95 |
| Figure 8.6: Our implemented video library .....  | 96 |
| Figure 8.7: WhatsApp chat screen. From <a href="https://www.howtogeek.com/333932/how-to-pin-whatsapp-conversations-to-the-top-of-the-chat-list/">https://www.howtogeek.com/333932/how-to-pin-whatsapp-conversations-to-the-top-of-the-chat-list/</a> ..... | 97 |
| Figure 8.8: DVD player remote. Taken from <a href="https://www.amazon.com/RMT-B118A-Blu-ray-Player-Remote-Control/dp/B00BA4K0H8">https://www.amazon.com/RMT-B118A-Blu-ray-Player-Remote-Control/dp/B00BA4K0H8</a> .....                                    | 98 |

# List of Tables

Table 5.1: Video runtimes, resolutions, and file size ..... 43

Table 5.2: Table showing codec, resolution, and resulting file size for WHO HIV self-testing video (short version) ..... 44

Table 5.3: Table showing the codec, resolution, and resulting file size for the WHO HIV self-testing video (full version)..... 44

Table 6.1: Table showing the videos we used for testing ..... 59

Table 6.2: Summary of observations in iteration 1 ..... 61

Table 8.1: Top 8 most visited video hosting services (List of video hosting services, n.d.) ..... 92

# 1 Introduction

Knowledge is an important feature of humanity and must be acquired, reinforced, and safeguarded throughout our lives. Several means that can help achieve this exists, such as illustrations, text, and interaction with other people. Video can leverage this and combine our senses to create a dynamic and entertaining experience. It is therefore a great medium that can be used for knowledge-building purposes.

This thesis looks at the possibility of implementing support for videos in the District Health Information Software 2 (DHIS2) Android Capture app. It also tries to explore how videos can be used by health workers in developing countries to assist them in their daily tasks.

## 1.1 Background

Community Health Workers (CHW) in Malawi live in the areas they serve. They have very short health educations, spanning from a few weeks to one year. Despite their short education, they provide health services in their area and deal with pregnancy check-ups, child nutrition, malaria treatment, HIV/AIDS care, and so on. In addition, they are also responsible for teaching their communities how to increase general health. For example, improving sanitation by building latrines, preventing pregnancy at an early age by providing awareness of the consequences, and so forth. Although the CHWs receive initial training, they still need to sustain their extensive knowledge when working with communities. The challenge is to ensure that the mode of continuous learning can captivate the CHWs.

Videos are described as a powerful medium, especially when it comes to conveying information. If videos, specifically training videos, were to be embedded in electronic patient information systems, it could contribute to sustainability in different ways. First of all, health workers who watch the videos may learn from them at a higher rate than with current strategies. The knowledge acquired from the videos can be preserved longer and not fade from memory as easily. Another important factor is that videos concerning health promotion may trigger a healthier behavior in the communities.

For the CHWs, a potential electronic patient information system is DHIS2. The Android Capture app created by DHIS2 currently has the ability to display help texts and illustrations. However, the current version of the app does not support video content. By implementing video support in the app, we enable CHWs to access and use videos, and thus the benefits videos entail.

## 1.2 Personal motivation

Early on in our master studies, we were introduced to the work of the Health Information Systems Programme (HISP) at the University of Oslo (UiO) through presentations from the Information Systems research group. Through it, we were introduced to the DHIS2 platform. Learning that UiO is part of a project that develops a system with the goal of improving health care in developing countries, intrigued us. As soon-to-be graduating developers in Norway, it is not often we get the opportunity to use our knowledge and skills for the betterment of underprivileged people, instead of commercial interests. When we were presented with the opportunity by the Information Systems group to do a masters project where our research involved fieldwork in Malawi, a developing country in Africa, we knew we had to take it.

A masters project like this would not only entail contributing towards a good cause but working with the DHIS system would also allow us to work with an open-source system. Open-source software often involves an open collaboration, where the source code is available for everyone. Working with such a system, getting an insight into an open-source community, and being able to contribute to a large system really piqued our interest.

## 1.3 Research questions

The purpose of this thesis is to explore how video functionality can be implemented in the DHIS2 Android Data Capture app, as well as how the CHWs can benefit from videos. Firstly, video functionality refers to the ability of access, play, present, and store videos. As a note, video production is not included in this thesis. Secondly, by implemented we also imply that the limitations of a rural settings are taken into consideration. Lastly, videos meant for CHWs refers to videos for them to use in the context of their jobs.

The three research questions presented below will be addressed in this thesis:

- How can videos be useful for community health workers?
- How should the interaction and user interface of the DHIS2 Android Capture app be extended to provide the videos in a usable way?
- How should the backend solution be designed to provide videos on cheap devices in rural settings?

## 2 Background

This chapter presents relevant background information for this thesis. Firstly, a brief introduction will be given on health-related information systems along with the specific system used in this thesis. Secondly, to get a better understanding of the context this research is conducted in, general information about the Republic of Malawi is given. Finally, aspects of learning and how videos as a medium can be a powerful tool for training is presented.

### 2.1 Health Information Systems

Health Information Systems are defined in Dictionary of Public Health (2018) as a set of methods and procedures used to both compile and present different statistics on population health. Lippeveld, Sauerborn, and Bodart (2000) state that the objective of these health information systems should be to improve health services management through optimal information support, rather than just gathering information. Having a system like this provides a good foundation for improved management, which is a prerequisite for increasing the efficiency of health services. WHO expresses the importance HIS has for decision-making and describes four key functions: data generation compilation analysis and synthesis, and communication and use. However, a HIS can be more detailed (see figure 2.1). The general flow of a HIS can roughly be outlined as collecting data from the health sector and analyzing it while making sure the quality is satisfactory. The data is then converted into relevant information for health-related decision-making.

Information systems concerning health services come in many forms, each a specialized type of HIS. Braa and Sahay (2012) mentions electronic medical records systems, logistics systems, laboratory systems as some of the different specialized component of HIS. Another well-known type is Health Management Information Systems, which is specially designed to assist in the management and planning of health programs, as opposed to delivery of care (WHO Regional Office for the Western Pacific, 2004). However, the type of HIS used in this thesis specializes in handling patient information and focuses more on the delivery of care.

## The Information Cycle

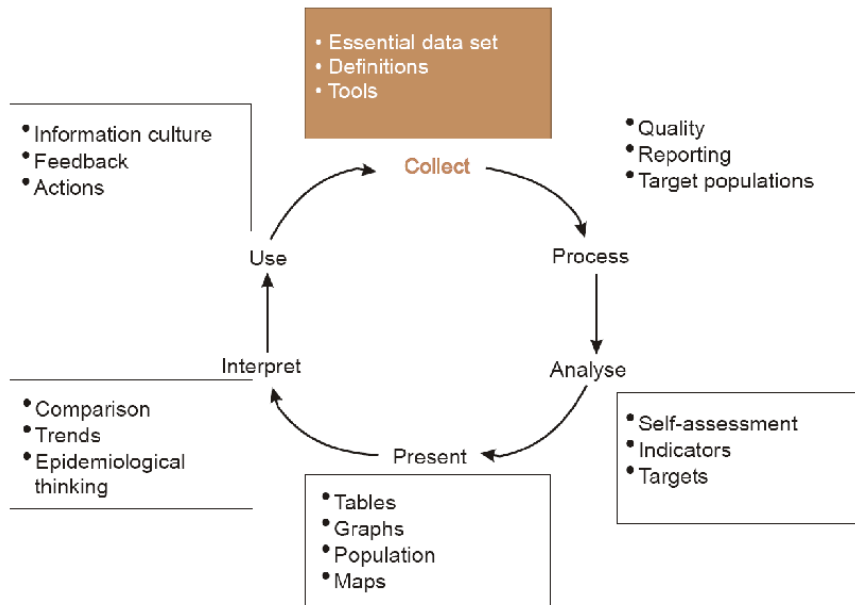


Figure 2.1: A detailed view of a health information cycle (DHIS 2 User Guide, n.d.)

## 2.2 DHIS2 and HISP

District Health Information Software 2 (DHIS2) is a free, open-source, software HMIS platform. This means that the software's source code is publicly available for study, distribution, and open for change. The platform can be described as having a "core", where generic functionality and settings are provided and implemented, which builds up the foundation of the software (DHIS 2 User Guide, n.d.). From the core, different applications can implement another layer in order to customize the software to fit different needs, thereby making the platform more flexible. Additionally, each country using DHIS2 can configure metadata, resulting in them having their own instance of the platform tailor-made in order to accompany their health system. They have full ownership of the application and data, as well as the responsibility of management and data storage.

The core of the software development is managed by the Health Information Systems Programme (HISP) at the University of Oslo. HISP is a global movement with the aim to strengthen HIS in developing countries. An important goal of HISP is to enable and support countries to strengthen their health systems and their capacity to govern their HIS in a sustainable way, thus consequently improving the management and delivery of health services. Through HISP's projects, the DHIS2 platform has been



introduced in almost 70 low and middle-income countries, making DHIS2 the world's largest HMIS platform.

DHIS2 is a modular web-based software designed to manage and present data through a dashboard, accessed through a web browser. However, the development of the software has led to solutions for mobile devices. By utilizing mobile technology, the reach of a HIS is significantly extended. The DHIS2 Mobile site (DHIS2, n.d.) promotes the ability to carry out important tasks on mobile phones. Among these tasks are registering cases, events, and personal information, tracking individuals, conducting surveys, and capturing aggregate data. The mobile solutions provided by DHIS2 can be SMS based, plain HTML and java for feature phones, or a high-end browser-based solution for smartphones. The specific mobile solution this thesis concerns is the DHIS2 Android Capture App. The Android application allows offline data to be captured across the different DHIS2 data models. Data collected are automatically synchronized whenever there is internet access, providing the necessary support for situations where an internet connection is an issue.

## 2.3 Malawi

The Republic of Malawi is a country located in Sub-Saharan Africa, in the southeast region of Africa, that shares a border with Mozambique to the east and south, Zambia to the east and Tanzania to the north (UN Malawi, 2014). The country does not have any coastline, however one of Africa's largest freshwater lakes, Lake Malawi, stretches along the east side of Malawi. The total surface area of the country is 118 484 km<sup>2</sup> and the estimated total population in 2019 was 18.6 million (UN Data Retrieval System, 2019). This makes Malawi a densely populated country, especially considering that the total surface area includes Lake Malawi. According to the UN, the estimated population density is 186 persons per km<sup>2</sup>. Given that the population is steadily growing, with an average annual growth rate of 2.8%, this density will only continue to increase in the years to come. The official language in Malawi is English. However, like many other African countries, Malawi is highly diverse which is reflected in the many spoken languages. In addition to an official language, Malawi has a national language called Chichewa. Though this language is common across the country, several more languages and dialects are spoken in different parts of Malawi.

Malawi is divided into three regions; the Northern, Central, and Southern (See figure 2.2). The regions are further divided into 28 districts. The geographical features of Malawi vary,

with high plateaus in the north and central areas and mountains to the south. Generally, the Malawian climate is tropical and is dominated by two main seasons; the dry season and the rain season. The dry season usually lasts from May to September and is relatively cold with almost no rainfall. On the contrary, the rain season from November to April is hot combined with typically tropical rain and thunderstorms.

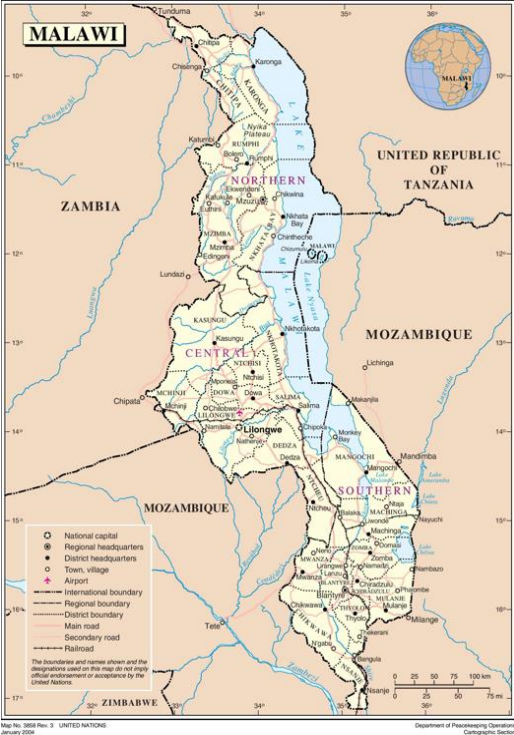


Figure 2.2: Map of Malawi (UN Malawi, 2004)

Malawi is a poor country, not only on a global scale but also compared to its neighboring countries, particularly Tanzania. The economy is heavily dependent on agriculture, which employs over 80% of the population (UN Data Retrieval System, 2019). Agriculture, and therefore the economy, is very vulnerable to external factors, especially abnormalities in the climate.

Malawi’s infrastructure was described in a report by Foster and Shkaratan (2010), as being relatively good when compared to other African countries in similar economic situations. Water is highlighted in the report as a factor Malawi exceeds in. Roads network in Malawi is improving, especially when it comes to paved roads. There is still a large part that is still unpaved and, according to the report, the preservation of the roads falls short of what is needed to maintain the network in good condition. The power sector has also made progress in the last two decades. However, the reliability of electricity is still a challenge and power failure is common, especially during the rainy seasons. The International

Telecommunication Union stated in their annual report from 2018 that the mobile coverage has reached over 80 percent of the country. The 3G coverage was estimated to 50 percent of the population. Utilizing mobile devices such as smartphones and tablets can, therefore, be a good solution in rural areas with unstable electricity and poor internet coverage (ITU, 2018).

The Human Development Index (HDI) is an indicator used by the UN to assess the general development status of a country. HDI does not only take the general income per capita into consideration but also other key dimensions of human development, such as good health and standards of living. In the HDI ranking report from 2019 Malawi was ranked 172 out of 189 countries, meaning that Malawi is one of the least developed countries (UN Development Programme, 2019). However, this is, unfortunately, the case for most of the countries in Africa, as Malawi's neighboring countries are also in the lower end of the ranking scale.

## 2.4 Current health situation in Malawi

Malawi has during these last years had a positive development when it comes to their health situation. The life expectancy at birth according to the UN (UN Data Retrieval System, 2019) is 62.5 years for females and 56.6 for males. This is a significant improvement compared to the expectancy of 48 and 43 years in 2005. The positive development is also reflected in other indicators, such as infant mortality rate. In 2005 the rate was 86.9 per 1000 live births, however, in 2015 the rate had decreased to 52.6. However, WHO still characterize Malawi as a country with a high prevalence of diseases such as tuberculosis, malaria, HIV/AIDS, and other tropical diseases. Maternal mortality is still among the highest in Africa (WHO, 2018). Besides direct causes, indirect causes such as delays in seeking care, poor referral systems, lack of equipment and staff capacity, also contribute to the high rate of maternal deaths.

The Malawian health care delivery system consists of 63% government facilities, 26% Christian Health Association of Malawi (CHAM), and some private for-profit providers (WHO, 2018). The health system is organized at four levels: Tertiary, secondary, primary, and community. The tertiary level consists of central hospitals, the secondary level consists of district hospitals and large CHAM hospitals, while the primary level consists of health centers and community hospitals. The health care at the community level is provided by health surveillance assistants (HSAs), commonly known as CHW, and maternity clinics, as well as dispensaries and health posts. The main task of CHW is to provide promotive and preventive health care, which can be done through door-to-door visitations, village and outreach

clinics, and mobile clinics. These levels are linked to each other through a referral system, meaning that, ideally, the higher the level, the more specialized health services provided. (Republic of Malawi Ministry of Health & Population, 2016)

WHO states in their Country Cooperation Strategy (2018) that some of the challenges in the Malawian health care delivery system are inadequate financing, infrastructure, and equipment. Another notable challenge is related to “inadequate human resources coupled with skewed distribution favoring the urban areas.” (WHO, 2018). This is despite a previously executed plan from 2005-2010 to have a 50% increase in the health workforce. The government has in other words not managed to absorb the new health workers coming out of the training institutions and sustain the new workforce (WHO, 2018).

Few qualified health workers are a general problem in Africa. According to the WHO the African Region had an average of 2.2 health workers per 1000 population (WHO, 2016). Comparing this statistic to the United Nations’ sustainable development goals of 4.45 doctors, nurses, and midwives per 1000 population, the African region is still in great need of health workers in order to provide universal health coverage as the population continues to grow.

The Malawian Ministry of Health (MoH) is aware of the importance of health information and its critical role as a resource for management and monitoring of health sector performance. A HIS is implemented but does unfortunately suffer from various problems. Some common problems the Minister of Health, Dr. Peter Kampalume, mentioned in the health ministry's national HIS Policy (MoH, 2015) were data accuracy, timeliness of reporting, analysis, and completeness. Despite efforts being underway, Kampalume states that the Malawian HIS is still unnecessarily fragmented and not capable of generating quality information at the time they are needed. The policy describes vertical health programs and fragmented systems which sometimes leads to parallel data collection and compromised data integrity, as well as an increased burden for health personnel conducting data collection. These factors contribute to the MoH questioning the reliability of the information generated. Another challenge with the HIS is that the necessary resources for information management have not evolved in line with the increased demand for data management and use.

Malawi is currently utilizing DHIS2 within its HMIS as a means to collect data, among other tasks. A national repository for patient-level data and aggregate data is managed with DHIS2. The repositories are described in MoH’s policies (Republic of Malawi MoH, 2015) as holding disaggregated time series data on all national health indicators. Accessing these repositories for e.g. data analysis or use is not

an availability every health facility has. To accommodate this, the DHO is responsible for preparing and sharing necessary reports for the facilities.

## 2.5 Video learning

Humans have since the earliest times had an incredible ability to adapt. Rather than relying on long-term physical changes through evolution, our adaptive specialization lies in the process of learning (Kolb, 2014). That is why Kolb describes human beings as the learning species. Kolb states that the learning process should be characterized by the texture and feeling of human experiences shared and interpreted through dialogue. His emphasis on experience and its impact on learning is why he has the following definition of learning: “learning is the process whereby knowledge is created through the transformation of experience.” (Kolb, 2014)

Though it is important to acquire experience, not everyone has the opportunity or resources to get firsthand experience for certain situations. This could for example be caused by the situation being a rare occurrence or perhaps the situation is too time-consuming to take part in. For that reason, sharing experience is a key factor for learning. As the world has become more and more globalized, experience can be shared with millions of people. One traditional way of communicating and sharing knowledge is through written mediums, such as books. On a global scale, it can be challenging to reach everyone with this type of medium. Though measures have been taken, there are still millions of people who can't read or write. However, by using visuals, such as pictures, the hindrance of lacking reading skills is negated. Additionally, by utilizing videos in an educational context, there have been many cases of more efficient processing and memory recall (Kay, 2012). Thus, one does not only learn faster, but it is easier to remember what has been learned. On top of that, watching videos often leads to the audience having a positive attitude towards videos (Kay, 2012). Hattie (2009), synthesizing over 800 meta-analyses, marks video as a powerful training medium.

In developing countries, videos can potentially be a powerful medium. As mentioned, using mediums with a visual nature is an advantage when dealing with analphabetism. The countries with the lowest literacy rate are usually developing countries. Globally, Africa is the continent with most countries with a low literacy rate (Roser & Ortiz-Ospina, 2016). In a study conducted by Holloway, Lee, and Mcconkey (1999) on introducing video based-training courses to community-based service personnel in Africa the overall ratings from the participants were positive. Another study by Grossman (2016) concluded that learning with the help of videos was effective in rural Gambia and the study participants had a

positive attitude regarding video learning. Van Mele (2010) highlights the highly effective, positive impact informational videos regarding rice production had in rural Bangladesh. Van Mele (2011) tells us about the impact of training videos for rural farmers in Bangladesh. They found that videos were a better format for distributing knowledge on farming than a more traditional farmer-to-farmer approach. They also measured the effects of viewing a training video on the outcome of the farmer's crop, reporting that the crops of 70% of the farmers exposed improved in-line with the topic of the videos.

Bello-Bravo and Baoua (2012) explored animated videos as a learning aid in developing nations. This study contains several points of interest to us. Firstly, it tells us that animated, informational videos were useful tools in spreading knowledge in areas with limited access to information and knowledge. Secondly, they found mobile phones to be adequate devices for delivering these videos. Thirdly, they touch on the importance of making the information available in the local language for increased an increased chance of success.

A literature review focusing on knowledge and skill retention of CPR for nurses (Hamilton, 2005), found that instructional videos were a viable strategy, depending on the quality of the video, for refreshing nurses' knowledge of CPR. (Hamilton, 2005)

Purchase and Worrill (2002) conducted an empirical study of on-line help design with a focus on the usability of certain on-line help features and principles. This study was conducted among students. They were asked to perform different tasks in a chosen system. Interestingly, the feature "Video help" was the third least favored feature as an aid for the tasks in the system, only beaten by "Web updates" and "Audio help". On the other hand, certain features relevant to our research, such as "Context sensitivity" and "Find" is among the top-rated features. When it comes to principles, the highest rated one is "Easy to understand", meaning that most of the participants value a system that is understandable. Two other highly ranked principles are, "Speak users' language" and "Context sensitive". These principles may be an obvious favored choice by the participants, nonetheless, it shows that taking context into consideration and facilitate local languages is important for the usability of the system.

## 3 Methodology

To better understand our research, it needed to first be situated in its context. This is necessary to not only get a better understanding of the technical situation but the social and cultural setting as well. Our thesis has mainly used two different qualitative methods of data collection: observations and interviews. During our research, we traveled two times to the southern region of Malawi, to the town and district of Zomba.

### 3.1 Research methodology

When conducting research, there are plenty of approaches to choose from. Qualitative and quantitative methods are some of the common ones. The following definitions can be used to describe the two approaches: “Qualitative research is an approach for exploring and understanding the meaning individuals or groups ascribe to a social or human problem. [...] Quantitative research is an approach for testing objective theories by examining the relationship among variables.” (Creswell, 2014). According to Creswell, qualitative research is often distinguished from quantitative research by being framed in terms of using words and open-ended questions, rather than numbers and closed-ended questions.

Our research has focused on the usefulness of videos for CHWs, as well as implementing videos in the DHIS2 Android Capture application to facilitate both the CHWs and their rural circumstances. We are interested in the CHWs’ opinions, thoughts, experience, and feelings regarding these aspects. Most of the questions in this research would be open-ended, with room for exploring the CHWs opinion. Therefore, we decided that a qualitative research approach would be suited for our research.

Just as there are different research approaches, there exist several research methodologies. One of which is Action Research. A traditional definition of Action Research is “Action research aims to contribute both the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework” (Rapoport, 1970, as cited in Susman and Evered, 1978, p.587). Action Research aims to solve current practical problems, often resulting in changes in the organization while expanding scientific knowledge. One of the goals is to first define and identify a problem and then coming up with a feasible solution for it. According to Susman and Evered (1978), Action Research can be seen as a cyclic process. These iterative cycles usually consist of five phases: diagnosing, action planning, action taken,

evaluating, and specifying learning. The diagnosing phase refers to the joint identification of situated problems and their underlying causes. The action planning phase refers to the process of specifying the actions that can improve the problem situation. In the action taking phase, the implementation of the intervention specified in the action planning phase should be done. The evaluation phase implies that there should be an assessment of the intervention by the practitioners and researchers. The last phase, specifying learning, summarizes the learning outcomes of the action research cycles, and signifies the ongoing process of documenting.

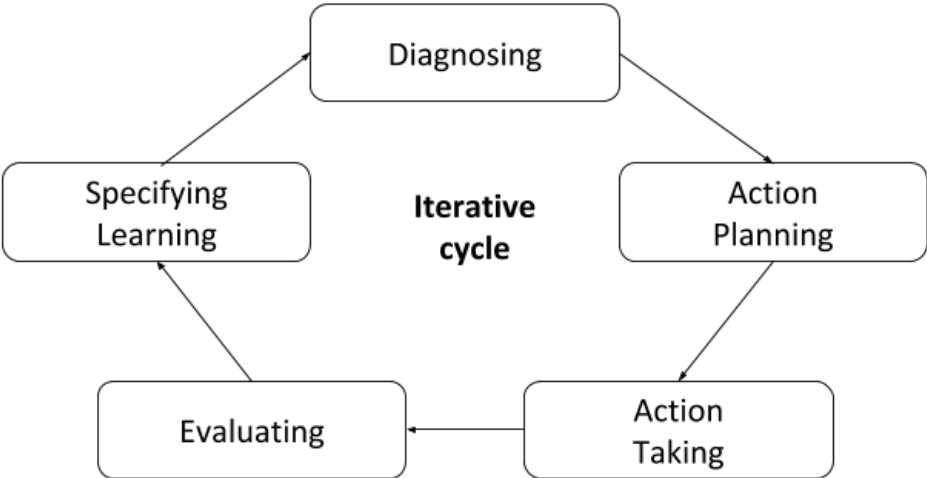


Figure 3.1: Illustration of our action research cycle

Since many of the projects at HISP uses action research methodology, it was natural for us to consider applying action research in our research. Our iterative cycle would be divided into five phases, which closely resembles the common five phases in Action Research. An illustration of our action research cycle can be seen in figure 3.1. The first phase, called diagnosing, would be determined by which iteration we were in. At first, this phase consisted of exploring and identifying the potential need for videos among CHW and took place during our initial trip to Malawi. In the following iteration, this phase was dependent on the evaluation and learning specified in the previous iteration. The second phase would mostly consist of developing our video feature based on the first phase. The third phase would be where our primary source of data collection would take place. In the fourth phase, the results from the CHWs would be evaluated. The last phase would involve summarizing and specifying learning, which would lead to establishing aims for the next iterative cycle. In total, we executed two iterative cycles. However, whereas AR results in changes in the organization, our research did not. The main reason was that we did not conduct or plan to have a deployment of our video functionality to the HIS



in Malawi. Our research is in that regard, part of a bigger AR project. More cycles of real change are needed to cause organizational changes.

## 3.2 Data collections methods

There are a variety of methods of data collection used in research. Our main approaches to collecting data in this thesis are by using two common types of methods in qualitative research: interviews and observation.

### 3.2.1 Interview

Interviews are according to Gill, Stewart, Treasure, and Chadwick (2008) one of the most common methods of data collection used in qualitative healthcare research. By using this method, the views and experiences, as well as motivations, of individual participants can be explored. Interviews can be divided into three fundamental categories: structured, semi-structured, and unstructured. Gill et al. (2008) points out that structured interviews by their very nature only allows for limited participant responses and consequently is unsuited where depth is required. In contrast to this structured approach, an unstructured interview is great where depth is required, but it is performed with little to no organization. It may be structured at the start, by asking an opening question, but the rest of the progress will mainly be based on the participant's response. This can be difficult to manage and participate in since there is a lack of guidelines to determine what to talk about. Additionally, unstructured interviews are usually very time-consuming. Taking elements from both of these approaches, semi-structured interviews consists of key questions but allows the participant or the interviewer to diverge. This contributes to defining the areas which are to be explored, while at the same time allows room to pursue information that may previously have been unforeseen.

Since we are interested in exploring the CHWs' opinions and motivations, structured interviews would not be the best choice for our data collection. On the other hand, unstructured interviews would take too much time and be difficult to execute without any guidelines. Therefore, we decided that the best type of interview for our data collection would be semi-structured interviews.

### 3.2.2 Observation

Observation as a method provides the researchers with a valuable approach to collect data from the non-verbal communication of the participant (Schmuck, 1997). This includes nonverbal expression of feelings and behaviors, as well as determining how much time is spent on various activities. As with

interviews, observation can also be divided into different forms, which is often determined by the role of the observers. One of these forms is passive observation. In this type of observation, the researcher does not participate in the activity and lets the participant carry out the task without any disturbances from the observer. Another form is participant observation, where the researcher has an active part in the observation with the participant and often involves natural conversations and informal interviews. Participant observation is often characterized by actions such as having an open, non-judgmental attitude, being interested in learning more about others, being aware of culture shock and for making mistakes, and being open to the unexpected of what is learned (Kawulich, 2005).

The form of observation we decided to use in our data collection would involve us taking part in the observation. Though we would not participate in the work being done by the CHWs, we did have many characteristics that define participant observation. We had casual and natural conversations with the CHWs that included details about their work and situation, as well as details about their actions during the observation. We would also demonstrate our video library features and observe the CHWs using the feature, while we conversed with the CHWs.

### 3.3 Research approach

One of the main goals of our research was to understand how health workers in Malawi would use the video library and their opinion and thoughts of using videos in general. In that regard, we also wanted to know how we best could facilitate this with our video functionality implemented in the DHIS2 Android Capture app.

Our first trip to Malawi took place in a span of two weeks in August 2019. The aim of this trip was to get a better understanding of the context our research would take place in. This would form the diagnosing phase of our first iteration. The data and knowledge gathered from this exploratory trip were analyzed upon returning to Norway and created a base for the functionality we would implement, which would be our first action planning phase. After implementing a prototype for the video functionality, we needed feedback in order to determine if we have achieved our goals. A second trip back to Malawi was therefore planned to take place for four weeks in January 2020. During our second trip, we would carry out the rest of the first iteration of the cycle, by testing our implementation (action taking), evaluate the data (evaluating) and identify improvements (specify learning). We would then start a new iterative cycle with a new phase of diagnosing, action planning, and action-taking. Evaluating and specifying learning from this new data would take place back in Norway. The findings

from both the first and second trip would form the basis of our thesis and be analyzed, discussed, and concluded on.

### 3.3.1 First trip – Exploring

Our first trip to Malawi took place at the end of August 2019 from the 18th to the 31st of August. The focus of this trip would mainly be on exploring and learning about the health clinics, the health workers, and their everyday work. We would also try to learn about their current solution and tools used in health training situations and health promotional situations. This trip would cover the diagnosing phase of our first iterative cycle.

Our first meeting with the Malawian healthcare system was a trip to a clinic called Magomero. The clinic was located outside Zomba and could be reached by 50-60 minutes of driving on a dusty dirt road. At Magomero clinic we were greeted with a sight that would become common to us at these clinics: a packed crowd of people, patiently waiting for their turn in line. However, we were not staying at the clinic, as we had scheduled to visit an outreach clinic they had arranged. We immediately drove to the outpost and were again greeted by a crowd of people. The CHWs at the outpost had sat up two “posts”. On one of these posts, they had a table with different registers and were diligently filling in new information. We observed their work and were shown a demonstration of some of their procedures for testing diseases. Back at the clinic, the crowd had diminished considerably. We were guided through the waiting area and to the health workers’ office, all the while we took notes of their working place. At the office, the CHWs introduced us to their working schedule and they happily answered any question we had.

The next day, we visited another health clinic located approximately 70 minutes outside Zomba: Bimbi clinic. When we arrived at the clinic, we joined a session of an ongoing campaign about family planning and nourishment. Both mothers and health promoters were attending this campaign that took place in an annex by the main clinic. We sat back, observed, and listened to the current lecture with the help of our local guide, and participated in the more practical aspect of the session: preparing nutritious food from local ingredients. After the session, we got the opportunity to ask some questions to the one responsible for the session, as well as some other health workers that were still at the clinic.

The next time we visited the Bimbi clinic, we tried to arrive earlier than the previous visit. When we entered the health facility, one of the clinic’s health workers was having a promotional lecture in the larger waiting area. The CHW stood in front of the crowd and lectured about HIV, according to our

local guide. This seemed like a perfect opportunity for us to see how these types of health promotional lectures were executed and what tools they utilized. The health promotion was taking place while it was checked that everyone at the presentation had been tested for HIV in the last 6 months (which was a requirement for receiving services). It was therefore a bit crowded and chaotic with people walking back and forth. Nonetheless, we listened to the lecture for a while, before we were led to the same annex as the previous day. This day they had arranged a nutrition program for children and pregnant women. Children were weighed and measures of their height and circumference of their upper left arm were taken, while the mothers were given advice. We took note of the different means and utilities they used and got the chance to ask the health workers at this program some questions. We were a bit taken aback when one of the female health workers started singing, inviting the others attending the program to join. Our local guide explained that it was a song for health promotion and not an uncommon thing to have at the clinics. We noted it down, a bit in awe as we listened to the song and watched their dance. The session continued with varied discussions about diet, correct use of supplements, and breastfeeding. When the session was over, we went back to the health workers' office, where we met two interns at the clinic. They were doing their practice at the clinic for 4 months. This presented an opportunity for us to observe and ask questions about the training the CHWs receive. The health workers happily answered our question before we returned to Zomba.

Our next scheduled meeting with Magomero clinic was a so-called "door to door visit". On the way to Magomero, we picked up the CHW we were going to accompany and drove to the area he was responsible for. First, we visited the chief of the village to state the reason for our stay in his village. After getting his permission, we proceeded on foot and arrived at one of the CHW's patients. She was pregnant and the health worker proceeded with a lecture about how to stay healthy while pregnant. We observed this promotional session and took notes. The next stop was at another pregnant woman's house. The aim of this visit was to find out about and document the general sanitary conditions in the household. At the next house we visited, a woman had recently given birth. Again, we witnessed a promotional lecture, but this time with more focus on the newborn's health. The door to door visits finished with a demonstration in front of a larger crowd. The topic was how to chlorinate the water from the nearby river. We drove the health worker back and returned to Zomba once again.

### 3.3.2 Second trip – Testing in the field

In January 2020 we had our second trip to Malawi, which took place during four weeks from the 4th of January to the 1st of February. This time we were also traveling with another master student from the University of Oslo and were accompanied by our supervisor during our first week. In comparison

to the first trip, the focus of this trip would be to test our implementation of the video library and get feedback on both the technical aspect of the feature and the idea of using videos in health training situations and in health promotion. Our plan was to first gather data through observations and interviews during the first two weeks. In other words, carrying out the action taking phase of the first iteration of the action research cycle. We would then continue with an evaluating phase by analyzing this data and finishing our first iteration by specifying learning. In our last two weeks, a second iteration would be carried out. With the specified learning from the previous iteration, we would diagnose what improvements we could make to our implementation. As part of the action planning phase, we would implement these improvements. With our new version, we would again test it with the CHWs.

With the help of the District Health Office, we were granted permission to conduct our research at four clinics. These were two we had visited before, Bimbi and Magomero, and two new ones, Likangala and Ngwelero. Our first week of the second trip consisted mainly of us introducing ourselves to these clinics and planning when we could meet the CHWs and if it would take place at the clinic or at an outreach clinic. All the clinics were welcoming and offered to help us by setting aside some of CHWs' time to participate in our research. We planned two field trips to each clinic during the first two weeks; one at the clinic itself and one at an outreach clinic. However, the health personnel at Bimbi needed to postpone the meetings due to a busy schedule both on their side and on our side. In addition, Ngwelero Clinic could only offer one day to take part in our research. Despite this, we had our schedule full and were ready to proceed with gathering data.

The process we followed when collecting data during the first iteration, as well as the second iteration, had the same structure and methods for each participant. First, we presented ourselves, our research, and briefly DHIS2. Then we presented the DHIS2 Android Capture app on a tablet and explained what it is and what features were part of our research. We then proceeded by demonstrating the features. The demonstration mainly consisted of us showing where the video library was located, how it could be navigated, and how the video controls worked. After demonstrating, we gave the device to the participant and asked them to carry out different tasks while we observed. During this observation we took notes about how well they performed and if they had any issues, and helped if there was anything they had problems with. The first task they were presented with was to locate the video library from the DHIS2 application home screen. Secondly, we asked them to open and play a video from the library. The next task was to use the video controls to do the following: Pause the video and fast-forward the video. After using these controls, the user was tasked to navigate back to the video library. The last task was to navigate the video library to locate the rest of the videos. After the user had tested the feature, we proceeded by conducting an interview.

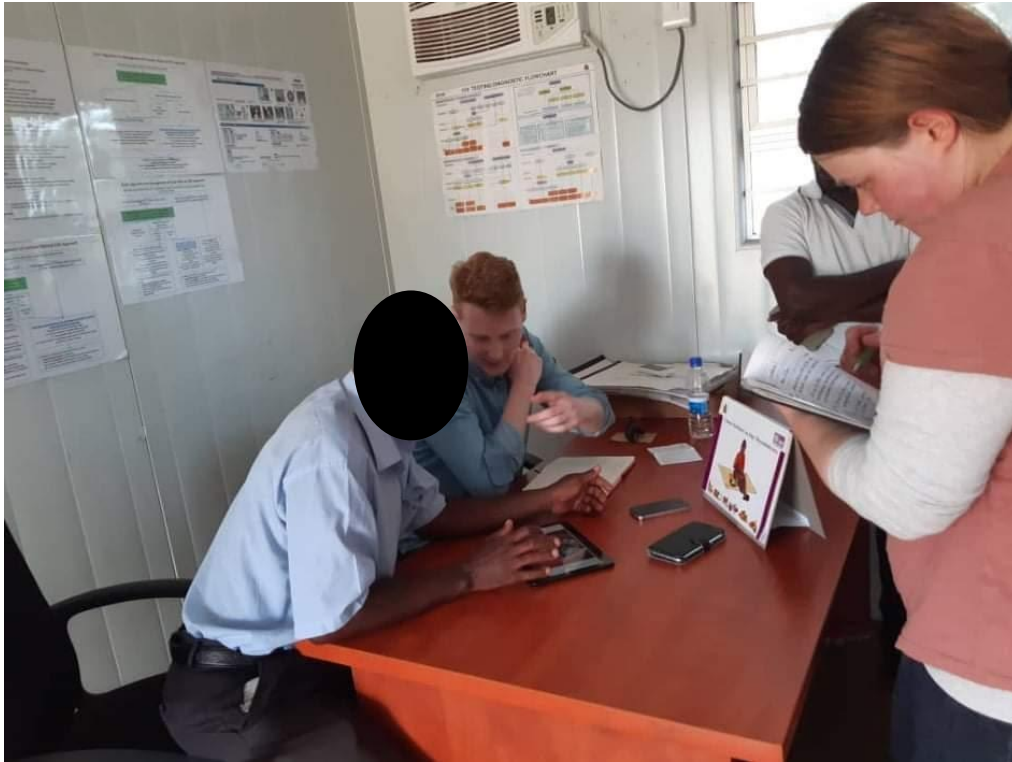


Figure 3.2: Observing an CHW testing the video library

In total, ten health workers, nine CHWs and one nurse, participated in the first part of our data gathering which took place during the first two weeks. A visit to one of Likangala's outreach clinics was the first item on our schedule. We arrived at the main clinic first, picked up one of the CHWs and drove to the point where the outreach clinic was located. The CHWs at this clinic were quite busy with the immunization of children, which was the day's program. We didn't want to disturb them and waited until they had time. Eventually, one CHW and a senior CHW had time for us and we proceeded with conducting two observations and interviews inside the building where the outreach work took place.

The day after we visited Ngwelero clinic and was welcomed by a nurse/midwife practitioner. Initially, we had only planned to introduce ourselves and schedule the actual days in which our data gathering should take place. However, we seized the opportunity and asked the nurse if she was willing to be interviewed right there and then, which she was.

The next week started with another visit to Likangala Clinic. This time we were staying at the clinic, as Monday was the day all health workers were gathered at the clinic for reporting. The staff was very helpful and offered to help us in any way they could. We were lucky and got three CHWs and one

senior CHW for our research. They showed us a room in an annex dedicated to HIV-related work, where we could conduct our data gathering process. To be more efficient, we demonstrated our feature to the whole group at once, before proceeding the user testing and interview one by one.

The following day, we had a scheduled meeting at Magomero clinic and was greeted by one of the CHWs we met in August. He and three other CHWs joined us in the waiting area at the clinic where we demonstrated, this time one-on-one, observed, and interviewed each of them. Despite having two more field trips planned, we decided to cancel those trips and begin to process the data rather than gathering more, as we now had conducted 10 interviews and observations. We had started to notice trends among the data gathered and decided that it was sufficient grounds to proceed to the next phase in our iteration: evaluation.

With the feedback we had gathered, we conducted an analysis of the data with a focus on technical improvements and our data gathering methods as well. We specified what improvements could be made and started a new iterative cycle by using these improvements as part of our diagnosing phase. From the diagnosing phase, we proceeded with action planning and spent the end of week 2 improving the video library. This meant that we could start the following week with a new action taking phase by gathering data on our improved implementation.



Figure 3.3: The road to Nwgelero Clinic after a heavy rainfall

The next week began with a true display of Malawi's rainy season. Though we had experienced periods of rainfall during the previous two weeks, it was nothing compared to the intensity and duration of this night's downpour. Since the roads that we had used to reach the clinics were already damaged by the rain, we were a bit concerned about what state the roads would be in now. Nonetheless, we proceeded with our schedule and managed to reach Bimbi Clinic without getting our vehicle stuck in the mud.

This time the staff at Bimbi Clinic were less busy and agreed to a meeting on Friday that week. Given that we didn't have any other scheduled meetings that week, we wanted to visit Likangala Clinic again to ask if they would help us with another round of data gathering. The road from Bimbi Clinic to Likangala Clinic showed clear signs of the ravages of the rain, as we needed to cross several puddles and streams that had carved its way through the road. Finally, we reached Likangala Clinic and was greeted by the sight of a crowded clinic. We waited a while before we eventually got hold of a health worker in charge. From what we understood it seemed like the CHWs were quite busy, both at the clinic and in the areas around, due to the heavy rainfall and the increased risk of Malaria that followed. They couldn't arrange a meeting until the next week, which we of course understood. The following days until Friday were without any planned field trip. Instead, we used the time to continue with the improvements to our video library and working on writing the thesis.



Figure 3.4: Interview session at Bimbi clinic

Back at Bimbi Clinic, the staff was busy wrapping up the programs for that week. Most of the CHWs were still out in the field, but we managed to get hold of two who were at the clinic. They showed us



to a bench under a tree so we could sit in the shadow since it was sunny. We demonstrated, observed, and interviewed them using our improved version of the video library. On the way back to Zomba, we stopped by a house of a CHW that was not at the clinic. The residents at the house welcomed us and arranged some chairs under a tree so that we could proceed with collecting data from this CHW as well.

Our last week in Malawi started with weather that was quite the opposite of the previous week's rain. Instead of heavy clouds and downpour, the weather was sunny and scorching as we drove on the now dry road to Ngwelero Clinic. Luckily, we found a place under a roof next to an annex at the clinic so we could be shaded from the searing sun as we proceeded with our data collection. The heat rose towards midday and it started to take its toll on both us and one of the CHWs. It was hard to stay focused, but despite that, we managed to collect data from three CHWs before we said farewell to Ngwelero Clinic and returned to Zomba. Our last visit to a health clinic in Malawi was at Magomero Clinic. As they were a bit occupied with the current vaccination that took place in the waiting area, they could only give us time with two CHWs. We gladly accepted the offer and proceeded with our data gathering. We were happy with the data collected and said goodbye to the friendly health workers at Magomero, before returning to Zomba, and eventually back to Norway.

## 4 Video as a tool

This chapter presents our findings on the CHWs' current context, their experience, and their opinions on the use of video as a tool. The chapter includes the results from both our first and second trip to Malawi. Our findings from the first trip will be presented first and are the base on which we formed the requirements for our prototype. The findings from our second trip are presented last and focus on the CHWs opinion on using video to aid their work.

### 4.1 Requirements gathering

During our first trip to Malawi in August 2019, we spent our time gathering information on how and what CHWs in Malawi do. This constituted the diagnosing phase of our first AR cycle. The duties of the CHWs are a large, diverse area. It includes performing tests, e.g. malaria tests, administering vaccines, collecting data about households in their catchment area, collecting data and monitoring the health status of children and newborns, and giving health counseling/health promotion.

These duties are performed in different settings. Some examples include work done at a clinic, outreach clinics where CHWs will travel to a village inside the clinic's catchment area for the day, and perform their duties there, and outreach where a single CHW will travel to clients homes to check up on them, register data about the household, or perform health promotion.

CHWs are often connected to a health clinic. These clinics serve a catchment area, an area and a group of the population, they are responsible for. At the clinics, a range of services are provided. We observed, amongst other things, health promotion, monitoring of newborn/child growth, vaccination, and HIV/AIDS testing.

A group of CHWs employed at a health clinic also do outreach clinics/village clinics. For this, they will travel to villages inside their catchment area for a day and perform several services there. This includes monitoring of newborn/child growth, vaccination, and health promotion. They also do the initial screening of clients and refer them back to the clinics if needed.

The last situation we observed CHWs in, was during home visits. A single CHW would travel to specific houses in their catchment area to perform their duties. These duties included data collection, for example a census or information about the hygiene facilities in a household, checking in on new mothers/newborns, and performing health promotion.

The CHWs also work with local volunteers that help the CHWs with their duties. We were told the volunteers are selected by village leaders and receive three days of training by the CHWs, before doing practical training. One CHW told us that he would train 20 volunteers at once.

#### 4.1.1 Health promotion

An important part of the work of CHWs is health promotion: the giving of health information and counseling to their clients. This encompasses a large span of topics, ranging from sexual health to the importance and proper use of mosquito nets. We observed this being done in a number of different ways and settings, ranging from a one-on-one with a mother in her house, to large gatherings of 40+ people assembled at the health clinic. In the case of larger assemblies, we mainly observed the usage of posters and songs (mnemonics). However, in situations with few clients, we observed them use flipcharts (also known as counseling cards).



Figure 4.1: Flipchart facing the client

These flipcharts are similar to a PowerPoint presentation in function. The CHW holds the flipchart between themselves and the client(s). On the side facing the client there are usually images describing certain situations, and sometimes sparse text (figure 4.1). The side facing the CHW will have text explaining the key points of the client-facing images, a manuscript for the CHW to read, or a combination (figure 4.2) The CHW would either ask the client to describe what they see, talk about

what the pictures show, or a combination. We observed several times that the CHW would quiz the client on either the contents of a page or the combined learnings of several pages. The discussion between the client and the CHW is guided by the contents of the flipcharts.

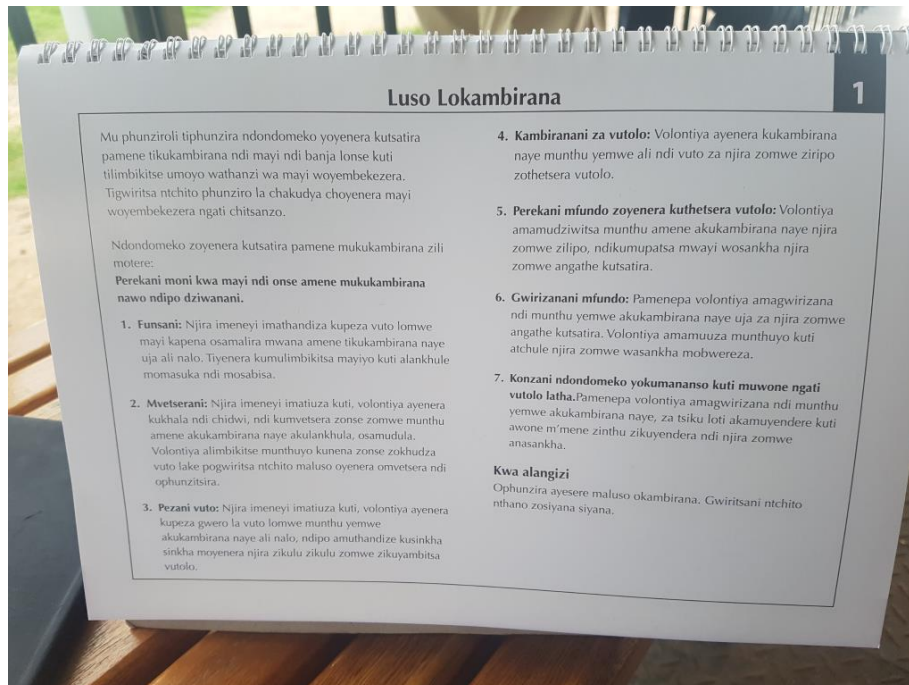


Figure 4.2: Flipchart facing the CHW

The CHWs will also sometimes adapt the content of their health promotion. We observed one example where the CHW was talking about the importance of eating food from all six food groups. They would list examples of foods in each of the groups, and these examples would be tailored to what was available in their area. We also observed a case where the flipchart in use was outdated or incomplete. Contraceptives were being discussed, and the CHW had to add forms of contraceptives that were available, but not covered by the flipchart being used, to the list.

Which of these (if any) learning aids will be used depends on several factors, such as the topic and availability of materials for said topic, the setting, and the number of concurrent clients. The flipcharts are used when the CHW is counseling single or small groups of clients. We were told it was often “per family”, which would mean 1-2 people. These flipcharts are common and are used for several different topics. As such, clinics and individual CHWs will have several different flipcharts covering different topics.



Figure 4.3: A CHW using a flipchart during counseling with a woman

Since these consultations are usually between a single CHW and the clients, the information given can differ from CHW to CHW. When asked how they use the text on the side of the flipchart that faces the CHW, one replied that “Health messages are supposed to be very precise, so [CHWs] are supposed to read the back”, but that after doing it many times they often “know it by heart”.

At one clinic we observed a pile of flipcharts (figure 4.4), titled “Pregnant Health Care” still in their plastic wrapping, being haphazardly stored in a hallway. It’s not clear why they were stored there and not in use.

We visited a small group of houses that did not have access to a well and instead got their water from a nearby river. Said river was, as this was in the dry season, little more than a stream. We also saw that they would wash their clothes a way downstream of where they would collect water (and that therefore, it’s not unlikely that another village would be doing the same upstream). Since they drew their water from this river, they had to treat it with chloride powder. While we were there the CHW explained and demonstrated how to safely purify this water in this way. He explained to us that it was very important that the process completed exactly as demonstrated since any mistakes could lead to the water becoming literal poison.



Figure 4.4: A stack of flipcharts stored at a clinic

From our conversations with the CHWs, we learned about a previous project they had been part of that was aimed at reducing the impact and risk of CHWs not executing their health promotion duties properly. We were told that there are cases where CHWs will do the bare minimum work needed so that their clients, if audited, would respond that they had been told about the topic.

We also learned that CHWs often carry these flipcharts with them to their clients and that they are often used outside in sandy or dirty conditions. Given this, it should come as no surprise that several of the CHWs we interviewed mentioned the flipcharts being damaged as an issue. Based on all these things it becomes apparent that the quality of the information given can differ greatly, despite the importance of accurate information.

Therefore, we believe it would be worthwhile to introduce videos intended for CHWs to show to their clients as part of their health promotion work. As part of our interactions with the CHWs, we asked two if they believed that videos on health promotion topics would be useful to their work. Both

responded positively and pointed out that it might be an improvement over the flipcharts since the videos can demonstrate and “[...] you get to see what is supposed to happen”.

#### 4.1.2 (Re)training

The CHWs have many duties beyond health promotion. In addition to this, CHWs in Malawi are also performing several tasks that are not part of their official duties, as described by their job descriptions. (Kok & Muula, 2013) This is something that was also mentioned to us. As a result of this, there is a large base of knowledge the CHWs need to have, despite only receiving 8-10 weeks of training, and re-training or refreshers are not widely available. (Kok & Muula, 2013)

We asked CHWs what they would do if they were uncertain how to perform a part of their duties. They replied that they would consult health manuals, their notes from their training, or ask a colleague. They also reinforced that they only perform things they are accredited to, and as such (according to them) usually know how it’s done. We also observed that certain parts of their work were repetitive, such as administering vaccines, and measuring and weighing children. Also of note is the pace of which they work. CHWs have large catchment areas (an area they are responsible for serving) and a large portfolio of duties and are as such quite busy.

At one of the clinics we visited they, according to one of the CHWs, only had one copy of most of the manuals and as such were kept at the clinic and not brought to outreach or door-to-door. In contrast to this, at another clinic we visited, we were told they had several copies of most manuals and would bring them into the field.

Some CHWs demonstrated to us how they perform malaria tests in the field. When we asked them what they would do if they needed to double-check the procedure for performing the test, they said that the box of test-kits came with an instruction pamphlet. When we asked to see this pamphlet, they were surprised to find out that the pamphlet was missing from the box.



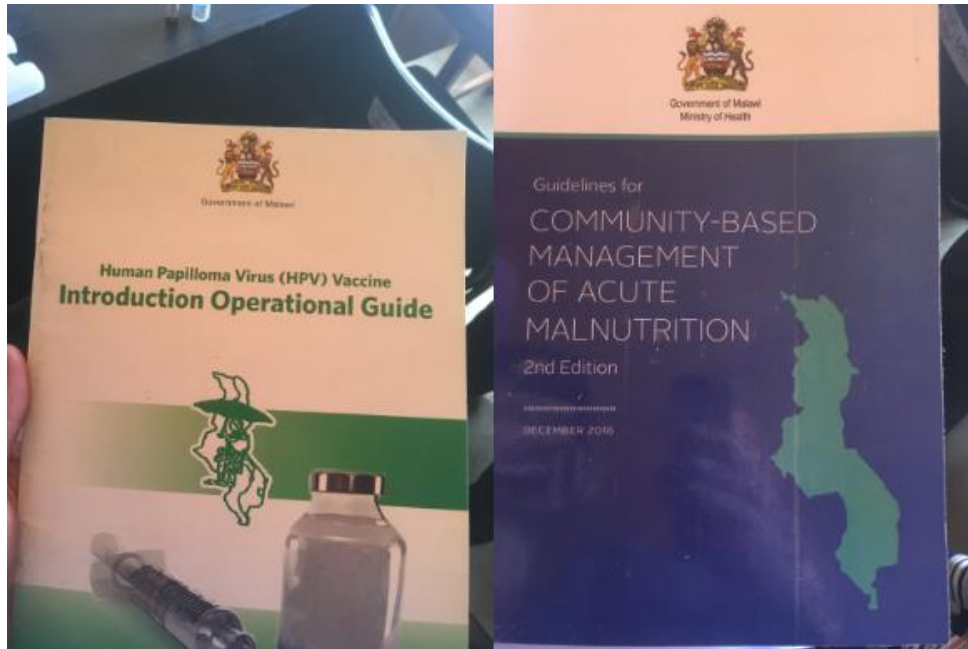


Figure 4.5: Health manuals found at a clinic

The initial idea of how videos could be implemented in the DHIS2 Android Capture app was to associate them with certain capture programs or data elements and make them accessible through “info”-buttons. The logic behind this was that the video would be accessible when the worker was registering related data, for example when the worker was registering vaccine administrations.

Based on the way and pace we observed the CHWs working at, we moved away from this idea. We see it as unlikely that the CHW would consult a video clip if they are uncertain of how to do something in “the field”.

One issue the CHWs raised to us was the lack of availability of re-training and refreshers. This has also been found in earlier research focused on the job satisfaction of CHWs in Malawi. (Kok & Muula, 2013) Creating an integration into the DHIS2 Android Capture app where the health worker could consult a video about a specific topic to refresh themselves before going into “the field”, was identified as a possible use of videos. We asked two CHWs whether they thought videos would be useful for this purpose. Both were positive, and one said that “[our] work and health messages [are] changing every day, so refreshers are needed”. He did however specify that only some topics would be appropriate for video-based refreshers, while others would still require hands-on training.

When new guidelines, procedures, and responsibilities for CHWs are introduced they often receive training for this. We were told that, depending on the available funding, the training could be organized



in several ways. Sometimes all the CHWs involved in the program are sent for training, sometimes the training is brought to their clinics, and sometimes only a few CHWs from each clinic/area goes, and they are then responsible for training their peers when they return.

A study from 2016 looking at the motivation of CHWs in Malawi found that training was highly sought after by the CHWs because training is “[...] a vehicle for career progression, distribution of materials, mentoring, capacity building, and income from allowances.” (Chikaphupha et al., 2016) Since the training was not always offered to all CHWs, “[...] some [CHWs] left out of training workshops insisted that only those who went for training should be the ones to provide the services”. (Chikaphupha et al., 2016)

While videos for training might be a good tool for enabling distributed training at scale (as in, a video is recorded once and can then be used by many individuals for training and thus reduce the workload of facilitators during training), it’s important to note that making available training material to a larger audience through video-based training will not be a complete solution to this issue. The same study found that most of the CHWs they asked were dissatisfied with their salary, and as such allowances from training was for many an important part of their income. They quote one CHW as saying “[...] HSAs [CHWs] tend not to work because they did not go to the trainings and did not get the allowances their friends got at the trainings. So, one may not be willing to work because they did not get that money”. (Chikaphupha et al., 2016)

We also experienced this ourselves. When we broached the subject of videos being used in place of going to training (and being awarded an allowance), one CHW we spoke to expressed concern about losing the opportunity to earn an allowance. Based on this, we have to be careful of the fact that if created and advertised as a tool to replace in-person training, there could be resistance and opposition to the introduction of the video functionality.

As a result of this, we decided to focus on enabling videos as supplements to training, or as a tool for lower-threshold refreshers and re-training.

## 4.2 Using video as a tool

During our second trip to Malawi in January 2020, we interviewed several CHWs at health clinics near Zomba. From the interviews with the CHWs, we got an insight into what experience with health-related

videos they currently have, and their thoughts and opinion on using videos in health training situations and in health promotional situations. Furthermore, we also got some information on what they envisioned the future content of the videos would be. In terms of our AR cycle, this subchapter covers the result from the action taking phase, which is not directly related to the software. The results from both the first and the second iteration are presented together.

#### 4.2.1 Previous knowledge/use and experience

To get a better understanding of the current position videos had among the CHWs, we needed to know what experience they had, both in terms of videos in general and its usage in a health-related context. In our prepared questions (see Appendix 1 and Appendix 2) we included two questions related to previous experience. First, we asked “Have you used videos in any previous health training situation?”, before we shifted the context by asking “Have you used videos in any previous health promotion situation?”

Most of the CHWs had some experience with videos in health training situations. During our first and second iteration we got answers from 21 CHWs in total, 10 from the first iteration and 11 from the second. 17 out of the 21 CHWs said that they had used videos as part of their initial training to become a CHW. When asked about their opinion on the use of videos in training situations, some CHWs mentioned that videos made them understand the topic better. This led to them stating that video was a good way of learning. One of the CHWs reasoned the statement by saying: “it was a good way of learning because it was like hinting – ah – emphasizing the topics for better understanding”. Another CHW said that videos are helpful when trying to retain information and reasoned: “because when you see something you cannot easily forget”. Using videos involves both visual and auditory senses and one CHW points out that it could make learning easier for those who have trouble reading: “Learning through videos is easier, so it’s good. We Malawians don’t like reading.”

Far fewer of the CHWs had experience with videos in health promotional situations prior to our interview. From the 21 answers we got on this question, 15 answered that they had not used video for health promotion before. Only 5 answered yes and we didn’t get any clear answer from 1 of the CHW. All of the CHWs with previous experience had participated in and been part of previous projects. One of these projects involved traveling to villages and mobilizing the people there to come and watch a health promotional video. Another project involved different programs where counseling was executed with the assist of the application CommCare.

Have used videos in previous health promotional situations

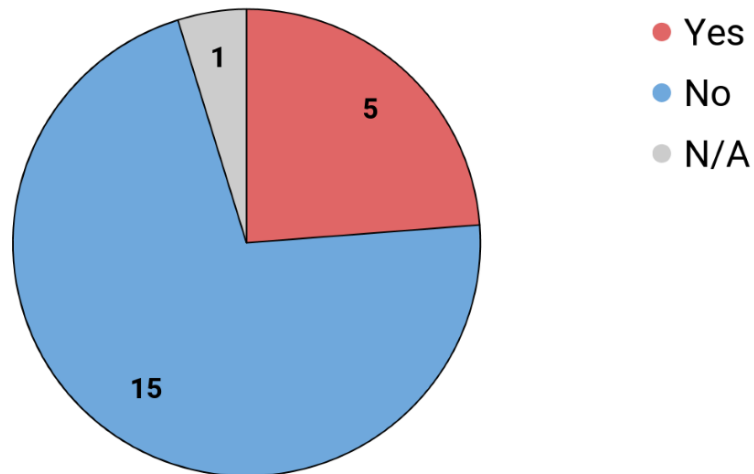


Figure 4.6: Diagram of how many CHWs have used videos in health promotion situations

Have used videos in previous health training situations

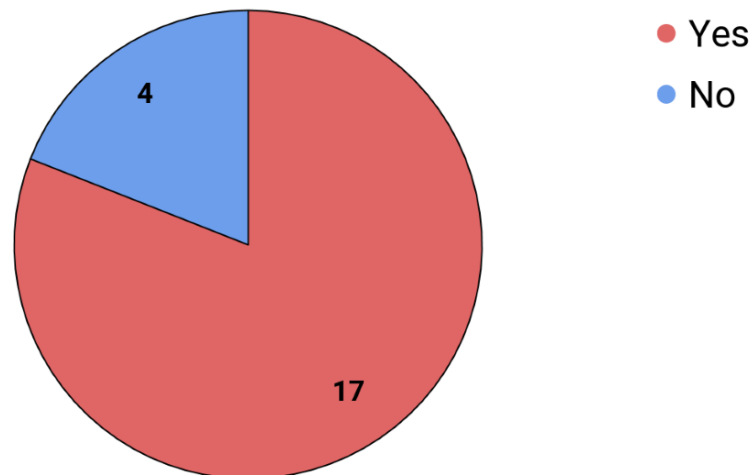


Figure 4.7: Diagram of how many CHWs have used videos in training situations

CommCare was brought up by 4 of the CHWs. They had all been part of the project and had used it in health promotion situations. We asked some of them how the CommCare application worked and how they as a CHW would use it. One CHW had his phone with him and proceeded with demonstrating exactly how it worked. He started by giving us an example of a situation he would use the application: "Let's assume I am in the field, and I want to visit an under 5 child. So as of now let's assume I am at the household for which I want to visit the child. So first when the child is available, we do like this". He then proceeded with filling out a survey of some sort, providing answers to question prompted by the application. The questioned varied from checking the symptoms the child is showing in a checkbox,

to answering yes/no questions about the different vaccines the child has received. The CHW then assumed the child in our example wasn't given a specific immunization. He answered "no" and the application gave a message that the clients should be given advice. On the next page the video could be played and shown to the clients. We observed that this video was a combination of a static picture and sounds.

The main procedure the CHWs explained to us when using CommCare was that the system will guide the interaction they have with the client based on the data they fill in during the interaction. The system will then inform the CHW what they should (if at all) give counseling on to the client and provides some videos the CHW can use. One of the CHWs explained this as CommCare recommending topics for them: "[...] The CommCare I am using at the village clinic, has topics. The topic will be determined depending on the counseling you are making to the client. So, the phone will guide me on which topic to play for the client."

We asked all the CHWs with CommCare experience, what they thought about needing to fill in all the data to get to the video, compared to a solution where the videos are directly accessed whenever they want. All 4 of the CHWs were positive to the idea of having more direct access to videos. A solution like that would in the CHWs opinion give them a chance to decide what topic to teach instead of having it determined by an application. In addition, it would be easier and more practical for some CHWs to play a video if it isn't locked behind several questions. One of the CHWs expressed that being guided by questions didn't always work out: "Whereas in [the case of CommCare] if you are lost, if you have lost one point [entered wrong data at some point in the consultation], then the phone will get you lost, will not direct you to where you wanted to emphasize".

However, the CHWs saw potential in both solutions. One CHW said: "I think these are two different applications of the same messages. And just what matters are the style of comparing the message or playing the videos. But I don't see any problem in either of the videos". Our translator explained to us that CommCare helps the CHWs analyze the situation. He told us that the clients usually don't have the expertise to guide the CHW to say what type of service they need, so it will be up to the CHW to decide in what way they would help the clients. Sometimes the CHW will only do the minimum work so that if anyone checks if they have actually talked to the clients, they would get a confirmation from the client, even though the CHW really hasn't delivered the message. "That's why the CommCare came so they can analyze the situation, so that if this is the issue, maybe we should talk about malaria because they don't even know about mosquito nets." our translator explains to us.

When asked if they had any experience with DHIS2, two answered yes, six answered that they had heard about it, while 13 answered no. However, we discovered that the CHWs are confusing DHIS2 with other health systems, as we were told by our guide and supervisor that they most likely are referring to CommCare or another system called Baobab. We did not encounter any CHWs using the DHIS2 Android Capture app.

#### 4.2.2 Videos for health training

After establishing the CHWs' existing experience the CHWs had with using videos in health training and health promotion settings, we wanted to know their opinion on using videos in training situations. This would give us insight into if videos are desired as an aid in health training and why this may be the case. In some of our interviews, we naturally touched on the subject when discussing the CHW's previous knowledge. However, we wanted to be sure that the CHW understood the question asked and could see it in a practical context. Hence the inclusion in our interview guides (see Appendix 1 and Appendix 2) of the question: "How is the use of videos compared to reading training materials?"

In total from both the first and second iteration, we got feedback from 21 CHWs located around Zomba. All of the CHWs were positive towards the concept of using videos in health training. While no one explicitly stated that videos should replace existing training materials, one of the CHWs pointed out that videos should be seen as a supplement to existing tools, not as a replacement. Another CHW expressed, according to our translator, that there are situations where videos may not be the solution: "He is saying that there are some issues that are generic, things they already know, maybe they just revise. Those things it's OK to have them on videos. But there are some, for example we were talking about the virus, if it's a new thing, then you need somebody to explain first, so that you can ask him the question that he can answer according to how he understands." In addition to this CHW, two other CHWs also pointed out that videos would work great as a means for retraining or refreshing previous knowledge. One of these CHWs also specified that videos would fit well for step-by-step instructions.

Given that everyone seemed positive about the idea of videos in training, we tried to find out what made videos appealing. 7 of the 21 CHWs we interviewed explained that videos made it easier to remember what was taught. In addition, it would be easier and faster to learn because it, according to one CHW, illustrates more than pictures. The information would as one CHW put it "Stick in your brain" because you would not only use your eyes but also your ears. Another CHW said "[...] because with the video, you can watch the actions", thus being able to understand it faster. During our interviews, we also picked up another factor that can contribute to why videos might be easier to understand than

training manuals and personal notes. Three of the CHWs mentioned that it can be tiresome and even difficult to read. One CHW comments: “We Malawians, reading is very difficult for us, to read. But when we see the videos it will help us to remember what we have done before”. This lower level of reading skills would naturally make it easier to gain knowledge by watching videos rather than reading.

The opinion that videos are easier to learn from wasn't the only factor we found that made videos appealing. One of the CHWs also brought up an interesting point: That videos, in general, can also be entertaining and thus more interesting to watch rather than reading:” Sometimes it's just interesting to watch the videos. So, they [who watches videos] will be interested in it, and they will be gaining knowledge.” Furthermore, it seems like it is not only videos themselves that are important for the CHWs. Three of them mention that the physical device that the video is on would also be beneficial for the CHWs. First of all, having access to videos covering different topics on a single device would mean that it would not be necessary to bring several books and health manuals. This would allow them to carry around fewer things when they are out in the field, while still having valuable information at hand if needed. Secondly, one CHW pointed out that the device the videos is on could lead to less material damage: “This one [tablet] in terms of wear and tear, books can get worn out, but this one, everything is intact”. Losing a page or having torn and stained pages in a manual would not be a challenge in having a device with videos instead.

#### 4.2.3 Videos for health promotion

In addition to using videos in health training situations, we also wanted to know the CHWs opinion on using videos in health promotional situations. To help the CHWs put their opinion into words, we specifically wanted to know how videos were compared to using the widely used flipcharts. That is why we included the following question in our interview guide (see Appendix 1 and Appendix 2): “How is the use of video compared to using physical assistive devices, e.g. flipchart?”

As with our interviews about videos in health training, we got 21 answers in total from the CHWs. The general opinion on using videos in health promotional situations was just like with the health training situation: positive. All 21 CHWs expressed that videos would in some way or another be beneficial as an aid in health promotion.

7 of the CHWs explained that using videos in health promotion would improve their own work. Videos will first of all be a new type of tool for most of the CHWs and one CHW explained that they had been using the flipcharts for a time, so having videos in their work would be exciting for him. One CHW

argued that they have to memorize the flipchart, which takes both valuable time and effort in their busy schedule. Another CHW elaborated on this by saying: “[...] When using the flipcharts, you have to explain a lot [more than] in the videos. In the videos it’s already explained. You only add when necessary.” We usually tried to follow this up with questions about how they would handle their relationship with the clients. We wanted to find out if the CHW would only show the video and take on a more passive role in these settings. One of the CHWs explained that if there is a need for interpreting/explaining, the CHW could pause the video and talk to the client where it is necessary. Another CHW also added that they could follow up with questions after the video.

Half of the CHWs we interviewed said that having videos in health promotion would be beneficial for their clients. The same benefits as with videos in health training situations were brought up: Videos are not static compared to pictures or texts but shows movement and combines the use of both sight and hearing. According to our translator, one of the CHWs expressed this as: “[...] This one [video] are better than the flipchart, because it’s like in action. They are talking but they are also doing action things. If they say the baby should sit like this, they are also showing.” Watching videos would in many of the CHWs’ opinion help the clients understand the topic faster and help increase recall of the health-related information. In addition, the clients can more easily relate to the current topic because they can see themselves in the videos, especially if it involves a real person and not illustrations. Not only did the CHWs mention the educational benefits of videos, but also that videos also could be a form of entertainment. Videos are, according to one of the CHWs, a sophisticated thing and would draw a lot of attention, especially in the villages. One CHW explained that he had participated in a project that involved traveling around and mobilizing people to come and watch a video they had put up. He told us that the videos attracted a lot of people since they were not only coming for the message but also because it was entertainment. Furthermore, they did not only attract a lot of people, but also those who are difficult to reach, mainly men. As one CHW said: “It is very difficult to catch men, but if there is a video, men will also come.” Videos could have a positive impact on a larger scale, as one CHW said that he thinks having videos explaining immunization programs and what it actually entails could help the clients and the public in general. Not only to understand the importance of immunization, but also reduce the skepticism that occurs. Videos could help reach out to many people and raise public awareness of the different immunization programs.

The benefits, in terms of the physical aspects of having video on a device, was again brought up by 5 of the CHWs when we asked this question. According to one of the CHWs, a negative side of flipcharts is that if one page is lost, you can’t use that flipchart anymore, or you would have to remember what was on it. Thus, having it all intact on a device would be a “Great improvement” according to the CHW.

Another CHW mentioned that having all the flipcharts in some form of video would improve their storage situation. All the flipcharts use storage-space, both when storing them in the clinics and when carrying them out in the field. Moreover, the flipcharts usually only cover one specific message according to one CHW, so having a device that can provide more than one flipchart would save space and in his opinion be easier. Additionally, some CHWs pointed out that flipcharts sometimes take time to use because they need to physically flip the pages. One of the CHWs explained that he sometimes when using the flipcharts, accidentally flips two pages at once instead of just one. However, with a video, this would be avoided.

#### 4.2.4 Desired video topics and use

After establishing the existing experiences the CHWs had with videos and their opinion on using videos in health promotion and in health-related training, we were interested in what kind of topic they would want the videos to cover. We wanted to know what the CHWs envisioned to use the video function for and get an idea of what the actual content of the videos would be like. Our second last question in the interview guide (See Appendix 1 and Appendix 2) is: “What would you have wanted presented in video form?”

All of the 21 CHWs gave their answer to this question as well. The topics we received varied in detail. Some CHWs went into detail about videos covering topics like the proper use of toilets, demonstration of breastfeeding, and danger signs for pregnant women. However, most of them mentioned overarching topics covered in the main modules and programs the CHWs were part of, such as immunization, maternal care, and malaria. The topics brought up by the CHWs could, therefore, be divided into six main categories: immunization, maternal and newborn care, hygiene and water sanitation, major health diseases, family planning, as well as nutrition.

For immunization we didn't get any more detailed answer of exactly what could be covered in a video. The same goes for family planning. We got a suggestion for water sanitation and hygiene from two CHWs, who explained that a video could show the importance and proper use of toilets. Videos covering the 6 food groups was suggested by a CHW when it comes to nutrition videos. Several CHWs mentioned that some videos covering the prevention of some of the major health diseases would be nice. We also got several suggestions for maternal and neonatal care. Some of these are:

- The importance of deliveries (birth) happening in health facilities
- Demonstration of breastfeeding
- Exclusive breastfeeding



- Resuscitation
- Warning signs for pregnant women and newborns
- Healthy nutrition for newborns
- Growth monitoring

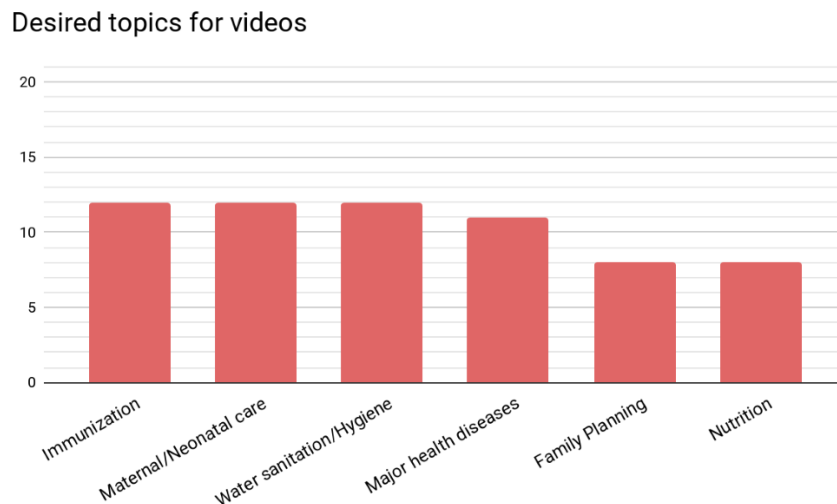


Figure 4.8: Diagram of the desired topics for videos

Given that there is more than one language in Malawi, we asked if the CHWs thought it was important to have different languages in a video. All the 16 answers we got deemed language as an important factor and that different languages are needed, ensuring that the local languages in different areas are represented. Most CHWs argued that it would make communication easier if the videos were in their local language and thus would help the clients understand and learn from the videos. One CHW said “For the user [to understand] the education [...] you have to use both English and the local language.” Another CHW stated that using the local language would make it more accessible.

## 5 Software

This chapter will discuss the underlying technical solutions that the video functionality is based on and supported by. It focuses on the very technical solutions that the user does not necessarily directly interact with.

The next two chapters, Iteration 1 and Iteration 2, focus more on the user interface, the graphical structuring of the video library, and changes that were made through the iterations we performed. It is in these areas that we did most of the practical work. To be able to support these features however, we had to think about several more technical solutions, many of which are not completely implemented or implemented at all. It is these decisions and designs this chapter focuses on, as well as the technical challenges we faced.

The finished version of all code can be seen in Appendix 3.

### 5.1 Remote video repository

For the proposed video functionality to be usable and able to scale, there are several technical and non-technical considerations to be taken into account. These systems surround the core-functionality of the video functionality in the DHIS2 Android Capture app, enabling them to work. Perhaps the most obvious part of these surrounding systems is the need to have a remote video repository.

While we could rely on bundling the video-files with the Android application and shipping them with the app, this would create a very rigid system where the user would have to re-download and re-install the app to update the library of videos available on the device. This approach could also be extended by enabling the end-user, or dedicated “technicians”, to through some menu, manually manage and download videos. This is however far too time- and resource consuming.

It thus becomes clear that some central, network-accessible (either over the internet or on a local network) repository of videos is preferable. The downloading of videos and updating of the library should also happen automatically in the DHIS2 Android Capture app, based on “instructions” provided by the remote video repository.

Initially, we looked at implementing this repository in the DHIS2 Core, using data elements. The DHIS2 Android Capture app already fetches data from the Core, and the DHIS2 Android SDK already provides the infrastructure for this communication.

However, the principles of DHIS2 focus on keeping Core as light as possible, both in size and complexity. Introducing a new use-case of hosting, managing, and providing the bandwidth for video transfers, of large video files, moves away from this principle. In addition, at the time of writing, the DHIS2 Core lacks support for storing video files, coupled with the Core development team being under heavy load. Instead, we created a solution built on a free-standing video repository server.

Due to the limited timeframe of our work, we focused most of our work on the implementation in the DHIS2 Android Capture app. When implementing the interaction with the remote video repository, we then made many assumptions on how it's exposed resources would look like. Based on these assumptions, or requirements, we created a bare-bones server to use for testing. The server we created had three simple endpoints.

### 5.1.1 Remote video repository endpoints

The first of the endpoints, `/videos`, returns a list of available videos, and includes the full meta-data structure for each video-entry, instead of a list of references and the info about individual entries being available at an endpoint such as `/video/{identifier}/info`. This break with general best practices for designing a RESTful API, but was deliberately done to limit the number of requests the DHIS2 Android Capture app needs to make to the remote video repository. However, as the repository gets larger, this might become an issue, depending on the size. Pagination could be an option to lessen the impact of this, or this API endpoint could be changed to serve a list of video IDs, which are then polled individually.

The endpoint returns data about all the videos the client making the request should have. As of now, this filtering is not implemented. In the future, the DHIS2 Android Capture app should include the user role and organization unit of the logged-in user in the request, allowing the video repository to sort based on this. This filtering should happen on the server-side, to reduce the amount of data that needs to be transferred, by applying the filters to the list of videos returned. The reduction in response payload size this enables is worth the comparatively small increase in data transferred with the request to enable this.

The second endpoint is `/video/[identifier]`, where the video identifier is a unique ID, identifying a specific video, received from the `/videos`-endpoint. This returns the video file associated with the ID. The last endpoint, `/thumbnail/[identifier]`, where the identifier is the same unique ID associated with the video, returns the video thumbnail.

### 5.1.2 Video meta-data

The meta-data about a video from the list returned by the `/videos`-endpoint is JSON data, and the format and expected fields are shown as in figure 5.1.

```
- {
  uid: "e6c0ccd0-074b-447a-9b4b-5b74f6802de2",
  fileName: "mafVideo1.mp4",
  videoTitle: "Breastfeeding: Food for Life",
  videoLanguages: [
    - {
      languageName: "English",
      index: 0
    }
  ],
  description: "Health promotion video about breastfeeding and reasons why it should be done.",
  version: "1",
  thumbnailFileName: "mafVideo1.png",
  tags: "baby, nutrition, newborn"
},
```

Figure 5.1: Video meta-data

The `uid`-field is the unique identifier for the specific resource (video). It should be generated by the video repository when a video is added. The identifier should be uniquely generated, for example using UUID version 4, which has minuscule chances of identifier collisions occurring.

The `filename`-field is whatever the file is stored as in the filesystem of the video repository. In our bare-bones server for testing, we kept the files and filenames as when they were updated, but when a proper remote video repository is made, it should also generate the filenames.

The `videoLanguages`-field contains an array of objects, each representing a language-track available in the video file. The `languageName` is the human-readable name of the language. The `index`-field of a `videoLanguage`-object is the index of the language track (in the video file) associated with the `languageName`. This is needed since the audio-tracks in video files are commonly identified using ISO 639-2 codes. However, the mapping between these codes and the human-consumable language name is not always as desired, as we experienced during our implementation. The local language of the part of Malawi we visited is Chichewa. Chichewa has ISO 639-2 code `NYA`. This code is associated with

three language names (Chichewa, Chewa, and Nyanja). When we tried programmatically converting the ISO 639-2 code to a language name, using 'java.lang.Locale', 'NYA' was displayed as Nyanja. As far as we're able to tell all of these languages are (mostly) the same, but despite this, we decided to implement this solution of 'languageName' and 'index' (corresponding to the index of the given language in the array of language-tracks in the video file) to avoid any end-user confusion (e.g. looking for a Chichewa video, but can only find a Nyanja video). This did however add more complexity to the displaying and user-selection of language track, which we will elaborate on in chapter 7.1.9.

The 'description'-field contains a human-readable description of the video. The 'version'-field is an incrementally increasing integer denoting the version of the video that is available. The 'thumbnailFileName'-field contains the filename on the filesystem for the thumbnail associated with the video. Like with the video file, these are in our bare-bones system just kept as what they are uploaded as but should have names generated for them.

The 'tags'-field is a comma-separated string containing tags/keywords for a video, and are used by the search functionality.

### 5.1.3 Video repository management

The remote video repository should include a management interface. The base-functionality it should have includes uploading videos (adding videos to the repository), deleting videos, updating videos, storing the previously described meta-data and updating it, versioning, and define which users need/should have which videos, decided by the user's user role and organization unit. Since the DHIS2 Android Capture app simply asks the remote repository which videos it should download, the repository needs to keep a record of which videos users of specific roles and organizational units should have.

While the actual video production is outside of the scope of this thesis, the videos should be produced so they contain separate audio tracks for each language the video should support. The videos are then uploaded and added to the video repository, along with meta-data for the video. Administrators at an appropriate level then assign which user groups (based on user role and organization unit) should receive which videos and add this information to the remote repository.

#### 5.1.4 Video format

To fulfill the requirement of the audio of videos existing in different languages, we initially planned to serve one video file, and then individual audio files for each language. When downloaded, the DHIS2 Android Capture app would then simultaneously play these files, seeming to the user as one file. This would minimize the size of files needed to be downloaded, which is something we need to be mindful of due to the availability and cost of stable, high-speed, internet connection in the areas the DHIS2 Android Capture app is intended to be used.

As we explored this, we learned that this syncing of playback of two media-files while possible, would be time-consuming, require third-party libraries, or both. We instead decided to package the audio-tracks with the video files, meaning each video file would contain the audio-tracks for each language the video is available in. This does slightly increase the total file size but was a necessary compromise to finish our prototype in the time available. Having one video file for each version (language) of the video makes the individual video files smaller but leads to a net increase in size if a device needs to have a video in several languages since the device then needs to store 2x the video and 2x audio track.

For testing our prototype, we used mainly videos sourced from our guide, Joshua, in Malawi. The videos were produced for Medical Aid Films (MAF) and were distributed on DVDs to be used in the communities. We converted the videos from VOB-files containing MPEG-2 Video (H.222/H.262) to mp4-format (H.264 AVC). H.264 AVC was 'chosen' as it was the default option for the conversion software we used, VLC media player.

In addition to this, we used two WHO videos about HIV-self testing (where one was a full version, and the other was a shorter version). These videos were not converted, as they were obtained in mp4-format (H.264 AVC). Due to their higher resolution than the other videos, their files were substantially larger. The runtime and size of some of the videos used are listed in table 5.1.

We did our testing on a Huawei MediaPad T5 tablet. This tablet is relatively cheap (1 300 NOK as of writing), and it would be logical that per-device cost would be a factor to consider for any large-scale roll-out of devices for CHWs in developing countries, leaning towards cheaper offerings. The device has 16 GB of internal storage, of which more than 6 GB was used by firmware on our device, leaving available storage at less than 10 GB. This storage can be expanded with external media (such as an SD-card).

| Title       | Runtime     | Resolution/Frame rate | Size     |
|-------------|-------------|-----------------------|----------|
| MAF video 1 | 14min 34sec | 720x576 / 25 FPS      | 83.7 MB  |
| MAF video 2 | 11min 12sec | 720x576 / 25 FPS      | 63.2 MB  |
| MAF video 3 | 19min 57sec | 720x576 / 25 FPS      | 98.5 MB  |
| MAF video 4 | 13min 14sec | 720x576 / 25 FPS      | 89.4 MB  |
| MAF video 5 | 15min 07sec | 720x576 / 25 FPS      | 84 MB    |
| HIV long    | 26min 50sec | 1280x720 / 25 FPS     | 504.1 MB |
| HIV short   | 2min 30sec  | 1280x720 / 25 FPS     | 28.5 MB  |

Table 5.1: Video runtimes, resolutions, and file size

We did our testing on a Huawei MediaPad T5 tablet. This tablet is relatively cheap (1 300 NOK as of writing), and it would be logical that per-device cost would be a factor to consider for any large-scale roll-out of devices for CHWs in developing countries, leaning towards cheaper offerings. The device has 16 GB of internal storage, of which more than 6 GB was used by firmware on our device, leaving available storage at less than 10 GB. This storage can be expanded with external media (such as an SD-card).

It is difficult for us to estimate how many videos would need to be available, locally on their device, to a CHW at a time for the solution, and intervention, to have the desired effect. This number also depends on how much of the CHWs outreach-work and training the health provider wishes to leverage the video-solution for. It is likely to be a case of the more the better, up to a certain threshold. However, we believe that it would not be unlikely that this number would be higher than 10-20 videos.

Based on this, and the file-sizes of the videos we used for testing, it becomes clear that we should try to keep file sizes as small as possible. This can be done in several ways, for example by using efficient video encoding formats and reducing the resolution of the video. During our testing, we converted some of the videos to mp4 H.264 AVC, as this was the default encoding settings in the tool used for the conversion, VLC Media Player. However, the H.265 HEVC-codec would be more efficient, but this was not used during our field testing, as we did not have issues with storage space and therefore simply did not think of it. As an informal test, we converted the short version of the WHO HIV self-testing video. These conversions were made using the x.264 and x.265 encoders. The results can be seen in table 5.2. Running the same conversions for the full version of the WHO HIV self-testing video yields the results presented in table 5.3.

| Codec               | Resolution | File size |
|---------------------|------------|-----------|
| H.264 – MPEG-4 AVC  | 1280x720   | 28.5 MB   |
| H.265 – MPEG-H HEVC | 1280x720   | 17.5 MB   |
| H.265 – MPEG-H HEVC | 720x406    | 7.9 MB    |

Table 5.2: Table showing codec, resolution, and resulting file size for WHO HIV self-testing video (short version)

| Codec               | Resolution | File size |
|---------------------|------------|-----------|
| H.264 – MPEG-4 AVC  | 1280x720   | 504.1 MB  |
| H.265 – MPEG-H HEVC | 1280x720   | 207.2 MB  |
| H.265 – MPEG-H HEVC | 720x406    | 86.4 MB   |

Table 5.3: Table showing the codec, resolution, and resulting file size for the WHO HIV self-testing video (full version)

The switch from H.264 to H.265 HEVC showed considerable savings in file size while maintaining the same video quality (Koumaras et al., 2012, De Cock et al., 2016). The further change from 1280x720px resolution to 720x406 yielded even further savings. In total, a size reduction of roughly 72% for a 2.5-minute video clip is quite substantial.

Due to the screen size of the devices the video library is intended for, and the often lower resolution of phone and tablet screens (especially those at the lower end of the market), we believe a reduction in resolution from, in this example, 1280x720 to 720x406 will not negatively impact the viewing experience and the videos ability to convey information.

### 5.1.5 Video propagation/downloading

Another important reason to limit video file size beyond the limited storage space on user devices is the infrastructure over which these videos will be transferred. As the DHIS2 Android Capture app is intended for use in developing areas, existing internet infrastructure might be unstable, slow, expensive, or unavailable.

Distributing the video files is obviously an important part of the system surrounding our video library. Unfortunately, it's not a topic we've had the time to explore. We have, however, made some



assumptions that guided our designs. One of them is that smaller file sizes would naturally be better in situations with constrained infrastructure.

In areas where the infrastructure allows for it, the syncing of the video library in the DHIS2 Android Capture app could happen over the internet. However, in situations that do not allow this, it is fully technically possible to physically move the files from a location with a better connection (where they can be downloaded over the internet), for example a district office, to an area the CHWs naturally gather, such as a clinic. This move could be done on a normal PC, which could also host an instance of the video repository and create a wireless-network for the client-devices to connect to, assuming the PC has a wireless networking interface.

To avoid having to reconfigure the DHIS2 Android Capture app for this purpose, the PC serving as the movable remote video repository might have to also run a DNS server, so that an IP/hostname can be pre-determined and configured in the app, so the connecting clients would connect to its server. This would however get very complicated when the communication between client and server runs over HTTPS. This would either then require the movable host to carry a valid, trusted, certificate, which would not be recommendable security-wise, or it might require the technician bringing the computer, or the device users, to reconfigure their (client) devices to hit the specific IP of the computer (on the local network)/toggle the app into a “local video sync”-mode.

## 5.2 Introducing videos to the DHIS2 Android capture app

Keeping in mind that the implementation we have done as part of prototyping, there are several sub-optimal or even wrong solutions in our implementations. Some of these mistakes were definitely also made due to inexperience with the Android APIs and lack of knowledge. As this sub-chapter progresses, we will comment on some of these, offering ways specific things should be improved or changed in the future.

In the following chapters, we will refer to ‘Context’ several times. This is ‘android.content.Context’, which is an “interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.” (Google, n.d. A)

### 5.2.1 Internal database and database communication

To keep track of which videos have been downloaded to the device, we implemented a very simple ‘android.room.Room’ database. The database contains two tables, or entities as they are called in Room-databases. Room is a “[...] persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite.” (Google, n.d. B)

The first of these entities (tables) is ‘StoredVideoEntity’. It uses the UID received from the remote video repository as a primary key and contains the fields ‘video\_title’, the title of the video to be displayed to the user; ‘file\_name’, the name of the file on the file-system; ‘description’, the human-readable description of the video; ‘version’, the version number of the video; and ‘thumbnail\_file\_name’, the filename of the thumbnail picture on the file-system. The ‘StoredVideoEntity’ Java class also contains a ‘List’ of ‘VideoLanguageEntity’-objects. This list is not written to or read from the database, as indicated by its ‘@Ignore’ annotation. This variable is instead set “manually” when reading the ‘StoredVideoEntity’ from the database by getting the relevant ‘VideoLanguageEntity’-entities from its database, and then using a setter-method to set it.

The other entity is ‘VideoLanguageEntity’. It stores the previously mentioned ‘videoLanguages’-object, one language per row, using the UID and an auto-incrementing field as the primary key. In addition, it contains ‘language\_name’, the human-readable name of the language; and ‘index’, the index of the specified language in the array of audio-tracks in the video container file. The ‘VideoLanguageEntity’ also defines the selected strategy for foreign keys being deleted, namely to cascade (delete the entities from the table).

In addition to this, there are Data Access Objects (DAO, ‘androidx.room.Dao’) which defines interaction with our database, through specifying database queries. There is one DAO for each Entity. These DAOs define the conflict resolution strategy, which are all set to replace. The abstract class ‘VideoDatabase’ represents the actual database, and “houses” these DAOs.

To manage the instance of this video database, and prevent the creation of multiple instances, we use the ‘VideoDatabaseInstance’ class. It stores a static reference to an instance of itself, ‘dbInstance’, and a reference to an instance of ‘VideoDatabase’ as its class variables. Its constructor uses the received ‘Context’ to create an instance of ‘VideoDatabase’, which gets stored in the ‘videoDatabase’ class variable. To avoid creating several such instances, we never call this constructor directly. Instead, we call the static ‘getInstance’-method of the class, which will return ‘dbInstance’ if it exists, and create it

and return it if it does not. From this instance, we can then get the instance of the database class by calling the 'getVideoDatabase'-method.

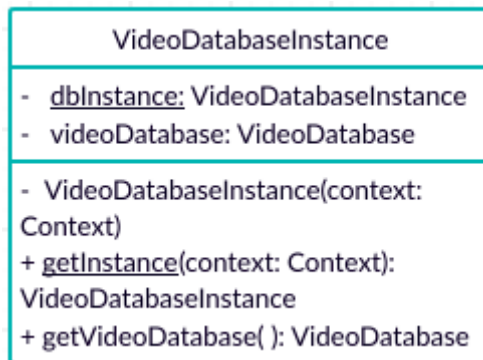


Figure 5.2: Class diagram of 'VideoDatabaseInstance'

Figure 5.3 shows an example of how the database is communicated with. The figure abstracts away the 'Room.databaseBuilder' creating a connection to the database using the 'VideoDatabase' class, which in turn uses the DAOs 'StoredVideoDao' and 'VideoLanguageDao' to interact with the database tables, represented by the entities 'StoredVideoEntity' and 'VideoLanguageEntity'.

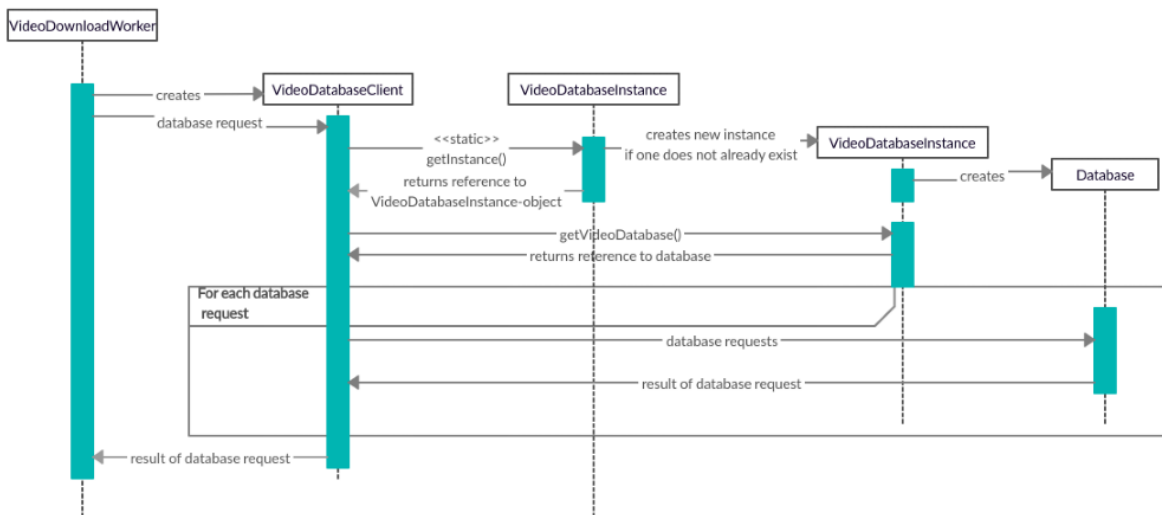


Figure 5.3: Sequence diagram showing database communication

```

1 package org.dhis2.data.videoDatabase;
2
3 import android.content.Context;
4 import androidx.room.Room;
5
6 public class VideoDatabaseInstance {
7     private static VideoDatabaseInstance dbInstance;
8     private VideoDatabase videoDatabase;
9
10    private VideoDatabaseInstance(Context context) {
11        videoDatabase = Room.databaseBuilder(context, VideoDatabase.class, "video-database")
12            .fallbackToDestructiveMigration()
13            .build();
14    }
15
16    public static synchronized VideoDatabaseInstance getInstance(Context context) {
17        if (dbInstance == null) {
18            dbInstance = new VideoDatabaseInstance(context);
19        }
20        return dbInstance;
21    }
22
23    public VideoDatabase getVideoDatabase() { return videoDatabase; }
24 }
25

```

Figure 5.4: Code of 'VideoDatabaseInstance'

The app communicates with the database through an instance of 'VideoDatabaseClient', which extends 'android.os.AsyncTask'. The use of 'AsyncTask' ensures that we do not perform database calls on the thread where the need for said database calls arises. This is mainly an issue when this "source thread" is the UI thread, which would be blocked if used to make database calls. The constructor of 'VideoDatabaseClient' receives the context of the caller and a taskID. Then, to execute the specified task, the 'executeOnExecutor'-method is called, using 'AsyncTask.THREAD\_POOL\_EXECUTOR'. This ensures that the executions can be run in parallel.

'VideoDatabaseClient' currently only does three tasks, namely get all the 'StoredVideoEntity' entities in the database and their related 'VideoLanguageEntity'-entities, insert a 'StoredVideoEntity' into the database (and it's related 'VideoLanguageEntity'-entities), and delete all 'StoredVideoEntity'-entities from the database (used during testing).

### 5.2.2 Video downloading and storage

Android provides two 'location' for storing data, namely internal and external storage. Internal storage should be used when storing data that should be private to the app and will not be accessible from any other app. External storage can be accessed by other apps. Since the video files and thumbnails we are storing are not sensitive, and do not need to be private, we opted to store them in the external storage.

The download process for both the video files and the thumbnails are the same. We open a `'java.net.HttpURLConnection'` to the remote resource, for example `'http://videosever.com/video/[identifier]'`. Then we create a file in the root of our app's external storage directory, using the UID provided by the remote repository as the filename (and appending the filetype). In the future, we should write these files to a sub-folder, and not the app's root directory.

We then open a `'java.io.FileOutputStream'` to said file, and a `'java.io.InputStream'` connected to the `'HttpURLConnection'`. We then read from the input-stream and write to the output-stream, until we have transferred the whole file.

In our implementation, this process will not recover intelligently from interruptions. `'HttpURLConnection'` will silently retry if a connection issue is encountered before it throws an exception. If the process is interrupted, the data that has already been written to the file will remain, as we do not delete the file. However, if either the video-file download or the thumbnail-file download fails (throws an exception) we do not add the video to the internal database listing available videos on the device. This process should be improved, for example by automatically deleting incomplete media and adding said file to a retry-queue.

We also discovered that instead of writing the files directly to external storage, we should have been using the `'android.provider.MediaStore'` API, at least for the video files. By writing files directly to external storage, they are stored in, for example, `'Android/data/com.dhis2.debug/'` in the device filesystem. Files stored in this way will not appear in the photo or video library app of the device, and manually navigating to them to view them (outside of the DHIS2 Android Capture app) is not very user friendly (using the "Files" app). Since the videos could be useful to users outside of the app, it could be an idea to allow this, which the `'MediaStore'` API achieves. However, implementing this should not be a large or difficult change to the code.

### 5.2.3 Video download Worker

The downloading of video and thumbnail files are handled by `'VideoDownloadWorker'`, which extends the `'android.work.Worker'`-class. By doing so we can ensure that the contents of its `'doWork'`-method is performed synchronously on a background thread, provided by a `'WorkManager'`. (Google, n.d. C)

The doing of work is triggered by creating an `'androidx.work.WorkRequest'` and enqueueing it in `'WorkManager'`, which then handles thread allocation and the lifecycle of the worker threads. Doing the downloading in a `'Worker'` offers several advantages beyond this. By using an

'androidx.work.PeriodicWorkRequest' we can schedule repeating of the work. Using a worker also allows us to use 'androidx.work.Constraints' to define requirements for when 'WorkManager' should do the defined work. These 'Constraints' can be things such as the presence of a network connection, what kind of network connection (metered/unmetered, etc.), battery level, and available storage space. A worker will also automatically, unless otherwise specified, be retried if it fails. The default retry strategy is exponential back-off.

The remote video repository server URL is stored in a property file, 'videoServer.properties'. The information is stored in key-value pairs, e.g. 'repository\_url=http://example.com'. The contents of this property-file are read into a 'java.lang.Properties'-object, through which we can read specific values by their key. The 'VideoDownloadWorker' loads this information, and then uses the URL to complete the video downloading.

First, we fetch the list of videos to download and the meta-data about them. We use 'Gson', a Java serialization/serialization library made by Google, to convert the JSON we receive into Java objects. 'Gson' automatically splits the list of objects in JSON-format into a 'java.util.List' of 'StoredVideoEntity'-objects. Since these JSON objects, as shown in figure 5.1, contains both the video meta-data and a list of available language tracks, we need a custom 'Gson' deserializer, 'StoredVideoEntityDeserializer'. This deserializer creates a 'List' of 'VideoLanguageEntity'-objects and associates it with the created 'StoredVideoEntity'.

We then compare this list with the list of all videos currently stored on the device, which is fetched from the database. This is done by overriding the 'equals'-method of 'StoredVideoEntity'. Two 'StoredVideoEntity'-entities are the same if they have the same UID and version number. This way we avoid downloading videos that we already have but ensure we download videos already stored if the version number received from the remote video repository indicates the video has been updated.

This filtered list is then sent to a method that loops through it, downloading the video file, thumbnail file, and adding a record to the database for each video. By adding a new record to the database for a video we already have a record for, we replace the existing record, instead of manually updating it. The records are primary-keyed to only the UID in the database, and not the version number. Once this is done, the worker ends with a 'Response.success()'-message. If any exceptions are thrown, the worker returns a 'Response.failure()'-message instead, and the worker is retried later.

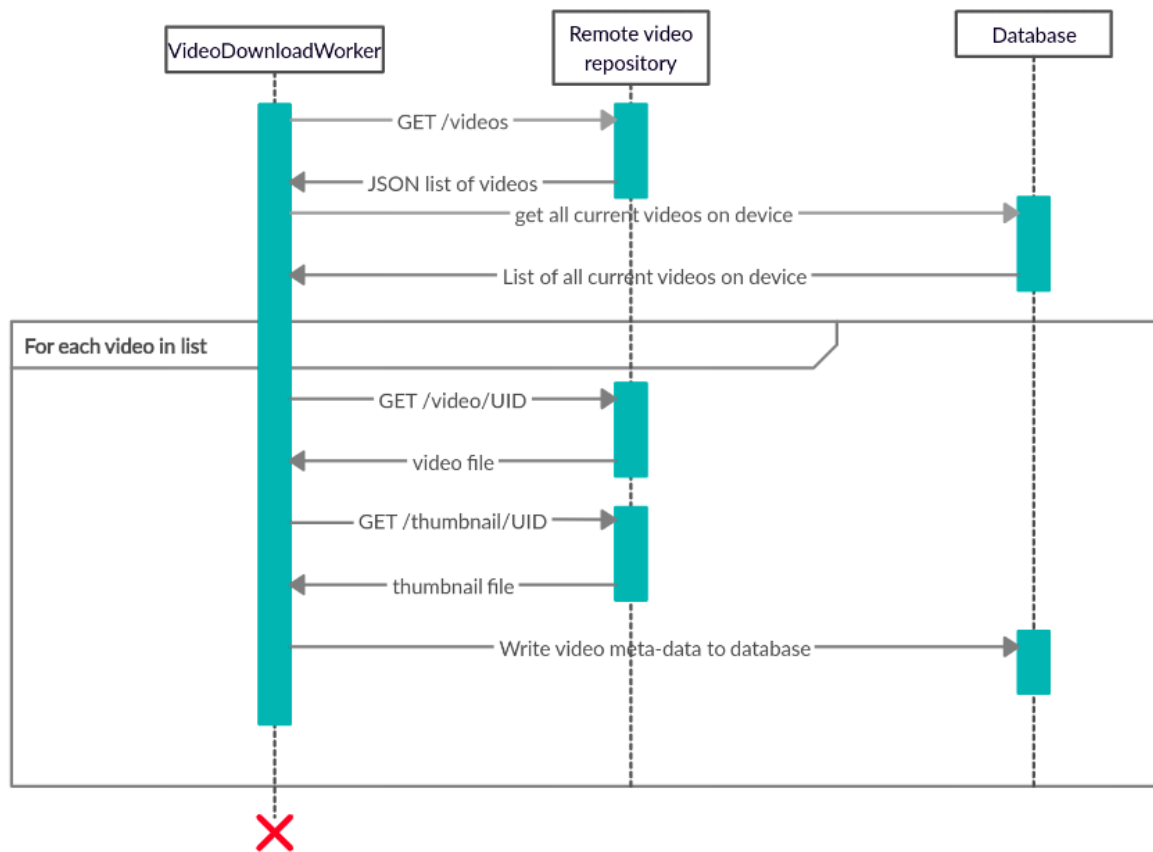


Figure 5.5: Sequence diagram showing the video downloading process

In its current form, one worker fetches the list of videos, and then downloads new/updated videos, synchronously. It should however be changed so that each video (as in, the download of the video file, the download of the thumbnail, and the writing of video meta-data to the database) is handled by its own worker. This way, one video download failing will not cause us to have to retry the entire list, as is done when everything is done by one worker. A worker is also only given 10 minutes to run (Google n.d. C), which would lead to problems with larger catalogs of videos to be downloaded.

In our prototype, we simply trigger 'VideoDownloadWorker' through pressing a button in the UI, since our goal was not to test the interaction between the DHIS2 Android Capture app and the remote video repository. This button was of course hidden when testing on users. In a finished version, the solution for triggering the downloading process should require as little work and know-how from the end-user as possible. One solution could be to use a 'PeriodicWorkRequest' and thereby scheduling the worker to run periodically, for example once per week. If there are no new videos or updated videos, the fetching of the list of videos is fairly lightweight.

The worker could also be triggered remotely through notifications, either from the DHIS2 Core or by having the DHIS2 Android Capture app subscribe to the remote video repository. These notifications can either be human-readable (“New videos available, click here to download”) or consumed by the app itself.

The selected strategy for triggering the worker could then also be combined with the use of ‘Constraints’, for example by only allowing the worker to run while the phone is charging or does not have a low battery charge.

Beyond those already described, we have taken several shortcuts in our implementation. Some examples of this follow.

The DHIS2 Android Capture app does have an already established database, but due to our time constraints, we elected to create a new database for our purposes. This should be rectified in the future.

There is currently no way to remotely remove videos from the device. One solution to this could be to delete any videos on the device that are not in the list of videos received from the remote video repository.

Another possible feature that currently does not exist but might be interesting in terms of the administration and oversight, is tracking video views and uploading this to the remote video repository, as these statistics might be useful for the administrators. This is however not something we have looked at whatsoever.



## 6 Iteration 1

As outlined in the Software chapter, this chapter and Iteration 2 will describe the workings of the graphical and UI parts of the video library addition to the DHIS2 Android Capture app. As such, we will begin with an outline of how these parts of the app worked and looked at the beginning of our first iteration. These designs were based on the data collected during our requirements gathering. After this we will present the data and observations collected during the iteration. Our user interface design strategy was to follow already established standards for similar interfaces and use the well-established Android interface elements. The reasoning behind this was the belief that the CHWs had prior experience with such interfaces, and that their knowledge and mental models would be transferred onto our implementation, thus increasing usability and decreasing the difficulty of learning our interface.

This chapter covers the action planning, as well as the evaluation and specifying learning phase in the first iteration. The data presented and discussed here is related to the software, while data related to the use of videos is presented in chapter 4.2.

### 6.1 Software implementation

Graphically, in the DHIS2 Android Capture app, our additions consist of mainly two parts: a video player, and a video library. In addition to this, we add a new option in the main app hamburger-menu. Since this is where the user's interaction with our additions begins, this is where we will begin our explanation as well. This additional menu option, "Video library", is shown in figure 6.1. Finding an appropriate icon for the menu option was not prioritized, and the menu icon for the "Log out"-button was reused.

This was achieved by adding an additional 'item' to the layout-file specifying the menu items of the hamburger menu, 'main\_menu.xml'. In addition to this, the required logic was added to 'MainActivity.kt', which handles interaction with the menu items in the menu.

The main activity of the DHIS2 Android Capture app uses 'Fragments' in order to dynamically change between several 'views' that the same Android 'Activity' is responsible for.

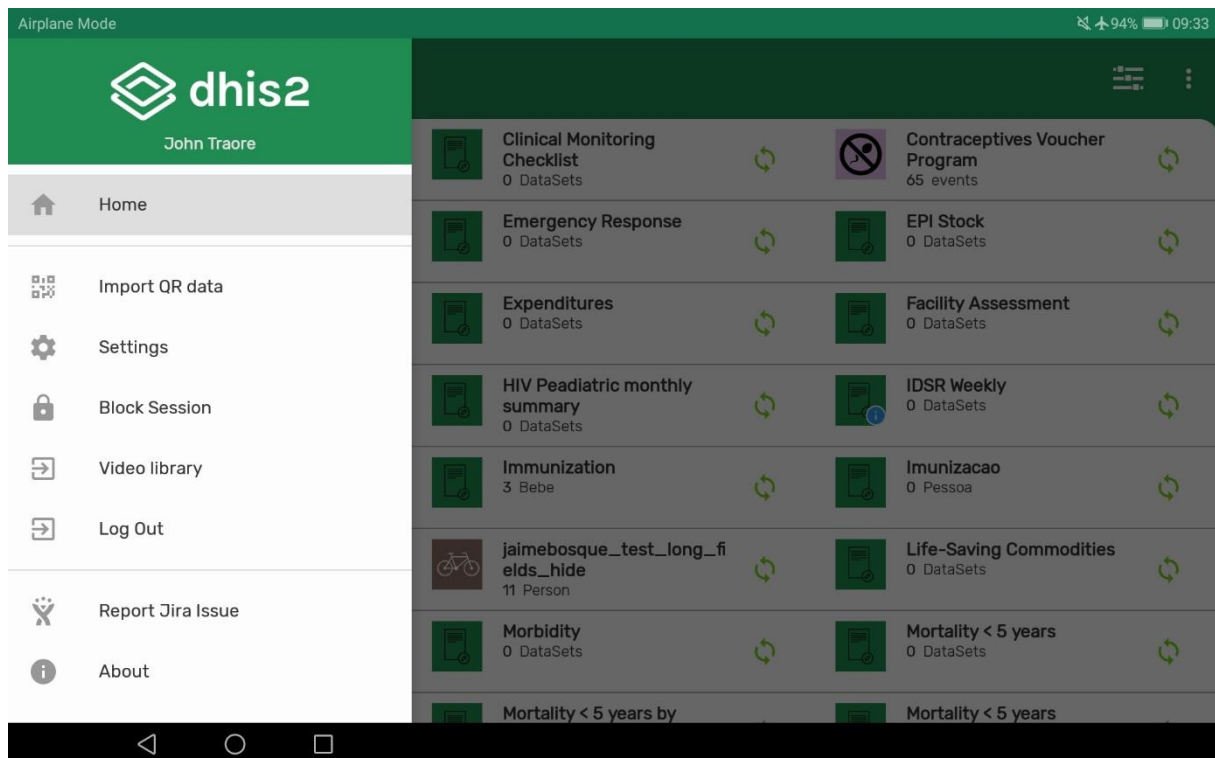


Figure 6.1: Screenshot of the main screen of the app with the hamburger-menu expanded

An ‘Activity’ is “[...] a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View).” (Google, n.d. D) An ‘Activity’ is one of the fundamental building blocks in creating UIs in Android.

A ‘Fragment’ “[...] represents a behavior or a portion of user interface in a FragmentActivity. [...] You can think of a fragment as a modular section of an activity, which has its own lifecycle, receives its own input events, and which you can add or remove while the activity is running (sort of like a "sub activity" that you can reuse in different activities).” (Google, n.d, E) Fragments are often used when you want several “screens” to contain some of the same elements but want to switch some of the other elements. A ‘Fragment’ is always contained in an ‘Activity’. To use a specific case as an example, ‘MainActivity’, an ‘Activity’, contains and controls the UI elements related to the hamburger menu and its functions. It also contains different ‘Fragments’, most of which represent screens for the different options in the main menu. When navigating between these ‘Fragments’ the main menu of the ‘MainActivity’ ‘Activity’ (as well as any other UI elements defined by the ‘Activity’) are still displayed and can be interacted with.

Pressing the “Video library” menu option changes the view to a new Fragment, ‘videoLibrary.VideoFragment’, which is the graphical view of the video library. Since the DHIS2 Android Capture app uses dependency injection, handled by the Dagger library, we for this fragment follow the architecture of the rest of the app. This is done since, while this fragment only handles our additional features, it is still controlled by the ‘MainActivity’-activity. As such, we have a ‘VideoPresenter’, a ‘VideoComponent’, a ‘VideoModule’, and a ‘VideoContracts’, which follows the architecture in the DHIS2 Android Capture app for injecting ‘Fragment’ objects.

The video library ‘Fragment’ is currently named ‘VideoFragment’ (located in the ‘videoLibrary’-package). Likewise, the video player class is currently named ‘VideoActivity’ (located in the ‘videoPlayer’-package). These are not good names and were left on accident. They should be changed to more descriptive names, such as ‘VideoLibraryFragment’ and ‘VideoPlayerActivity’.

The reason the video library view was made as a fragment, was to keep the Android app bar (shown in figure 6.2. The app bar is the green bar at the top of the screen which houses the hamburger-menu and “Video Library” heading), which includes the hamburger main menu. This app bar and menu is handled by the ‘MainActivity’-activity, and as such the video library needed to be a fragment presented by this activity.

First, the fragment uses ‘VideoDatabaseClient’ to get the list of videos from the database. Using ‘VideoDatabaseClient’ ensures that this work is not done on the UI thread. The fragment uses a ‘RecyclerView’ to list all the videos. A ‘RecyclerView’ is a view object where a specific layout is recycled, or repeated, for a list of objects. To achieve this, an ‘Adapter’ is used. We send the adapter, ‘VideoListAdapter’, our list of ‘StoredVideoEntity’-objects, as well as a lambda expression defining what should happen if a video is clicked on.

The adapter is then responsible for iterating over this list, and for each object populating the object’s data into a specific item layout, ‘item\_video\_list’. This group of populated layouts is then added to the RecyclerView and displayed in the fragment.

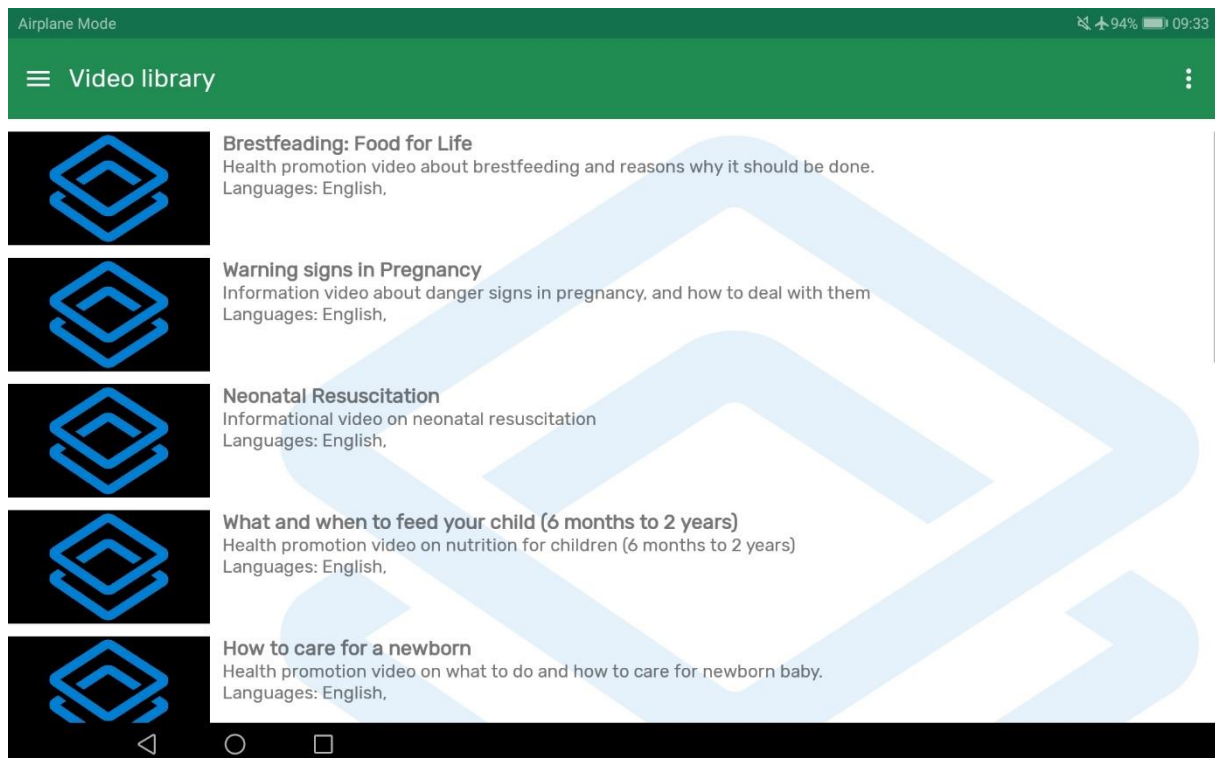


Figure 6.2: Screenshot of video library

Figure 6.2 shows how the video library looked at the beginning of user testing in iteration 1. For each video in the list, there is a title, a description text, and a list of languages the video is available in. At this stage in development, support for selecting a language track was not implemented, but the feature was planned and as such the information is displayed. In addition to this, to the left of the entries, was a video thumbnail. This thumbnail was at this point just a placeholder image, the same for each video, as the logic for downloading and handling thumbnails had yet to be implemented.

At this point in the development (beginning of user testing in iteration 1), only clicking the video thumbnail would “open” the video (start playing it). When this is done, we launch a new activity, ‘videoPlayer.VideoActivity’. When the new ‘Activity’ is started, we use an ‘android.content.Intent’ to transfer information to the new activity. The information transferred is the filename of the video selected. This information can then be retrieved using the ‘getIntent’-method of an ‘Activity’-class.

The video player is implemented as its own ‘Activity’, rather than as a ‘Fragment’. This is because the video player will be handling functionality that is substantially different to that of ‘MainActivity’. In addition, we did not want the app bar with the hamburger menu on this screen.

The creation routine of the video player 'Activity' is defined in the 'onCreate'-method of the class. When created, the video player does several things in addition to retrieving the filename of the video selected from the 'Intent'. First, we use the 'getWindow'-method of an 'Activity' to get a reference to the 'android.view.Window' of the activity. We need this reference to be able to add a "flag" to the 'Window', namely the 'FLAG\_KEEP\_SCREEN\_ON'-flag. Normally, the screen of an Android device will dim and then turn off if the user does not interact with the device for a set time. This is obviously not optimal for video viewing, and by setting this flag, we avoid this. The flag ensures that, as long as the 'Window' is visible to the user, the screen remains on and is not dimmed.

We then set the "content view" of the 'Activity'. This view is defined by a pre-defined layout-file. This step is standard in 'Activities'. We then create a reference to a specific part of the layout, namely the 'android.widget.VideoView', which is responsible for displaying and playing the video file.

The 'VideoView'-class uses an 'android.media.MediaPlayer'. We need direct access to methods of this class ('MediaPlayer') to be able to read a list of available audio-tracks and change the selected audio track. Since we planned for implementing language selection later, we at this point implemented functionality to retrieve a reference to 'MediaPlayer' from the 'VideoView' instance.

We then create an 'android.widget.MediaController', and add it to the 'VideoView'. Doing this adds video controls to the video player, such as play/pause, which is handled by the 'MediaController'. This functionality does not come standard with a 'VideoView'. When we create the 'MediaController' we also override its 'show' function. This function is called (for example) when the user taps the video, and triggers the video controls to be shown. By default, they then fade out after a set time. By overriding this method, we can change this set time. The controls can be hidden by tapping the video again. This was done since some users might not be familiar with video controls and having the video controls fade away while the user is processing them and deciding how to act is not a good idea.

Notice that we have still not set the 'VideoView' to play our video. This is done in its own method, 'initializePlayer'. This method also handles requesting permission from the user, if required, to read from external storage (where the videos are stored). This method is called from the 'onStart'-method, which is called when the 'Activity' is started.

The 'VideoView' was at this point in time set to have an aspect-ratio of 4:3, and full-screen functionality did not exist.

In addition, changing the device orientation (landscape/vertical mode) would start the video playing from the start. This is because an orientation change is a “configuration change” (Google, n.d. F), and Android (temporarily) destroys (and recreates) ‘Activities’ on configuration changes, which causes the loss of the current state of the video being played. We handled this by ensuring the instance state is correctly saved and loaded by using the ‘onSaveInstanceState’- and ‘onRestoreInstanceState’- methods. By storing the current playback position in this way, and then restoring it when the ‘Activity’ is created again by setting the playback position in the video player to the given position we solve this problem.

To navigate back to the list of videos, the user would have to click the back button in the Android navigation bar, which an example of is highlighted in figure 6.3.

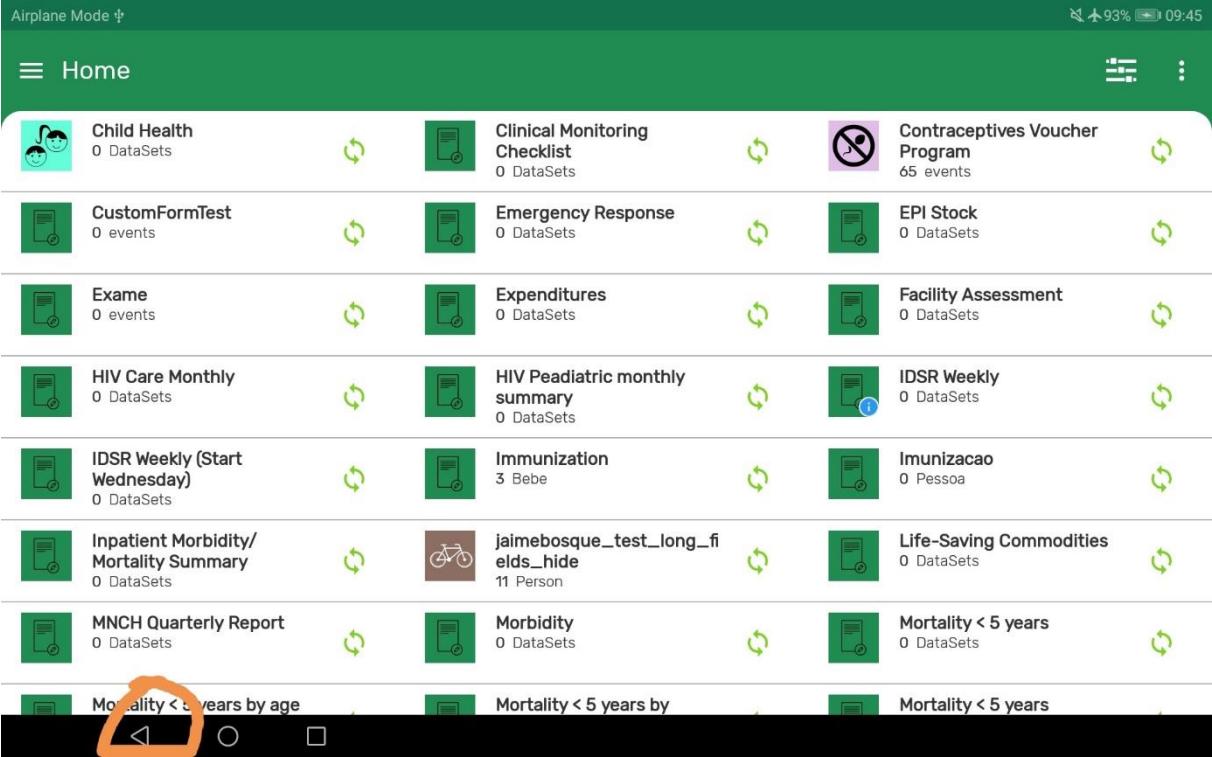


Figure 6.3: Example of Android back button highlighted

## 6.2 Data collection

Based on the implementations described in the software chapter and those described so far in this chapter, we performed observations of CHWs using the functionality and then interviewed them. First, we would demonstrate the library and player functions, showing them how it worked. We would then ask them to perform specific tasks. For the first two sessions, the videos we used for testing were simple videos recorded using a web camera of one of us saying, essentially, “hello, hello this is a test video”. After these two sessions, our guide, Joshua, sourced some more relevant videos for us to use in testing. These videos were produced by Medical Aid Films. Some of them were targeted at health workers, or CHWs specifically, while some of them were targeted at patients/clients. The videos are in English. In addition to this, we had two videos made by WHO, about HIV self-testing (a long and short version). These two videos from WHO were in Chichewa, the local language in the region of Malawi we visited. All of these videos were very good for testing, as they were good examples of videos the library is likely to include. At the time of writing, all of these videos could be easily found with an online search. To make the list in the video library longer (to test scrolling), we added each video to the list twice. The list of videos used can be seen in table 6.1.

| <b>Title</b>  | <b>Language</b>             | <b>Source</b>     |
|---|-----------------------------|-------------------|
| Food for life: Breastfeeding  | English                     | Medical Aid Films |
| Warning signs in pregnancy  | English                     | Medical Aid Films |
| Neonatal Resuscitation  | English                     | Medical Aid Films |
| Food for life: What and when to feed your child (6 months to 2 years) | English                     | Medical Aid Films |
| How to care for a newborn   | English                     | Medical Aid Films |
| In our hands (HIV self-testing)                                       | Chichewa, English subtitles | WHO               |
| In our hands (HIV self-testing) (short version)                       | Chichewa, English subtitles | WHO               |

Table 6.1: Table showing the videos we used for testing

**In this iteration, we performed 10 interviews and observations. We made the following observations:**

- 1.** We observed two users attempting to open the video by clicking on the video title. This did not work, as expected.
- 2.** We observed two users having issues finding the video controls. Users unfamiliar with how video playing in smartphones work might not know that tapping the video generally toggles the video controls showing.
- 3.** We also observed two users being confused by the main menu of the DHIS2 Android Capture app (the hamburger-menu). This would be a potential issue with the DHIS2 Android Capture app itself, as this is outside of what we've added.
- 4.** Two users were also confused about what the "video library" was, as opposed to the main view of the DHIS2 Android Capture app, and if the video library was inside one of these programs.

We believe this stems from our demonstrations since the main menu button is close to one of the programs (and it was this program the users tried to open to find the video library), and that the users might have thought we clicked this program to access the video library.

As part of our introduction, we would quickly explain the basics of what DHIS2 is, and what the DHIS2 Android Capture app was. As part of this, we would show them the main screen, telling them that it contained a lot of "programs" where data would be entered.

- 5.** We also observed one user having trouble using the Android navigation bar back-button to exit a playing video. This could be because of unfamiliarity (since different Android-devices implement these buttons differently), or since the buttons are half grayed-out when the video controls are not showing.



| Observation                                      | Frequency |
|--|-----------|
| Clicking video title to open video               | 2         |
| Issues finding video controls                    | 2         |
| Confused by main-menu                            | 2         |
| Confused about video library VS main view of app | 2         |
| Difficulties with back button                    | 1         |

Table 6.2: Summary of observations in iteration 1

**From our interviews we gathered the following information:**

1. One user pointed out that the list of videos in the video library was missing thumbnails, and thought that thumbnails would make the video library easier to use, saying “If I come to the videos, in most videos you find there is [a thumbnail]. ‘Warning signs in pregnancy’ I will read, but I don’t know if there will be some picture [describing the video, a thumbnail]. [...] I see a picture [and] I [understand] ‘Ok, this is about pregnancy’.”

2. One user pointed out that new users of the system would require training.

3. One user pointed out that the effectiveness and usability of using videos would depend on the screen size of the device used. A (too small) screen might not be large enough for showing videos to crowds of people and only be good for showing videos at the household-level.

When asked if he had any comments about what we were doing, he said “Yeah the application is good. But I don’t know, if we have a large crowd of people, and they want to watch on maybe a small phone like this [points to iPhone SE], [I] don’t know how it will work, because you might need a larger screen. But if it is at the household level, because most of the times we also go to the household level, maybe when we are doing... sometimes we may have very few people and we can be able to watch on a very small phone, but I don’t know if it can be OK teaching a lot of people using a small smartphone. “

Outside of the above, we did not observe or discover any other issues during this iteration. Outside of the issues outlined above, all the users were able to use the features tested on them without difficulty.

## 6.3 Action items for Iteration 2

Based on the gathered data during interviews and observations during iteration 1, we have identified the following action items:

### **1. Clicking the video title to open the video**

*Action to be taken:*

Make clicking video title/the whole tile open the video.

### **2. Issues finding video controls**

*Action to be taken:*

We could make changes so that the video controls are always showing. However, this would take up screen real-estate, and might not be the best idea. A better idea might be to show a text on the screen for a few seconds explaining how to open the controls.

### **3. Confusion with the main menu (hamburger menu)**

Some users are confused by how the main menu (hamburger menu) in the DHIS2 Android Capture app works. One would click open the menu, then immediately click the “Home” entry.

*Action to be taken:*

This is more of an issue with the choice to use a hamburger menu for the main menu in the DHIS2 Android Capture app, and outside of the scope of our implementation. However, we think this issue would be fairly easy to address by simple, short, training.

### **4. Confused about what is the video library VS the main view of the DHIS2 app**

2 test users were confused about what the video library was VS the screen showing different programs in the DHIS2 Android Capture app main screen. There was also confusion as to whether the video library was inside one of these programs.

*Action to be taken:*

We think this issue stems from our demonstrations, since the main-menu button is close to one of the programs, and they might not have seen that we clicked the main menu and not on a program. By more clearly showing where we clicked during our demonstrations, we think this issue will be addressed.

## 5. Trouble using the (Android) back-button.

We observed one user having trouble when told to “go back”.

### *Action to be taken:*

We will address this by adding a back arrow in the title field (app bar) in the video player. This is implemented in the capture programs in the DHIS2 Android capture app, of which figure 6.4 shows an example. Implementing this button will reduce the reliance on the device-specific back-button.

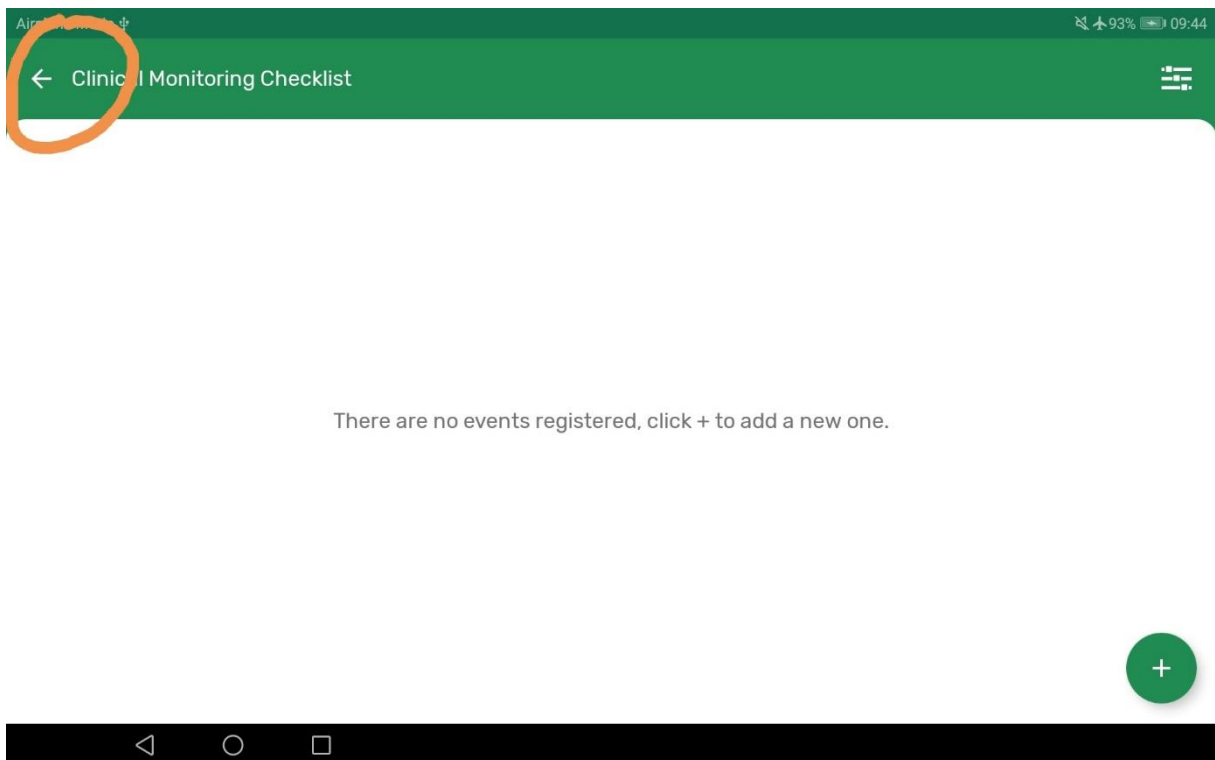


Figure 6.4: Back arrow in the app bar highlighted

## 6. Video library is missing thumbnails

The version of the video library used for user testing has placeholder thumbnails. (see figure 6.2).

### *Action to be taken:*

We will implement video thumbnails.

## 7. New users might require training

We assume that when the DHIS2 Android Capture app is rolled out for more extensive testing/use, training will be provided. Based on the current iteration, and planned features for the video library,

the training for the video library itself would be light compared to the training for the remaining features of the app.

#### **8. Screen size vs crowd size.**

*Action to be taken:*

To do what we can to minimize this issue (outside of recommending a tablet-sized device be used), we should maximize our usage of the available screen space. We also added questions about screen-size to our interviews for iteration two to further explore this potential issue.

**Additional action items based on features we have discovered to be required based on our requirements gathering:**

#### **9. Language select**

Based on the requirements gathering we have done we know that there needs to exist support for different language tracks in the videos.

#### **10. Search functionality**

When the video library is expanded and contains a larger number of videos, a search-functionality will be needed. We will implement this. This also requires the ability to add tags to the video, (so that searching for HIV will show a video about AIDS, for example).

# 7 Iteration 2

In this chapter, we describe the changes we made to our implementations based on the findings of iteration 1. This consisted of both improvements to (for example) navigation, but also some new features. After that, we present the data gathered in our second round of observations and interviews, and lastly, we finish by summarizing the learnings of this iteration and recommendations of further work. This chapter covers the action planning, as well as the evaluation and specifying learning phase in the second iteration. The diagnosing phase in this iteration was based on the action items (chapter 6.3) outlined in the specifying learning phase of the previous iteration. Like with chapter 6, the data presented and discussed here is related to the software, while data related to the use of videos is presented in chapter 4.2.

## 7.1 Software implementation

In this section, we will use the same numeration used in the Action items section of Iteration 1.

### 7.1.1 Clicking the video title to open the video

As described in the previous chapter, we observed several users having trouble opening a video from the video library. The videos were organized in a list with a thumbnail that when clicked on would take the user to the video player. From the observations in our first iteration, we noticed some of the CHW attempted to click on either the title or the row in the list where the video was located. In order to make the video library more intuitive, we decided to address this instinctive behavior the CHWs were expressing, by making the whole row shown in figure 7.1, meaning the title, thumbnail, and description area in the library clickable.

The changes done in our software to accomplish the clickable row were minimal. Originally a 'Listener' detecting clicks was attached to the thumbnail element in each row. To make the row itself clickable, we simply changed the element the 'Listener' was attached to from the thumbnail to the row-container element. That way, the CHWs can click on any element in the row, including thumbnail and video title, and still be taken to the corresponding video.

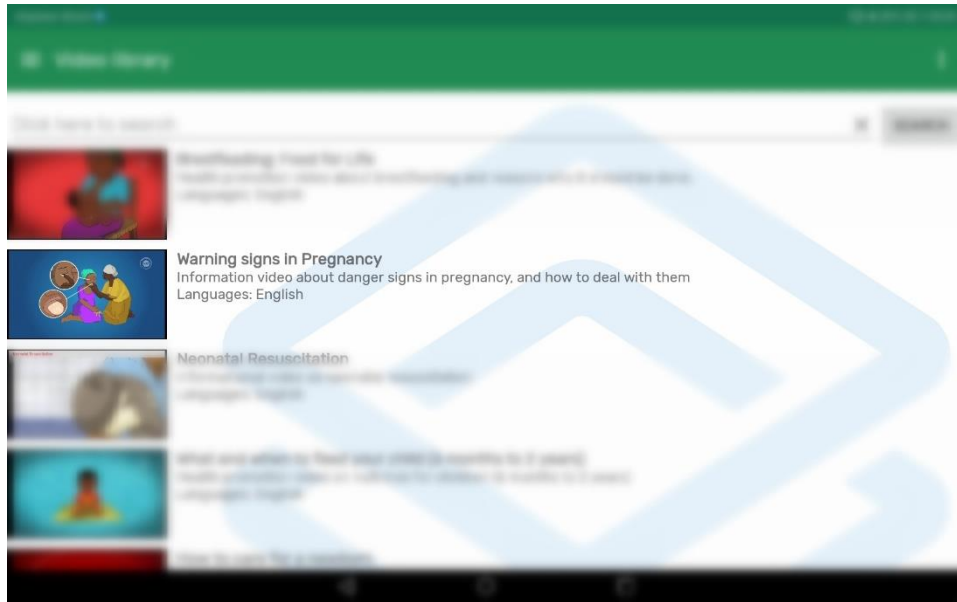


Figure 7.1: A row in the video library

### 7.1.2 Issues finding video controls

We also observed users having trouble finding the video controls. To combat this, we created a message that is displayed in the middle of the screen when a video initially starts playing, telling the user to 'Click the screen to open the video controls'. When the user taps the screen, the video controls are shown and stay showing until the user taps the screen again.

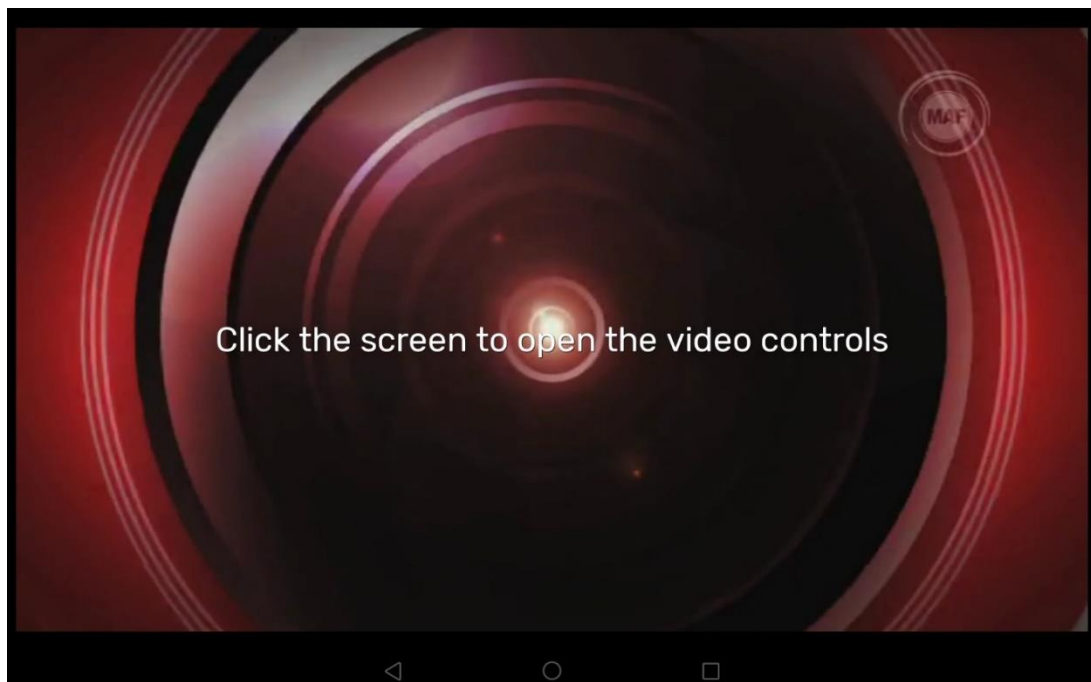


Figure 7.2: Screenshot showing the added instructions about displaying video controls

### 7.1.3 Confusion about main menu (hamburger menu)

Some users were having trouble using the main menu (hamburger menu) of the DHIS2 Android Capture app. As described in the previous chapter, this is an issue related to the core DHIS2 Android Capture app, and not our additions. However, to avoid this issue in future data collection, we will take extra care to slowly demonstrate the menu.

### 7.1.4 Confusion related to the video library VS the main view

Some users exhibited and expressed confusion about the relation between the DHIS2 Android Capture app's main screen (shown in figure 6.3). This issue is likely due to the quality of our demonstrations, and as such should be addressed by ensuring the test subjects have a clear view of the navigation steps taken to reach the video library, and to put more emphasis on the main screen being the main screen during our introduction.

### 7.1.5 Trouble using the (Android) back-button

Some CHWs had problems with navigating back to the video library from the video player. To achieve this, the CHW would need to press the back button on the Android navigation bar. The implementation of this button depends on the device used. On our testing device, the navigation bar is a digital element in the UI. However, on some devices, particularly older devices, this navigation bar consists of physical buttons on the device. Because of this, when using an unfamiliar device, the user might have trouble quickly locating the button. We, therefore, wanted to reduce the reliance on the device-specific back-button.

In order to make the navigation from the video player back to the library intuitive, we decided to add a back arrow on the screen. To better integrate our video library in the DHIS2 Android Capture App, we looked at existing solutions and standards within the application. One of the common solutions featured an arrow next to the title, located in the app bar of the screen (see figure 6.4).

To achieve this, we simply added an app bar to the video player layout file and added the image of a left-pointing arrow to this app bar. We then add a click listener to the back arrow, which triggers the 'Activity.finish()' function. Calling this function signals that whatever the 'Activity' it is being called on was doing, is now done. This destroys the video player 'Activity' and returns the user to the previously active 'Activity' (the 'Activity' that started the video player).

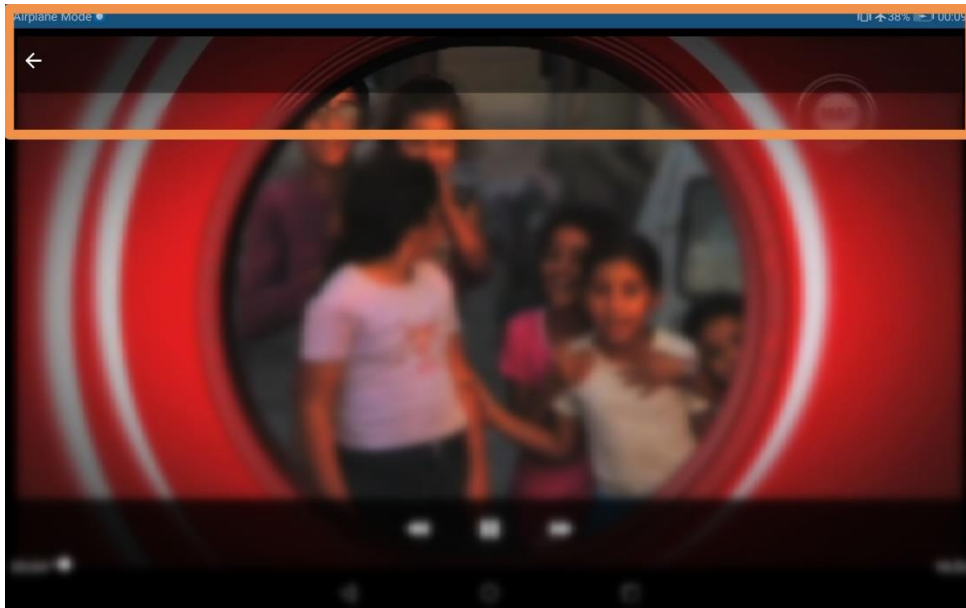


Figure 7.3: App bar with a back arrow in video player highlighted

### 7.1.6 Missing thumbnails

As pointed out by a user during our first iteration, the video library was missing (dynamic) thumbnails. Thumbnails have the potential to efficiently inform the user about a video's content, and as such was always a planned feature. In the Software chapter, we describe the technical solution of associating thumbnails with videos, and their propagation from the remote video repository to the device. Beyond this, we simply updated the 'Adapter' creating the list of videos in the video library to use the downloaded thumbnail file for each video, instead of the placeholder DHIS2 logo.

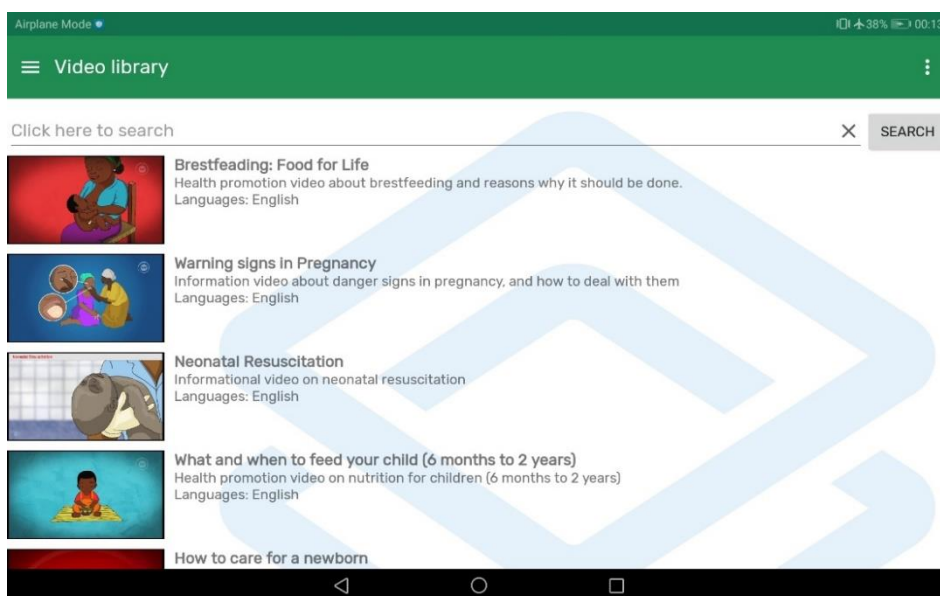


Figure 7.4: The video library with thumbnails



### 7.1.7 New users might need training

One of the users interviewed during iteration 1 pointed out that new users of the video library functionality might require training to effectively use the features. This is something we agree with and recommend any future use to include training.

### 7.1.8 Screen size vs crowd size

Another feature we implemented in this iteration was full-screen viewing in the video player. Based on data collected during interviews in the first iteration, we assume that the usability of the video library, in terms of which situations it can be used, depends on the screen size of the device displaying the video. This limitation, controlled by the devices the user organizations decide to buy, is beyond our ability to control, outside of recommending larger devices such as tablets. We can however lessen this constraint to the best of our ability by maximizing the screen real-estate we use for displaying videos. Because of this, it was important to implement full-screen video playback.

To achieve this, we made several changes. First, we changed the hard-coded aspect ratio of our 'VideoView' from 4:3 to 16:9, which is much closer to the aspect ratio of modern mobile and tablet devices in landscape mode. The 'VideoView' will then enforce a 16:9 aspect ratio within the area of screen real-estate it has available. We then maximize the screen real-estate available to the 'VideoView' by removing as many other UI elements from the screen as possible, thus increasing the space available to 'VideoView'. This is done by using 'setSystemUiVisibility' to set the 'View.SYSTEM\_UI\_FLAG\_LOW\_PROFILE'- and 'View.SYSTEM\_UI\_FLAG\_FULLSCREEN'-flags. The first flag requests that the status bar and/or navigation items dim. The second flag causes the same items to be hidden. Of note, both these flags have since been tagged for deprecation with the (at the time of writing) upcoming Android API level R ("Android 11"). In addition to this, we also hide the previously described app bar with our back button, by toggling its visibility.

The flags and app bar visibility are toggled when the user clicks the video, through an 'onClickListener' attached to the 'VideoView'. They are initially set to the "hidden" state when the 'Activity' starts. The flags are automatically un-set when the user interacts with the screen (such as by clicking it). We keep track of the full-screen status through a boolean and use it to toggle the visibility of the app bar.

The end result is that the status bar, action bar, and navigation bar are all hidden when the user opens a video and are all shown when the user clicks the window (which also toggles the video controls to

become visible). Clicking the screen another time will again hide the status bar, action bar, and navigation bar, as well as the video controls.

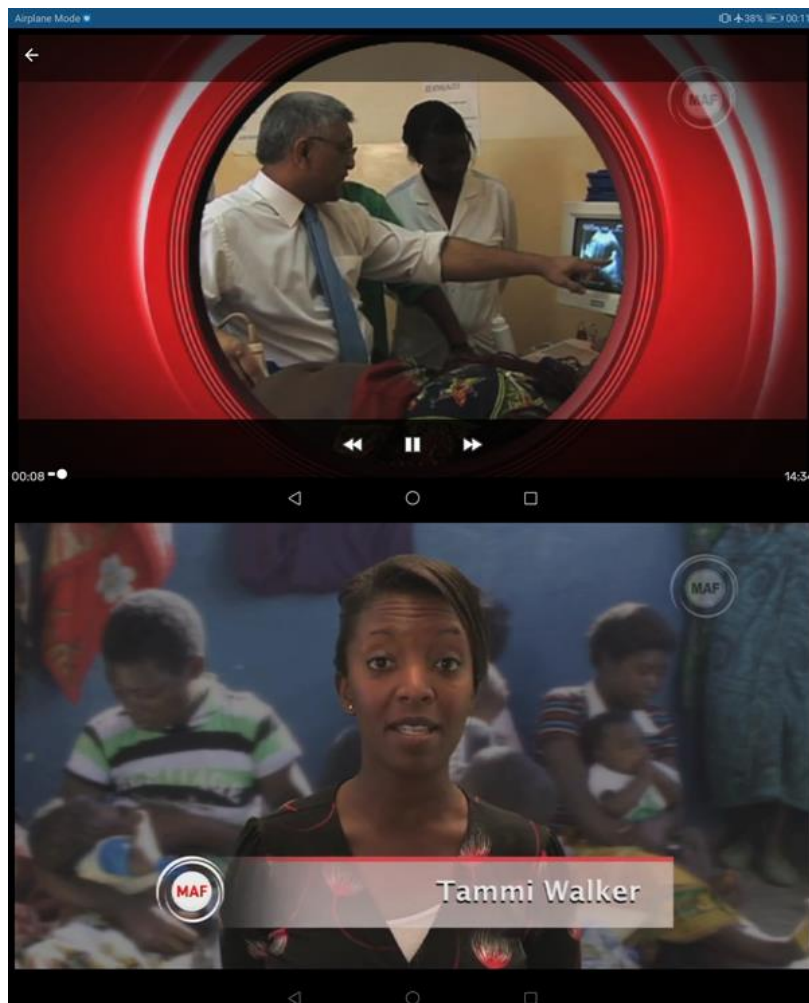


Figure 7.5: Showing hideable elements (top), hiding hideable elements (bottom)

### 7.1.9 Language select

Facilitating the option to choose from multiple language tracks on a video had always been a planned feature we wanted in our video library.

As mentioned earlier, the video files supplied by the remote video repository includes a distinct audio track for each language it supports. In order to implement the language selection, we created a box on top of the video player, in the video player 'Activity'. This box contains instructions to select a language, the available languages to select in a drop-down, and a "Select" button. The user will only be shown this interface if the video has more than one language track, and a language track has not already been selected. A language already being selected is something that occurs on a configuration change (as

discussed in chapter 6.1), such as when the orientation of the device changes. If the user goes back to the video library and then opens the same video again, a language is **not** “already selected”.

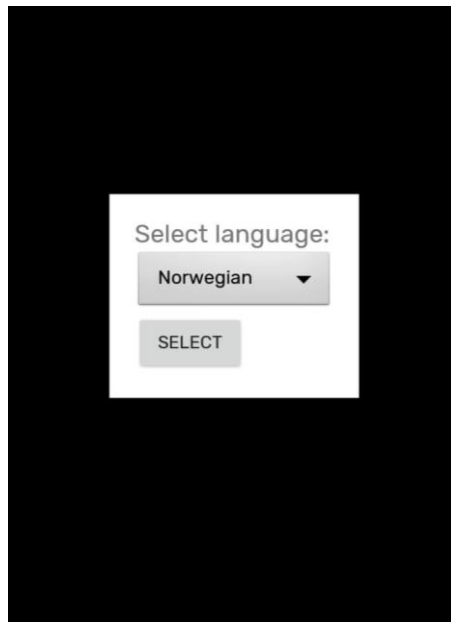


Figure 7.6: Implemented language selection dialog

When the user selects a language, we set the visibility of the language selection dialog to ‘View.GONE’, meaning it is no longer part of the view. The video then starts automatically playing.

The list of languages is transferred to the video player by adding the list of ‘VideoLanguageEntity’-objects to the ‘Intent’ sent to the ‘Activity’. ‘Intents’ can by default handle primitive data types, but not other objects. Because of this, serialization and deserialization are needed to transfer this list of objects. This is currently done by serializing and deserializing the list into a JSON-string, using ‘Gson’. This is done because we had already written the serialization code for these objects earlier and reused it here, despite this not being the best solution. A better solution would be to have ‘VideoLanguageEntity’ implement ‘java.io.Serializable’. ‘java.io.Serializable’ is however comparatively slow, so for a video with very many languages this might be slow. ‘android.os.Parcelable’ would be a faster solution, but it is not realistic that the list of languages for a video grows so large that the performance issues of ‘java.io.Serializable’ becomes an issue.

In order to make this audio selection work, we need to associate the index of the specific languages in the ‘VideoLanguageEntity’-array associated with a video with the index of the corresponding language track in the video file. As described in chapter 5.1.2, due to issues we encountered in the mapping of

language names to ISO 639-2 codes, this is not as simple as comparing the language name string of the 'VideoLanguageEntity' to the human-readable language name string mapped from the ISO 639-2 code. Instead, we have to rely on the order of languages in the 'VideoLanguageEntity'-array being the same as the order of the language tracks in the video file.

However, the video file also contains one (or more) video tracks, so mapping index 0 to index 0 would potentially map the language "English" to the actual video track, and index 1, the language "Chichewa" to index 1 of the video file, the English language track. To avoid this issue, we also filter through the video tracks and only include audio tracks. We then store this information in an object of a private class within the video player 'Activity', which contains the language name, its index in the 'VideoLanguageEntity'-array, and its index in the array of tracks within the video. The dropdown selection in the language selection dialog is then populated based on an array of objects of this private class, and the video track selection is made based on the option the user selects, again using the information stored in this private class.

All of this is done by an 'onPreparedListener' attached to the 'VideoView' since within this function is (one of the) only way(s) to extract the "lower-level" 'MediaPlayer' interface from the 'VideoPlayer' instance. We need access to the 'MediaPlayer' interface to extract the list of available tracks, using the 'MediaPlayer.getTrackInfo'-method.

In order for the language selection to "survive" orientation changes (configuration changes), we also persist the selected language through the previously mentioned 'onSaveInstanceState'- and 'onRestoreInstanceState'-methods.

#### 7.1.10 Search functionality

We decided to introduce and implement a search function in our video library during this iteration. Having a search function is fundamental when working with a large number of elements. In our case, the video library will ideally contain several videos for different topics and potentially be tailored for both health promotion and health training. For the users of this library, the CHWs, finding a specific video in such a large library would be challenging and consume valuable time. We implemented a search function to lessen these challenges and to provide a tool for the CHWs so that they could search and find a specific video or topic that fit their needs.

The videos in our video library are presented as entries in a long list. These entries contain a title and description that ideally should provide a good indication and explanation of the content of the corresponding video. Matching what the user is searching for with words in the title and description of a video would provide the basis for a general searching function. However, this assumes that the quality of the title and the description is good enough. Additionally, videos may cover several topics or be comprehensive to the point where having a small description will not be sufficient to explain what the video is about. Furthermore, the title and the description may not necessarily contain specific words that the video is actually about. For example, a video covering how to administer a special vaccine to prevent malaria may have the words “Malaria”, “Vaccine” and “Administer” in the title and description, making the video searchable by those words. However, if a CHW wants to search for every video that concerns immunization, the video will not be in the results, even though the video is about immunization.

We decided to address these problems by introducing keywords, or “tags”, that the videos can be tagged with. Every video will be associated with different tags. These tags are stored as an attribute of the video at the same line as description and title (as meta-data). The keywords that these tags represent are meant to be words that describe the video’s main content. They are also meant to describe relevant topics or themes the video presents that is not necessarily explicitly mentioned in the title or description. Since these keywords are provided manually through the attributes of a video, they should be based on a common set of relevant keywords. This is in order to establish a common vocabulary and avoid faults such as misspelling since these types of mistakes will have a negative impact on the search.

To implement the search functionality, we added a search bar to the top of the video library, which can be seen in figure 7.4. This bar contains a text input field, with the placeholder text “Click here to search”, a search button, and a button to clear the search. When the user clicks the search button, the search is applied. If the user has not entered anything in the search field, the search button does nothing. When the user clicks the clear search button, the filter is removed.

Instead of having the search filtering be done through the database query used to fetch data to populate the video library list, we instead made the ‘VideoListAdapter’ implement ‘android.widget.Filterable’. By doing this, we can use an ‘android.widget.Filter’ to filter the entries the ‘Adapter’ returns. This filter, ‘VideoSearchFilter’, filters videos by matching the “constraint”, the users search term, against the title, description, and tags of a video. If the search term is contained in the title, description, or tags of a video entry, the entry will be included in the filtered list. By using this

method, we avoid having to query the database for the data, instead just using the 'Adapter' we have already created to display the entries.

## 7.2 Data collection

Just as in the first iteration, we got feedback on our video library from CHWs through observations and interviews. Our demonstration of the library was the same as last time, though with more focus on our new features, and an increased focus on the aspects mentioned in points 3, 4, and 5 of the previous section. The interviews also included the same questions as last time, however, we added another question specifically asking the CHW of their opinion on our language selection feature (see Appendix 2). Additionally, we added a question concerning our new search function. In total, we got feedback from 16 CHWs. Five of these were CHWs we had interviewed in iteration 1, meaning that they could compare their experience from the old video library with the new one we had improved. We also asked the CHWs if they had a smartphone during this iteration. We were told by our local guide that smartphones were known as "WhatsApp-phones", indicating WhatsApp's wide-spread adoption in the area. 14 responded they did, one responded they did until recently, and one responded that they did not, and had never used one.

A question in our interview focused on the new search functionality and asked what the CHW thought of the functionality. Out of the 12 answers we got on this question, ten answered that it was easy to use. However, we observed that many of them struggled to remove the search term and get back to the original unfiltered list of videos. 8 of them tried to clear the search using the back button. One CHW said that the "x" symbol for removing the search term and thus returning to the full list of videos, was "not a good sign for coming back." He suggested that a left-facing 'back'-arrow, similar to the symbol we implemented for navigating from the video player back to the library, could be a better idea. We had observed that he, and many others, instinctively pressed the Android back button. We discussed the possibility of implementing the same function as the "x" symbol on that button and he agreed that it could be a good solution. This behavior was observed in eight other CHWs. One of these CHWs explained that it was difficult for her because people will automatically press the back button since they are used to the button behaving that way on their own phones. Another piece of feedback we got from two CHWs was that accessing the keyboard when searching was a bit confusing. One of the CHWs had trouble with getting the keyboard to appear because he clicked the search button without realizing that it is possible to click on the input field on the search bar to start typing. He agreed when we suggested that the keyboard could pop up when the search button is clicked while the input

field is empty. In total, 8 people tried to click the “Search”-button to initiate the searching process, instead of clicking the input field. All the others figured out by themselves, some after trying the “Search”-button once or twice, that they needed to click the input field.

Our solution for choosing the video language and the CHWs’ opinions on this feature was something we were interested in, which is why we included a question about this in our interviews (See Appendix 2). In one of the CHW’s opinion the default value in the dropdown menu of the current solution (see figure 7.7) should not be a specific language. He explained that it was confusing to already have a language selected (since the dropdown selection has the first option selected by default), which another CHW echoed. He suggested that the default value should help give the user instruction, for example by having the value “Select language” be the default selection in the dropdown.

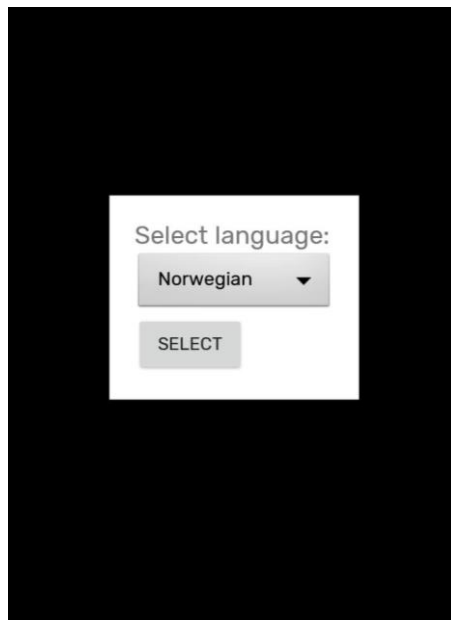


Figure 7.7: Current language selection dialog with a dropdown menu

We also had two additional alternatives to our current solution to “Language selection” that we wanted feedback on, to know which one was preferred. The CHWs were presented with an illustration and an explanation of how the different selection would work. One alternative featured a list of the language track available with a radio button to each language (see figure 7.8). The language that is selected is indicated by the circle that is filled in, and the user needs to press the “select” button to finalize their selection. The other alternative also featured a list of language tracks, but instead of a radio button, each language had a corresponding button marked with “select” (see figure 7.9). This alternative also had a variant where the button is marked with the corresponding language and not “select” (and the language name is not printed in front of the button, see figure 7.10). In total, we got

answers from 12 CHWs on this question. Three of them preferred the radio buttons and thought it was the easiest of the three alternatives. The other nine CHWs preferred the button list. This alternative was in their opinion a much more direct approach to selecting the video language, which made it easier to use and understand. Out of the nine who preferred the button list, 7 answered that they would rather have the variant where the buttons are marked with the corresponding language. In addition, we also had two other CHWs suggesting improvements to the language selection in another question, which was similar to this alternative.



Figure 7.8: Alternative language selection dialog with radio buttons

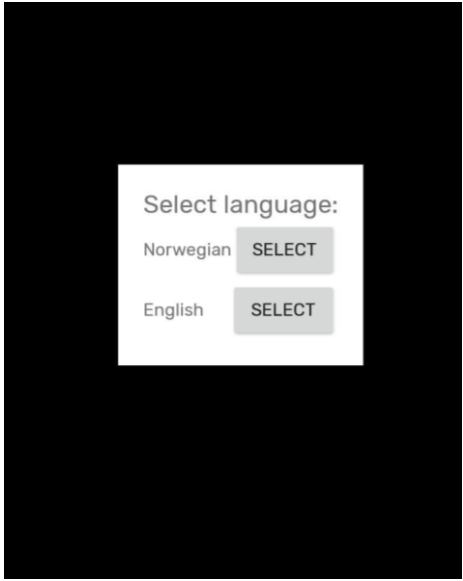


Figure 7.9: Alternative language selection dialog with a list of buttons



One of the last questions we included in our interview guide was how large of a screen the CHWs would need for showing videos to clients. We wanted to know their view on the practical use of videos concerning the actual device they would use and provided two example screen sizes, that of a smartphone, and a tablet. The general opinion of most of CHWs was that a bigger screen would be better. This was especially important when counseling medium- and large sized-groups. How large of a group the screen size examples could support, varied. One CHW said that for one or two clients, the small screen of a phone would be sufficient, but a tablet would be needed for a group of three or more. Another CHW expressed that a tablet would be too small in size to show videos on to a group and that phones were too small even when consulting one client. However, most of the CHWs agreed that the size of a phone screen would only be enough when handling one client.

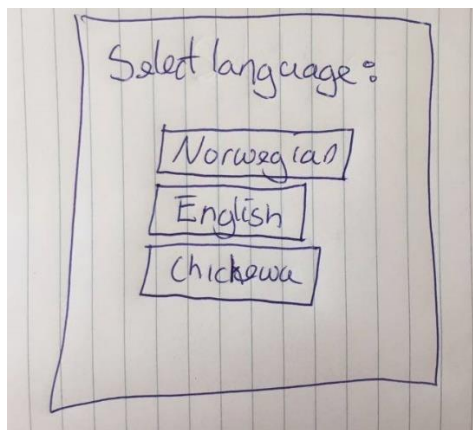


Figure 7.10: Sketch of variation of the third option

Like in the previous iteration, we asked the CHWs if they had any improvement for our video library. Several CHWs misunderstood the question and pointed out improvements regarding the test videos and their content, instead of improvements of the library and player functionality like we intended. However, two of the CHWs we interviewed expressed that one of the improvements that could make our video library better is having the option to view the list of videos by categories. One of them gave an example where he only wanted to see the videos concerning children who are undernourished or videos of children who are HIV-positive, instead of being presented the full list of videos and needing to search through that. We discussed with him the idea of having buttons with different main health topics, thus essentially structuring the library by categorizing the videos. He seemed quite positive about that idea.

In interviews after this, we started asking if the CHWs thought organizing the videos in categories in the library would be a good idea. The solution we proposed was that the video library screen would

initially contain buttons with category names on them and clicking on the button would take you to a list of videos in that category. Figure 7.11 shows a sketch that was made in the field of this and used to explain this concept during the interviews.

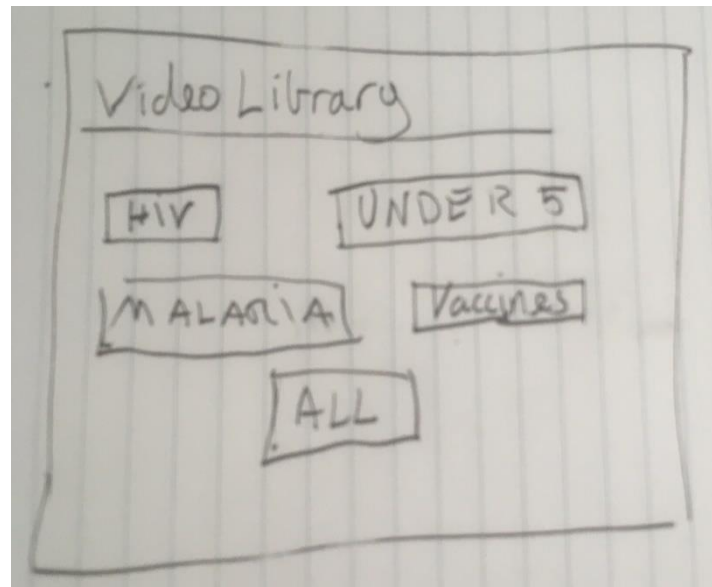


Figure 7.11: Sketch of video categorization

Since we did not immediately add this question to our interview guide, we, unfortunately, forgot to ask this question of a few CHWs. The CHWs we did ask were very positive to this categorization, however. One CHW suggested that the categories should match with the health programs that the CHWs are used to.

When it came to accessing and using the video controls, 2 of the CHWs found the video controls themselves difficult to use (not as in finding them, but the icons themselves). 3 found the video controls difficult to find, 2 of which were among our repeat interviewees. This seems to indicate that those who were introduced to the “new” system (the help-text displaying on top of the video) without any previous experience, had a lower ratio of having issues with the video controls. Of course, these are rather small sets of observations, so these “statistical” findings should be taken with several grains of salt.

We observed 5 users having trouble with the main menu (hamburger menu). 3 of these users were among our repeat interviewees, and we only observed 2 of them having trouble with the main menu during iteration 1. Since the ratio of users having trouble with the hamburger menu increased from iteration 1 to iteration 2, and 2 of them being repeat “offenders”, it seems that our “improved”

explanation of the main menu was not at all improved. Again, it should be mentioned that this is a small set of observations.

The one user we interviewed who had never used a smartphone had trouble using the video controls, had trouble with the icons (play/pause, fast-forward, etc.) in the video controls, and initially had trouble scrolling in the video library. The scrolling, however, he figured out by himself without our intervention.

Outside of this, we observed that the users largely were able to work the features we had implemented as intended. This time, we recorded no users having difficulty navigating from the video player back to the video library, seeming to indicate our addition of a back arrow worked. We specifically observed several users use the back arrow with (little to no) hesitation.

### 7.3 Challenges and action items

The videos in our video library are at this point presented as entries in a long list. They are not structured or divided in subsections by topics. Nor are they sorted in a logical way, for example alphabetically. The library should be reorganized to better present them. This is especially important when the library contains several more videos than our current library, as handling and navigating these videos in a disorganized list would be challenging. As some of the feedback we got from the CHWs indicates, the videos should be divided into categories. The exact division of these categories could be done in several ways, however, we believe that the CHWs would benefit from having categories based on the health programs they are part of. These categories could (and should) be implemented in a generic fashion so that they can be created and managed in the remote video repository. This categorization could also help alleviate or at least reduce the need to use the search functionality, which many users had trouble with.

Our (frontend) solution for the search, especially the way of clearing a search, did not mesh with the expectations of the CHWs. Several users tried to initiate the searching process by clicking the “Search” button. Therefore, changes should be made so the search field gains focus and the keyboard opens if the user clicks the “Search” button while the search field is empty.

Another, but smaller change that should be considered is improving the navigation from a filtered list of videos when searching, to the original unfiltered list (clearing the search). Given that 8 of 16 pressed

the back button on the device when attempting this navigation, the functionality that the “x” button in the search bar provides (clearing the search) should also be triggered when the back button (both the back button on the Android navigation bar and the back button arrow in the app bar) is clicked. In addition to this, it might be an idea to replace the “x”-button with the text “Clear” (or something similar).

We still observed some users having trouble with the showing and hiding of the video controls, indicating this interaction needs more work. We also observed some users having trouble with the hamburger menu (DHIS2 Android Capture app main menu), despite our increased efforts to better explain this menu interaction. While this is not directly related to our work, it indicates that this menu element might require more training than one assumes.

## 8 Discussion

In this chapter, we will discuss our findings from the chapters iteration 1, iteration 2, and Video as a Tool. This chapter is divided into seven topics where the relevant findings are discussed.

### 8.1 Compared to similar application

The DHIS2 Android Capture app is not the only mobile platform that is used for data collection within the health sector of developing countries. One of the rival Android applications, that we even encountered among the CHWs in Malawi, is CommCare. The application had been used in a project that some of the CHWs we spoke to, were a part of. This project was brought up when we asked them about any previous experiences they had with videos in health promotional situations, indicating that the project had in some degree involved videos. Our understanding of the video implementation in Commcare is based on the descriptions and demonstrations we received from the CHWs we spoke to (see chapter 4). As described in chapter 4.2.1, the application guides the CHW through a series of questions, some of which they ask their clients. The application processes the information and if certain conditions are met, for example if the client lacked specific immunizations, the application would recommend giving a consultation of the relevant topic and present a video for the CHW to show the client.

CommCare in other words has a different approach to presenting videos compared to our implementation in the DHIS2 Android Capture app. However, both applications have in common that they are meant as an aid for the CHWs in their daily tasks, mainly in delivering educational health messages. Though videos are featured in both applications, how the CHWs navigate to a video is quite different. Based on what the CHWs showed us, the videos in CommCare are only accessible after going through a survey. Being available only in certain conditions makes the video less easy to access. This factor also contributes to the CHW being in less control of what video the application recommends. Given the information we got from our translator (see chapter 4.2.1) this can sometimes be beneficial. If the CHW has trouble providing the clients the correct type of service in accordance with their needs, being recommended a video from the application would help ensure that the CHW gives the client what they need. This is also the case in situations where CHWs may forget to talk to the client about certain topics. However, less control over choosing videos and what health messages to give could also be problematic. As written in chapter 4, one CHW said being guided by questions did not always work out. A mistake in the entering of information could get them lost and make it difficult to be directed

to the points they wanted to emphasize. This contrasts with our solution where videos are accessed through a video library. In our solution, the CHWs would have no need to go through any surveys to play a certain video. They can choose from all the available videos and decide which one fits the situation best. This requires that the CHWs have assessed the situation themselves in order to choose the correct video since the video library lacks the analytical part that the CommCare solution has. However, based on the feedback from the CHWs who have used CommCare, the CHWs are not opposed to a more direct approach would, which would for some be easier and more practical.

When looking closer at the differences between the CommCare solution and our video library, we discovered that the videos used in CommCare are not videos. Based on our observation of the videos in the CommCare application in chapter 4, the “videos” we saw consist of a static illustration combined with a soundtrack. Though it is certainly a step towards combining visual and auditory senses, as videos do, the visual sense is only stimulated by a static illustration. One of the main advantages the CHWs brought up (see chapter 4) when talking about the use of videos, was that videos are not static and show movement. Taking that into consideration, having a “video” with only a static picture would be less advantageous for the CHWs and their clients, compared to using a video with movement.

Despite our endeavors to compare how the applications handle videos, it can be argued that the CommCare project has a different aim than our video library. While videos and their use are the focus in our implementation, the CommCare project seems to focus more on helping the CHWs analyze the situation. By having a survey that the CHWs must fill in, certain issues may be brought up that the CHW could have difficulty discovering. Based on the explanations from our translator, CommCare seems more like an analytical tool for aiding the CHWs when assessing situations, rather than a tool for accessing videos related to health training or health promotion. Therefore, it is important to keep in mind that the CommCare project should not be judged based on the ability to present videos, as this seems to not be the main focus of the project. Likewise, our video library is made with videos in focus. Therefore, our solution is arguably more suited for handling videos compared to CommCare’s solution.

In short, our video library is not the only application for CHWs that involves videos. CommCare has a solution for videos, even though it can be argued that it is mostly audio clips rather than videos. The applications have different approaches to navigation, where our solution is more direct than CommCare’s. This results in both advantages and disadvantages to the CHWs. Despite this we believe our video library is better suited when considering videos. On the other hand, CommCare provides assistance through analysis and recommendations, which is beyond what our library offers. However,

in the end, both CommCare and our additions to the DHIS2 Android Capture app aims to achieve the same goal: help the CHWs deliver health educational messages.

## 8.2 Illiteracy

Videos have many benefits, especially in an educational setting. As mentioned in chapter two; combining both visual and auditory stimulation will result in overall improved learning and memorization. This was also pointed out by CHWs and is reflected in their opinion on the use of videos both in health training situations and in health promotional situations.

Another benefit of videos, especially in Malawi and other developing countries, is the potential positive effect it can have for those who have trouble reading. As expressed by the CHWs there are those in Malawi who find reading difficult, and thus find it challenging to learn and understand the health messages provided in written materials such as posters, flyers, and pamphlets. It is also not unthinkable that some of the CHWs themselves might have a lower reading proficiency. Videos would be helpful as they, given the right content, will not require reading, but still provide the same message and information.

A study by Ditai et al. (2018) among women with little or no literacy in Uganda evaluated whether a video or illustrative slideshow improved the participants' understanding of information, compared to the traditional method of reading the information out loud. The slideshow included the use of flipcharts with bullet points and pictorial illustrations, while the video showed the information being read. There was no significant difference regarding understanding the information between these three methods. However, most of the women preferred the slideshow compared with both video and standard methods. One of the main reasons was the use of visual illustrations. In comparison to Ditai et al.'s study (2018), the CHWs and their community in our research already utilize flipcharts. Though our video library was not tested with people that have little or no literacy, the CHWs believed videos would have a positive effect on those types of clients. If the clients in Malawi prefer slideshows over videos, as in the result in Ditai et al. (2018), remains to be seen. However, it is important to take note of how the content of the video where a person only reads the information aloud, could make the flipcharts preferable. Therefore, it would be wise to focus on the visual aspects of the videos to facilitate those who have little or no literacy.

### 8.3 Training and refreshers

Most of the CHWs in our study have personal experience with the use of videos in an educational context, more specifically in the health-related training that is required to become a CHW. The CHWs with this experience all had positive associations with it, mentioning mainly the educational benefits. Thus, it is apparent that videos, as they are already included in the training CHWs receive, are working well for the CHWs as an aid in their health training. However, despite the fact that the majority of CHWs have used videos in their initial training with great success, the CHWs usually resort to consulting written materials, such as health manuals and their notes from their previous training, or their colleagues when they are uncertain on the procedure and needs a refresher. Why is that the case, when videos are in their opinion beneficial? The main reason is of course their current lack of solutions and resources to enable them to use videos for this purpose. However, another reason may be the inconvenience of videos in certain situations. An example of such a situation could be when the CHWs are out in the field and when they are in the middle of consulting clients. Taking out a phone/tablet and watching videos on, for example, how to administer a certain vaccine while taking care of a planned vaccination, uses valuable time for both the CHW and their clients. Additionally, it is not something that the clients would take pleasure in and may have consequences for the relationship between the CHW and the client. So, while videos should be used in the CHWs initial health training and as refreshers, it should not be used as the first choice of tool when a CHW is in the middle of tending to their clients. Videos should rather be used as a means in the preparation for these situations.

As mentioned in chapter 2, Holloway, Lee, and Mcconkey (1999) conducted a study of video-based training courses in Africa. Though we did not test how videos would impact the CHWs and the community, we found the impact the CHWs expected to be largely similar to the results in the study by Holloway et al. (1999). The results of their study showed that the video courses created more enthusiasm and increased motivation from the participants. Motivation and enthusiasm are similar to one of the benefits a CHW we spoke to (see chapter 4) brought up: that videos are interesting to watch. This increased interest could arguably lead to more enthusiasm and motivation further down the line.

As we learned from Hamilton (2005), properly made videos were adequate for re-training nurses in CPR (cardiopulmonary resuscitation), even potentially performing better than in-classroom training. They also stress that video-based training cannot completely replace in-classroom or practical training. This largely coincides with what the CHWs told us.



The CHWs did stress that the videos should not completely replace training, however. As we learned, and also read in Chikaphupha et al. (2016), training is “[...] a vehicle for career progression, distribution of materials, mentoring, capacity building, and income from allowances (Chikaphupha et al., 2016). We also had one CHW express fear that the videos would lead to them receiving less traditional training, and thus decrease their income.

## 8.4 Health promotion

Health promotion is an important task for the CHWs and a task where videos would be a powerful tool. Unlike with health training, very few CHWs have any experience with using videos in health promotion. The reason is most likely the same as with health training: lack of solutions and resources for using videos for this purpose. Nevertheless, the CHWs see the benefits of videos in health promotion situations and are very positive to the use of videos in their health promotional work.

There are different reasons why videos fit so well to this kind of task. Like with health training, the previously mentioned educational benefits of videos also applies here, as it is essentially health training for the community. Some of the educational benefits, especially the effect videos can have for those who have difficulty reading, are arguably more important in videos in health promotion than health training. Promotional videos are meant to reach a wider audience than the training videos and have to appeal to a wide variety of people. Having videos with visuals and sound creates a medium that can relay the health message in a context that is familiar to clients who are watching. That would most likely be more relatable to the clients and therefore more appealing.

Videos are also often viewed as a form of entertainment. Being seen as a sophisticated thing in the villages, videos for health promotion would attract a lot of attention according to one of the CHWs. Therefore, using videos will attract a lot of people and could appeal to clients who are traditionally difficult to reach, for example men. Furthermore, all of these outcomes will possibly lead to the health message having a greater impact on the client, thus making the health promotional work the CHWs do more successful. These findings are corroborated in a study by Coetzee et al. (2018) which looked at the views of CHWs on using videos on tablets for health promotion work. In the study, they gave a group of CHWs in South Africa tablets with health promotion videos for use in home-visits and interviewed them about their experiences using the tablets and videos. In the study, the CHWs reported that their work was seen as more important by their clients when they used the videos, which in turn lead to an increase in confidence and morale amongst the CHWs.

In Coetzee et al. (2018) it is also reported, just like we were told by the CHWs we interviewed, that their clients found the videos themselves very interesting, leading to increased interest and learning. It also mentions that the CHWs experienced that the health message they had delivered with the help of the videos would be spread by the clients themselves, to other community members. Kay (2012) found that video-based content often leads to improved memory recall.

One topic of health promotion that was frequently brought up as fitting for videos by the CHWs was vaccination/immunization. Several of the CHWs expressed that there is some resistance to vaccinations in their communities, but that they believed that videos showing why vaccines were so important (for example showing people afflicted with preventable diseases) would be very helpful in overcoming this.

The current solution for assistive devices in health promotion is through the use of flipcharts, which in addition to providing relevant illustration, encourages communication between the CHW and the client(s) through explanation, retelling, and interactive questions throughout the consultation. This may be more difficult to achieve with videos. Depending on the content of the video, there may be some sort of narrating voice explaining what is going on in the video as it plays. It may explain and answer common questions, but it can't have a two-way conversation with the audience as the CHWs do. It may, therefore, be difficult to know if the client has actually understood the message or if the video has left the client confused.

Several of the CHWs we interviewed mentioned multiple drawbacks with the flipcharts, such as them becoming damaged, having to carry many around, having to memorize them, etc. In addition to this, in Coetzee et al. (2018) the CHWs describe their clients progressively becoming less interested and invested in health promotion messaging the CHWs themselves provided verbally, and that the use of videos would reinvigorate the client's interest and allow them to capture their attention again. It is also mentioned that the videos increased the participation of others in the household, such as the men/husbands. This is further supported by Kay (2012), which found that the audience will often have a positive attitude towards video-based learning. This aligns well with information we gathered from our interviews about men being harder to capture for health promotion messaging. As we came to understand it, receiving the health promotional messaging was largely seen as the woman's job in the household.

It is clear that using videos in health promotion have both advantages and disadvantages. However, the positive effects videos would have outweighed the negative. Videos should not be thought of as something the CHW should just play and then not interact with the client. It is mentioned in Coetzee et al. (2018) that some of the CHWs were of the notion that the videos would “take over” on days when they were too tired to verbally deliver the messages and interact with the clients. As pointed out in the study this is something that should be avoided, as the question-asking and conversation are just as important to ensure the viewer is engaged and consuming the information given. Additionally, there are measures that can be taken to lessen the negative effect videos can have. To ensure that the clients receive the correct message from a video, the CHWs can make sure to ask questions at the end of the video and encourage a discussion about the content of the video. The video can also be paused at certain points when necessary to check if the clients understand what they are shown. These measures would ensure that communication between CHWs and their clients are still kept. In short, videos would be a very good aid for the CHWs in health promotion.

In summary, our findings with regards to using videos delivered on mobile devices (phones/tablets) is in line with the findings of Coetzee et al. (2018). The study also reports that the CHWs were largely positive and excited about using a tablet and health promotion videos, which matches the feedback we got from our interviews.

Van Mele (2010 and 2011) shows the effectiveness of using videos to promote and spread information to rural farmers in Bangladesh. While this study focused on agriculture, we believe these learnings are transferrable to health promotion. Bello-Bravo and Baoua (2012) also highlight video as an important tool for spreading knowledge in areas with limited access to information.

This previous research lines up with what we found the CHWs beliefs were in terms of videos being a powerful tool in conveying information effectively and capturing the patient’s attention.

In a study on using mobile-based audio for health promotion in Malawi (Pimmer & Mbvundula 2018), which seems to be about the CommCare project described in 8.1 (or a project very similar to it), three main benefits were found.

Firstly, the audio clips helped legitimize to the clients/patients the CHW’s use of a smartphone as a tool in their duties. It explains how the more traditional uses of the mHealth application, such as data collection, would often create a barrier between the CHWs and their clients since the CHW would

spend more time looking at their phone. Using the device to play the audio messages helped remove these misunderstandings, by highlighting the use of the phone as a professional tool of the CHWs.

Second, the audio clips helped the CHWs deliver the health promotion messages. They quote one of the CHWs as saying *“You have somebody [!] to support you in case you have forgotten any point, to remind you”* (Pimmer & Mbvundula 2018), meaning they saw the audio clips as a colleague they brought with them. The CHWs would leverage the audio clips by playing them and then asking questions about their contents, and potentially expand upon the information if needed.

Third, the audio clips helped persuade the community. Some of the CHWs had told them that persuading their communities to make changes could be difficult, but that the audio clips helped. To explain this, they quote one of the CHWs as saying *“[...] so the community feels it [the phone] is like a doctor”* (Pimmer & Mbvundula 2018).

These findings match many of the beliefs of the CHWs we spoke to and we believe these benefits will also be transferrable to video-based health promotion messages. One important factor we had not considered, was that the use of audio/video-based health promotion messages might help explain and legitimize the CHWs use of the smartphone/tablet as a tool of their job, to their clients.

## 8.5 Language

The content of a video can be expressed in different languages. Though English is commonly seen as a universal language and often the choice when attempting to reach a wide audience, we found that it would not be beneficial to have all the videos in English in the context of Malawi. To reach out to the whole community, the CHWs may need to deal with several languages, as their communication with the clients is usually done in the local language. In chapter 4, we found that all of the CHWs thought it was important to have different languages in videos, and not only English. The main reason was that the local language was naturally easier to understand than English in the communities. It would make videos more accessible for the community and make communication easier, thus having a positive effect on the knowledge the community gains. As one of the CHWs expressed it: *“[...] So that they can have a chance of learning, we have to use the local language”*. Facilitating different languages in our video library was therefore essential, and we solved it by implementing functionality allowing the selection of which language track from the video to play. This belief the CHWs presented is supported in Bello-Bravo and Baoua (2012), which found that the chance of success of video-based learning aids

was greatly increased by producing videos in the local language(s). The study by Purchase and Worrill (2002) has “Speak user’s language” as a highly ranked principle for the usability of an on-line help system. While we did not perform any ranking of relevant features or principles, the option to have different languages was clearly important for all of the CHWs. Thus, to have this option is, both in Purchase and Worril’s study (2002), and our own research, seen as a highly valued feature.

As described in Iteration 2 we implemented a solution for selecting different language tracks in our video library. In addition to our initial language selection interface, we presented three alternatives to the CHWs, as described in chapter 7.2. The three alternatives have different advantages and we found out that there is improvement to be made by choosing one of these other alternatives over our initial design.

The first alternative (see figure 7.7) is the language selection implemented with a dropdown menu. In contrast to the other two alternatives, this solution does not immediately present all the available languages in a list, whether it be a list of radio buttons or normal buttons. The different languages are available in a list that pops up when the user clicks on the dropdown selection element. When the language is selected, the user needs to click on a “Select” button to be taken to the video with the correct language track. Compared to the other alternatives, this means that this solution takes less space on the screen, at least at the start, when the user is first prompted to select language. This does not only make the view of the language selection less messy but can also be an advantage when a video has several language tracks that need to be listed. The full list is hidden and managed in the dropdown menu, rather in the language selection itself. However, based on the feedback from the CHWs in chapter Iteration 2, the CHWs found it a bit confusing when using this menu to select a language. Especially when the default value shown is one of the available languages, rather than an informative message, for example, “Select a language”.

The second alternative (see figure 7.8) is the language selection implemented with a list of radio buttons. Like with the first solution, the user needs to first click/select the language they want, before clicking a “Select” button to be taken to the video with the selected language track. However, this solution is a bit more direct than the dropdown menu, as the languages are presented in a visible, expanded list as soon as the user is prompted to choose a language track. Given the feedback we got from CHWs in chapter 7.2 the CHWs prefer the more direct alternatives for language selection. This alternative is arguably more direct than the previous. This solution also requires less interaction. The user only needs to click once to select the language, and then a second time on the “Select” button to

be taken to the video. The dropdown menu needs to be interacted with an additional time to access the list of languages.

Likewise, with the second alternative, the third alternative (see figure 7.9) is also a list, but with buttons rather than radio buttons. As with the second alternative, this means that the languages are directly accessible when the user is prompted to select the language track. However, because each language has its own “Select” button, it is even more direct than the other two alternatives. A user only needs to look at the list and find the language they want, before clicking the associated “Select” button. This potentially means that the user only needs to click one time in the language selection to be taken to the video with the correct language track. As the results in chapter 7.2 suggests we discovered that most of the CHWs preferred this alternative. They agree that this makes the language selection more direct and thus more understandable.

When we presented these three alternatives to the CHWs, we found that most of them preferred the third alternative. The list solution with “Select” buttons requires fewer clicks than with the use of radio buttons and doesn’t hide alternatives in another list that must be opened, which is the case for the dropdown menu. This makes the solution direct and more intuitive, which the CHWs finds easier to use and understand. Several of the CHWs also preferred an even more direct approach: having the language as the text on each button, rather than “Select” (see figure 7.10). Though it does not reduce the number of clicks needed to choose a language, it can be easier to understand as it is not necessary to connect the language and the corresponding “Select”.

General internationalization in the app (e.g. the heading “Video Library”) can be done as normal in Android, which the DHIS2 Android Capture app already uses. However, our solution does not support different languages in the video meta-data. This means, that while the video contains audio tracks for several languages, the video title, description, and tags only exist in one language. This is problematic beyond just that the user might not understand the language chosen for this information, but in that the user might attempt to search for the video using words in one of the languages the video has an audio-track for, but which is not the language the meta-data is in. To improve this, the video library (and remote video repository) probably would need to contain the meta-data in several languages. Which of these languages are then displayed could then be made dependent upon the language settings the user has in the app, with a pre-selected language as a default.

## 8.6 Implementation

As mentioned earlier, we split our software implementation work into two parts, the “backend” components which surround and support the “frontend” which faces the user and the user directly interacts with. Most of our implementation work was done in the latter category, while our work with the “backend” components was largely limited to a basic theoretical design and an as bare-bones implementation as possible, when required at all.

### 8.6.1 Frontend

Our frontend implementation work was largely limited by the time we in Malawi, thus limiting our ability to lean on the “heavier” tools and processes of frameworks such as Participatory Design, which often rely on repeated interaction with the user group, while also creating a working, high-fidelity prototype. Our goal was to create an easy-to-use interface that fulfilled the requirements we had discovered during our requirements gathering phase.

To enable us to quickly construct a usable, easy-to-learn interface, we decided to lean on already familiar user interfaces and their associated mental models. As Don Norman writes, *“With standardization, once you have learned to drive one car, you feel justifiably confident that you can drive any car, anyplace in the world. Standardization provides major breakthroughs in usability”*. (Norman, 2013) This is based on consistency increasing the ability for the user to transfer skills from one system to another, thus making the new system easier to learn. (Nielsen, 1993) Consistency has the potential to drastically reduce this required time to learn new systems. (Polson, 1988).

This also extends to the standardization of specific user interface elements, as shown in Nielsen (2004), which shows that the elements of large parts of modern websites are (more or less) standardized.

Not only can adhering to established interface design patterns have positive results on the usability and learnability of a new interface, but **not** following them can adversely impact these same goals, when the user incorrectly tries to apply rules learned in previously used interfaces to the new interface (Satzinger & Olfman, 1998), highlighting the importance of adhering to the user’s already existing mental model.

We, therefore, decided to largely base our user interface design on existing, established designs from similar applications. In order to establish these pre-existing, established designs, we explored several platforms. To collect a sample of existing video hosting services, we consulted the Wikipedia entry “List

of video hosting services”, which ranks the video hosting websites based on their Alexa traffic rank from November 2<sup>nd</sup>, 2019. (List of video hosting services, n.d).

Table 8.1 shows the 8 top video hosting websites by Alexa traffic rank, according to the Wikipedia list. (List of video hosting services, n.d.) We excluded twitch.com as it is mainly focused on watching live videos, as well as websites dedicated to pornographic content, despite many of them having higher Alexa traffic ranks than the websites we used. We also decided to look at the search result pages of the websites, instead of the front-pages, since many of them had ads, featured/highlighted content, and the like. All of the pictures were taken on a tablet in the landscape orientation. We also include a screenshot from the YouTube Android app and a screenshot of Google video search results.

| Name            | Language     | Alexa traffic rank | Included in comparison |
|-----------------|--------------|--------------------|------------------------|
| YouTube.com     | Multilingual | 2                  | Yes                    |
| Twitch.com      | Multilingual | 41                 | No                     |
| Aparat.com      | Persian      | 102                | Yes                    |
| Bilibill.com    | Mandarin     | 110                | Yes                    |
| Vimeo.com       | Multilingual | 170                | Yes                    |
| Niconico.com    | Japanese     | 178                | Yes                    |
| IQiyi.com       | Mandarin     | 200                | Yes                    |
| Dailymotion.com | Multilingual | 201                | Yes                    |

Table 8.1: Top 8 most visited video hosting services (List of video hosting services, n.d.)

We found that all the services we visited followed the same basic design (figures 8.1 to 8.5). They all were laid out in rows, with one video entry per row. They all had a thumbnail, onto which the video runtime was stamped in the bottom corner. With the exception of aparat.com they all had the video title, in larger/bold text, to the right of the thumbnail, with additional information, such as the channel that uploaded the video, and stats (e.g. how many views the video has), placed in smaller text under the title. Aparat.com places this section to the left of the video thumbnail, and we believe this is a result of the website’s language being Farsi, which is read and written right-to-left. Several of the extra pieces of information displayed is not relevant to us, such as the channel that uploaded a video, how many views the video has, etc. They all also had a search bar at the top of the screen, and 5 of them used an “x”-icon in the search bar as a button to clear the current search query.

Based on this, we conclude that there is more or less a defined, widespread standard for how interfaces displaying an (online) list of videos look and work. This also differs from how mobile devices list videos and photos that are stored locally on the device, for example in a “gallery app”, as they differ intensely in mission. While the galleries are focused on showing you content (photos and videos) that you are



already familiar with, the listing-interfaces we have focused on is concerned with showing the user content they are not familiar with, and thus includes more information about the content to help the user find what they want/need.

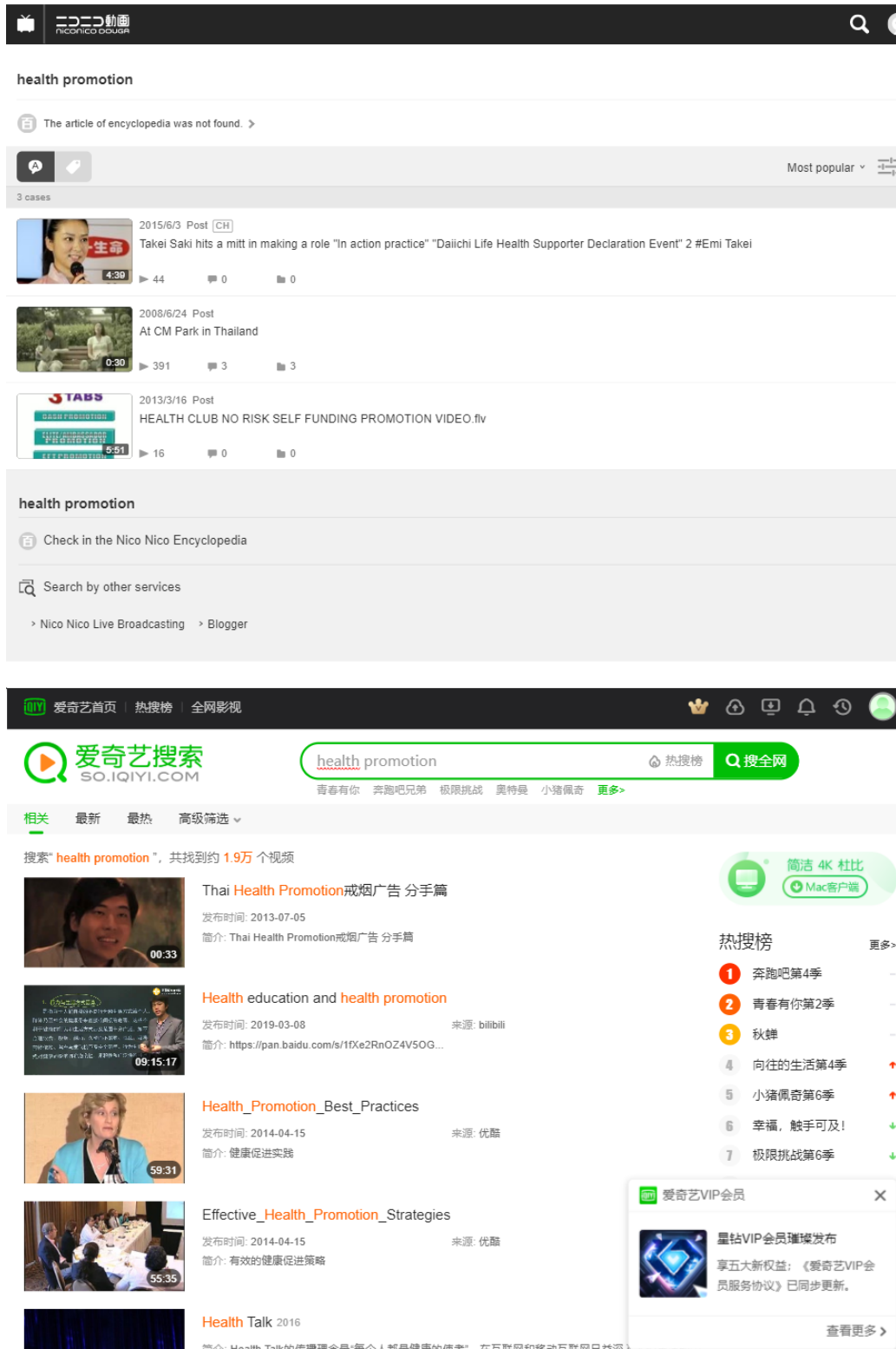


Figure 8.1: Niconico.com (top), iQIYI.com (bottom)

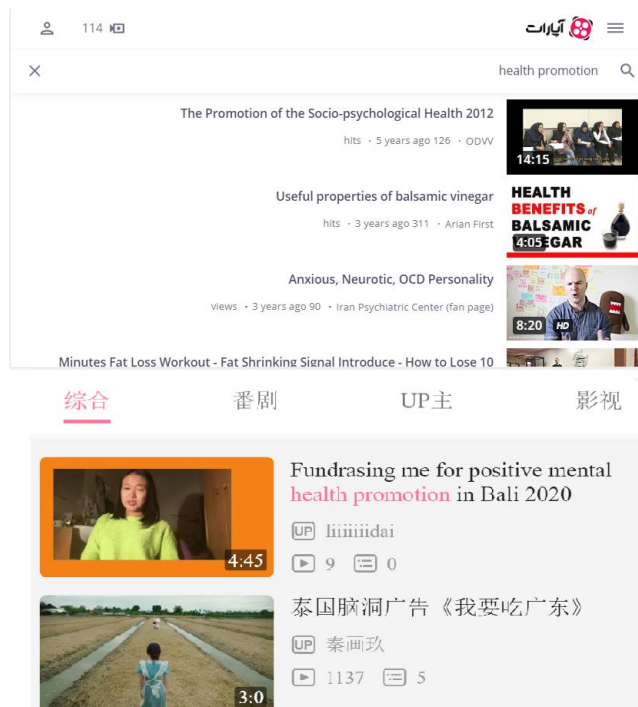


Figure 8.2: Aparat.com (top), bilibili.com (bottom)

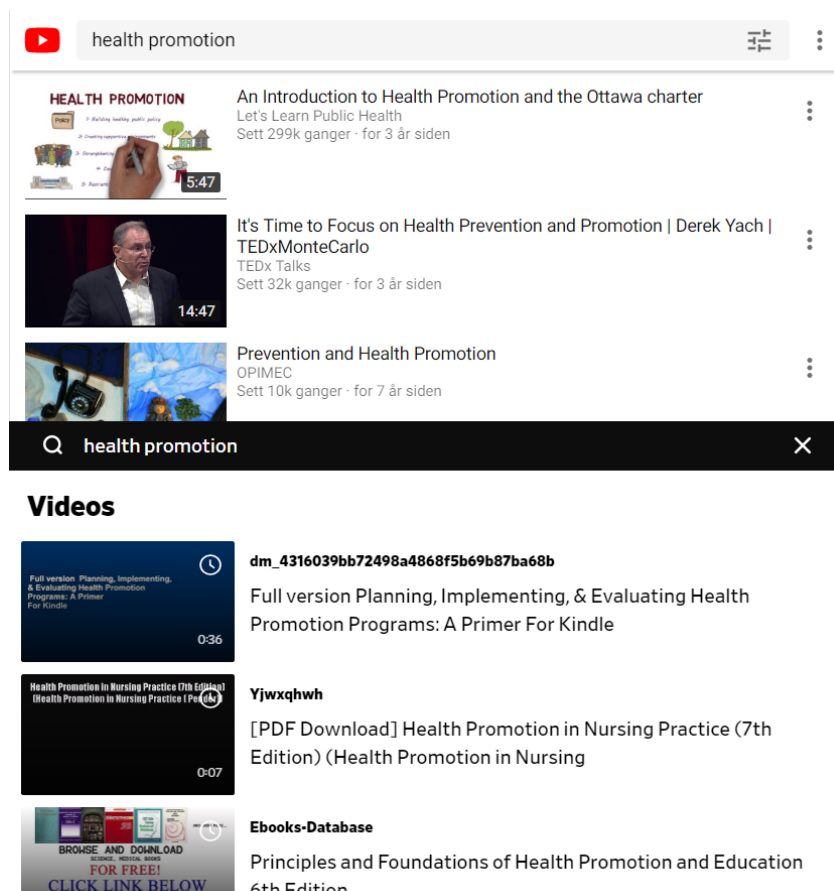


Figure 8.3: YouTube.com (top), dailymotion.com (bottom)

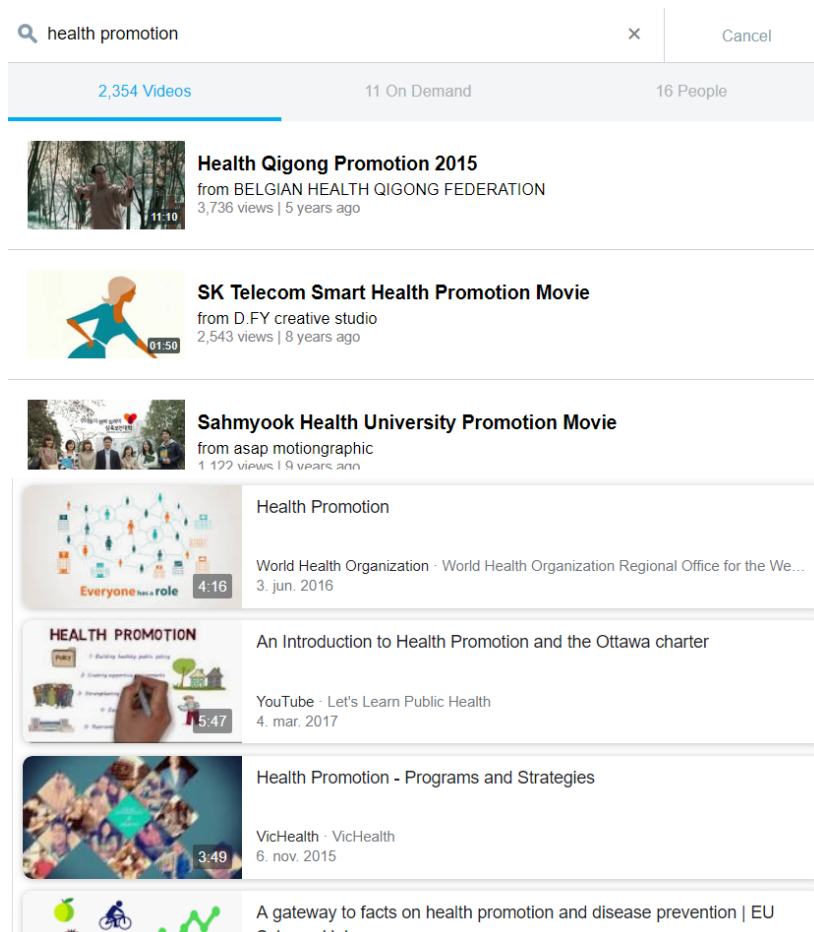


Figure 8.4: vimeo.com (top), Google video search (bottom)

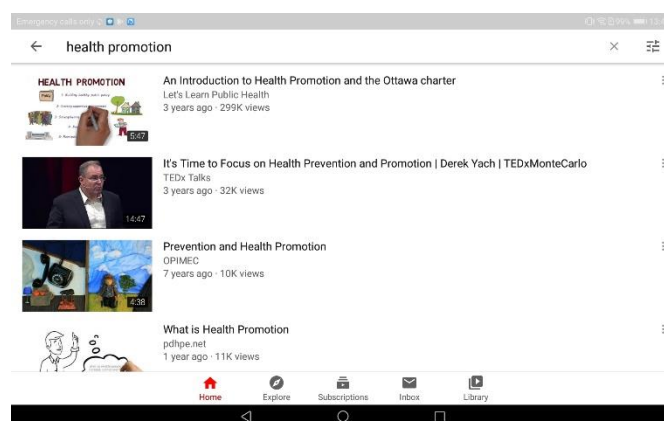


Figure 8.5: YouTube Android app

Of the CHWs we interviewed, all but two had a smartphone. Of those two, one had previously had a smartphone, and one had never used a smartphone. Based on this we believe that the CHWs are fairly familiar with relatively modern smartphones and modern app design.

Despite these websites having a different underlying goal than we do, namely that they are driven by commercial interests, we can take advantage of the standard design they all seem to follow, in order to leverage the familiarity a user would have with these interfaces and thusly reduce training time. As shown in figure 8.6 our design is largely similar to the designs we explored and have described. For each video entry (row) we have a thumbnail, and to the right of it, a title in bold, under which is a description, and a list of languages the video is available in. Of the relevant elements from our external examples, we're only missing the runtime of the video stamped onto the thumbnail, which was not consciously excluded but rather excluded by time constraints.

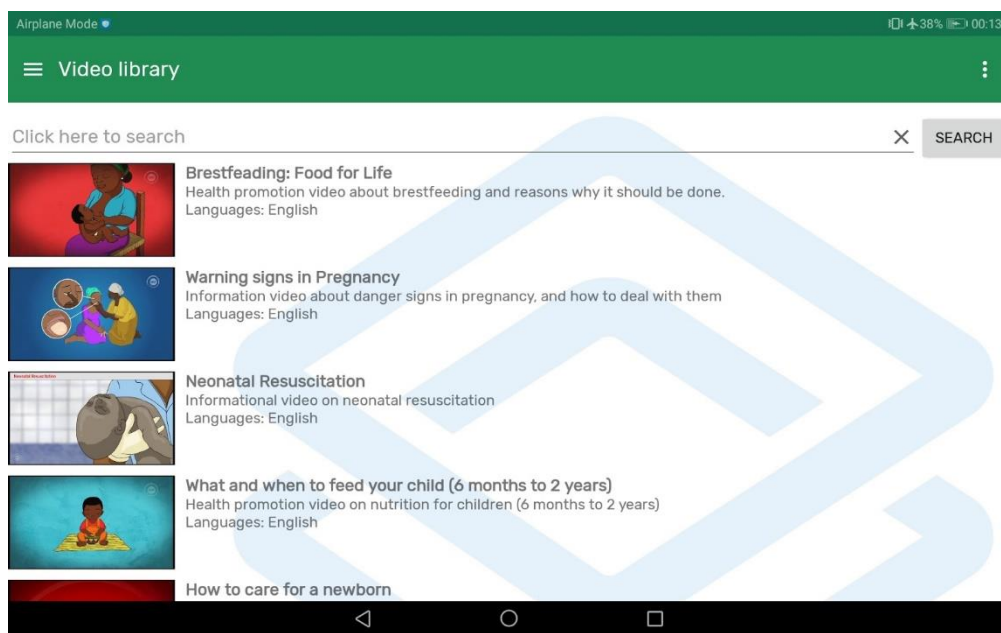


Figure 8.6: Our implemented video library

As presented in our findings, this approach to the design of the video library interface, and the design itself, was largely successful, with all users being able to work the library. Even one user, during the second iteration, which said he had never used a smartphone, was able to scroll in the video library and select a video to play. But why did it work? While the design was based on an established design standard used by the largest websites in the video hosting sphere, we are not sure if our interviewees would have ever, or frequently, been exposed to these websites. While in Malawi, we were told by our guides that mobile data was considered expensive. We also experienced that, while we usually had mobile coverage, the connection quality and available speeds were frequently poor. So, while many people have smartphones capable of streaming videos on, for example, YouTube, we are unsure how

many actually have done this, or do it with any regularity. Instead, we observed several times that people would send each other video clips on WhatsApp.

So how can we explain that most of the users seemed instantly familiar with the interface design of our video library? As highlighted by smartphones often being called “WhatsApp-phones”, the use of the WhatsApp app is widespread. If we look at the interface listing your chats in WhatsApp (figure 8.7) we can see several similarities to the established standard for listing videos. There is a profile picture of the person the chat is with, to the right of which is the person’s name in larger, bold letters, and under which is more details about the conversation (usually the last message).

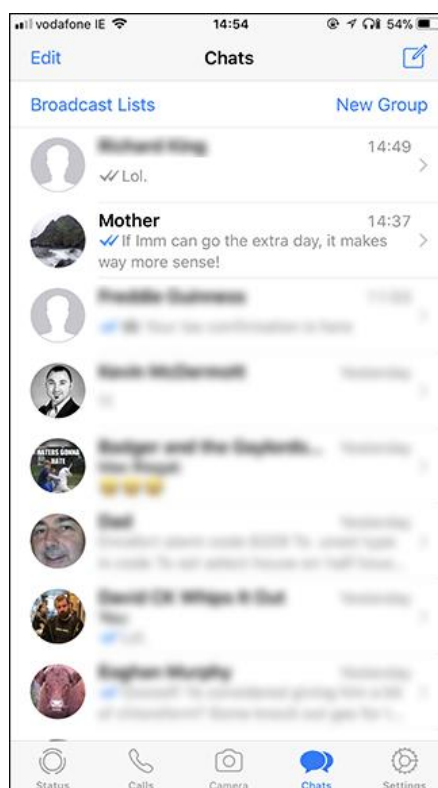


Figure 8.7: WhatsApp chat screen. From <https://www.howtogeek.com/333932/how-to-pin-whatsapp-conversations-to-the-top-of-the-chat-list/>

As we can see, this is quite similar to the thumbnail, title, information pattern we found on the video hosting websites.

Our addition of the back arrow in the app bar for navigation back from the video player to the video library and making the whole row in the video library clickable, also eliminated most instances of users having in navigating between the video library, the video player, and back. The back arrow in the app bar is a very common design pattern in Android apps.

In keeping with using familiar UI design, we opted to use Android’s standard video controls, ‘MediaController’, which contains a scrub controller/timeline, rewind button, play/pause button, and a fast forwards button. These control buttons are similar to most video control designs and thus familiar to most people that have previously used a video player on a smartphone or computer. These icons are generally well established and have been used for a long time, for example on VCR/DVD-player remotes, as highlighted in figure 8.8. As expected, most of our test subjects were able to use these controls without issue.



Figure 8.8: DVD player remote. Taken from <https://www.amazon.com/RMT-B118A-Blu-ray-Player-Remote-Control/dp/B00BA4K0H8>

These observations largely coincide with what we know about the transfer of knowledge and skills between applications (Nielsen, 1993, Polson, 1988, Satzinger & Olfman, 1998).

The mechanics for showing and hiding the video controls turned out to be an issue for some users. This is despite the “tap anywhere on the video surface to show/hide the controls”-pattern of doing this is standard across most video players. In an attempt to decrease this issue, we added a help text to the video player, explaining this interaction. This did reduce the occurrences of this issue. However, given the low amount of data points on this (with a total of 26 observations, 10 before this addition and 16 after), we cannot safely conclude whether this was an effect of this change, us unintentionally providing a better explanation of how this interaction worked during our demonstrations, or simply the experience level of the interviewees. This added text message may also end up being an issue in itself, distracting from the contents or leading to user irritation as users grow tired of seeing the message.

Our language selection dialog used a dropdown selection menu, a standard Android UI element. Several users had trouble with this selection interface, however. The knowledge we have about the transfer of knowledge between applications (Nielsen, 1993, Polson, 1988, Satzinger & Olfman, 1998), might however also explain this. It does not help that the dropdown select is a standard Android UI element if the user has not previously encountered one. Given the extensive use of WhatsApp, it is interesting to note that there is no dropdown selection featuring prominently in WhatsApp (that we could find). Drawing the line from this to the target users not having used dropdown menus before is a stretch though, as we do not have any knowledge of what other apps are common and popular in the area. We did receive suggestions and information through our interviews about how this selection dialog could be improved, in which most users were in favor of moving away from using a dropdown selection, in favor of other selection elements.

The last element of our interface, which we did observe many users having issues with, was the search feature. Some of the facets of such a search bar are well established, as we saw by looking at the other video hosting websites we explored, such as the location of the search bar being at the top of the screen, other facets of it are not as well established as standards. For example, while 5 of the platforms we looked at uses an “x”-button to clear the search, not all did. This is also something that varies between different apps. In WhatsApp for example, a “Cancel” button is used, appearing in the location of our “Search”-button (and triggering a search is done by hitting the submit/enter button on the keyboard). Here we also see a very clear example of our design not aligning with the behavior the users expect. Several of them tried, and confirmed when asked about it that they, intuitively expected the back button to clear the search, to “go back” to the unfiltered list. Instead, doing this would return the user to the main screen of the app.

The initial idea for integrating videos in the DHIS2 Android Capture app was for videos to be integrated as “tooltips” within data capture, similar to how it’s done in the CommCare project described earlier. That integration focuses on supporting the CHWs by doing analysis for them and then recommending them a video to show to the client. It is easy to argue that this is a good solution. It places less responsibility on the CHW to decide on an appropriate action by guiding them through the interaction and into exposing the patient to the selected media. The process of patient interaction guided by the mobile device can also be used to ensure the level of care provided to the patient by the CHW. We believe this is part of the reason the specific CommCare project we were told of did this.

Despite these advantages, we believe that housing the videos in a video library such as the one we developed to be a better solution. Since we were focusing on both videos for health promotion and training, we would need a place to house training videos anyways. Instead of requiring training videos to be introduced through a tracker program, we decided to create the video library. We also believe that the video library makes it easier for the user to find, and especially re-find, a specific video. By having the videos gathered in one central place, we also believe it encourages the users to expose themselves to more of the videos.

A study by Purchase and Worrill (2002) shows that “Video help” is one of the least favorite functions in an on-line help system, way lower ranked than other types of help such as tutorials and information ‘bubbles’ on the screen. This arguably can strengthen our decision to move away from a tooltip solution. However, as a final point, there is nothing about our implementation that restricts future implementers from also triggering videos from tooltips or as parts of guided patient interactions.

## 8.6.2 Backend

As described earlier, the main focus on our work on the backend portion of the technical work was focused on outlining a general, theoretical design for the communication between the DHIS2 Android Capture app and a remote video repository, and the system surrounding the remote video repository. We implemented a bare-bones server functioning as the remote video repository and implemented the client-side (in the app) functionality to poll the repository, and download video-files, thumbnail-files, and metadata from the remote server. We have not, besides from a few recommendations, found a good solution for when and how the client should be triggered to poll the remote server.

In addition, as we’ve touched on, mobile data is expensive and of unreliable speed, and wired internet is not ubiquitous at many clinics. Given these constraints, we have not had time to explore and evaluate how videos should be propagated to the mobile devices. We have provided some suggestions here as well, but we do not have any data on which to evaluate these proposals since our data collection work was mainly focused on the end-user facing implementations in the DHIS2 Android Capture app, and the intended users’ ability to work them.

One substantial decision we have made is to base our architecture on the remote video repository being a stand-alone server and not handled by the DHIS2 Core. This was initially done as we were told, at the beginning of our work with this thesis, that DHIS2 Core did not at the time support video files, and that the core development team did not have time to work on this support.



Beyond this, the DHIS2 Core was not made to handle large multimedia files. By creating a standalone remote video repository, we have decoupled it from the DHIS2 Core. It also enabled us to save on implantation time.

It is also important to note that this decoupling the video repository from the DHIS2 Core, we create a new system that needs to be maintained by the already (often) stretched thin maintenance personnel, and another system for managers to learn. And it's not as if the system will be completely decoupled, either. As we have proposed earlier, part of the remote video repository's system of managing which videos should be downloaded by which users, should rely on the user role and organization group of the user using the DHIS2 Android Capture app. This requires the remote video repository to be connected to the DHIS2 Core through, for example, the DHIS2 Core REST API.

Bucher et al. (2019) describe the implementation of a mobile device-based implementation of videos for training CHWs, using DHIS2. In it they implement their video library as its own app, using the DHIS2 SDK, and communicating with an instance of the DHIS2 Core, where the video files are hosted, using the "resource" feature, which they query using the REST API. They did end up moving away from using the DHIS2 SDK for their app throughout their process, as requirements were discovered that they were not able to fulfill using the SDK. It is also worth noting that their "video library" interface looks quite similar to ours.

Their solution is also based on providing the user with a total list of available videos, and then having the user select which ones to download. Our solution removes this responsibility from the end-user by leaning on the remote video repository to be a management portal for managers to select which users should have which videos on their devices, as described earlier.

Their implementation was focused on providing content for a specific topic ("helping babies survive"), while we focused on creating a solution that would lend itself to both training videos for CHWs on (potentially) any topic, and health promotion videos. While we do not believe that these differing topics made a big difference when it comes to the interface and technical designs chosen by them and us, we did collect data about other possible video topics that the CHWs themselves identified they would like to have training and/or health promotion videos about.

They also mention having originally implemented the local storage of video files on the device in such a way that these could be accessed from outside of the app via the Android gallery and file manager,

and that they eventually disabled this feature as it did not allow for tracking the usage of the content. As we have mentioned earlier, this feature (having the content available outside of the app) is something we are in support of. We have also discussed the topic of tracking video plays and reporting them back to the remote repository, or the DHIS2 Core, but have made no effort on this area beyond these brief discussions. It is our initial opinion that the advantages of having the videos available outside of the app, for example in the case that the app breaks, or the project that used the app ends, outweigh the negatives of not being able to track interactions of the media outside of the app.

We do however acknowledge that we might not have the same focus on metrics and tracking as a manager. As part of the next batch of development, when the remote video repository is further expanded upon, care should be taken to involve the managers in this decision.

## 8.7 Device

One possible detractor from using videos in the previously described manners, is the device itself. As we've described, several of a devices' physical characteristics are important to the usefulness of a video library.

When asked, the CHWs indicated that the tablet we used for testing (Huawei MediaPad T5) would be large enough to show videos to small gatherings. Of a normal-sized smartphone, they said it might be big enough for one person to watch a video. This is slightly contrary to the findings in Bello-Bravo and Baoua (2012), which concluded that mobile phones were adequate devices for delivering informational videos. It's not completely clear, but it seems like the study showed a video to one participant at a time, while in our use-case we need to enable small groups to watch the videos at the same time. As such, we recommend purchases for this use to be tablets.

The adverse effect of increasing the screen size will be a decrease in battery life. Any devices purchased for this should at a minimum be able to survive the workday of a CHW. It should also have a screen bright enough to use in the sun. If one wants to store several videos locally on the tablet, one might run into issues of storage space, depending on the device chosen. All these factors should be taken into consideration when deciding which tablets to purchase for this use.

## 9 Conclusion

This chapter concludes our thesis by summarizing our findings while addressing the research questions presented at the start of this thesis. This chapter also includes our suggestion for further improvements based on our findings, as well as limitations of the study.

### 9.1 How can videos be useful for community health workers?

We identified two main areas into which we explored the usage of videos to be used by CHWs, namely videos for health promotion, and videos for training/re-training. This does not mean this is an extensive list of possible areas, only the ones we explored.

In both areas, the feedback of CHWs was very positive. In terms of videos for training/re-training, many of the CHWs had already experienced videos used this way during their training. Based on these experiences, most of them shared sentiments regarding videos being a good tool for learning and knowledge retention. Previous research, as well as what we were told by CHWs, has also found that the CHWs are not satisfied with the amount of refreshers available to them (Kok & Muula, 2013). Facilitation videos for this purpose could help alleviate this issue.

The CHWs also highlighted that they did not think videos should be used to replace more traditional in-person training, but used in conjunction with, and added to, it. Aside from this all of the CHWs we interviewed were positive to using videos as a tool for their own training.

Several of the CHWs mentioned benefits to themselves, such as not having to carry several flipcharts and less memorization of these. They also mentioned benefits to their clients, such as videos being a medium for information transfer, increased knowledge retention, videos being a novelty and therefore interesting. Another important factor is that the inclusion of videos for health promotion might lead to an increased legitimization and acceptance of the CHWs using a smartphone/tablet in an mHealth capacity, in the eyes of their clients/patients.

The sentiments provided by CHWs on both videos for their own training, and videos for them to use in health promotion largely agrees with existing literature.

One important comment made was the belief that videos for health promotion would help capture men in health promotion situations, which we have been told is often difficult. Videos related to immunization was mentioned by CHWs several times, with them explaining that video material on the subject could help alleviate speciesism around the topic amongst their patients.

We also compiled a list of topics that the CHWs identified as topics which they would like to have videos covering.

## 9.2 How should the interaction and user interface of the DHIS2 Android Capture app be extended to provide the videos in a usable way?

After our first batch of requirements gathering, we decided to move away from implementing the videos as help elements inside of tooltips, which was originally the idea. This choice was made as we found that uses of videos beyond this, as well as many of the uses initially pitched as such tooltip information for how to administer a vaccine as a help element in the digital form for capturing vaccinations performed, was not compatible with the way and speed of which the CHWs worked.

The design of our implemented interfaces was largely based on the interfaces of existing, similar, well-known applications, and the usage of standard Android UI elements. The belief was that the CHWs would be able to transfer existing knowledge and mental models of how to work these applications, into our application. Of the 16 CHWs we asked about it, all of them except two reported owning a smartphone. Of these two, one of them had previously owned one.

In the case of the video listing part of our video library, this strategy proved to work, with almost all users being able to use it without any training, beyond being shown the app and the features we were testing one time.

During our first iteration, users had to click on the video thumbnail (placeholder) to open the video, and only there (due to time constraints). This clearly did not match the mental model of the CHWs, as many of them tried to click on the title, which we expected.

In the case of our video player, most of the users were able to use the controls themselves, which were the standard Android video controls. One CHW, who did not own a smartphone, was confused about the icons in the video controls. However, some of the CHWs had trouble with the interaction for showing/hiding the video controls, despite it being standard.

Our language selection dialog, using a standard Android UI element, proved difficult to understand for some of the CHWs. This might indicate that they were not familiar with this particular UI element, despite it being a standard Android UI element. This might be due to them not being exposed to it in apps they already are familiar with.

In the case of the search bar, our design, again based on existing designs, clearly did not match the mental models of the CHWs, as several of them had trouble working it.

Our initial strategy for navigation from the video player, back to the video library, (using the back button on the Android navigation bar, which used to be physical buttons on older devices) did not work. When questioning the CHWs about this, it became clear that they were looking for a back button in the shape of a left-facing arrow in the app bar (top of the screen). This is a standard design pattern in Android apps.

This highlights that while the CHWs have an established base of knowledge/mental models based on apps they are familiar with that we can draw on when creating interfaces, we, coming from a western country where the use of smartphones and computers are ubiquitous, cannot assume that their existing experiences, knowledge, and mental models are the same as ours.

This means that, while the process of drawing on existing, widespread standard designs is still valid in the context of CHWs in LMICs, we need to supplement this process by either a) employing frequent user testing with the end-users, b) discovering and documenting the “universally understood” design patterns, and the “standards” and established expectations the CHWs have in terms of user interface elements and interactions, or a combination of both of these.

In our case, our testing with the CHWs enabled us to quickly discover these deficiencies and allowed us to work with the CHWs to find solutions to the problems.

### 9.3 How should the backend solution be designed to provide videos on cheap devices in rural settings?

Several factors are important when trying to provide videos to CHWs for health promotion and training, first among them the device itself. Great care should be taken when selecting devices for this use. While it might initially seem pertinent to aid the scaling of an intervention based on providing thousands of CHWs with smart devices by limiting the cost of the devices, several considerations should be made. For example, the size of the screen will decide how big of an audience can view the media at once, the larger the bigger. The screen size will again adversely affect the battery life of the device.

Care should be taken to limit the size of the video files themselves. Providing videos in the local language(s) is important for outreach. A management system should be created for use by managers, to administrate which CHWs in which areas should have which videos. We were unfortunately not able to investigate methods of propagating the videos to the devices in low-bandwidth areas.

We learned several lessons with regards to the unseen solutions that need to be made in order to enable the users and managers of a CHW-facing video library.

### 9.4 Limitations

Our observations of CHWs using our implemented features always came after our introduction and demonstration of the app. We were not able to observe them using it for example after a week had passed. We were not able to leave the devices with them for them to use and then report back to us. Further, we never observed our implemented features in actual use, neither real nor simulated.

We did not interview the CHWs' clients about video-based health promotion, nor did we expose them to it. We did not meaningfully interact directly with their clients at all.

We were not able to take action on the action items collected in iteration two and gather feedback on these. These actions were largely tied to the new features in iteration two, such as the search feature and the language select. We received some good feedback on these topics and have made recommendations based on these in chapter 7.3.

During our data collection phase, the focus of the thesis had not been fully formed. Therefore, the data we collected was of a more general nature, which likely reduced the quality and relevance of the collected data.

## 9.5 Further work

### 9.5.1 Development

While our prototype was suitable for small scale testing with few CHWs, more implementation work is required. The end of chapter 7.3 makes recommendations for further implementation work in the app. The end of chapter 8.5 also discusses how our implementation of handling several languages in the videos could be strengthened.

Further development of the remote video repository is needed, especially in terms of an interface for uploading and managing videos and users. Some of these areas that need further work are outlined in chapter 5. The communication between the app and the remote video repository should also be expanded to include the needed user role and organization unit information needed.

Serious work is needed on the topic of how to propagate the videos from the remote video repository to the devices of CHWs, beyond doing so at initial setup. Part of the idea is to be able to push out new and/or revised videos and change the videos on the device as needed. Transferring these videos in a low-bandwidth setting, where mobile data is expensive, is a serious challenge.

### 9.5.2 Design and use

As mentioned earlier, if the goal is to leverage the know-how users already have in terms of working known applications, in order to support rapid development of UI design, surveys on which apps the CHWs are familiar with, and their level of familiarity and use-patterns would be very useful. In addition, surveying their general familiarity with the standard Android UI elements would also be helpful.

Further user testing with the system should be conducted, preferably in more realistic settings. Being able to observe CHWs use the app in real or simulated training and health promotion situations would likely uncover hidden issues and further potential for improvement. Observations of the CHW's clients being exposed to this form of health promotion should also be performed. Data should be collected on the experiences and opinions of both CHWs and their clients, and the merits of video-based health promotion compared with currently in-use methods.

Creating training videos/refreshers and testing these on CHWs should also be done, to document the usefulness of this.

Beyond this, the processes of how to best produce these videos should also be explored.



## References

- Bello-Bravo, J., & Baoua, I. (2012). Animated Videos as a Learning Tool in Developing Nations: A Pilot Study of Three Animations in Maradi and Surrounding Areas in Niger. *Electronic Journal of Information Systems in Developing Countries*, 55(1), 1-12.
- Braa, J., & Sahay, S. (2012). *Integrated health information architecture : Power to the users : Design development and use*. New Dehli: Matrix Publ.
- Bucher, S., Meyers, E., Kshatriya, B.S.A., Avanigadda, P.C. & Purkayastha, S. (2019). Development of an Innovative Mobile Phone-Based Newborn Care Training Application. In: Abraham A., Gandhi N., Pant M. (eds) *Innovations in Bio-Inspired Computing and Applications*. IBICA 2018. Advances in Intelligent Systems and Computing, vol 939. Springer, Cham
- Chikaphupha, K. R., Kok, M. C., Nyirenda, L., Namakhoma, I., & Theobald, S. (2016). Motivation of health surveillance assistants in Malawi: A qualitative study. *Malawi medical journal: the journal of Medical Association of Malawi*, 28(2), 37–42. <https://doi.org/10.4314/mmj.v28i2.2>
- Coetzee, B., Kohrman, H., Tomlinson, M., Mbewu, N., Le Roux, I., & Adam, M. (2018). Community health workers' experiences of using video teaching tools during home visits—A pilot study. *Health & Social Care in the Community*, 26(2), 167-175.
- Creswell, J. (2014). *Research design : Qualitative, quantitative, and mixed methods approaches* (4th ed.; International student ed.). Los Angeles, Calif: SAGE.
- De Cock, J., Mavlankar, A., Moorthy, A., Aaron, A. (2016). A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications. *Applications of Digital Image Processing XXXIX*, 997116 <https://doi.org/10.1117/12.2238495>
- DHIS2. (n.d.) DHIS2 Mobile. Retrieved March 7, 2020 from <https://www.dhis2.org/mobile>
- DHIS2. (n.d.) DHIS2 User Guide. Retrieved June 4, 2020 from [https://docs.dhis2.org/2.34/en/dhis2\\_user\\_manual\\_en/what-is-dhis2.html](https://docs.dhis2.org/2.34/en/dhis2_user_manual_en/what-is-dhis2.html)
- Ditai, J., Kanyago, J., Nambozo, M.R. et al. (2018) Optimising informed consent for participants in a randomised controlled trial in rural Uganda: a comparative prospective cohort mixed-methods study. *Trials*, 19, 699. <https://doi.org/10.1186/s13063-018-3030-8>
- Foster, V., & Shkaratan, M. (2010). *Malawi's Infrastructure: A Continental Perspective*. The International Bank for Reconstruction and Development
- Gill, P., Stewart, K., Treasure, E. & Chadwick, B. (2008). *Methods of data collection in qualitative research: Interviews and focus groups*. *BDJ*, 204(6), 291-295.
- Google (n.d. A). Retrieved May 29, 2020 from the Android Developer Website: <https://developer.android.com/reference/android/content/Context>
- Google (n.d. B). Retrieved May 29, 2020 from the Android Developer Website: <https://developer.android.com/topic/libraries/architecture/room>
- Google (n.d. C). Retrieved May 29, 2020 from the Android Developer Website: <https://developer.android.com/reference/androidx/work/Worker>

- Google (n.d. D) Retrieved May 29, 2020 from the Android Developer Website:  
<https://developer.android.com/reference/android/app/Activity>
- Google (n.d. E) Retrieved May 29, 2020 from the Android Developer Website:  
<https://developer.android.com/guide/components/fragments>
- Google (n.d. F) Retrieved May 29, 2020 from the Android Developer Website:  
<https://developer.android.com/guide/topics/resources/runtime-changes>
- Grossman, H. (2016). *Contextualization in Video Education in Africa: A Participatory Applied Cognitive Science Approach* (Doctor dissertation, University of California). Retrieved from:  
<https://escholarship.org/uc/item/0j95v6pd>
- Hamilton, R. (2005). Nurses' knowledge and skill retention following cardiopulmonary resuscitation training: A review of the literature. *Journal of Advanced Nursing*, 51(3), 288-297.
- Hattie, J. (2009). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. London: Routledge.
- Holloway, S., Lee, L., & Mcconkey, R. (1999). Meeting the training needs of community-based service personnel in Africa through video-based training courses. *Disability and Rehabilitation*, 21(9), 448-454
- International Telecommunication Union. (2018). *Measuring the Information Society Report - Volume 2*. Switzerland: ITUPublications
- Kawulich, B. (2005). Participant Observation as a Data Collection Method. *Forum: Qualitative Social Research*, 6(2), Art. 43
- Kay, R. (2012). Exploring the use of video podcasts in education: A comprehensive review of the literature. *Computers in Human Behavior*, 28(3), 820-831.
- Kok, M. C., & Muula, A. S. (2013). Motivation and job satisfaction of health surveillance assistants in Mwanza, Malawi: an explorative study. *Malawi medical journal: the journal of Medical Association of Malawi*, 25(1), 5–11.
- Kolb, D. (2014). *Experiential Learning: Experience as the Source of Learning and Development, Second Edition*. PH Professional Business.
- Koumaras, H., Kourtis, M., & Martakos, D. (2012). Benchmarking the encoding efficiency of H.265/HEVC and H.264/AVC. *2012 Future Network & Mobile Summit (FutureNetw)*, 1-7.
- Lippeveld, T., Sauerborn, R., & Bodart, C. (2000). *Design and implementation of health information systems*. Genève: World Health Organization.
- List of video hosting services. (n.d). In Wikipedia. Retrieved 19:34, May 28, 2020, from  
[https://en.wikipedia.org/w/index.php?title=List\\_of\\_video\\_hosting\\_services&oldid=95806700](https://en.wikipedia.org/w/index.php?title=List_of_video_hosting_services&oldid=95806700)  
1
- Nielsen, J. (1993). *Usability engineering*, Chapter 8. Boston: Academic Press
- Nielsen, J. (2004, September 12<sup>th</sup>) The Need for Web Design Standards. Retrieved from  
<https://www.nngroup.com/articles/the-need-for-web-design-standards/?lm=do-interface-standards-stifle-design-creativity&pt=article>
- Norman, D. (2013). *The design of everyday things (Rev. and exp. ed.)*. New York: Basic Books

- Pimmer, C., & Mbvundula, F. (2018). One Message, Many Voices: Mobile Audio Counselling in Health Education. *Journal of Health Care for the Poor and Underserved*, 29(1), 463-480.
- Polson, P.G. (1988). *The Consequences of Consistent and Inconsistent User Interfaces*. In Guindon, R. (Ed), *Cognitive Science and Its Applications for Human-computer Interaction* (pp 59-106). New York: Psychology Press
- Porta, M. (2018). Health Information System. In J.M. Last (Ed.), *A Dictionary of Public Health*. (2nd ed.) Retrieved June 4, 2020 from <https://www.oxfordreference.com/view/10.1093/acref/9780191844386.001.0001/acref-9780191844386-e-1927?rskey=HiaRo3&result=1>
- Purchase, H., & Worrill, J. (2002). An empirical study of on-line help design: Features and principles. *International Journal of Human - Computer Studies*, 56(5), 539-567.
- Republic of Malawi Ministry of Health & Population. (2016). The health care system. Retrieved March 9, 2020 from: <https://www.health.gov.mw/index.php/2016-01-06-19-58-23/national-aids>
- Republic of Malawi Ministry of Health. (2015). Malawi National Health Information System Policy. Retrieved from [https://www.healthdatacollaborative.org/fileadmin/uploads/hdc/Documents/Country\\_documents/September\\_2015\\_Malawi\\_National\\_Health\\_Information\\_System\\_Policy.pdf](https://www.healthdatacollaborative.org/fileadmin/uploads/hdc/Documents/Country_documents/September_2015_Malawi_National_Health_Information_System_Policy.pdf)
- Roser, M., & Ortiz-Ospina, E. (2016). Literacy. Retrieved March 23, 2020 from: <https://ourworldindata.org/literacy>
- Satzinger, J., & Olfman, L. (1998). User Interface Consistency across End-User Applications: The Effects on Mental Models. *Journal of Management Information Systems*, 14(4), 167-193.
- Schmuck, R. (1997). *Practical action research for change*. Arlington Heights, IL: IRI/Skylight Training and Pub.
- Susman, G., & Evered, R. (1978). *An Assessment of the Scientific Merits of Action Research*. *Administrative Science Quarterly*, 23(4), 582-603.
- United Nations Data Retrieval System. (2019). Malawi. Retrieved March 9, 2020 from: <https://data.un.org/en/iso/mw.html>
- United Nations Development Programme. (2019). *Human Development Report 2019*. New York: United Nations Development Programme
- United Nations Malawi. (2014). Malawi Country Profile. Retrieved March 9, 2020 from: <https://mw.one.un.org/country-profile/>
- Van Mele, P. (2011). Video-mediated farmer-to-farmer learning for sustainable agriculture. Ghent, Belgium: *Agro-Insight*.
- Van Mele, P., Wanvoeke, J., & Zossou, E. (2010). Enhancing rural learning, linkages, and institutions: The rice videos in Africa. *Development in Practice*, 20(3), 414-421.
- WHO Regional Office for the Western Pacific. (2004). *Developing Health Management Information Systems – A practical guide for developing countries*, World Health Organization.

World Health Organization. (2016). Health workforce requirements for universal health coverage and the Sustainable Development Goals. Retrieved from <https://apps.who.int/iris/bitstream/handle/10665/250330/9789241511407-eng.pdf>

World Health Organization. (2018). WHO country cooperation strategy at a glance: Malawi. Retrieved March 9, 2020 from: <https://apps.who.int/iris/handle/10665/136935>

# Appendix 1 – Interview guide iteration 1

1. What is your name?
2. What is your role?
3. How long have you works as <role>?
4. Do you have ant experience with DHIS2?
5. Have you used videos in any previous health training situation?
  - a. If yes, how did you experience that compared to our application, if you don't focus on the content of the videos?
6. Have you used videos in any previous health promotion situation?
  - a. If yes, how did you experience that compared to our application, if you don't focus on the content of the videos?
7. What can be improved?
8. How is the use of video compared to reading training materials, e.g. personal notes from previous training lectures, health manuals?
9. How is the use of video compared to using physical assistive devices, e.g. flipchart?
10. What would have wanted presented in video form?
11. Any other comments?

## Appendix 2 – Interview guide iteration 2

1. What is your name?
2. What is your job title?
3. How long have you works as <title>?
4. Do you have a smart phone/WhatsApp phone?
5. Do you have any experience with DHIS2?
6. Have you used videos in any previous health training situation?
  - a. If yes, how did you experience that compared to our application, if you don't focus on the content of the videos?
7. Have you used videos in any previous health promotion situation?
  - a. If yes, how did you experience that compared to our application, if you don't focus on the content of the videos?
8. What can be improved?
9. Is it important to have different language?
10. What do you think of our solution for selecting language?
11. What do you think of our solution for searching?
12. How is the use of video compared to reading training materials, e.g. personal notes from previous training lectures, health manuals?
13. How is the use of video compared to using physical assistive devices, e.g. flipchart?
14. Is this screen(tablet) big enough for health promotion? What about <phone screen>?
15. What would you have wanted presented in video form?
16. Any other comments?

## Appendix 3 – Code

# Repos:

<https://github.com/arneroan/dhis2-android-capture-app/tree/dev-videos>

<https://github.com/arneroan/dhis2-video-server/>

## Code, android app:

### app/src/main/java/org/dhis2/data/service/VideoDownloadWorker.java:

*Whole file by us.*

```
package org.dhis2.data.service;

import android.content.Context;
import android.content.res.AssetManager;
import android.os.AsyncTask;

import androidx.annotation.NonNull;
import androidx.work.Worker;
import androidx.work.WorkerParameters;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import org.dhis2.data.videoDatabase.VideoDatabaseClient;
import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
```

```

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.lang.reflect.Type;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;
import java.util.concurrent.ExecutionException;

import javax.inject.Inject;

public class VideoDownloadWorker extends Worker {

    public VideoDownloadWorker(
        @NonNull Context context,
        @NonNull WorkerParameters workerParams) {
        super(context, workerParams);
    }

    @NonNull
    @Override
    public Result doWork(){
        Properties properties = getConfigValue(getApplicationContext());
        if (properties == null || properties.getProperty("library_url") == null) {

```



```

        return Result.failure();
    }

    String server = properties.get("library_url").toString();

    try {
        URL videoListUri = new URL(server + "/videos");
        List<StoredVideoEntity> videos = getAllVideosToDownload(videoListUri);
        List<StoredVideoEntity> videosToDownload = getVideosToDownload(videos);
        downloadVideos(server, videosToDownload);
        return Result.success();

    } catch (MalformedURLException ex) {
        ex.printStackTrace();
    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
    } catch (ExecutionException e) {
        e.printStackTrace();
    }
    return Result.failure();
}

private void downloadVideo(URL url, String fileName) throws IOException {
    //create the new connection
    HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

    //set up some things on the connection
    urlConnection.setRequestMethod("GET");
    urlConnection.setRequestProperty("Accept", "*/*");
    urlConnection.setRequestProperty("User-agent", "mozilla");
}

```

```

urlConnection.setDoOutput(false);

//and connect!
urlConnection.connect();

File file = new File(getApplicationContext().getExternalFilesDir(null), fileName);

//this will be used to write the downloaded data into the file we created
FileOutputStream fileOutput = new FileOutputStream(file);

//this will be used in reading the data from the internet
InputStream inputStream = urlConnection.getInputStream();

//this is the total size of the file
int totalSize = urlConnection.getContentLength();
//variable to store total downloaded bytes
int downloadedSize = 0;

//create a buffer...
byte[] buffer = new byte[1024];
int bufferLength = 0; //used to store a temporary size of the buffer

//now, read through the input buffer and write the contents to the file
while ( (bufferLength = inputStream.read(buffer)) > 0 ) {
    //add the data in the buffer to the file in the file output stream (the file on the sd card
    fileOutput.write(buffer, 0, bufferLength);
    //add up the size so we know how much is downloaded
    downloadedSize += bufferLength;
    //this is where you would do something to report the progress, like this maybe
    //updateProgress(downloadedSize, totalSize);
}

```

```

    }
    //close the output stream when done
    fileOutput.close();
}

private StoredVideoEntity getVideoInfo(URL url, String uid) throws IOException {
    InputStream is = url.openStream();
    Reader reader = new InputStreamReader(is, StandardCharsets.UTF_8);

    Gson gson = new Gson();
    StoredVideoEntity storedVideoEntity = gson.fromJson(reader, StoredVideoEntity.class);

    storedVideoEntity.setUid(uid);

    String[] fileNameSplit = storedVideoEntity.getFileName().split("\\.");
    String fileName = uid + "." + fileNameSplit[fileNameSplit.length - 1];
    storedVideoEntity.setFileName(fileName);

    return storedVideoEntity;
}

private List<StoredVideoEntity> getAllVideosToDownload(URL url) throws IOException {
    InputStream is = url.openStream();
    Reader reader = new InputStreamReader(is, StandardCharsets.UTF_8);
    Gson gson = new Gson();
    Type listType = new TypeToken<List<StoredVideoEntity>>().getType();
    List<StoredVideoEntity> videos = gson.fromJson(reader, listType);
    return videos;
}

```

```
private List<StoredVideoEntity> getVideosToDownload(List<StoredVideoEntity> videos) throws  
ExecutionException, InterruptedException {
```

```
    List<StoredVideoEntity> storedVideos = (List<StoredVideoEntity>) new  
VideoDatabaseClient(getApplicationContext(),  
0).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR).get();
```

```
    List<StoredVideoEntity> videosToDownload = new ArrayList<>();
```

```
    for (StoredVideoEntity video : videos) {
```

```
        if (!storedVideos.contains(video)){
```

```
            videosToDownload.add(video);
```

```
        }
```

```
    }
```

```
    return videosToDownload;
```

```
}
```

```
private void downloadVideos(String server, List<StoredVideoEntity> videos) throws IOException {
```

```
    for (StoredVideoEntity video : videos) {
```

```
        URL url = new URL(server + "/video/" + video.getUid());
```

```
        String[] fileNameSplit = video.getFileName().split("\\.");
```

```
        String fileName = video.getUid() + "." + fileNameSplit[fileNameSplit.length -1];
```

```
        video.setFileName(fileName);
```

```
        String[] thumbnailFileNameSplit = video.getThumbnailFileName().split("\\.");
```

```
        String thumbnailFileName = video.getUid() + "." +  
thumbnailFileNameSplit[thumbnailFileNameSplit.length -1];
```

```
        video.setThumbnailFileName(thumbnailFileName);
```

```
        downloadVideo(url, video.getFileName());
```

```
        downloadThumbnail(server, video);
```

```

        new VideoDatabaseClient(getApplicationContext(),
1).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, video);
    }
}

private void downloadThumbnail(String server, StoredVideoEntity video) throws IOException {
    URL url = new URL(server + "/thumbnail/" + video.getUid());
    downloadVideo(url, video.getThumbnailFileName());
}

private Properties getConfigValue(Context context) {
    AssetManager assetManager = context.getAssets();
    Properties properties = new Properties();
    try {
        InputStream inputStream = assetManager.open("videoServer.properties");
        properties.load(inputStream);
        return properties;
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}
}

```

## **app/src/main/java/org/dhis2/data/service/VideoDownloadWorkerComponent.java:**

*Whole file by us.*

```

package org.dhis2.data.service;

import androidx.annotation.NonNull;

import org.dhis2.data.dagger.PerService;

import dagger.Subcomponent;

```

```
@PerService
@Subcomponent(modules = ReservedValuesWorkerModule.class)
public interface VideoDownloadWorkerComponent {
    void inject(@NonNull VideoDownloadWorker videoDownloadWorker);
}
```

## **app/src/main/java/org/dhis2/data/service/VideoDownloadWorkerModule.java:**

*Whole file by us.*

```
package org.dhis2.data.service;

public class VideoDownloadWorkerModule {
}
```

## **app/src/main/java/org/dhis2/data/user/UserComponent.java:**

*Our changes marked in green*

```
package org.dhis2.data.user;

import androidx.annotation.NonNull;

import org.dhis2.data.dagger.PerUser;
import org.dhis2.data.service.ReservedValuesWorkerComponent;
import org.dhis2.data.service.ReservedValuesWorkerModule;
import org.dhis2.data.service.SyncDataWorkerComponent;
import org.dhis2.data.service.SyncDataWorkerModule;
import org.dhis2.data.service.SyncGranularRxComponent;
import org.dhis2.data.service.SyncGranularRxModule;
import org.dhis2.data.service.SyncInitWorkerComponent;
import org.dhis2.data.service.SyncInitWorkerModule;
import org.dhis2.data.service.SyncMetadataWorkerComponent;
```

```
import org.dhis2.data.service.SyncMetadataWorkerModule;
import org.dhis2.usescases.about.AboutComponent;
import org.dhis2.usescases.about.AboutModule;
import org.dhis2.usescases.datasets.dataSetTable.DataSetTableComponent;
import org.dhis2.usescases.datasets.dataSetTable.DataSetTableModule;
import org.dhis2.usescases.datasets.dataSetTable.dataSetSection.DataValueComponent;
import org.dhis2.usescases.datasets.dataSetTable.dataSetSection.DataValueModule;
import org.dhis2.usescases.datasets.datasetDetail.DataSetDetailComponent;
import org.dhis2.usescases.datasets.datasetDetail.DataSetDetailModule;
import org.dhis2.usescases.datasets.datasetInitial.DataSetInitialComponent;
import org.dhis2.usescases.datasets.datasetInitial.DataSetInitialModule;
import org.dhis2.usescases.enrollment.EnrollmentComponent;
import org.dhis2.usescases.enrollment.EnrollmentModule;
import org.dhis2.usescases.events.ScheduledEventComponent;
import org.dhis2.usescases.events.ScheduledEventModule;
import org.dhis2.usescases.eventsWithoutRegistration.eventCapture.EventCaptureComponent;
import org.dhis2.usescases.eventsWithoutRegistration.eventCapture.EventCaptureModule;
import org.dhis2.usescases.eventsWithoutRegistration.eventInitial.EventInitialComponent;
import org.dhis2.usescases.eventsWithoutRegistration.eventInitial.EventInitialModule;
import org.dhis2.usescases.eventsWithoutRegistration.eventSummary.EventSummaryComponent;
import org.dhis2.usescases.eventsWithoutRegistration.eventSummary.EventSummaryModule;
import org.dhis2.usescases.main.MainComponent;
import org.dhis2.usescases.main.MainModule;
import org.dhis2.usescases.main.program.ProgramComponent;
import org.dhis2.usescases.main.program.ProgramModule;
import org.dhis2.usescases.programEventDetail.ProgramEventDetailComponent;
import org.dhis2.usescases.programEventDetail.ProgramEventDetailModule;
import org.dhis2.usescases.programStageSelection.ProgramStageSelectionComponent;
import org.dhis2.usescases.programStageSelection.ProgramStageSelectionModule;
import org.dhis2.usescases.qrCodes.QrComponent;
import org.dhis2.usescases.qrCodes.QrModule;
```

```

import org.dhis2.usescases.qrCodes.eventsworegistration.QrEventsWOREgistrationComponent;
import org.dhis2.usescases.qrCodes.eventsworegistration.QrEventsWOREgistrationModule;
import org.dhis2.usescases.qrReader.QrReaderComponent;
import org.dhis2.usescases.qrReader.QrReaderModule;
import org.dhis2.usescases.reservedValue.ReservedValueComponent;
import org.dhis2.usescases.reservedValue.ReservedValueModule;
import org.dhis2.usescases.searchTrackEntity.SearchTEComponent;
import org.dhis2.usescases.searchTrackEntity.SearchTEModule;
import org.dhis2.usescases.settings.SyncManagerComponent;
import org.dhis2.usescases.settings.SyncManagerModule;
import org.dhis2.usescases.sms.SmsComponent;
import org.dhis2.usescases.sms.SmsModule;
import org.dhis2.usescases.sync.SyncComponent;
import org.dhis2.usescases.sync.SyncModule;
import org.dhis2.usescases.teiDashboard.TeiDashboardComponent;
import org.dhis2.usescases.teiDashboard.TeiDashboardModule;
import org.dhis2.usescases.teiDashboard.nfc_data.NfcDataWriteComponent;
import org.dhis2.usescases.teiDashboard.nfc_data.NfcDataWriteModule;
import org.dhis2.usescases.teiDashboard.teiProgramList.TeiProgramListComponent;
import org.dhis2.usescases.teiDashboard.teiProgramList.TeiProgramListModule;
import org.dhis2.usescases.videoLibrary.VideoComponent;
import org.dhis2.usescases.videoLibrary.VideoModule;
import org.dhis2.utils.optionset.OptionSetComponent;
import org.dhis2.utils.optionset.OptionSetModule;

import dagger.Subcomponent;

@PerUser
@Subcomponent(modules = UserModule.class)
public interface UserComponent {

```



@NonNull

MainComponent plus(@NonNull MainModule mainModule);

@NonNull

ProgramEventDetailComponent plus(@NonNull ProgramEventDetailModule  
programEventDetailModule);

@NonNull

SearchTEComponent plus(@NonNull SearchTEModule searchTEModule);

@NonNull

TeiDashboardComponent plus(@NonNull TeiDashboardModule dashboardModule);

@NonNull

QrComponent plus(@NonNull QrModule qrModule);

@NonNull

QrEventsWORegistrationComponent plus(@NonNull QrEventsWORegistrationModule qrModule);

@NonNull

TeiProgramListComponent plus(@NonNull TeiProgramListModule teiProgramListModule);

@NonNull

ProgramComponent plus(@NonNull ProgramModule programModule);

@NonNull

EventInitialComponent plus(EventInitialModule eventInitialModule);

@NonNull

EventSummaryComponent plus(EventSummaryModule eventInitialModule);

@NonNull

SyncManagerComponent plus(SyncManagerModule syncManagerModule);

@NonNull

ProgramStageSelectionComponent plus(ProgramStageSelectionModule  
programStageSelectionModule);

@NonNull

QrReaderComponent plus(QrReaderModule qrReaderModule);

@NonNull

AboutComponent plus(AboutModule aboutModule);

@NonNull

VideoComponent plus(VideoModule videoModule);

@NonNull

DataSetDetailComponent plus(DataSetDetailModule dataSetDetailModel);

@NonNull

DataSetInitialComponent plus(DataSetInitialModule dataSetInitialModule);

@NonNull

DataSetTableComponent plus(DataSetTableModule dataSetTableModule);

@NonNull

DataValueComponent plus(DataValueModule dataValueModule);

@NonNull

ReservedValueComponent plus(ReservedValueModule reservedValueModule);

@NonNull

SyncDataWorkerComponent plus(SyncDataWorkerModule syncDataWorkerModule);

@NonNull

SyncMetadataWorkerComponent plus(SyncMetadataWorkerModule syncDataWorkerModule);

@NonNull

ReservedValuesWorkerComponent plus(ReservedValuesWorkerModule reservedValuesWorkerModule);

@NonNull

EventCaptureComponent plus(EventCaptureModule eventCaptureModule);

@NonNull

SmsComponent plus(SmsModule smsModule);

NfcDataWriteComponent plus(NfcDataWriteModule nfcModule);

@NonNull

SyncGranularRxComponent plus(SyncGranularRxModule syncGranularRxModule);

@NonNull

SyncComponent plus(SyncModule syncModule);

@NonNull

SyncInitWorkerComponent plus(SyncInitWorkerModule syncInitWorkerModule);

@NonNull

EnrollmentComponent plus(EnrollmentModule enrollmentModule);

```

@NonNull
ScheduledEventComponent plus(ScheduledEventModule scheduledEventModule);

@NonNull
OptionSetComponent plus(OptionSetModule optionSetModule);
}

```

## app/src/main/java/org/dhis2/data/videoDatabase/VideoDatabase.java:

*Whole file by us.*

```

package org.dhis2.data.videoDatabase;

import androidx.room.Database;
import androidx.room.RoomDatabase;

import org.dhis2.data.videoDatabase.daos.StoredVideoDao;
import org.dhis2.data.videoDatabase.daos.VideoLanguageDao;
import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

@Database(entities = {StoredVideoEntity.class, VideoLanguageEntity.class},
version = 5)
public abstract class VideoDatabase extends RoomDatabase {
    public abstract StoredVideoDao storedVideoDao();
    public abstract VideoLanguageDao videoLanguageDao();
}

```

## app/src/main/java/org/dhis2/data/videoDatabase/VideoDatabaseClient.java:

*Whole file by us.*

```

package org.dhis2.data.videoDatabase;

import android.app.Activity;
import android.content.Context;
import android.os.AsyncTask;

```

```

import android.util.Log;

import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

import java.util.List;

public class VideoDatabaseClient extends AsyncTask<Object, Void, Object> {

    private int taskID;
    private final static String TAG = "dbClientAsync";

    private Context context;

    public VideoDatabaseClient(Context context, int taskID) {
        this.context = context;
        this.taskID = taskID;
    }

    @Override
    protected void onPreExecute() {

        super.onPreExecute();
    }

    @Override
    protected Object doInBackground(Object... params) {
        Log.i("button4", "We should do database stuff now!");

        VideoDatabaseInstance dbInstance =
VideoDatabaseInstance.getInstance(context);

        switch (this.taskID) {
            case 0: //get all videos in DB
                List<StoredVideoEntity> videos =
dbInstance.getVideoDatabase()
                .storedVideoDao().getAll();

                for (StoredVideoEntity video : videos) {
                    List<VideoLanguageEntity> videoLanguages =
dbInstance.getVideoDatabase()
                    .videoLanguageDao().getAll(video.getUid());

                    video.setVideoLanguages(videoLanguages);
                }
                return videos;
            case 1: //insert a video.
                StoredVideoEntity videoEntity = (StoredVideoEntity)
params[0];

                dbInstance.getVideoDatabase()
                    .storedVideoDao().insert(videoEntity);
                dbInstance.getVideoDatabase()
                    .videoLanguageDao().insertAll(videoEntity.getVideoL
anguages());
                break;
            case 2: //delete all entries
                dbInstance.getVideoDatabase()
                    .storedVideoDao().deleteAll();
            default:

```

```

        break;
    }
    return null;
}
}

```

## **app/src/main/java/org/dhis2/data/videoDatabase/VideoDatabaseInstance.java:**

*Whole file by us.*

```

package org.dhis2.data.videoDatabase;

import android.content.Context;
import androidx.room.Room;

public class VideoDatabaseInstance {
    private static VideoDatabaseInstance dbInstance;
    private VideoDatabase videoDatabase;

    private VideoDatabaseInstance(Context context) {
        videoDatabase = Room.databaseBuilder(context, VideoDatabase.class,
"video-database")
            .fallbackToDestructiveMigration()
            .build();
    }

    public static synchronized VideoDatabaseInstance getInstance(Context
context) {
        if (dbInstance == null) {
            dbInstance = new VideoDatabaseInstance(context);
        }
        return dbInstance;
    }

    public VideoDatabase getVideoDatabase() { return videoDatabase; }
}

```

## **app/src/main/java/org/dhis2/data/videoDatabase/daos/StoredVideoDao.java:**

*Whole file by us.*

```

package org.dhis2.data.videoDatabase.daos;

import androidx.room.Dao;

```

```

import androidx.room.Insert;
import androidx.room.OnConflictStrategy;
import androidx.room.Query;

import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;

import java.util.List;

@Dao
public interface StoredVideoDao {
    @Query("SELECT * FROM storedVideoEntity")
    List<StoredVideoEntity> getAll();

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insert(StoredVideoEntity storedVideoEntity);

    @Query("DELETE FROM storedvideoentity")
    void deleteAll();
}

```

## app/src/main/java/org/dhis2/data/videoDatabase/daos/VideoLanguageDao.java:

*Whole file by us.*

```

package org.dhis2.data.videoDatabase.daos;

import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.OnConflictStrategy;
import androidx.room.Query;

import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

import java.util.List;

@Dao
public interface VideoLanguageDao {
    @Query("SELECT * FROM videoLanguageEntity")
    VideoLanguageEntity[] getAll();

    @Query("SELECT * FROM videoLanguageEntity WHERE video_uid == :uid")
    List<VideoLanguageEntity> getAll(String uid);

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insert(VideoLanguageEntity videoLanguageEntity);

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insertAll(List<VideoLanguageEntity> videoLanguageEntities);

    @Delete
    void delete(VideoLanguageEntity videoLanguageEntity);
}

```

## app/src/main/java/org/dhis2/data/videoDatabase/entities/StoredVideoEntity.java:

*Whole file by us.*

```
package org.dhis2.data.videoDatabase.entities;

import androidx.annotation.NonNull;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.Ignore;
import androidx.room.PrimaryKey;

import com.google.gson.annotations.JsonAdapter;

import java.util.List;

@Entity
@JsonAdapter(StoredVideoEntityDeserializer.class)
public class StoredVideoEntity {

    @NonNull
    @PrimaryKey
    public String uid;

    @NonNull
    @ColumnInfo(name = "video_title")
    public String videoTitle;

    @NonNull
    @ColumnInfo(name = "file_name")
    public String fileName;

    @ColumnInfo(name = "description")
    public String description;

    @NonNull
    @ColumnInfo(name = "version")
    public int version;

    @ColumnInfo(name = "thumbnail_file_name")
    public String thumbnailFileName;

    public String tags;

    @Ignore
    public List<VideoLanguageEntity> videoLanguages;

    public StoredVideoEntity(String uid, String videoTitle, String
fileName, String description, int version, String thumbnailFileName, String
tags) {
        this.uid = uid;
        this.videoTitle = videoTitle;
        this.fileName = fileName;
        this.description = description;
    }
}
```



```

        this.version = version;
        this.thumbnailFileName = thumbnailFileName;
        this.tags = tags;
    }
    public StoredVideoEntity(String uid, String videoTitle, String
fileName, String description, int version, String thumbnailFileName, String
tags, List<VideoLanguageEntity> videoLanguages) {
        this.uid = uid;
        this.videoTitle = videoTitle;
        this.fileName = fileName;
        this.description = description;
        this.version = version;
        this.thumbnailFileName = thumbnailFileName;
        this.tags = tags;
        this.videoLanguages = videoLanguages;
    }

    public String getUid() { return uid; }

    public String getVideoTitle() { return videoTitle; }

    public String getFileName() { return fileName; }

    public List<VideoLanguageEntity> getVideoLanguages() { return
videoLanguages; }

    public String getDescription() { return description; }

    public String getThumbnailFileName() {
        return thumbnailFileName;
    }

    public int getVersion() { return version; }

    public String getTags() {
        return tags;
    }

    public void setFileName(String fileName) { this.fileName = fileName; }

    public void setUid(String uid) { this.uid = uid; }

    public void setVideoLanguages(List<VideoLanguageEntity> videoLanguages)
{
        this.videoLanguages = videoLanguages;
    }

    public void setThumbnailFileName(String thumbnailFileName) {
        this.thumbnailFileName = thumbnailFileName;
    }

    @Override
    public boolean equals(Object other){
        if (!(other instanceof StoredVideoEntity)) return false;
        return (this.uid.equals(((StoredVideoEntity) other).getUid()) &&
this.version == ((StoredVideoEntity) other).getVersion());
    }
}

```

## app/src/main/java/org/dhis2/videoDatabase/entities /StoredVideoEntityDeserializer.java:

*Whole file by us.*

```
package org.dhis2.data.videoDatabase.entities;

import com.google.gson.JsonArray;
import com.google.gson.JsonDeserializationContext;
import com.google.gson.JsonDeserializer;
import com.google.gson.JsonElement;
import com.google.gson.JsonObject;
import com.google.gson.JsonParseException;

import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

import java.lang.reflect.Type;
import java.util.ArrayList;
import java.util.List;

public class StoredVideoEntityDeserializer implements
JsonDeserializer<StoredVideoEntity> {
    @Override
    public StoredVideoEntity deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) throws JsonParseException {
        JsonObject jsonObject = json.getAsJsonObject();
        JsonArray jsonLanguages =
jsonObject.get("videoLanguages").getAsJsonArray();

        List<VideoLanguageEntity> languages = new ArrayList<>();
        jsonLanguages.get(0).getAsJsonObject();

        for (int i = 0; i < jsonLanguages.size(); i++) {
            languages.add(new VideoLanguageEntity(
                jsonObject.get("uid").getAsString(),
                jsonObject.get(i).getAsJsonObject().get("languageName").getAsString(),
                jsonObject.get(i).getAsJsonObject().get("index").getAsInt()
            ));
        }

        return new StoredVideoEntity(
            jsonObject.get("uid").getAsString(),
            jsonObject.get("videoTitle").getAsString(),
            jsonObject.get("fileName").getAsString(),
            jsonObject.get("description").getAsString(),
            jsonObject.get("version").getAsInt(),
            jsonObject.get("thumbnailFileName").getAsString(),
            jsonObject.get("tags").getAsString(),
            languages
        );
    }
}
```

## **app/src/main/java/org/dhis2/data/videoDatabase/entities/VideoLanguageEntity.java:**

*Whole file by us.*

```
package org.dhis2.data.videoDatabase.entities;

import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.ForeignKey;

import org.jetbrains.annotations.NotNull;

@Entity(foreignKeys = @ForeignKey(entity = StoredVideoEntity.class,
parentColumns = "uid", childColumns = "video_uid", onDelete =
ForeignKey.CASCADE),
primaryKeys = {"video_uid", "language_name"})
public class VideoLanguageEntity {

    @NotNull
    @ColumnInfo(name = "video_uid")
    private String videoUid;

    @NotNull
    @ColumnInfo(name = "language_name")
    private String languageName;

    @NotNull
    private int index;

    public VideoLanguageEntity(String videoUid, String languageName, int
index) {
        this.videoUid = videoUid;
        this.languageName = languageName;
        this.index = index;
    }

    public String getVideoUid() {
        return videoUid;
    }

    public String getLanguageName() {
        return languageName;
    }

    public int getIndex() {
        return index;
    }
}
```

## **app/src/main/java/org/dhis2/usecases/main/MainActivity.kt:**

*Our additions marked in green.*

```
package org.dhis2.usescases.main

import android.Manifest
import android.app.Activity
import android.content.Context
import android.content.Intent
import android.content.SharedPreferences
import android.content.pm.PackageManager
import android.os.Bundle
import android.transition.ChangeBounds
import android.transition.TransitionManager
import android.view.View
import android.widget.TextView
import android.widget.Toast
import androidx.constraintlayout.widget.ConstraintSet
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.databinding.DataBindingUtil
import androidx.databinding.ObservableInt
import com.andrognito.pinlockview.PinLockListener
import com.android.dbexporterlibrary.ExporterListener
import javax.inject.Inject
import org.dhis2.App
import org.dhis2.R
import org.dhis2.data.prefs.Preference
import org.dhis2.databinding.ActivityMainBinding
import org.dhis2.usescases.about.AboutFragment
import org.dhis2.usescases.development.DevelopmentActivity
import org.dhis2.usescases.general.ActivityGlobalAbstract
import org.dhis2.usescases.general.FragmentGlobalAbstract
import org.dhis2.usescases.jira.JiraFragment
import org.dhis2.usescases.main.program.ProgramFragment
import org.dhis2.usescases.qrReader.QrReaderFragment
import org.dhis2.usescases.settings.SyncManagerFragment
import org.dhis2.usescases.teiDashboard.nfc_data.NfcDataWriteActivity
import org.dhis2.usescases.videoLibrary.VideoFragment
import org.dhis2.utils.Constants
import org.dhis2.utils.DateUtils
import org.dhis2.utils.analytics.BLOCK_SESSION
import org.dhis2.utils.analytics.CLICK
import org.dhis2.utils.analytics.CLOSE_SESSION
import org.dhis2.utils.filters.FilterManager
import org.dhis2.utils.filters.FiltersAdapter

private const val FRAGMENT = "Fragment"
private const val PERMISSION_REQUEST = 1987

class MainActivity : ActivityGlobalAbstract(), MainView, ExporterListener {
    private lateinit var binding: ActivityMainBinding
    @Inject
    lateinit var presenter: MainPresenter

    private var programFragment: ProgramFragment? = null

    var activeFragment: FragmentGlobalAbstract? = null

    private var currentFragment = ObservableInt(R.id.menu_home)
```

```

private var isPinLayoutVisible = false

private var fragId: Int = 0
private var prefs: SharedPreferences? = null
private var backDropActive = false
var adapter: FiltersAdapter? = null
    private set

// -----
//region LIFECYCLE

override fun onCreate(savedInstanceState: Bundle?) {
    (applicationContext as
App).userComponent()?.plus(MainModule(this))!!.inject(this)

    super.onCreate(savedInstanceState)
    binding = DataBindingUtil.setContentView(this,
R.layout.activity_main)
    binding.presenter = presenter
    binding.navView.setNavigationItemSelectedListener { item ->
        changeFragment(item.itemId)
        false
    }

binding.pinLayout.pinLockView.attachIndicatorDots(binding.pinLayout.indicatorDots)
    binding.pinLayout.pinLockView.setPinLockListener(object :
PinLockListener {
        override fun onComplete(pin: String) {
            presenter.blockSession(pin)
        }

        override fun onEmpty() {}

        override fun onPinChange(pinLength: Int, intermediatePin:
String) {}
    })

    if (savedInstanceState != null) {
        val frag = savedInstanceState.getInt(FRAGMENT)
        currentFragment.set(frag)
        binding.currentFragment = currentFragment
        changeFragment(frag)
    } else {
        binding.currentFragment = currentFragment
        changeFragment(R.id.menu_home)
    }

    prefs = applicationContext.getSharedPreferences(
        Constants.SHARE_PREFS, Context.MODE_PRIVATE
    )

    adapter = FiltersAdapter()
    binding.filterLayout.adapter = adapter

    binding.moreOptions.setOnLongClickListener {
        startActivity(DevelopmentActivity::class.java, null, false,
false, null)
        false
    }
}

```

```

}

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    outState.putInt(FRAGMENT, fragId)
}

override fun onResume() {
    super.onResume()
    presenter.init()
    presenter.initFilters()

    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    ) != PackageManager.PERMISSION_GRANTED ||
ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.CAMERA
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.CAMERA),
            PERMISSION_REQUEST
        )
    }
    binding.totalFilters = FilterManager.getInstance().totalFilters
    adapter!!.notifyDataSetChanged()
}

override fun onPause() {
    presenter.onDetach()
    super.onPause()
}

//endregion

/*User info methods*/

override fun renderUsername(username: String) {
    binding.userName = username

(binding.navView.getHeaderView(0).findViewById<View>(R.id.user_info) as
TextView)
        .text = username
    binding.executePendingBindings()
}

/*End of user info methods*/

override fun openDrawer(gravity: Int) {
    if (!binding.mainDrawerLayout.isDrawerOpen(gravity)) {
        binding.mainDrawerLayout.openDrawer(gravity)
    } else {
        binding.mainDrawerLayout.closeDrawer(gravity)
    }
}

override fun showHideFilter() {

```

```

        val transition = ChangeBounds()
        transition.duration = 200
        TransitionManager.beginDelayedTransition(binding.backdropLayout,
transition)
        backDropActive = !backDropActive
        val initSet = ConstraintSet()
        initSet.clone(binding.backdropLayout)
        if (backDropActive) {
            initSet.connect(
                R.id.fragment_container,
                ConstraintSet.TOP,
                R.id.filterLayout,
                ConstraintSet.BOTTOM,
                50
            )
        } else {
            initSet.connect(
                R.id.fragment_container,
                ConstraintSet.TOP,
                R.id.toolbar,
                ConstraintSet.BOTTOM,
                0
            )
        }
        initSet.applyTo(binding.backdropLayout)
        programFragment!!.openFilter(backDropActive)
    }

    override fun onLockClick() {
        if (prefs!!.getString(Preference.PIN, null) == null) {
            binding.mainDrawerLayout.closeDrawers()
            binding.pinLayout.root.visibility = View.VISIBLE
            isPinLayoutVisible = true
        } else {
            presenter.blockSession(prefs!!.getString(Preference.PIN, ""))
        }
    }

    override fun onBackPressed() {
        when {
            fragId != R.id.menu_home -> changeFragment(R.id.menu_home)
            isPinLayoutVisible -> {
                isPinLayoutVisible = false
                startActivity(Intent(this@MainActivity,
MainActivity::class.java))
                finish()
            }
            else -> super.onBackPressed()
        }
    }

    override fun changeFragment(id: Int) {
        fragId = id
        binding.navView.setCheckedItem(id)
        activeFragment = null
        var tag: String? = null

        when (id) {
            R.id.sync_manager -> {
                activeFragment = SyncManagerFragment()
                tag = getString(R.string.SYNC_MANAGER)
            }
        }
    }

```

```

        binding.filter.visibility = View.GONE
    }
    R.id.qr_scan -> {
        activeFragment = QrReaderFragment()
        tag = getString(R.string.QR_SCANNER)
        binding.filter.visibility = View.GONE
    }
    R.id.nfc_scan -> {
        val intentNfc = Intent(this,
NfcDataWriteActivity::class.java)
        startActivity(intentNfc)
    }
    R.id.menu_jira -> {
        activeFragment = JiraFragment()
        tag = getString(R.string.jira_report)
        binding.filter.visibility = View.GONE
    }
    R.id.menu_about -> {
        activeFragment = AboutFragment()
        tag = getString(R.string.about)
        binding.filter.visibility = View.GONE
    }
    R.id.block_button -> {
        analyticsHelper.setEvent(BLOCK_SESSION, CLICK,
BLOCK_SESSION)
        onLockClick()
    }
    R.id.video_button -> {
        activeFragment = VideoFragment()
        tag = getString(R.string.video_dashboard);
        binding.filter.visibility = View.GONE
    }
    R.id.logout_button -> {
        analyticsHelper.setEvent(CLOSE_SESSION, CLICK,
CLOSE_SESSION)
        presenter.logout()
    }
    R.id.menu_home -> {
        activeFragment = ProgramFragment()
        programFragment = activeFragment as ProgramFragment?
        tag = getString(R.string.done_task)
        binding.filter.visibility = View.VISIBLE
    }
    else -> {
        activeFragment = ProgramFragment()
        programFragment = activeFragment as ProgramFragment?
        tag = getString(R.string.done_task)
        binding.filter.visibility = View.VISIBLE
    }
}

if (activeFragment != null) {
    currentFragment.set(id)
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, activeFragment!!,
tag).commitAllowingStateLoss()
    binding.title.text = tag
}
binding.mainDrawerLayout.closeDrawers()

if (backDropActive && activeFragment !is ProgramFragment) {

```



```

        showHideFilter()
    }
}

override fun updateFilters(totalFilters: Int) {
    binding.totalFilters = totalFilters
}

override fun showPeriodRequest(periodRequest:
FilterManager.PeriodRequest) {
    if (periodRequest == FilterManager.PeriodRequest.FROM_TO) {
        DateUtils.getInstance()
            .showFromToSelector(this)
    } else {
        DateUtils.getInstance()
            .showPeriodDialog(
                this,
                { datePeriods ->
FilterManager.getInstance().addPeriod(datePeriods) },
                true
            )
    }
}

fun setTitle(title: String) {
    binding.title.text = title
}

override fun showTutorial(shaked: Boolean) {
    when (fragId) {
        R.id.menu_home -> (activeFragment as
ProgramFragment).setTutorial()
        R.id.sync_manager -> (activeFragment as
SyncManagerFragment).showTutorial()
        else -> showToast(getString(R.string.no_instructions))
    }
}

public override fun onActivityResult(requestCode: Int, resultCode: Int,
data: Intent?) {
    if (requestCode == FilterManager.OU_TREE && resultCode ==
Activity.RESULT_OK) {
        adapter!!.notifyDataSetChanged()
        updateFilters(FilterManager.getInstance().totalFilters)
    }
    super.onActivityResult(requestCode, resultCode, data)
}

override fun fail(message: String, exception: String) {
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show()
}

override fun success(s: String) {
    Toast.makeText(this, s, Toast.LENGTH_SHORT).show()
}
}

```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoComponent.java:**

*Whole file by us.*

```
package org.dhis2.usecases.videoLibrary;

import org.dhis2.data.dagger.PerFragment;
import dagger.Subcomponent;

@PerFragment
@Subcomponent(modules = VideoModule.class)
public interface VideoComponent {
    void inject(VideoFragment programFragment);
}
```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoContracts.java:**

*Whole file by us.*

```
package org.dhis2.usecases.videoLibrary;

public class VideoContracts {
    public interface VideoView{
        void renderVideoLibrary();
    }

    public interface VideoPresenter {
        void init(VideoView videoFragment);
        void onPause();
    }
}
```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoFragment.java:**

*Whole file by us.*

```
package org.dhis2.usecases.videoLibrary;

import android.app.Activity;
import android.content.Context;
```

```

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewTreeObserver;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;

import androidx.annotation.NonNull;
import androidx.databinding.DataBindingUtil;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import androidx.work.OneTimeWorkRequest;
import androidx.work.WorkManager;

import com.google.gson.Gson;

import org.dhis2.Components;
import org.dhis2.R;
import org.dhis2.data.service.VideoDownloadWorker;
import org.dhis2.data.videoDatabase.VideoDatabaseClient;
import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
import org.dhis2.databinding.FragmentVideoBinding;
import org.dhis2.usescases.general.FragmentGlobalAbstract;
import org.dhis2.usescases.videoPlayer.VideoActivity;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutionException;

import javax.annotation.Nullable;
import javax.inject.Inject;

public class VideoFragment extends FragmentGlobalAbstract implements
VideoContracts.VideoView {
    @Inject
    VideoContracts.VideoPresenter presenter;

    private FragmentVideoBinding videoBinding;
    private VideoListAdapter videoListAdapter;

    @Override
    public void renderVideoLibrary() {
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        ((Components) context.getApplicationContext()).userComponent()
            .plus(new VideoModule()).inject(this);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
@androidx.annotation.Nullable ViewGroup container,
@androidx.annotation.Nullable Bundle savedInstanceState) {

```

```

        videoBinding = DataBindingUtil.inflate(inflater,
R.layout.fragment_video, container, false);
        videoBinding.setPresenter(presenter);

        RecyclerView rv = videoBinding.videoListRecyclerView;
        LinearLayoutManager layoutManager = new
LinearLayoutManager(this.getActivity());
        rv.setLayoutManager(layoutManager);

        List<StoredVideoEntity> videoList = new ArrayList<>();
        try {
            videoList = (List<StoredVideoEntity>) new
VideoDatabaseClient(getContext(),
0).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR).get();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (ExecutionException e) {
            e.printStackTrace();
        }
    }

    videoListAdapter = new VideoListAdapter(videoList, (uid, fileName,
languages) -> {
        Log.d("d", "we got it?");
        //connect to starting the video activity here, with the
filename!
        Intent intent = new Intent(getContext(), VideoActivity.class);
        intent.putExtra("fileName", fileName);
        intent.putExtra("languages", (new Gson()).toJson(languages));
        startActivity(intent);
    });
    rv.setAdapter(videoListAdapter);

    videoBinding.searchText.setOnTouchListener((v, event) -> {
        if(event.getAction() == MotionEvent.ACTION_UP &&
event.getRawX() >= (videoBinding.searchText.getRight() -
videoBinding.searchText.getCompoundDrawables()[2].getBounds().width())) {
            videoBinding.searchText.setText(new char[0], 0, 0);
            videoListAdapter.getFilter().filter("");
            InputMethodManager imm = (InputMethodManager)
getContext().getSystemService(Activity.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(getView().getWindowToken(), 0);
            return true;
        }
        return false;
    });

    videoBinding.searchButton.setOnClickListener(view -> {
        String searchText =
videoBinding.searchText.getText().toString();
        videoListAdapter.getFilter().filter(searchText);
        InputMethodManager imm = (InputMethodManager)
getContext().getSystemService(Activity.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
    });

    videoBinding.searchText.setOnEditorActionListener((v, actionId,
event) -> {
        if (actionId == EditorInfo.IME_ACTION_DONE) {
            videoBinding.searchButton.performClick();
            return true;
        }
    });

```

```

        }
        return false;
    });

    videoBinding.downloadAllButton.setOnClickListener(view -> {
        Log.i("button3", "we are in button 3 download terretory!");
        OneTimeWorkRequest uploadWorkRequest = new
OneTimeWorkRequest.Builder(VideoDownloadWorker.class)
            .build();

WorkManager.getInstance(getContext()).enqueue(uploadWorkRequest);
    });

    videoBinding.getExistingButton.setOnClickListener(view -> {
        Log.d("stuff", "it's getting clicked!");
        try {
            List<StoredVideoEntity> ents = (List<StoredVideoEntity>)
new VideoDatabaseClient(getContext(),
0).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR).get();
            Log.i("d", "we did it!");
        } catch (ExecutionException e) {
            e.printStackTrace();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    });

    videoBinding.deleteAllButton.setOnClickListener(view -> {
        new VideoDatabaseClient(getContext(),
2).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    });

videoBinding.searchLayout.getViewTreeObserver().addOnGlobalLayoutListener(n
ew ViewTreeObserver.OnGlobalLayoutListener() {
    @Override
    public void onGlobalLayout() {

videoBinding.searchLayout.getViewTreeObserver().removeOnGlobalLayoutListene
r(this);

        float buttonWidth = videoBinding.searchButton.getWidth();
        float layoutWidth = videoBinding.searchLayout.getWidth();
        float searchWidth = layoutWidth - buttonWidth;
        ViewGroup.LayoutParams params =
videoBinding.searchText.getLayoutParams();
        params.width = (int) Math.floor((double) searchWidth);
        videoBinding.searchText.setLayoutParams(params);
    }
});

    return videoBinding.getRoot();
}
}

```

## app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoListAdapter.java:

*Whole file by us.*

```
package org.dhis2.usecases.videoLibrary;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import org.dhis2.R;
import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;
import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

import java.util.ArrayList;
import java.util.List;

public class VideoListAdapter extends
RecyclerView.Adapter<VideoListAdapter.MyViewHolder> implements Filterable {

    private VideoSearchFilter filter;
    private List<StoredVideoEntity> originalVideoList;
    public List<StoredVideoEntity> videoList;

    private OnRowClickListener onRowClickListener;

    @Override
    public Filter getFilter() {
        if (filter == null) {
            filter = new VideoSearchFilter(this, originalVideoList);
        }
        return filter;
    }
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView titleText, descriptionText, languagesText;
    public ImageView videoThumbnail;
    private ImageView thumbnail;
    private View currentView;

    public MyViewHolder(View view) {
        super(view);
        titleText = view.findViewById(R.id.video_title);
        descriptionText = view.findViewById(R.id.video_description);
        videoThumbnail = view.findViewById(R.id.video_thumbnail);
        languagesText = view.findViewById(R.id.video_languages);
    }
}
```

```

        thumbnail = view.findViewById(R.id.video_thumbnail);
        this.currentView = view;
    }
}

public VideoListAdapter(List<StoredVideoEntity> videoList,
OnRowClickListener onRowClickListener) {
    this.originalVideoList = videoList;
    this.videoList = new ArrayList<>();
    this.videoList.addAll(videoList);
    this.onRowClickListener = onRowClickListener;
}

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.item_video_list, parent, false);

    return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(@NonNull MyViewHolder holder, int
position) {
    StoredVideoEntity video = videoList.get(position);
    holder.titleText.setText(video.getVideoTitle());
    holder.descriptionText.setText(video.getDescription());

    List<VideoLanguageEntity> langs = video.getVideoLanguages();
    String langsString = "";

    for (VideoLanguageEntity lang : langs){
        if (!langsString.equalsIgnoreCase("")){
            langsString += ", ";
        }
        langsString += lang.getLanguageName();
    }
    langsString = "Languages: " + langsString;
    holder.languagesText.setText(langsString);

    if (video.getThumbnailFileName() != null
    && !video.getThumbnailFileName().equalsIgnoreCase("")) {
        Bitmap thumbnailBitmap =
BitmapFactory.decodeFile(holder.currentView.getContext().getExternalFilesDi
r(null).getAbsolutePath() + "/" + video.getThumbnailFileName());
        holder.videoThumbnail.setImageBitmap(thumbnailBitmap);
    }
    holder.itemView.setOnClickListener(v -> {
        onRowClickListener.onRowClicked(video.getId(),
video.getFileName(), video.getVideoLanguages());
    });
}

@Override
public int getItemCount() {
    return videoList.size();
}
}

```

```

    public interface OnRowClickListener {
        void onRowClicked(String uid, String fileName,
List<VideoLanguageEntity> languages);
    }
}

```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoModule.java:**

*Whole file by us.*

```

package org.dhis2.usecases.videoLibrary;

import androidx.annotation.NonNull;

import org.dhis2.data.dagger.PerFragment;
import org.hisp.dhis.android.core.D2;

import dagger.Module;
import dagger.Provides;

@Module
public class VideoModule {

    @Provides
    @PerFragment
    VideoContracts.VideoPresenter providesPresenter(@NonNull D2 d2) {
        return new VideoPresenterImpl(d2);
    }
}

```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoPresenterImpl.java:**

*Whole file by us.*

```

package org.dhis2.usecases.videoLibrary;

import androidx.annotation.NonNull;

import org.hisp.dhis.android.core.D2;

import io.reactivex.disposables.CompositeDisposable;

public class VideoPresenterImpl implements VideoContracts.VideoPresenter {

    private final D2 d2;

```



```

private CompositeDisposable compositeDisposable;

VideoPresenterImpl(@NonNull D2 d2) {
    this.d2 = d2;
}

@Override
public void init(VideoContracts.VideoView view) {
    compositeDisposable = new CompositeDisposable();
}

@Override
public void onPause() {
    compositeDisposable.clear();
}
}

```

## **app/src/main/java/org/dhis2/usecases/videoLibrary/ VideoSearchFilter.java:**

*Whole file by us.*

```

package org.dhis2.usescases.videoLibrary;

import android.widget.Filter;

import org.dhis2.data.videoDatabase.entities.StoredVideoEntity;

import java.util.ArrayList;
import java.util.List;

public class VideoSearchFilter extends Filter {

    private final VideoListAdapter adapter;
    private final List<StoredVideoEntity> originalList;
    private final List<StoredVideoEntity> filteredList;

    public VideoSearchFilter(VideoListAdapter adapter,
List<StoredVideoEntity> originalList) {
        super();
        this.adapter = adapter;
        this.originalList = originalList;
        this.filteredList = new ArrayList<>();
    }

    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        filteredList.clear();
        final FilterResults results = new FilterResults();

        if (constraint.length() == 0) {
            filteredList.addAll(originalList);
        }
        else {

```

```

        final String filterPattern =
constraint.toString().toLowerCase().trim();
        for (StoredVideoEntity videoEntity : originalList) {
            if (searchFilterMatch(videoEntity, filterPattern)) {
                filteredList.add(videoEntity);
            }
        }
    }
    results.values = filteredList;
    results.count = filteredList.size();

    return results;
}

@Override
protected void publishResults(CharSequence constraint, FilterResults
results) {
    adapter.videoList.clear();
    adapter.videoList.addAll((List<StoredVideoEntity>) results.values);
    adapter.notifyDataSetChanged();
}

private boolean searchFilterMatch(StoredVideoEntity storedVideoEntity,
String constraint) {
    return (
storedVideoEntity.getVideoTitle().toLowerCase().contains(constraint)
        ||
storedVideoEntity.getDescription().toLowerCase().contains(constraint)
        ||
storedVideoEntity.getTags().toLowerCase().contains(constraint)
    );
}
}

```

## **app/src/main/java/org/dhis2/usecases/videoPlayer/ VideoActivity.java:**

*Whole file by us.*

```

package org.dhis2.usecases.videoPlayer;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.Button;

```

```

import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.MediaController;
import android.widget.Spinner;
import android.widget.VideoView;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import org.dhis2.R;
import org.dhis2.data.videoDatabase.entities.VideoLanguageEntity;

import java.io.File;
import java.lang.reflect.Type;
import java.util.ArrayList;
import java.util.List;

import static android.content.pm.PackageManager.PERMISSION_GRANTED;

public class VideoActivity extends AppCompatActivity {

    private VideoView mVideoView;
    private String videoFileName;
    private MediaPlayer mediaPlayer;
    private ImageView backButton;
    private int videoPos;
    private boolean isFullscreen = true;
    private List<VideoLanguageEntity> langs;
    private TrackPosition selectedLang;
    private LinearLayout languageSelect;
    private int numAudioTracks;
    private LinearLayout toolbar;
    private LinearLayout clickForControls;

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        outState.putInt("videoPos", mVideoView.getCurrentPosition());
        if (selectedLang != null) {
            outState.putString("videoLang", (new
Gson()).toJson(selectedLang));
        }
        super.onSaveInstanceState(outState);
    }

    @Override
    protected void onRestoreInstanceState(Bundle inState) {
        super.onRestoreInstanceState(inState);
        videoPos = inState.getInt("videoPos");
        Type trackPositionType = new TypeToken<TrackPosition>() {
        }.getType();

        selectedLang = (new
Gson()).fromJson(inState.getString("videoLang"), trackPositionType);
        int i = 1;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

```

```

        if (savedInstanceState != null) {
            videoPos = savedInstanceState.getInt("videoPos");
        }
        this.videoFileName = getIntent().getExtras().getString("fileName");
        Type listType = new TypeToken<List<VideoLanguageEntity>>() {
        }.getType();
        this.langs = (new
Gson()).fromJson(getIntent().getExtras().getString("languages"), listType);
        setContentView(R.layout.activity_video);
        mVideoView = findViewById(R.id.videoview);
        Spinner dropdown = findViewById(R.id.spinner1);
        Button playButton = findViewById(R.id.playButton);
        languageSelect = findViewById(R.id.languageSelect);
        backButton = findViewById(R.id.backArrow);
        toolbar = findViewById(R.id.toolbar);
        clickForControls = findViewById(R.id.clickForControls);

        playButton.setOnClickListener(v -> {
            TrackPosition trackPosition = (TrackPosition)
dropdown.getSelectedItem();
            selectedLang = trackPosition;
            languageSelect.setVisibility(View.GONE);
            mediaPlayer.selectTrack(trackPosition.getMpPosition());
            mVideoView.start();
            displayHelpText();
        });

        backButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Log.i("Back_Button", "Button was clicked");
                finish();
            }
        });

        mVideoView.setOnPreparedListener(mp -> {

            MediaPlayer.TrackInfo[] tracks = mp.getTrackInfo();
            int trackPos = 0;
            List<TrackPosition> trackPositions = new ArrayList<>();
            numAudioTracks = 0;

            for (int i = 0; i < tracks.length; i++) {
                if (tracks[i].getTrackType() ==
MediaPlayer.TrackInfo.MEDIA_TRACK_TYPE_AUDIO) {
                    trackPositions.add(new TrackPosition(i, trackPos));
                    numAudioTracks++;
                    trackPos++;
                }
            }
            for (VideoLanguageEntity lang : langs) {
                for (TrackPosition pos : trackPositions) {
                    if (lang.getIndex() == pos.getListPosition()) {
                        pos.setLangName(lang.getLanguageName());
                    }
                }
            }
            if (isFullscreen) {
                toolbar.setVisibility(View.GONE);
            }
        });
    }
}

```

```

        ArrayAdapter<TrackPosition> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, trackPositions);
        dropdown.setAdapter(adapter);

        mediaPlayer = mp;

        if (numAudioTracks <= 1) {
            languageSelect.setVisibility(View.GONE);
            mVideoView.start();
            displayHelpText();
        } else if (selectedLang != null) {
            languageSelect.setVisibility(View.GONE);
            mediaPlayer.selectTrack(selectedLang.getMpPosition());
            mVideoView.start();
            displayHelpText();
        }
    });
    mVideoView.setOnClickListener(v -> {
        if (!isFullscreen) {

mVideoView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LOW_PROFILE
| View.SYSTEM_UI_FLAG_FULLSCREEN);
            toolbar.setVisibility(View.GONE);
            isFullscreen = true;
        } else {
            toolbar.setVisibility(View.VISIBLE);
            isFullscreen = false;
        }
    });

    MediaController controller = new MediaController(this) {
        @Override
        public void show() {
            super.show(0);
        }
    };

    controller.setMediaPlayer(mVideoView);
    mVideoView.setMediaController(controller);
}

@Override
protected void onStart() {
    super.onStart();
    initializePlayer();
}

@Override
protected void onStop() {
    super.onStop();

    releasePlayer();
}

@Override
protected void onPause() {
    super.onPause();

    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.N) {
        mVideoView.pause();
    }
}

```

```

    }
}

private void initializePlayer() {
    File file = new File(getExternalFilesDir(null).getAbsolutePath() +
"/" + videoFileName);

    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) == PERMISSION_GRANTED) {
        Log.i("work", "we dont have permission?!");
        Uri uri = Uri.fromFile(file);
        mVideoView.setVideoPath(uri.getPath());
        mVideoView.seekTo(videoPos);
    } else {
        Log.i("work", "we have permission");
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 9909);
    }
    mVideoView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LOW_PROFILE
| View.SYSTEM_UI_FLAG_FULLSCREEN);
}

private void releasePlayer() {
    mVideoView.stopPlayback();
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == 9909 && grantResults[0] == PERMISSION_GRANTED) {
        initializePlayer();
    }
}

private void displayHelpText(){
    clickForControls.setVisibility(View.VISIBLE);
    Thread thread = new Thread() {
        @Override
        public void run() {
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
            }

            runOnUiThread(() -> {
                clickForControls.setVisibility(View.GONE);
            });
        }
    };
    thread.start();
}

private final class TrackPosition {
    //position in the mediaPlayer trackInfo-array.
    private final int mpPosition;
    //Position of the language gotten from VideoLanguageEntity
    private final int listPosition;
    private String langName;
}

```

```

    public TrackPosition(int mpPosition, int listPosition) {
        this.mpPosition = mpPosition;
        this.listPosition = listPosition;
    }

    public void setLangName(String langName) {
        this.langName = langName;
    }

    public String toString() {
        return langName;
    }

    public int getMpPosition() {
        return mpPosition;
    }

    public int getListPosition() {
        return listPosition;
    }
}
}

```

## Layout

### app/src/main/res/layout/activity\_video.xml:

*Whole file by us.*

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#000000"
tools:context="org.dhis2.usescases.video.VideoActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <VideoView
            android:id="@+id/videoview"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="8dp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintDimensionRatio="16:9"

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
<LinearLayout
    android:id="@+id/languageSelect"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="#ffffff"
    android:orientation="vertical"
    android:padding="20dp">

    <TextView
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select language:" />

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@android:drawable/btn_dropdown"
        android:spinnerMode="dropdown" />

    <Button
        android:id="@+id/playButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="play" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/clickForControls"
    android:layout_centerInParent="true"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:visibility="gone">
    <TextView
        android:id="@+id/clickForControlsText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click the screen to open the video controls"
        android:textSize="30sp"
        android:textColor="#ffffff"
        android:shadowColor="#000000"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="1"
        android:gravity="center"/>
</LinearLayout>
</RelativeLayout>
<LinearLayout
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/actionBarSize"
    android:background="#80000000"

```



```

        android:gravity="center_vertical">

        <ImageView
            android:id="@+id/backArrow"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_marginStart="16dp"
            app:srcCompat="@drawable/ic_arrow_back"
            tools:ignore="ContentDescription" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

## app/src/main/res/layout/fragment\_video.xml:

*Whole file by us.*

```

<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <data>

        <variable
            name="presenter"

type="org.dhis2.usescases.videoLibrary.VideoContracts.VideoPresenter" />
    </data>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@android:color/white">

        <ImageView
            android:id="@+id/videoThumbnail"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_alignParentEnd="true"
            android:layout_alignParentBottom="true"
            android:layout_marginEnd="-200dp"
            android:layout_marginBottom="-200dp"
            android:alpha=".1"
            app:srcCompat="@drawable/ic_dhis" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:id="@+id/buttonLayout">

            <Button
                android:id="@+id/downloadAllButton"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        android:text="download all"
        tools:layout_editor_absoluteX="114dp"
        tools:layout_editor_absoluteY="-8dp" />

<Button
    android:id="@+id/getExistingButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone"
    android:text="Get existing"
    tools:layout_editor_absoluteX="114dp"
    tools:layout_editor_absoluteY="-8dp" />

<Button
    android:id="@+id/deleteAllButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="gone"
    android:text="Delete all!"
    tools:layout_editor_absoluteX="114dp"
    tools:layout_editor_absoluteY="-8dp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_below="@+id/buttonLayout"
    android:layout_marginTop="10dp">
    <LinearLayout
        android:id="@+id/searchLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/searchText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="text"
            android:maxLines="1"
            android:hint="Click here to search"
            android:drawableRight="@drawable/ic_close"
            android:imeOptions="actionDone"/>
        <Button
            android:id="@+id/searchButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Search" />
    </LinearLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/videoListRecyclerView"
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:scrollbars="vertical"/>
</LinearLayout>

```

```
</RelativeLayout>
</layout>
```

## app/src/main/res/layout/item\_video\_list.xml:

*Whole file by us.*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginBottom="10dp">

    <ImageView
        android:id="@+id/video_thumbnail"
        android:layout_width="160dp"
        android:layout_height="90dp"
        app:srcCompat="@drawable/ic_dhis"
        android:padding="1dp"
        android:background="#000000"
        android:layout_marginRight="10dp"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/video_title"
            android:textSize="16sp"
            android:textStyle="bold"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/video_description"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/video_languages"/>

    </LinearLayout>

</LinearLayout>
```

## app/src/main/res/menu/main\_menu.xml:

*Our changes marked in green.*

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group
    android:id="@+id/main_sections"
    android:checkableBehavior="single">
    <item
      android:id="@+id/menu_home"
      android:icon="@drawable/ic_menu_home"
      android:title="@string/menu_home" />
    </group>

    <group
      android:id="@+id/close_section"
      android:checkableBehavior="single">
      <item
        android:id="@+id/qr_scan"
        android:icon="@drawable/ic_qr"
        android:title="@string/QR_SCANNER" />

        <item
          android:id="@+id/nfc_scan"
          android:icon="@drawable/ic_nfc"
          android:visible="false"
          android:title="NFC Reader" />
        <item
          android:id="@+id/sync_manager"
          android:icon="@drawable/ic_menu_settings"
          android:title="@string/SYNC_MANAGER" />
        <item
          android:id="@+id/block_button"
          android:icon="@drawable/ic_menu_lock"
          android:title="@string/block_session" />

        <item
          android:id="@+id/video_button"
          android:icon="@drawable/ic_menu_logout"
          android:title="@string/video_dashboard" />
        <item
          android:id="@+id/logout_button"
          android:icon="@drawable/ic_menu_logout"
          android:title="@string/log_out" />

      </group>

    <group
      android:id="@+id/extra_section"
      android:checkableBehavior="single">
      <item
        android:id="@+id/menu_jira"
        android:icon="@drawable/ic_jira"
        android:title="@string/jira_report" />

        <item
          android:id="@+id/menu_about"
          android:icon="@drawable/ic_menu_info"
          android:title="@string/about" />

      </group>
    </menu>

```

# Other

## app/src/main/assets/videoServer.properties:

*Whole file by us.*

library\_url=<http://192.168.12.1:8081>

## app/build.gradle:

*Our changes marked in green.*

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.firebase.firebase-perf'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-kapt'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'io.fabric'

def libraries = rootProject.ext.libraries
def versions = rootProject.ext.versions

repositories {
    // required by crashlytics
    maven {
        url("https://maven.fabric.io/public")
    }

    // used by sdk
    maven {
        url("https://oss.sonatype.org/content/repositories/snapshots")
    }
    mavenCentral()
}

android {

    testOptions {
        unitTests {
            includeAndroidResources = true
            returnDefaultValues = true
        }
    }

    lintOptions {
        checkReleaseBuilds false
        abortOnError false
    }
}
```

```

compileSdkVersion versions.sdk
buildToolsVersion versions.tools
defaultConfig {
    applicationId "com.dhis2"
    minSdkVersion versions.minSdk
    targetSdkVersion versions.sdk
    versionCode versions.vCode
    versionName versions.vName
//    testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
    testInstrumentationRunner "org.dhis2.usescases.MockRunner"
    vectorDrawables.useSupportLibrary = true
    multiDexEnabled true

    setProperty("archivesBaseName", "dhis2-v" + versions.vName)

    buildConfigField "String", "SDK_VERSION", "\"" + libraries.dhis2sdk
+ "\""
    buildConfigField "String", "MAPBOX_ACCESS_TOKEN",
"\"pk.eyJ1Ijoib3Jpa2dnIiwiaSI6ImNqdWpqb24zYTFFsOXo0M2xwcXRseGx3aXkifQ.6N9aqv
pqdrrSBwO9GyfjUw\""

    javaCompileOptions {
        annotationProcessorOptions {
            includeCompileClasspath true
        }
    }
}

testOptions {
    unitTests.returnDefaultValues = true
}

configurations.all*.exclude module: 'jsr305'

packagingOptions {
    // excluding duplicate license files
    // from jackson modules
    exclude "META-INF/LICENSE"

    // excluding duplicate rxjava properties file
    // see
https://github.com/ReactiveX/RxJava/issues/4445#issuecomment-243591879
    exclude 'META-INF/rxjava.properties'
}

signingConfigs {
    release {
        storeFile file('dhis_keystore.jks')
        storePassword KEYSTORE_PASSWORD
        keyAlias KEY_ALIAS
        keyPassword KEYSTORE_PASSWORD
    }
}

buildTypes {
    debug {
        // custom application suffix which allows to

```

```

        // install debug and release builds at the same time
        applicationIdSuffix = ".debug"

        // jacoco test coverage reports both for
        // androidTest and test source sets
        testCoverageEnabled = false

        // Disable fabric build ID generation for debug builds
//
        ext.enableCrashlytics = false

        // Using dataentry.jks to sign debug build type.
        signingConfig = signingConfigs.debug

        resValue "string", "content_provider",
"com.dhis2.debug.provider"
    }
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        ext.enableCrashlytics = true
        signingConfig signingConfigs.release

        resValue "string", "content_provider", "com.dhis2.provider"
    }

    training {
        applicationIdSuffix = ".training"
        matchingFallbacks = ['release']
        signingConfig = signingConfigs.debug
    }

    experimental {
        // custom application suffix which allows to
        // install debug and release builds at the same time
        applicationIdSuffix = ".experimental"

        // jacoco test coverage reports both for
        // androidTest and test source sets
        testCoverageEnabled = false

        // Using dataentry.jks to sign debug build type.
        signingConfig = signingConfigs.debug
        debuggable = true
        jniDebuggable = true
        renderscriptDebuggable = true
        matchingFallbacks = ['debug']

        resValue "string", "content_provider",
"com.dhis2.experimental.provider"
    }

    beta {
        applicationIdSuffix = ".beta"
        matchingFallbacks = ['debug']
        signingConfig signingConfigs.debug
        resValue "string", "content_provider",
"com.dhis2.beta.provider"
    }
}

```

```

        alpha {
            applicationIdSuffix = ".alpha"
            matchingFallbacks = ['debug']
            signingConfig signingConfigs.debug
            resValue "string", "content_provider",
"com.dhis2.alpha.provider"
        }
    }

    flavorDimensions "default"
    productFlavors {
        dhis {
            applicationId "com.dhis2"
            dimension "default"
            versionCode versions.vCode
            versionName versions.vName
        }
    }

    compileOptions {
        targetCompatibility = "8"
        sourceCompatibility = "8"
    }

    dataBinding {
        enabled = true
    }

    packagingOptions {
        exclude 'LICENSE.txt'
        exclude 'META-INF/DEPENDENCIES'
        exclude 'META-INF/ASL2.0'
        exclude 'META-INF/NOTICE'
        exclude 'META-INF/LICENSE'
        exclude 'META-INF/rxjava.properties'
    }

    dexOptions {
        javaMaxHeapSize "4g"
    }

    configurations.all {
        resolutionStrategy {
            preferProjectModules()

            force 'junit:junit:4.12', 'com.squareup.okhttp3:okhttp:3.12.0'

            forcedModules = ['com.squareup.okhttp3:okhttp:3.12.0']
        }
    }

    kotlinOptions {
        jvmTarget = "1.8"
    }
}

sonarqube {
    properties {
        property "sonar.language", "java"
    }
}

```



```

dependencies {
    debugImplementation project(':core')
    implementation project(':customFab')
    debugImplementation project(':dhis2-rule-engine')
    implementation project(':viewpagerdotsindicator')
    implementation project(':tableview')

    //region AndroidX
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.annotation:annotation:1.1.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation 'com.google.android.material:material:1.1.0-beta01'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation "androidx.multidex:multidex:${libraries.multidex}"
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation "androidx.lifecycle:lifecycle-
extensions:${libraries.lifecycle}"
    //JOB SCHEDULER
    implementation "androidx.work:work-runtime:2.2.0"
    implementation "androidx.work:work-rxjava2:2.2.0"
    //endregion

    //region DHIS2 LIBRARIES

    releaseImplementation("org.hisp.dhis:android-
core:${libraries.dhis2sdk}") {
        exclude group: "org.hisp.dhis", module: "core-rules"
    }

    releaseImplementation "org.hisp.dhis.rules:rule-
engine:${libraries.ruleEngine}"
    //endregion

    //region UI LIBRARIES
    //AndroidTreeView
    implementation "com.github.bmelnychuk:atv:${libraries.atv}"
    //FAB
    implementation "com.github.dekoservidoni:omfm:${libraries.omfm}"
    //FLEXBOX
    implementation "com.google.android:flexbox:${libraries.flexbox}"

    //PIN LOCK
    implementation 'com.andrognito.pinlockview:pinlockview:2.1.0'
    //endregion

    //region RXJAVA
    implementation "io.reactivex.rxjava2:rxandroid:${libraries.rxandroid}"
    implementation "io.reactivex.rxjava2:rxjava:${libraries.rxjava}"
    implementation "com.github.akarnokd:rxjava2-
interop:${libraries.rxjavainteroperop}"
    implementation 'io.reactivex.rxjava2:rxkotlin:2.3.0'

    //rxbinding
    implementation
"com.jakewharton.rxbinding2:rxbinding:${libraries.rxbindings}"
    implementation "com.jakewharton.rxbinding2:rxbinding-appcompat-
v7:${libraries.rxbindings}"
    //endregion

    //region DAGGER

```

```

implementation "com.google.dagger:dagger:${libraries.dagger}"
kapt "com.google.dagger:dagger-compiler:${libraries.dagger}"
//endregion

//region GOOGLE AUTO VALUE
// Google's auto-value. Annotations are used only during compilation
time, as well as
// annotation processor which hoops up into compilation process
api "com.google.auto.value:auto-value:${libraries.autovalue}"
kapt "com.google.auto.value:auto-value:${libraries.autovalue}"

// Extension which generates mappers for work with cursors
kapt "com.gabrielittner.auto.value:auto-value-
cursor:${libraries.autovaluecursor}"
kapt "com.ryanharter.auto.value:auto-value-
parcel:${libraries.autovalueparcel}"
//endregion

//region BRITEDATABASE
//square
implementation "com.squareup.sqlbrite3:sqlbrite:${libraries.sqlbrite}"
implementation 'com.github.lykmapipo:sqlbrite-migrations:v1.0.1'

//endregion

//region PLAY SERVICES

//LOCATION
implementation 'com.google.android.gms:play-services-basement:17.1.1'
implementation 'com.google.android.gms:play-services-location:17.0.0'
//endregion

// Kujaku dependencies
implementation 'io.ona.kujaku:library:0.7.7'
implementation 'com.mapbox.mapboxsdk:mapbox-android-sdk:8.3.0'
implementation 'com.mapbox.mapboxsdk:mapbox-android-plugin-markerview-
v8:0.3.0'

// Jake Wharton libraries
//timber
implementation "com.jakewharton.timber:timber:${libraries.timber}"

//GSON
implementation 'com.google.code.gson:gson:2.8.5'

//CRASHLYTICS
implementation('com.crashlytics.sdk.android:crashlytics:2.9.5@aar') {
    transitive = true
}

//FIREBASE
implementation 'com.google.firebase:firebase-core:17.2.0'
implementation 'com.google.firebase:firebase-perf:19.0.0'
implementation 'com.google.firebase:firebase-messaging:20.0.0'

//Test
implementation fileTree(include: ['*.jar'], dir: 'libs')
androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
androidTestImplementation "androidx.test:core:1.2.0"

```

```

androidTestImplementation 'androidx.test:rules:1.2.0'
androidTestImplementation "androidx.test:core-ktx:1.2.0"
androidTestImplementation 'androidx.test.ext:junit:1.1.1'
androidTestImplementation "androidx.test.ext:junit-ktx:1.1.1"

testImplementation 'junit:junit:4.12'
implementation "org.jetbrains.kotlin:kotlin-stdlib-
jdk7:$kotlin_version"
testImplementation 'org.robolectric:robolectric:4.2.1'
testImplementation "org.robolectric:shadows-multidex:4.0.1"
androidTestImplementation 'org.mockito:mockito-android:2.28.2'
androidTestImplementation 'org.mockito:mockito-core:2.28.2'
testImplementation 'org.mockito:mockito-core:2.28.2'
testImplementation 'org.mockito:mockito-inline:2.13.0'
testImplementation "com.nhaarman.mockitokotlin2:mockito-kotlin:2.1.0"

testImplementation 'com.google.truth:truth:0.43'
implementation 'com.facebook.stetho:stetho:1.5.0'
implementation 'com.facebook.stetho:stetho-okhttp3:1.5.0'
implementation 'com.squareup.okhttp3:okhttp:3.12.0'

//RXLint
implementation 'nl.littlerobots.rxlint:rxlint:1.6'

//JODA-TIME
implementation 'joda-time:joda-time:2.9.9'

// Zxing library
implementation 'com.google.zxing:core:3.3.0'
implementation 'me.dm7.barcodescanner:zxing:1.9.8'
implementation 'com.journeyapps:zxing-android-embedded:3.2.0@aar'

//FancyShowCase
implementation 'me.toptas.fancyshowcase:fancyshowcaseview:1.1.5'

//Lottie animations
implementation 'com.airbnb.android:lottie:2.8.0'

//Glide load images
implementation 'com.github.bumptech.glide:glide:4.9.0'
kapt 'com.github.bumptech.glide:compiler:4.9.0'

//Paperwork
implementation "hu.supercluster:paperwork:${libraries.paperwork}"

implementation 'com.google.guava:guava:27.0.1-android'

implementation 'cat.ereza:customactivityoncrash:2.2.0'

implementation "com.google.android.gms:play-services-
analytics:${libraries.analytics}"
//implementation 'com.github.pwittchen:rxbiometric:0.1.0'
implementation 'co.infinum:goldfinger-rx:1.2.1'
implementation
"de.adorsys.android:securestorage:library:${libraries.latestSecureStorageVer
sion}"

implementation 'com.github.prabhat1707:DbExporterHelper:v1.0'

/*region SECURITY*/

```

```

implementation 'com.scottyab:rootbeer-lib:0.0.7'

//annotationProcessor for room, since there's also kotlin.
kapt 'android.arch.persistence.room:compiler:1.1.1'
}

apply plugin: 'com.google.gms.google-services'

```

## app/src/main/AndroidManifest.xml:

*Our changes marked in green.*

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="org.dhis2">

    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.USE_BIOMETRIC" /> <!--
- <uses-permission android:name="android.permission.NFC"/> -->
    <!-- <uses-feature android:name="android.hardware.nfc" /> -->
    <uses-feature
        android:name="android.hardware.sensor.accelerometer"
        android:required="false" />

    <application
        android:name=".App"
        android:allowBackup="false"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:usesCleartextTraffic="true"
        tools:replace="allowBackup">
        <activity
android:name=".usescases.videoPlayer.VideoActivity"></activity>

        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />
    </application>
</manifest>

```

```

<activity
    android:name=".usescases.splash.SplashActivity"
    android:theme="@style/SplashTheme">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER"
/>
    </intent-filter>
    <intent-filter android:autoVerify="true">
        <category android:name="android.intent.category.DEFAULT" />

        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.BROWSABLE"
/>

        <data
            android:host="play.dhis2.org"
            android:scheme="https" />
    </intent-filter>
</activity>
<activity
    android:name=".usescases.login.LoginActivity"
    android:windowSoftInputMode="adjustPan" />
<activity android:name=".usescases.main.MainActivity" />
<activity
android:name=".usescases.programEventDetail.ProgramEventDetailActivity" />
    <activity
        android:name=".usescases.searchTrackEntity.SearchTEActivity"
        android:configChanges="keyboardHidden|screenSize" />
    <activity
android:name=".usescases.teiDashboard.TeiDashboardMobileActivity" />
    <activity android:name=".usescases.qrScanner.QRActivity" />
    <activity
android:name=".usescases.teiDashboard.teiProgramList.TeiProgramListActivity
" />
        <activity android:name=".usescases.map.MapSelectorActivity" />
    <activity
android:name=".usescases.programStageSelection.ProgramStageSelectionActivit
y" />
        <activity android:name=".utils.WebViewActivity" />
    <activity

android:name=".usescases.eventsWithoutRegistration.eventInitial.EventInitia
lActivity"
        android:windowSoftInputMode="stateHidden" /> <!-- <activity
android:name=".usescases.teiDashboard.TeiDashboardActivity" /> -->
        <activity android:name=".usescases.qrCodes.QrActivity" />
        <activity
android:name=".usescases.qrCodes.eventsworegistration.QrEventsWOREgistratio
nActivity" />

    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="AIzaSyAGNpaLadezvUQFPLMpWEHqsTxLdNWhPGk" />
    <meta-data
        android:name="io.fabric.ApiKey"
        android:value="a5caa46009119f5e584e4964c30a922095b1075c" />

```

```

        <activity
android:name=".usescases.datasets.datasetDetail.DataSetDetailActivity" />
        <activity
android:name=".usescases.datasets.datasetInitial.DataSetInitialActivity" />
        <activity
android:name=".usescases.datasets.dataSetTable.DataSetTableActivity"
        android:configChanges="keyboardHidden|orientation|screenSize"
/>

        <receiver android:name=".widgets.DhisCustomLauncher">
            <intent-filter>
                <action
android:name="android.appwidget.action.APPWIDGET_UPDATE" />
            </intent-filter>

            <meta-data
                android:name="android.appwidget.provider"
                android:resource="@xml/dhis_custom_launcher_info" />
        </receiver>

        <activity
android:name=".usescases.reservedValue.ReservedValueActivity" />
        <activity
android:name=".usescases.eventsWithoutRegistration.eventCapture.EventCaptureActivity"
        android:configChanges="keyboardHidden|orientation|screenSize"
/>
        <activity android:name=".usescases.sync.SyncActivity" />
        <activity android:name=".usescases.development.DevelopmentActivity"
/>
        <activity
            android:name=".usescases.sms.SmsSubmitActivity"
            android:theme="@style/Theme.AppCompat.Translucent" />

        <service android:name=".usescases.sms.SmsSendingService" />

        <activity
android:name=".usescases.teiDashboard.nfc_data.NfcDataWriteActivity" />

        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="@string/content_provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/file_paths" />
        </provider>

        <activity android:name=".usescases.orgunitselector.OUTreeActivity"
/>
        <activity
            android:name=".usescases.enrollment.EnrollmentActivity"
            android:windowSoftInputMode="adjustResize|stateHidden" />
        <activity android:name=".usescases.events.ScheduledEventActivity"
/>

        <service
            android:name=".data.notifications.DhisFirebaseMessagingService"

```

```

        android:exported="false">
        <intent-filter>
            <action android:name="com.google.firebase.MESSAGING_EVENT"
/>
        </intent-filter>
    </service>

    <meta-data

android:name="com.google.firebase.messaging.default_notification_channel_id"
"
        android:value="@string/default_notification_channel_id" />
    </application>

</manifest>

```

app/src

## app/src/main/res/values/strings.xml:

*Our changes marked in green.*

```

<resources>
    <!--GENERAL-->
    <string name="app_name">Dhis2</string>
    <string name="logo_text">dhis</string>
    <string name="logo_number">2</string>
    <string name="google_api"
translatable="false">AIzaSyATNJAcNH6YB8ZzkJbzh1vWOgMAKi7KEHQ</string>
    <string name="tab" translatable="false">\u0009</string>
    <string name="default_notification_channel_id"
translatable="false">20180901</string>
    <!-- Warnings and error messages -->
    <string name="error">Error</string>
    <string name="error_wrong_server_url">Wrong server url</string>
    <string name="error_wrong_credentials">Wrong credentials</string>
    <string name="error_unexpected_error">Unexpected error</string>
    <string name="error_internal_server_error">Internal server
error</string>
    <string name="yes">yes</string>
    <string name="no">no</string>
    <string name="no_value">no value</string>
    <string name="no_reserved_values">There are no more reserved values.
Contact your system administrator</string>
    <string name="no_reserved_values_connection">There are no more reserved
values. Connect to the network to refill them</string>

    <!-- SyncService strings: -->
    <string name="sync_now">Synchronise now</string>
    <string name="sync_text">Synchronising with server</string>
    <string name="sync_title">Syncing now</string>

    <string name="sync_metadata">Metadata</string>
    <string name="sync_events">Events</string>

```

```

<string name="sync_tei">Tracked entities</string>

<string name="sync_complete_title">Finished</string>
<string name="sync_complete_text">Sync complete</string>

<string name="sync_error_title">Sync Error</string>
<string name="sync_error_text">An error has occurred during
synchronisation. Please retry.</string>
<string name="metadata_sync_error">FAIL: Something went wrong in the
sync process. If you have connection please try again. If the error
persist, contact your administrator.</string>
<string name="data_sync_warning">WARNING: Most of your data is synced
but some fields presented conflicts. Sync your configuration and try again.
Check their sync status: programs, TEI\'s and events with conflicts are
marked as @.</string>
<string name="data_sync_warning_program">WARNING: Most of your data is
synced but some fields presented conflicts. Sync your configuration and try
again. If warning persists, check their sync status: programs, TEI\'s and
events with conflicts are marked as @.</string>
<string name="data_sync_error">ERROR: Some of your data failed to sync.
Check their sync status: programs with errors are marked as @, TEI\'s and
events as $.</string>
<string name="data_sync_error_program">ERROR: Some of your data failed
to sync. Check their sync status: TEI\'s and events with conflicts are
marked as $.</string>

<string name="search_result_text">%s results found</string>
<string name="add_new">Add new</string>
<string name="add">Add</string>
<string name="schedule_new">Schedule new</string>
<string name="lorem" translatable="false">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat</string>
<string name="referral">Referral</string>

<!-- Login -->
<string name="url_hint">Server url</string>
<string name="user_hint">Username</string>
<string name="password_hint">Password</string>
<string name="log_in_button">Log In</string>
<string name="unlock_button">Unlock session</string>
<string name="syncing_configuration">Syncing your
Configuration</string>
<string name="syncing_dataShortly">Your Data will shortly be
synced</string>
<string name="syncing_data">Syncing your Data</string>
<string name="configuration_ready">Your Configuration is ready</string>
<string name="data_ready">Your Data is ready</string>
<string name="configuration_sync_failed">Configuration sync
failed</string>
<string name="data_sync_failed">Data sync failed</string>
<string name="account_recovery">Account recovery</string>

<!--PERIOD-->
<string name="DAILY">Daily</string>
<string name="WEEKLY">Weekly</string>
<string name="MONTHLY">Monthly</string>
<string name="YEARLY">Yearly</string>
<string name="clear_filters">Clear filters</string>
<string name="action_accept">Accept</string>
<string name="action_share_zero_items">Share</string>

```



```

<string name="action_share_single_item">Share %d item</string>
<string name="action_share_multiple_item">Share %d items</string>

<!-- Input hints -->
<string name="search">Search</string>
<string name="enter_text">Enter text</string>
<string name="enter_long_text">Enter long text</string>
<string name="enter_number">Enter number</string>
<string name="enter_integer">Enter integer</string>
<string name="enter_negative_integer">Enter negative integer</string>
<string name="enter_positive_integer_or_zero">Enter positive integer or
zero</string>
<string name="enter_positive_integer">Enter positive integer</string>
<string name="filter_options">Filter options</string>
<string name="choose_date">Choose date</string>
<string name="change_calendar">Change calendar</string>

<!-- Status -->
<string name="program_completed">Program Completed</string>
<string name="program_inactive">Program inactive</string>

<string name="event_open">Open</string>
<string name="event_completed">Event Completed</string>
<string name="event_skipped">Event Skipped</string>
<string name="event_schedule">Event Schedule</string>
<string name="event_expired">Event Expired</string>
<string name="event_editing_expired">Editing Expired</string>
<string name="event_accepted">Event Accepted</string>
<string name="event_approved">Event Approved</string>
<string name="event_inactive">Event Inactive</string>
<string name="read_only">Read only</string>
<string name="complete">Complete</string>
<string name="finish_and_complete">Finish and complete</string>
<string name="completed">Completed</string>
<string name="cancelled">Cancelled</string>
<string name="re_open">re-open</string>
<string name="activate">activate</string>

<!-- State -->
<string name="state_synced">Sync</string>
<string name="state_to_post">To post</string>
<string name="state_to_update">To update</string>
<string name="state_to_delete">To delete</string>
<string name="state_error">Error</string>
<string name="action_sync">SYNC</string>
<string name="action_sync_sms">SYNC SMS</string>

<!--Completion-->
<string name="completion">Completion</string>

<!--Event initial-->
<string name="create">Create</string>
<string name="update">Update</string>
<string name="date">Date</string>
<string name="org_unit">Org unit</string>
<string name="no_org_units">There aren't any open Organization Units
for the selected date. Please select another date or another Organisation
Unit.</string>
<string name="category_option">Category option</string>
<string name="latitude">Latitude</string>
<string name="longitude">Longitude</string>

```

```

    <string name="new_event">New event</string>
    <string name="failed_insert_event">Failed to insert new Event</string>
    <string name="event_created">Event created</string>
    <string name="event_updated">Event updated</string>
    <string name="fake_program_stage_description"
translatable="false">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna
aliquam erat volutpat.</string>
    <string name="one_time">One time</string>
    <string name="permanent">Permanent</string>

<!--DataSet initial-->
    <string name="report_period">Report period</string>
    <string name="cat_option">Cat option</string>
    <string name="field_required">If you set a value, the whole row needs
all of values to complete the Data Set.</string>
    <string name="field_mandatory">Some values required are missing. You
need set mandatory values to complete the Data Set.</string>
    <string name="datavalue_saved">Value is saved!</string>
    <string name="there_is_no_available_period">There is no available
periods to select.</string>
<!--Event Detail-->
    <string name="event_date">Event date</string>
    <string name="due_date">Due date</string>
    <string name="cat_combo">Cat combo</string>
    <string name="date_default">DD/MM/YYYY</string>
    <string name="cancel">Cancel</string>
    <string name="delete">Delete</string>
    <string name="delete_event">Delete event</string>
    <string name="confirm_delete_event">Are you sure that you want to
delete this event?</string>
    <string name="event_was_deleted">Event deleted successfully</string>
    <string name="total">Total</string>

<!--TEI Enrollment list-->
    <string name="enrollment_list">Enrollment List</string>
    <string name="active_programs">Active programs</string>
    <string name="to_enroll_programs">Programs to enroll</string>
    <string name="enroll">Enroll</string>
    <string name="last_visit">Last Visit:</string>
    <string name="apply">apply</string>

<!--TEI DASHBOARD-->
    <string name="see_details">See Details</string>
    <string name="view_more">VIEW MORE</string>
    <string name="edit">EDIT</string>
    <string name="enrolling_ou">Enrolling OU:</string>
    <string name="follow_up">Follow up</string>
    <string name="follow_up_enabled">Follow up enabled</string>
    <string name="follow_up_disabled">Follow up disabled</string>
    <string name="deactivate">Deactivate</string>
    <string name="program_not_allow_events">This program does not allow
more events</string>

<!--DONE TASKS-->
    <string name="events">events</string>
    <string name="done_task">Home</string>
    <string name="app_info">App Info</string>
    <string name="log_out">Log Out</string>
    <string name="block_session">Block Session</string>

```

```

<!--MAP SELECTION-->
<string name="select_location">Select location</string>

<!-- SYNC -->
<string name="a_minute">1 Minute</string>
<string name="fifteen_minute">15 Minutes</string>
<string name="a_hour">1 Hour</string>
<string name="a_day">1 Day (Default)</string>
<string name="Automatic_Sync_Data">Automatic Sync Data every:</string>
<string name="Manual">Manual</string>
<string name="Automatic_Sync_Metadata">Automatic Sync Configuration
every:</string>
<string name="a_week">1 Week</string>
<string name="sync_max_parameters">Current sync parameters:</string>
<string name="sync_default_parameters">Reset to default</string>
<string name="wipe_data_meesage">Database will be deleted.\nYou will be
asked to log in again.</string>
<string name="wipe_data_ok">Accept</string>
<string name="wipe_data_no">Cancel</string>
<string name="SYNC_DATA">SYNC Data Now</string>
<string name="SYNC_META">SYNC Configuration Now</string>
<string name="SYNC_MANAGER">Settings</string>
<string name="last_data_sync_date">Last sync on: %s</string>
<string name="last_metadata_sync_date">Last sync on: %s</string>
<string name="refill_reserved_values">Refill</string>
<string name="settings_sms_module_title">SMS module:</string>
<string name="settings_sms_receiver_number">SMS gateway</string>
<string name="settings_sms_result_sender_number">SMS result
sender</string>
<string name="settings_sms_result_timeout">SMS result timeout</string>
<string name="settings_sms_result_timeout_hint">seconds</string>
<string name="settings_sms_result_sender_hint">number</string>
<string name="settings_sms_receiver_number_hint">number</string>
<string name="settings_sms_module_switch">Enable SMS
submission</string>
<string name="settings_sms_response_wait_switch">Wait for SMS result
response</string>
<!-- SEARCH TRACK ENTITY -->

<!--region SETTINCS-->
<string name="settingsSyncData">Sync data</string>
<string name="settingsSyncMetadata">Sync configuration</string>
<string name="settingsSyncParameters">Sync parameters</string>
<string name="settingsReservedValues">Reserved values</string>
<string name="settingsReservedValues_descr">Here you can see the number
of values available offline and reserve new ones.</string>
<string name="settingsErrorLog">Open sync error log</string>
<string name="settingsErrorLog_descr">This log contains all errors
caused by syncing your data and metadata.</string>
<string name="settingsDeleteLocalData">Delete local data</string>
<string name="settingsDeleteLocalData_descr">All Capture App data
stored in your device will be deleted. Data which is not synced to the
server will be lost.</string>
<string name="settingsReset">Reset app data &amp;
configuration</string>
<string name="settingsReset_descr">Your configuration and App data
stored in the device will be deleted. You will be asked to log in again and
data which is not synced to the server will be lost.</string>
<string name="settingsSms">SMS Settings</string>
<string name="settingsSms_descr">Check and edit parameters related to
the sms gateway</string>

```

```

<!--endregion-->

<string name="enrollmment_date">Enrollment date</string>

<!-- SEARCH OPTION SET -->
<string name="search_optionset">Search</string>
<!--FORM FRAGMENT-->
<string name="report_date">Report date</string>
<string name="add_file">Add file</string>
<string name="write_new_note">Write new note</string>
<string name="clear">clear</string>
<string name="add_note">Add note</string>
<string name="revert">Revert</string>
<string name="activated">Activated</string>
<string name="enroll_in">Enroll in %s</string>
<string name="missing_mandatory_fields_title">Mandatory fields are
missing</string>
  <string name="missing_mandatory_fields_text">Some mandatory fields are
missing. If you want to go back now, all saved information will be
deleted.</string>
  <string name="missing_mandatory_fields_text_not_delete">Some mandatory
fields are missing. If you want to go back, please fill them.</string>
  <string name="missing_mandatory_fields_events">Some mandatory fields
are missing. You can go back now, but this event won\'t be able to be
completed.</string>
  <string name="title_delete_go_back">Your data is not saved</string>
  <string name="delete_go_back">If you go back now all information
entered will be deleted.</string>
  <string name="missing_mandatory_fields_events_2.0">Some mandatory
fields are missing:\n%s\nYou can go back now, but this event won\'t be able
to be completed.</string>
  <string name="missing_mandatory_fields_go_back">Delete and go
back</string>
  <string name="error_fields">Some fields have errors:\n%s\nPlease, check
them.</string>
  <string name="missing_mandatory_fields_finish">Some mandatory fields
are missing. You can finish data entry and exit, but the event cannot be
completed.</string>
  <string name="enrollment_coordinates">Enrollment coordinates</string>
  <string name="tei_coordinates">Coordinates for the</string>

<!--EVENT SUMMARY-->
<string name="section_list">Section list</string>
<string name="event_summary_header" formatted="false">%s questions
of %s completed</string>
<string name="event_summary_sections_header" formatted="false">%s of %s
completed</string>
<string name="complete_and_close">Complete and close</string>

<!--AGE CUSTOM VIEW-->
<string name="hint_day">Days</string>
<string name="hint_month">Months</string>
<string name="hint_year">Years</string>
<string name="week">Weeks</string>
<string name="incident_date">Incident date</string>
<string name="local_results">Local results</string>
<string name="online_results">Online results</string>
<string name="past_programs">Past programs</string>

<!--PERMISSIONS-->

```

```

<string name="permission_denied">You don\'t have the required
permission to perform this action</string>
<string name="add_relation_button">Add as...</string>

<string name="date_dialog_clear">Clear</string>
<string name="period">Period</string>
<string name="org_unit_filter">(%d) Org Unit</string>
<string name="org_unit_select_level">Level %d</string>
<string name="tuto_main_1">Welcome to dhis2</string>
<string name="tuto_main_2">Here you can see your activities</string>
<string name="tuto_main_3">You can apply a filter...</string>
<string name="tuto_main_4">...by changing the period and selecting
dates...</string>
<string name="tuto_main_5">...or filter by Organisation Unit</string>
<string name="tuto_main_6">Check other actions on the main
menu</string>

<!-- RELATIONSHIPS -->
<string name="default_relationship_label">Create relationship</string>
<string name="next">Next</string>
<string name="search_min_num_attr">You need to enter at least %d of the
attributes to do a search</string>
<string name="search_attr">You need to enter at least one of the
attributes to do a search</string>
<string name="search_max_tei_reached">There is more than %d results.
Please enter further search criteria</string>
<string name="search_criteria_not_met">Your search criteria didn\'t
return any result.\nRevise your criteria or enroll a new %s</string>
<string name="search_init">Enter your search criteria</string>

<!-- QR -->
<string name="QR_SCANNER">Import QR data</string>
<string name="SAVE">Save</string>
<string name="button_ok">OK</string>
<string name="qr_id">Identification</string>
<string name="qr_attributes">Attributes</string>
<string name="qr_enrollment">Enrollments</string>
<string name="qr_events">Events</string>
<string name="qr_relationships">Relationships</string>
<string name="read_more_qr">Read next QR to get more info.</string>
<string name="save_qr">Stop reading and save data</string>
<string name="qr_no_data">no data</string>
<string name="qr_error_id">Invalid QR code</string>
<string name="qr_error_attr">This attribute can\'t be added</string>
<string name="qr_error_enrollment">This enrollment can\'t be
added</string>
<string name="qr_error_event">This event can\'t be added</string>

<string name="jira_report">Report Jira Issue</string>
<string name="about">About</string>
<string name="programs_to_enroll">Programs to enroll</string>
<string name="select_all">select all</string>
<string name="clear_all">clear selection</string>
<string name="empty_program_list">There is no information for the
filters selected. Please try changing them.</string>
<string name="share">Share</string>
<string name="share_qr">Share QR</string>
<string name="teiType_search_online">Search by type is not possible for
online search.\nPlease select a program from the list.</string>
<string name="max">Max</string>
<string name="current">Current</string>

```

```

    <string name="tei">TEI</string>
    <string name="limit_by_organisation_unit">Limit by organisation
unit</string>
    <string name="limit_by_program">Limit by program</string>
    <string name="programs_list">Program list</string>
    <string name="wipe_data">Reset app data & configuration</string>
    <string name="add_relationship_error">It\'s not possible to add this
relationship</string>
    <string name="warning_error_on_complete_title">Check this!</string>
    <string name="warning_on_complete_title">Warning</string>
    <string name="error_on_complete_title">Error</string>
    <string name="save">Save</string>
    <string name="reserved_value">Reserved Values</string>

    <string name="downloaded_confirm">Do you want to add more data from
QR?</string>

    <!--VIDEO-->
    <string name="video_dashboard">Video library</string>

    <!--ABOUT-->
    <string name="about_app">App Version: %s</string>
    <string name="about_sdk">SDK Version: %s</string>
    <string name="about_connected">Connected to: %s</string>
    <string name="about_user">Current user: %s</string>
    <string name="about_more">More about this app: read <a
href="https://drive.google.com/open?id=1YHRBk6LSp6uROfs19-wCfQGc-
rgl2f5iDrhCDasTR8M">here</a> what\'s supported and what\'s not
supported</string>
    <string name="about_git">This and previous versions of this
application, as well as the source code are available at <a
href="https://github.com/dhis2/dhis2-android-capture-app">GitHub</a>. DHIS2
Android Capture is licensed under the terms of the GNU General Public
License as published by the Free Software Foundation - see <a
href="http://www.gnu.org/licenses/gpl-
3.0.html">http://www.gnu.org/licenses/gpl-3.0.html</a>.</string>
    <string name="about_dev">Developed and maintained by the <a
href="https://www.dhis2.org/">DHIS2</a> project, part of University of
Oslo, <a href="https://www.mn.uio.no/ifi/english/">department of
informatics</a>.</string>
    <string name="about_contact">For more information or questions about
this application, please write to <a
href="mailto:android@dhis2.org">android@dhis2.org</a></string>

    <string name="login_error_null_pass">Password is missing. Please fill
this field.</string>
    <string name="login_error_null_username">Username is missing. Please
fill this field.</string>
    <string name="login_error_dhis_version">This server is running a dhis
version different than 2.29. Update your server version or use the testing
environment to proceed.</string>
    <string name="login_error_dhis_version_v2">This server is running a
dhis version different than %s. Update your server version or use the
testing environment to proceed.</string>
    <string name="login_error_unsuccessful_response">Something went wrong.
Please check that server url, username and password are correct and try
again.</string>
    <string name="login_error_error_response">The server could not complete
this request.</string>
    <string name="login_error_default">There has been a problem while
trying to log into the server.</string>

```

```

    <string name="login_error">Login error</string>
    <string name="missing_mandatory_fields">Mandatory fields are
missing</string>
    <string name="online_search_unexpected">Unexpected error while
searching online</string>
    <string name="online_search_parsing_error">Error while parsing
results</string>
    <string name="online_search_error">Searching online returned an
error</string>
    <string name="online_search_response_error">Something went wrong in the
server</string>
    <string name="search_program_not_selected">Select a program to enable
enrolling</string>
    <string name="search_access_error">You don\'t have the requiered
permission</string>
    <string name="feature_unavaible">This functionality is not ready
yet.</string>
    <string name="relationship_label_error">Error getting relationship
label</string>
    <string name="all_tei_type">All %s</string>
    <string name="invalid_phone_number">This is not a valid phone
number</string>
    <string name="invalid_email">This is not a valid email</string>
    <string name="invalid_negative_number">Only negative numbers are
allowed</string>
    <string name="invalid_possitive_zero">Only positive numbers or zero
allowed</string>
    <string name="invalid_possitive">Only positive numbers are
allowed</string>
    <string name="invalid_interval">Only values from 0 to 1 are
allowed</string>
    <string name="invalid_percentage">Only values from 0 to 100 are
allowed</string>
    <string name="date_dialog_tigle">Select dates to filter</string>
    <string name="select_option">Select option</string>
    <string name="issue_reported">Issue reported</string>
    <string name="jira_credential_error">Check that your jira user and
password are correct</string>
    <string name="jira_issue_error">Error reporting the issue</string>
    <string name="jira_fields_error">All fields are mandatory</string>
    <string name="dialog_generate_new_event">Generate new event</string>
    <string name="message_generate_new_event">Do you want to create another
event?</string>
    <string name="event_completed_title">Events Completed</string>
    <string name="event_completed_message">All events in this program are
completed. Would you like to close the program as well?</string>
    <string name="complete_enrollment_message">Would you like to complete
the enrollment as well?</string>
    <string name="check_event">Check</string>

    <string name="menu_home">Home</string>
    <string name="appwidget_text">EXAMPLE</string>
    <string name="add_widget">Add widget</string>
    <string name="showHelp">Show help</string>
    <string name="tuto_settings_1">Synchronize your data...</string>
    <string name="tuto_settings_2">...your configuration...</string>
    <string name="tuto_settings_3">...set how many records to download from
server...</string>
    <string name="tuto_settings_reserved">Check and manage how many
reserved values are left</string>

```

<string name="tuto\_settings\_errors">Check if a connection to the server failed</string>

<string name="tuto\_settings\_reset">Free some disk space</string>

<string name="tuto\_settings\_4">...or delete everything to start over.</string>

<string name="tuto\_search\_1\_v2">Find your entities both in your local device and the online server.</string>

<string name="tuto\_search\_2">You can switch to other programs</string>

<string name="tuto\_search\_3\_v2">Search or Add new entities. You need to search before you can add a new one!</string>

<string name="tuto\_search\_4\_v2">Click here to clean the fields and restart the search.</string>

<string name="tuto\_dashboard\_1">This is the entity dashboard</string>

<string name="tuto\_dashboard\_2">See entities profile information...</string>

<string name="tuto\_dashboard\_3">...Share information via QR or SMS...</string>

<string name="tuto\_dashboard\_4">...Or mark the entity for follow-up</string>

<string name="tuto\_dashboard\_5">Click to create or schedule an event or refer to another Org Unit.</string>

<string name="tuto\_dashboard\_6">Here you will see all events. Click to see an event detail.</string>

<string name="tuto\_dashboard\_7">Navigate through the menu to add relationships, see indicators or write notes.</string>

<string name="tuto\_dashboard\_8">Click to see all enrollments of your entity or create a new one.</string>

<string name="tuto\_tei\_event\_1">Click to delete this event. This action can not be undone.</string>

<string name="tuto\_tei\_event\_2">Complete the event.</string>

<string name="tuto\_program\_event\_1">All events related to this program will be listed here.</string>

<string name="tuto\_program\_event\_2">Click to add a new event</string>

<string name="tuto\_event\_initial\_new\_1">Enter all data to create the event</string>

<string name="tuto\_event\_initial\_1">Click to check the event completion status.</string>

<string name="tuto\_event\_summary">Click to complete or re-open an event</string>

<string name="tuto\_main\_7">Read and store information from other devices via QR</string>

<string name="tuto\_main\_8">Synchronize your data with the server</string>

<string name="tuto\_main\_9">Block session with a 4 digit code. Do not forget your code!</string>

<string name="tuto\_main\_10">Log out to change user or server. You will be able to log in again even if there is no internet connection.</string>

<string name="text\_plus\_colon">%s:</string>

<string name="relationships">relationships</string>

<string name="empty\_relationships">There are no relationships, click "+" to add a new one.</string>

<string name="empty\_tei\_add">There are no events registered, click "+" to add a new one.</string>

<string name="empty\_tei\_no\_add">There are no events registered.</string>

<string name="qr\_data\_values">Data values</string>

<string name="info">Info</string>

<string name="qr\_no\_id\_error">Could not read this Id</string>

<string name="delete\_local\_data">Delete local data</string>

<string name="dashboard\_overview">Overview</string>

<string name="dashboard\_relationships">Relationships</string>



<string name="dashboard\_indicators">Indicators</string>  
 <string name="empty\_indicators">There are no indicators for this  
 program.</string>  
 <string name="dashboard\_notes">Notes</string>  
 <string name="unable\_to\_complete">Unable to complete</string>  
 <string name="field\_errors">Some fields have errors. Please, check them  
 to be able to complete</string>  
 <string name="unselect\_enrollment">All enrollments</string>  
 <string name="unselect\_enrollment\_description">Show dashboard with all  
 active programs</string>  
 <string name="CHECK\_SYNC\_ERRORS">Open sync errors log</string>  
 <string name="error\_dialog\_title">Sync error log</string>  
 <string name="share\_with">Share with...</string>  
 <string name="errors\_during\_sync">Your configuration sync has failed.  
 There could be errors in your server. Click here for more  
 information.</string>  
 <string name="sectionSelectorPrev">PREV</string>  
 <string name="sectionSelectorNext">NEXT</string>  
 <string name="event\_overview">Details</string>  
 <string name="event\_event">Event</string>  
 <string name="event\_indicators">Indicators</string>  
 <string name="event\_notes">Notes</string>  
 <string name="complete\_add\_new"><![CDATA[Complete & add new]]></string>  
 <string name="complete\_later">Complete later</string>  
 <string name="finish\_add\_new"><![CDATA[Finish & add new]]></string>  
 <string name="finish">Finish</string>  
 <string name="error\_fields\_title">Fields with errors</string>  
 <string name="error\_fields\_events">Some fields has errors. They have  
 not been saved. Please, check them.</string>  
 <string name="event\_reopened">This event can now be edited</string>  
 <string name="unique\_warning">The value should be unique, a coincidence  
 has been found. It has not been saved. Please check it.</string>  
 <string name="please\_wait">Please wait</string>  
 <string name="skip">Skip</string>  
 <string name="reschedule">Reschedule</string>  
 <string name="event\_was\_skipped">The event was marked as  
 skipped</string>  
 <string name="no\_intructions">There is no information for this  
 screen</string>  
 <string name="event\_overdue">Overdue</string>  
 <string name="unsupported\_program\_rule">There is one program rule which  
 is not supported. Please check the documentation</string>  
 <string name="delete\_event\_error">This event has not been created  
 yet</string>  
 <string name="empty\_description">No info for this field</string>  
 <string name="biometrics\_dialog\_title">Empty Credentials</string>  
 <string name="biometrics\_first\_use\_text">To be able to log in using the  
 fingerprint scanner you first need to use the default way</string>  
 <string name="biometrics\_security\_title">Biometric Security</string>  
 <string name="biometrics\_security\_text">Use your fingerprint to log in  
 with this user?</string>  
 <string name="tab\_tables">Tables</string>  
 <string name="table">Table</string>  
 <string name="login\_https">https://</string>  
 <string name="delete\_local\_data\_message">Non synchronized data will be  
 lost. Do you really want to continue?</string>  
 <string name="local\_data\_accept">Accept</string>  
 <string name="delete\_local\_data\_error">Data could not be deleted.  
 Please retry.</string>  
 <string name="delete\_local\_data\_done">Delete local data finished  
 successfully.</string>

```

    <string name="org_unit_selection_warning">At least one organisation
unit must be selected</string>
    <string name="send_user_name_title">Send data</string>
    <string name="send_user_name_message">Help DHIS2 improve this app by
sharing error logs associated to your user id.</string>
    <string name="check_mandatory_field">Check fields</string>
    <string name="error_no_network_during_sync">You lost your internet
connection while performing your configuration sync. Please try again
later</string>
    <string name="action_agree">Agree</string>
    <string name="login_server_info_title">Server url</string>
    <string name="login_server_info_message">"Enter the DHIS 2 server url
for your organisation as provided by your system
administrator.\n\nAlternatively, you can scan a QR code containing the URL
of your server"</string>
    <string name="forgot_your_code">Forgot your code?</string>
    <string name="event_form_end_options_title">What would you like to do
for this event?</string>
    <string name="sectionSelectorEnd">END</string>
    <string name="resource_not_found">%s not found</string>
    <string name="relationship_not_found_message">The record you are trying
to access is not available. Would you like to search for it?</string>
    <string name="sync_by_sms">Synced by SMS</string>
    <string name="state_warning">Warning</string>
    <string name="no_conflicts_message">There are no conflicts for this
program</string>
    <string name="network_unavailable_sms">Network connection is not
available, but it looks like there is phone services - you can use
SMS.</string>
    <string name="network_unavailable">Network connection is not
available.</string>
    <string name="action_close">Close</string>
    <string name="tuto_main_11">Now you can check sync information by
clicking here</string>
    <string name="no_conflicts_update_message">This data is stored in your
device only, it is not synced with the server.</string>
    <string name="no_conflicts_synced_message">You are up to date! All your
data is sent to the server.</string>
    <string name="relationship_error_recursive">You can\'t relate a record
with itself</string>
    <string name="unblock_session">Unblock session</string>
    <string name="data_set_closed">Close</string>
    <string name="data_set_open">Open</string>
    <string name="unsupported_value_type">Unsupported value type</string>
    <string name="change_calendar_view">Change calendar view</string>
    <string name="data_every_fifteen_minutes">15 minutes</string>
    <string name="data_every_hour">Every hour</string>
    <string name="data_every_day">Daily</string>
    <string name="data_every_week">Weekly</string>
    <string name="period_week">Week</string>
    <string name="sync_setting">Synchronization setting: %s</string>
    <string name="relationship_without_enrollment">The record you are
trying to access has no enrollment. You can\'t access the
dashboard.</string>
    <string name="formatting_error">Formatting error</string>
    <string name="dashboard_menu_delete_tei">Delete TEI</string>
    <string name="dashboard_menu_delete_enrollment">Delete enrollment
only</string>
    <string name="delete_authority_error">You do not have the authority to
delete this record</string>

```

```

    <string name="record_marked_for_deletion">This record is marked for
deletion. You can\'t access it.</string>
    <string name="validation_url">This is not a valid url</string>
    <string name="uncomplete_authority_error">You do not have the authority
to re-open this event</string>
    <string name="adjusting_form">Adjusting form...</string>

    <string name="sms_receiver_number">Receiver number</string>
    <string name="sms_amount_question">Do you accept to send %d
messages?</string>
    <string name="sms_sending">Sending %1$d/%2$d</string>
    <string name="sms_waiting_amount_confirm">Waiting amount confirmation
- %d</string>
    <string name="sms_all_sent">All sent</string>
    <string name="sms_submission_completed">Submission completed</string>
    <string name="sms_waiting_confirmation">Waiting for confirmation
SMS</string>
    <string name="sms_waiting_timeout">Not received confirmation
message</string>
    <string name="sms_confirmed">Confirmation received</string>
    <string name="sms_error">Error sending</string>
    <string name="sms_error_item_not_ready">Data item not ready to be
sent</string>
    <string name="sms_error_no_network">No network</string>
    <string name="sms_error_no_check_network_permission">No check network
permission</string>
    <string name="sms_error_no_receive_sms_permission">No receive sms
permission</string>
    <string name="sms_error_no_send_sms_permission">No send
permission</string>
    <string name="sms_error_no_gateway_set">No gateway set</string>
    <string name="sms_error_no_user_login">User not logged in</string>
    <string name="sms_error_module_disabled">SMS module is
disabled</string>
    <string name="sms_error_ongoing_submission">Another SMS submission
ongoing</string>
    <string name="sms_metadata_empty">SMS module not initiated, not found
required metadata</string>
    <string name="sms_completed">Completed</string>
    <string name="sms_count_error">SMS count not accepted</string>
    <string name="sms_state_started">Started sms submission</string>
    <string name="sms_state_converted">Data read and converted. Submission
id: %d</string>
    <string name="sms_general_error">Error sending sms submission</string>
    <string name="sms_title_enrollment">SMS TEI registration and
enrollment</string>
    <string name="sms_title_event">SMS event submission</string>
    <string name="sms_title_data_set">SMS data set submission</string>
    <string name="sms_bar_state_starting">Starting</string>
    <string name="sms_bar_state_sending">In progress</string>
    <string name="sms_bar_state_sent">Sent</string>
    <string name="sms_bar_state_failed">Failed</string>

    <string name="start_sync_granular">Starting synchronization.</string>
    <string name="syncing">Synchronizing...</string>
    <string name="end_sync_granular">Synchronization completed.</string>
    <string name="error_sync">Error has happened during
synchronization.</string>
    <string name="cancel_sync">Cancelled synchronization.</string>

    <string name="copied_text">Copied</string>

```

```

<string name="image_selected">Image selected</string>
<string name="show_in_map">Show in map</string>
<string name="show_list">Show list</string>

<!--region TRACKER_IMPORT_CONFLICT_MESSAGES-->
<string name="value_required_but_not_provided">%s is mandatory but has
not been provided.</string>
<string name="polygon_captured">Polygon captured</string>
<string name="polygon">Polygon</string>
<string name="map_layers">Map layers</string>
<!--endregion-->

<string name="wrong_pattern">It is not a correct format.</string>
<string name="fingerprint_title">Fingerprint</string>
<string name="fingerprint_message">Please, put your finger in your
fingerprint reader.</string>
<string name="security_title">Security</string>
<string name="security_rooted_message">For security reasons, this
version of the app is not allow to be used in rooted devices. Please use
the training version.</string>
<string name="fix_error">There are some errors. Please fix them before
continuing.</string>

<string name="filters_active">Filters applying</string>
<string name="filters_disabled">No filters applied</string>
<string name="filters_title_period">Period</string>
<string name="filters_title_org_unit">Org. Unit</string>
<string name="filters_title_status">Status</string>
<string name="filters_title_state">State</string>
<string name="filter_period_today">Today</string>
<string name="filter_period_yesterday">Yesterday</string>
<string name="filter_period_tomorrow">Tomorrow</string>
<string name="filter_period_this_week">This week</string>
<string name="filter_period_last_week">Last week</string>
<string name="filter_perdior_next_week">Next week</string>
<string name="filter_period_this_month">This month</string>
<string name="filter_period_last_month">Last month</string>
<string name="filter_period_next_month">Next month</string>
<string name="filter_period_form_to">From - To</string>
<string name="filter_period_other">Other</string>
<string name="filter_period_any">Anytime</string>
<string name="data_values_error_sync_message">Sync failed. Sync failed.
Check the data values to try to fix the problem and try again. If it keeps
failing contact your system administrator</string>

```

## Code, remote video repository:

### index.js:

*Whole file by us.*

```
const express = require('express');
```

```
const http = require('http');
```

```

const app = express();
const fs = require('fs');
const uuidv4 = require('uuid/v4');

app.use(express.static('src'));

function readData() {
  const rawData = fs.readFileSync("videos.json");
  return JSON.parse(rawData);
}

let videos = readData();
app.listen(8081);

function saveVideos(){
  const rawData = JSON.stringify(videos);
  fs.writeFileSync("videos.json", rawData);
}

function addVideo(filename, title, langs, description, version, thumbFileName, tags) {
  const uuid = uuidv4();
  let languages = [];
  for (i = 0; i < langs.length; i++){
    languages.push({"languageName": langs[i], "index": "" + i});
  }
  videos.push({
    "uid": uuid,
    "fileName": filename,
    "videoTitle": title,
    "videoLanguages": languages,

```

```

    "description": description,
    "version": version,
    "thumbnailFileName": thumbFileName,
    "tags": tags
  });
  saveVideos();
}
function getVideoInfo(uid) {
  for (i = 0; i < videos.length; i++){
    if (videos[i].uid === uid) {
      return videos[i];
    }
  }
}
//statically serve files.
app.get('/video/:videoid/info', function(req, res){
  const videoInfo = getVideoInfo(req.params.videoid);
  if (videoInfo) {
    res.send(videoInfo);
  } else {
    res.send({"error": "Video id not found"});
  }
})

app.get('/video/:videoid', function(req, res){
  const videoInfo = getVideoInfo(req.params.videoid);
  if (videoInfo) {
    const file = `${__dirname}/src/videos/${videoInfo.fileName}`;
    res.download(file);
  }
})

```

```

    } else {
      res.sendStatus(404);
    }
  })
  app.get('/thumbnail/:videoid', function(req, res){
    const videoInfo = getVideoInfo(req.params.videoid);
    console.log(videoInfo);
    if (videoInfo) {
      const file = `${__dirname}/src/thumbnails/${videoInfo.thumbnailFileName}`;
      res.download(file);
    } else {
      res.sendStatus(404);
    }
  })

```

```

app.get('/videos', function(req, res){
  res.send(videos);
})

```

//root, API index.

```

app.get('/', function(req, res){
  res.send({"test": "hello"});
});

```

## **videos.json:**

*Whole file by us. Example of video.json file.*

```

[
  {

```

```
"uid": "e6c0ccd0-074b-447a-9b4b-5b74f6802de2",
"fileName": "mafVideo1.mp4",
"videoTitle": "Brestfeeding: Food for Life",
"videoLanguages": [
  {
    "languageName": "English",
    "index": 0
  }
],
"description": "Health promotion video about brestfeeding and reasons why it should
be done.",
"version": "1",
"thumbnailFileName": "mafVideo1.png",
"tags": "baby, nutrition, newborn"
},
{
  "uid": "80d80b32-8f55-4ae4-a41a-426c0bbe67a8",
  "fileName": "mafVideo2.mp4",
  "videoTitle": "Warning signs in Pregnancy",
  "videoLanguages": [
    {
      "languageName": "English",
      "index": 0
    }
  ],
  "description": "Information video about danger signs in pregnancy, and how to deal
with them",
  "version": "1",
  "thumbnailFileName": "mafVideo2.png",
  "tags": ""
```



```

},
{
  "uid": "176a1751-0245-45e8-8036-9f57011fc15d",
  "fileName": "mafVideo3.mp4",
  "videoTitle": "Neonatal Resuscitation",
  "videoLanguages": [
    {
      "languageName": "English",
      "index": 0
    }
  ],
  "description": "Informational video on neonatal resuscitation",
  "version": "1",
  "thumbnailFileName": "mafVideo3.png",
  "tags": "baby, newborn, CPR"
},
{
  "uid": "3f246473-6abb-4f77-992c-4f63fe717ba3",
  "fileName": "mafVideo4.mp4",
  "videoTitle": "What and when to feed your child (6 months to 2 years)",
  "videoLanguages": [
    {
      "languageName": "English",
      "index": 0
    }
  ],
  "description": "Health promotion video on nutrition for children (6 months to 2 years)",
  "version": "1",
  "thumbnailFileName": "mafVideo4.png",

```

```

    "tags": "baby, food, feed "
  },
  {
    "uid": "878b88d2-9ecb-43c9-9da6-9f45bbb561d2",
    "fileName": "mafVideo5.mp4",
    "videoTitle": "How to care for a newborn",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ],
    "description": "Health promotion video on what to do and how to care for newborn
    baby.",
    "version": "1",
    "thumbnailFileName": "mafVideo5.png",
    "tags": ""
  },
  {
    "uid": "90d23f83-dd12-4a56-a556-4cf55bf43db4",
    "fileName": "WHO__In our hands_ - HIV self-testing in Malawi.mp4",
    "videoTitle": "HIV self-testing-kits",
    "videoLanguages": [
      {
        "languageName": "Chechewa",
        "index": 0
      }
    ],
    "description": "Health promotion video on HIV self-testing kits.",
    "version": "1",

```

```

    "thumbnailFileName": "hivSelfTest.png",
    "tags": "AIDS, NRTI, ART"
  },
  {
    "uid": "4a7c573a-32a5-4166-a8b4-e0ebc2736385",
    "fileName": "mafVideo1.mp4",
    "videoTitle": "Brestfeeding: Food for Life (Duplicate)",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ],
    "description": "Duplicated to extend list! Health promotion video about brestfeeding
and reasons why it should be done.",
    "version": "1",
    "thumbnailFileName": "mafVideo1.png",
    "tags": "baby, nutrition, newborn"
  },
  {
    "uid": "04ccec1a-af07-4e3e-a079-091cc1ebe472",
    "fileName": "mafVideo2.mp4",
    "videoTitle": "Warning signs in Pregnancy (Duplicate)",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ],
  },

```

```

    "description": "Duplicated to extend list! Information video about danger signs in pregnancy, and how to deal with them",
    "version": "1",
    "thumbnailFileName": "mafVideo2.png",
    "tags": ""
  },
  {
    "uid": "b4bc6f87-d9b8-48f3-9c76-54fa4ac103c6",
    "fileName": "mafVideo3.mp4",
    "videoTitle": "Neonatal Resuscitation (Duplicate)",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ],
    "description": "Duplicated to extend list! Informational video on neonatal resuscitation",
    "version": "1",
    "thumbnailFileName": "mafVideo3.png",
    "tags": "baby, newborn, CPR"
  },
  {
    "uid": "3850fbeb-fd31-4567-868a-dd46942879c5",
    "fileName": "mafVideo4.mp4",
    "videoTitle": "What and when to feed your child (6 months to 2 years) (Duplicate)",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ]
  }

```

```

    ],
    "description": "Duplicated to extend list! Health promotion video on nutrition for
children (6 months to 2 years)",
    "version": "1",
    "thumbnailFileName": "mafVideo4.png",
    "tags": "baby, food, feed "
  },
  {
    "uid": "ea141493-e65d-499d-b695-b5d96f198171",
    "fileName": "mafVideo5.mp4",
    "videoTitle": "How to care for a newborn (duplicate)",
    "videoLanguages": [
      {
        "languageName": "English",
        "index": 0
      }
    ],
    "description": "Duplicated to extend list! Health promotion video on what to do and
how to care for newborn baby.",
    "version": "1",
    "thumbnailFileName": "mafVideo5.png",
    "tags": ""
  },
  {
    "uid": "41cb2652-0cc9-420b-a821-3c8250827ae2",
    "fileName": "WHO__In our hands_ - HIV self-testing in Malawi.mp4",
    "videoTitle": "HIV self-testing-kits (Duplicate)",
    "videoLanguages": [
      {
        "languageName": "Chechewa",

```

```

        "index": 0
    }
],
    "description": "Duplicated to extend list! Health promotion video on HIV self-testing
kits.",
    "version": "1",
    "thumbnailFileName": "hivSelfTest.png",
    "tags": "AIDS, NRTI, ART"
},
{
    "uid": "96c66497-01ff-4288-9850-f85b1c661649",
    "fileName": "langTest.mp4",
    "videoTitle": "Language test video",
    "videoLanguages": [
        {
            "languageName": "Norwegian",
            "index": "0"
        },
        {
            "languageName": "English",
            "index": "1"
        }
    ],
    "description": "Language test video (long)",
    "version": "1",
    "thumbnailFileName": "mafVideo1.png",
    "tags": ""
}
]

```