Tampere University

Raisa Turpeinen

OPEN SOURCE TELEMEDICINE AND LOW SIGNAL CONDITIONS:

An overview of current software

Faculty of Information Technology and Communication Sciences
M. Sc. Thesis
December 2023

ABSTRACT

Raisa Turpeinen: Open source telemedicine and low signal conditions: An overview of current software
M.Sc. Thesis
Tampere University
Master's Degree Programme in Human-Technology Interaction
December 2023

This thesis investigates current telemedicine open source software (OSS) from the perspective of how well suited they are for use in remote, low signal settings such as rural India. There is a need for better access to healthcare in these settings, and digital services especially could be the answer by making communication between patients and health care workers (HCW) easier. Additionally, easy communication between different levels of HCWs is vital in order to bring the necessary variety of expertise to these remote locations, which would further improve the quality of care for patients.

Finding a suitable OSS base for a MomConnect and Neuvola -inspired telemedicine application intended for pregnancy and mother-and-child healthcare is a primary focus of this thesis and it is conducted in parallel to a larger research project, in order to inform its approach regarding the application. To achieve its goal this thesis seeks out and analyzes a selection of the current (2022-2023) open-source healthcare software, and evaluates their suitability for the purposes of the main project. Considerations such as can any of that pre-existing software be used without modification in the intended research setting, or will a new application need to be developed to meet the requirements will be at the heart of the discussion chapter.

During the analysis process this thesis aims to answer the following research questions: 1. What telemedicine OSS are currently available? 2. What are some necessary qualities for a mobile application to function in low signal environments?

The findings of this thesis will provide insight into currently available OSS and their suitability to India's low signal rural areas.

Key words and terms: telemedicine, OSS, low signal compatibility, rural settings.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Contents

1. Introduction

In the recent years since the start of the pandemic the implementation of telemedicine has become increasingly common and relevant to everyone regardless of demographics. It has been a way to work with the safety regulations and still provide quality care and services for those who need them, presenting a remote alternative to traditional healthcare services. Remote access can be especially beneficial to people living in areas where in-person healthcare can be a challenge to access due to long distances and the cost of travel. Especially such patient groups as expecting mothers may need continuous monitoring to ensure a safe pregnancy but can be limited by their condition to refrain from frequent trips to a distant hospital and then back again. More so than well-equipped hospitals, smartphones tend to be widely available even in remote low resource settings and this is how telemedicine becomes an important aspect of the solution to bringing accessible healthcare to those who need it.

This thesis investigates and analyzes recent open-source telemedicine applications and it is conducted in parallel to the research project "*Saving newborn and maternal life in remote "aspirational districts" in context of Covid-19 in India*" (Atkins et al., 2022) by Salla Atkins, Vinod Diwan, Kalpana Mahadik and Ashish Pathak. Following the needs of the research project, this thesis is specifically focused on finding suitable applications to aid the health care workers (HCW) in mother-and-child care in the context of rural India. It takes a look at currently available open-source software (OSS) in an attempt to find the best suited option to aid with communication and health care services in this setting. As a secondary area of interest, an effort will also be made to map some central requirements for mobile applications to work reliably in low signal conditions.

Similar overviews of currently available open source applications have been conducted before (Kiah et al., 2014) but none in recent years. They cover many of the same applications from the viewpoint of the implementer, while this study reviews these applications from the angle of their apparent compatibility with remote and low resource settings, and in this way provides new insight into the field of current OSS. These goals will be approached by studying the end-users and the use environment, selecting suitable mobile applications for analysis, conducting the analysis and then discussing the results. This thesis covers this undertaking and each of its stages, respectively in the following chapters.

2. Background

This chapter provides terminology that will be helpful in understanding the topic matter at hand. The methodology of the study and the criteria definitions can be found in chapter 3.

2.1 Parallel research project

This thesis is written to complement an international research project called "*Saving newborn and maternal life in remote "aspirational districts" in context of Covid-19 in India*", by Salla Atkins, Vinods Diwan, Kalpana Mahadik and Ashish Pathak. (Atkins et al., 2022)

The project aims to address access to care and perceptions of care in the chosen setting through empowering pregnant or postnatal women and their households through mobile services, and it also aims to address quality of care by health workers through mentoring, supportive supervision and accurate, up-to-date information through mobile applications. These goals will be approached by implementing an open source mobile application that will help the healthcare providers and patients to have better communication and more information available to them. The resulting application is meant to have a similar concept as MomConnect in South Africa or Neuvola services in Finland – to aid in mother-and-child care during and after pregnancy. The project targets India's rural areas and their populations and the application needs to adjust to the needs of the environment and the target user group.

The goal of this thesis is to give an overview of the current landscape of telemedicine OSS available and analyze a section of it. The results of this analysis will help to inform the approach for the application selection part of the research project.

2.2 Telemedicine

Telemedicine – or telehealth – refers to the use of electronic information and communications technologies to provide and support health care when distance separates the participants. (Institute of Medicine (US), 1996) This can include both treatment and diagnoses, and it has become much more commonplace all over the world in the recent decade.

Telemedicine can help healthcare providers in many ways: in addition to evaluating and diagnosing patients over distances, it can be used to efficiently and securely store, access and manage all kinds of medical and patient data; for communication between HCWs and HCW and patient; for monitoring patients; as an aid in clinical decision-making; and as a tool for long distance medical education and training. Telemedicine allows smaller clinics to borrow the expertise of HCWs in larger hospitals and thus making it possible to give more specialized care even in remote locations that otherwise would not be able to provide it. Physical data storage is tied to a single location, takes up a lot of space and is a lot of extra work to maintain and consult, taking up significant amounts of the HCWs' time away from patient care.

Healthcare in rural and remote settings has special needs compared to more densely populated areas, and the use of telemedicine can be the answer to resolving many of these needs. These needs generally have to do with long travel distances and inaccessibility of specialized services, and long travel distance especially can often translate into issues of the patient not being able to afford visits, either in terms of money or time. In some cases these issues can become the deciding factor in the patient being able to receive the kind of specialized care they need. With help from telemedicine much of the variety and quality of care offered by large hospitals can be brought to patients everywhere regardless of location, thus avoiding the abovementioned issues and also providing the local HCWs with more opportunities for learning and consulting as needed. Therefore using telemedicine as tool in healthcare benefits both the health care provider and the patient and it should be implemented in such settings where it is feasible to do so.

There are some challenges in making this shift to digitalized healthcare, which will be of interest in this study as they are relevant to the context setting and complicate the selection process for a suitable piece of software to use. The implementation of telemedicine in remote locations and especially in undeveloped countries can face a lack of sufficient hardware both for the server equipment, cabling systems, and the HCWs' personal devices. This can mean a complete lack of devices, or a lack of more modern and powerful devices that may be required for certain features such as live video calls, or other resource-heavy actions.

However, in spite of these challenges telemedicine has already been approved and adopted into use in numerous countries worldwide even before the Covid-19 pandemic, and it is to be hoped that this will be the direction of development all over the world to ensure more accessible health care for all. (Senbekov et al., 2020) (Ventola, 2014)

2.3 Open source software

Open source is a software philosophy and a type of software license that determines the software to be freely redistributable and allows the modification and any derivative works based on it, and its source code is available for free or a very low cost to anyone who wants to access it. It is in these terms defined by the Open Source Initiative (OSI) (Open Source Initiative), which is one of the leading two communities advocating for open-source methodology. The other one is the Free Software Foundation (FSF) (Free

Software Foundation) and it emphasizes the programmer’s freedom to run the software, to study and edit it, and to redistribute it as they wish. OSI is more concerned with the implementation of open source in the corporate world, and therefore more relevant here as the healthcare industry can be a big market for software. (AlMarzouq et al., 2005)

OSS has many benefits that include but are not limited to: low cost compared to commercial options as there are no associated licensing fees; no lock-in with a specific company or piece of software; the source code is available to all users; OSS can be customized as desired, as opposed to proprietary software which typically does not allow this; projects are highly community-focused and support for the application may be found for free through the community; there are many people working on the code and it is constantly being peer-reviewed, which in theory leads to good reliability and issues getting noticed and fixed quickly.

On the other hand, there are certain challenges as well that are sometimes associated with OSS: it can be challenging to discover some OSS and to know which option is the best fit for your needs; there often are more features available in proprietary software than open source; there is a lack of thorough documentation for many OSS; OSS often requires more in-house knowledge and skill to install and maintain the software manually, compared to proprietary options. (Trappler, 2009)

The choice to use OSS has already been made for this study through the parallel research project so it will not be analyzed in great detail here, but it is valuable to be aware of the advantages and shortcomings of the choice nonetheless when building a framework for the analysis.

2.4 End-users and the setting

India is chosen as the setting for this study through the premise of the parallel research project, and this will dictate who its end-users are and which aspects of analysis will require special care and attention. In addition to knowing the target users, any specific differences relevant to this use environment, as well as the local networks and available hardware are other important factors to consider. This section gives a brief overview of some considerations specific to India in relation to the interests of this study.

The laws regarding health care and software related matters in India may differ from those in the EU and the USA and that is important to take into consideration, but the details go beyond the primary focus of this study. It will simply be acknowledged as a factor to be considered when developing an application intended for public and international use.

When designing a healthcare application the general assumption is that it is for the use of normal people and should not require the most recent and optimized technology from its user. Rather, it should be usable even on more modest devices to ensure accessibility of service to all. It is therefore important to consider the question of what kind of hardware is the end-user likely to have access to, and then design the application accordingly. Fortunately for this study, smart phones and other mobile devices are increasingly common in India, as evidenced by Telecom's mobile subscription data report from 2023. (Institute of Medicine (US), 1996)

International OSS usually are by default available in English, but in India's case English may not be sufficient to reach all the intended users. Hindi and English are the official languages of India, but there are over 60 native languages used accross the geographic area that constitutes modern day India. In fact, 22 out of those 60 are recognized as official regional languages, which implies they all have a very significant number of speakers. With such a wide variety of languages, English alone should not be assumed to be sufficient for this application and instead the application should have suitable localization options to make it more accessible to those who do not speak English fluently.

Additionally to the obstacle of the language barrier, India has an approximate literacy rate of 70-80% out of the entire population. The corresponding percentage is on the lower end of that range for women and on the higher end for men, and there are significant regional differences as well – the more rural areas have lower literacy rates than the national average. In this light it is reasonable to expect that a significant section of the intended user base for this application is illiterate. It is important that service is provided for this user group as well, and it is a challenge that anyone designing an application for this setting must take on to make their application illiterate user friendly. (Shah, 2013)

2.5 Mobile Networks in India

Opensignal (Opensignal) provides public reports on wireless and broadband operators and network experience worldwide. Their customer-centric regional reports include information about the availability of mobile networks, their consistent performance, as well as download and upload speeds. Most of Opensignal's parameters have the word *experience* in their name, and this means they aim to represent the status of a typical user experience with their testing, rather than the optimal performance.

Some of their parameters are self-explanatory, but some are more opaque in what exactly they are used to measure. An example of this are the excellent and core consistent quality parameters. They measure how well applications are likely to perform in the networks and these scores are calculated using six indicators: download speed, upload speed, latency, jitter, packet loss, and time to first byte. The scores mean the percentage of user tests related to these six indicators that reached the minimum performance tresholds. Core consistent quality measures common but less demanding use of an application, such as standard video streaming, web browsing, email use and photo sharing. Excellent consistent quality measures more demanding use cases such as live video streaming and streaming video games.

The mobile network report from India (Opensignal, 2022a) illustrates the recent performance situation of local mobile networks there. There is the option to view either the national analysis or a regional analysis, and each one is organized into three categories called consistency, coverage and overall experience, and each category is displayed by operator. Some scores are not important for this study, but a summary of the more relevant information can be found below in table 2.1. The included parameters show, for example, the networks' ability to support different communication functions such as voice and video.

Mobile Network Experience	Airtel	BSNL	Jio	Vi	
Video Experience	41.0	15.4	40.4	38.9	0-100 points
Voice App Experience	77.4	67.7	77.2	76.4	0-100 points
Download Speed Experience	13.6	3.0	12.9	13.2	Mbps
Upload Speed Experience	4.1	1.1	3.9	4.7	Mbps
Availability	98.5	82.1	99.4	95.4	% of time
Coverage Experience	8.0	2.6	9.5	6.3	0-10 points
Excellent Consistent Quality	54.2	12.2	62.6	48.5	% of tests
Core Consistent Quality	76.9	37.0	85.7	69.1	% of tests

Table 2.1: India in October 2022

The stability and availability of the mobile networks in India seem to be highly region and operator dependent. This analysis does not consider region in detail as no specific region was selected by the parallel research project. However, on the national analysis the choice of operator already causes significant variation in coverage and consistency of mobile networks; for example, coverage experience score can go from 2.6 to 9.5 by switching from the operator BSNL to Jio. Similarly, core consistent quality would go from 37.0% to 85.7% of tests. Overall, BSNL is the lowest performing operator in every single category, while Jio is the highest performing in more categories than any of the other operators. Excellent core consistency is at worst at 12.2% and at best it is at 62.6%, and core consistent quality ranges from 37.0% to 85.7%. The difference between the highest and lowest performing operators is staggering and raises the question of which user case should a healthcare application be designed for.

Next compare to the report from Finland (Opensignal, 2022b) in table 2.2:

Mobile Network Experience	DNA	Elisa	Telia	
Video Experience	64.5	64.9	66.2	0-100 points
Voice App Experience	82.3	81.9	82.1	0-100 points
Download Speed Experience	54.5	51.6	46.0	Mbps
Upload Speed Experience	15.8	12.2	13.7	Mbps
Availability	98.8	99.3	99.0	% of time
Excellent Consistent Quality	92.0	90.9	91.6	% of tests
Core Consistent Quality	96.4	97.2	97.3	% of tests

Table 2.2: Finland in August 2022.

The scores are quite different, according to these reports the recent overall network performance in Finland is clearly on a higher and more consistent level than that in India. There are some comparable statistics, such as the the voice app experience and availability. From this it follows that voice related communication should work similarly to what could be expected in Finland, but little else will and that has to be taken into consideration when designing an application for such a different setting.

2.6 Low Signal Conditions

Low signal conditions are a major resource limitation affecting the premise of this study, and therefore it warrants a closer look. Here it refers to conditions where the network signal is frequently weak or inconsistent. This disrupts the normal use of applications and services with an online connection and therefore it must be taken into consideration when developing software to be used in such conditions. For example, developers can put extra care into error recovery and offline functionality to get around the challenges caused by low signal conditions and thus maintaining a good quality of service for the user. What exactly is sufficient to achieve that goal will depend on the application in question, although there are some requirements that will apply no matter what.

Having offline functionality could help in low signal conditions by not tying the entirety of an application's usability to having internet access and a stable signal. Offline functionality lets the user access necessary information no matter the location. This manner of functionality could, for example, allow a HCW or a patient to see the patient's information or other shared medical information while offline. Maybe the users could even type and choose to send messages while offline, and they would eventually get sent once the internet connection is restored.

Error handling – while always very important – becomes critical in settings with low signal networks where network failures cause frequent issues with the software. The robustness of a piece of software means its ability to keep operating even under challenging conditions, such as frequently occurring loss of network signal. It is an obstacle that can in many cases be worked around so that the harm caused by the loss of signal is minimized as much as possible.

Having a means of communication between HCWs and patients is at the center of the goal application, but not all means of communication are built equal – some require more resources from the network and hardware than others. Generally, text format communication is the least expensive, while high-quality live streamed video is among the most resource-heavy. The choice of what kind of communication to include should be carefully considered as this will greatly affect the end-user's experience of how usable the application is, and therefore their willingness to use it at all.

3. Methodology

This chapter documents the methodology of the research process and analysis that are at the very center of this thesis. This is done by first examining the essence of the research questions and determining a suitable approach for the research. This is followed by a description of the data gathering journey and its challenges, and also that of the analysis itself, from framework building to defining criteria.

3.1 Research questions

Before the analysis it is important to examine the research questions. A better understanding of the premise will guide the analysis to be more in line with the goals that have been set, which in turn will foster more purposeful results overall. In other words, it is very helpful to have a good idea of what it will take to answer these questions, and why the answers matter. Next, each research question will be inspected in this way:

1. What telemedicine OSS are currently available?

The first research question can be answered by first defining what is meant by open-source and telemedicine in this context, and then conducting an online search for any and all open-source telemedicine applications. Important terminology is already covered in the previous chapter, and therefore what is left is the search for applications. For the purposes of the related research project, this thesis will also aim to find the best application to meet their requirements. The answer to this research question will provide a better understanding of the current selection of mobile open-source telemedicine applications and their discoverability.

2. What are some necessary qualities for a mobile application to function in low signal environments?

The second research question can be answered by first defining what in this context is considered to be a low signal environment, and then identifying its requirements. This will be done through studying relevant literature and mobile network reports. The answer to this question will provide direction for designing mobile software for remote and underdeveloped environments, as their needs can be quite different from those in areas with robust and reliable internet connection at all times.

An additional research goal for this thesis is to estimate how well the existing open-source applications match the needs of the main project, and to determine whether they could be combined in some way to better meet said needs.

3.2 Finding applications

This section explains how the search for applications went. At the time of conducting these searches there was no comprehensive listing of all current open-source telemedicine software – official or otherwise – so the approach to finding applications had to be manually driven and organic, acting on any clues that could be found and trying a multitude of different search words and going through any and all results. It also had to be accepted that the chance of missing some applications was high and that it was important to focus on what coould be discovered within a reasonable time frame and analyze that.

Some incomplete software listings could be found, however, and one of them was comprehensive enough that after a significant amount of time dedicated to searching the internet for other, additional options, it was eventually chosen as the main source of candidate applications for this analysis. It is a github page with a wide collection of open-source health care software for various purposes, software that are being used in hospitals and clinics around the world which already is a form of testimony to their effectiveness and usability.

Applications chosen for the analysis mostly come from this list, with the addition of an example application (Intelehealth) provided by the main project and one other being found through researching one of the applications on the list (OpenClinic GA). Other searches resulted in no free-to-use options that had enough of the required features for it to be purposeful to include them in the analysis. Only non-specific healthcare applications were chosen as there were no appropriate pregnancy-focused applications to be found in the first place. There was one application on the github list that was focused solely on ophthalmology and as such it was removed from the analysis.

The final selection of applications consisted of 25 candidates, which is significantly more than the expected amount at the beginning of this project and therefore it became necessary to narrow the selection down to only include the best suited examples for a full analysis. Therefore the research process was modified to consist of a two-step analysis in order to save resources and direct those towards the most likely candidates. The following chapter describes the frame of analysis itself in more detail.

3.3 Building a framework

For the analysis to be effective, a set of apt criteria must be established. These criteria should measure the right things, for them to point to the intended results. As the analysis in this case is divided into two separate steps, they will have different areas of focus and therefore, two separate sets of criteria.

For step one the focus is to narrow down the selection of applications before a more detailed inspection is carried out. Therefore the criteria for this step cover topics such as availability of information in English, recent activity within the project, whether it has mobile support or any means of communication between HCW and patient.

For step two the requirements translated from the main project's needs provided a base that could be built upon, and some additional criteria must be derived from the use context as well as other relevant factors. This is how the criteria for step two of analysis were arrived at, and their purpose is to show which application is the closest match to what the main project needs.

The following chapters explain the criteria in more detail, as well as the reasoning and concerns behind them. The tables summarizing the results of this analysis can be found in chapter 4.

3.4 Step one of analysis

Step one filters the applications for general suitability and whether there is enough information available to do a full analysis on them. Applications that fulfill all criteria are selected for the second part of analysis. Following are the definitions for the criteria of step one:

Enough information here refers to sufficient online information available in English to analyze the applications. This analysis only considers information available in English.

The *complete application* criterion means that the software is a full, ready to use application and not a part of one, such as just the back-end or a toolbox for building applications. This condition is fulfilled if it seems that the software needs no additional parts to function as required for the purposes of this study.

Recent activity, here defined as activity from the past 2 years, somewhere in the project that implies it is being currently maintained. Unmaintained software can become a security risk (Open Source Security Foundation Best Practices Working Group, 2023) and therefore such will not be chosen for analysis. A sufficient form of activity could be an update on the website, blog, code repository, discussion forum or social media that implies it is still being maintained and the project is still alive. Actively maintained applications tend to be more secure, they get bug fixes and their compatibility issues are more likely to get addressed.

Having *mobile support* refers to the application containing some form of support for use on mobile devices. This could be either a mobile version of the application that you download and install on your device or a browser application with support for a mobile-friendly user interface.

Communication at this stage of analysis refers to any means of customizable communication from the HCW to the patient that is available in the free version of the application. This could be an instant chat, other messaging, voice or video calls, SMS, or email. This criterion is fulfilled as long as the communication can be done from inside the application and it goes at least in the direction of HCW to patient.

3.5 Step two of analysis

The applications that passed the first step of analysis next undergo this second step in which they are inspected in greater detail and tested for more nuanced criteria. The following sections divide these criteria into themed groups and provide a definition for each criterion.

3.5.1 General features

All the criteria related to important application features that do not belong under the other themes are included in this section.

The first two categories show whether the application provides a *user portal* for HCWs as well as for patients. Having at least a HCW portal is assumed, and all evaluated applications have that. Ideally they will also have a patient portal so that patients can have independent access to their own information and to the means of communicating with the HCWs all in one place. Checkmarks in these sections mean that the corresponding user portal is available.

The localization and *language options* sections show whether the application is easily localizable and how many language options it has available already. While localization is a wider topic than just language, here it is viewed through the lens of how easy it would be to translate the application into any desired language. Checkmark for localization means that localization seems reasonably doable if needed, and the numbers in the language options row indicate how many language options are currently available. The first number shows overall options and the one in parentheses shows how many out of the official regional languages in India are available if that information is available. The details of available language options and anything important related to localization will be opened up in the individual application summaries.

The *low-literacy friendly* category is concerned with how well the application could be used by low-literacy or illiterate users. Some example ways the applications can be more friendly to these user groups are: lots of multimedia content such as videos; intuitive icons; minimal amounts of text and what text there is should be available to listen as audio playback; touch based interaction and no typing required; calls to HCWs directly from within the application without navigating menus to find the right page; audio reminders of upcoming appointments, pictorial login. (Medhi et al., 2006) (Katusiime & Pinkwart, 2019) A checkmark in this category means it has some illiterate user friendly features and it is at least partly possible for an illiterate person to navigate the application and use it. In this study's case the application will not be just for illiterate users and not all text can reasonably be omitted and therefore none of the applications are expected to be fully illiterate user friendly.

3.5.2 Communication

The categories under the *communication* section specify the direction and means of communication available through this application. It focuses on the available types of communication and their direction – whether they go both ways or one way only, and whether they are between HCWs or HCW and patient.

The *information sharing* category is related to the ability to share different formats of information between HCWs and from the HWC to the patient within the application. A checkmark in this category means that information sharing can be done in at least one way. More details about how it can be done will be included in the application-specific sections of this chapter.

3.5.3 Low signal compatibility

Low signal setting was defined earlier in this thesis, and *low signal compatibility* at its simplest means the applications' compatibility with that setting – whether they can function with that low network signal. In practise this consists of the application having relatively low resource requirements and at least some offline functionality.

From the Opensignal mobile network report (Opensignal, 2022a) it becomes evident that the quality of network signal in India is highly dependent on which operator's services the user is relying on. This raises the question of whether applications like this one should be designed for the lowest end of network performance to ensure everyone can use it – at the expense of available features and complexity of the application – or should they be designed for the average performance experience overall. According to the 2023 Telecom mobile subscription report BSNL – the lowest scoring operator in India in the Opensignal report – is also the least popular by far. (Institute of Medicine (US), 1996) Here this will be interpreted as a sign that the majority of mobile subscribers will be able to perform at a higher level than what BSNL offers and that will not be treated as a significant treshold. It will still be considered where necessary, however.

The *low requirements* category refers to whether the application requires more from the network connection than it is likely to be able to support. OSS are usually very customizable and it becomes highly individual to each implementer how resource heavy the end result is. However, there are some features that are necessary in the best suited application and this is what this criterion is based on. One of these is a form of communication, such as messaging between users. Text and voice based communication are within the low resource treshold, but live video calls and streaming videos through the application are not. A checkmark in this category means that the application is likely to have low enough resource requirements to work in the intended local networks.

Having *offline features* means the application has functionality offline as well and will not be completely dependent on the network signal being alive. This will greatly increase its usability and compatibility with low signal environments as the users can still access at least some parts of the application when the network signal is too weak for online actions. A checkmark in this category means there are at least some meaningful offline features included in the application.

The last three categories in this section are for *web, mobile*, and *progressive web application* (PWA) versions of the software. It is a meaningful difference in terms of when and how the service can be accessed and what platforms it can be used on. The advantage that mobile applications have over web applications is that they can be used offline as well, if such functionality has been included. Web applications on the other hand are independent of the device and operating system as they are accessed from the browser and only need an internet connection to work, no installation required. PWAs combine the benefits of the other two by being platform neutral and also having the potential for offline functionality. A checkmark in these rows means the corresponding form of application is available.

3.5.4 Security

The need for data security is heightened in medical care due to the sensitivity of the data that is being handled on a daily basis and it is for this reason that security has its own entire section in this analysis. However, when talking about software that is implemented and used in real life contexts by real people, evaluating its comprehensive state of security consists of more variables than this thesis can cover. How an application is set up and how it is used contributes greatly to its overall security, as does the hardware and many other factors. These are instance specific and user dependent factors and will not be considered here. This thesis focuses on the application itself; its code, documentation and the community’s efforts to provide a secure and up-to-date service. Features that fall into this area of security include access control, authentication, encryption of data, application security testing, logging, security information and OpenSFF scores. The checkmarks in the security section represent the reported presence of the corresponding security feature in the application. Following are the explanations of the criteria:

Authentication ensures that the user is who they say they are, and the most common way to implement this security feature is requiring a combination of user name and password when accessing the application. Multi-factor authentication utilizes multiple forms of authentication and therefore is more secure. A checkmark in this category means the application has some form of authentication in use.

Through *access control* an authenticated user can be authorized to access the application itself and the specific parts of its content that are relevant to them. For example both HCWs and patients can have access to the application, but the patient only has access to their own information while the HCWs will have acess to multiple patients' information and, on the other hand, outsiders have no access to the application at all. A checkmark in this category means the application supports access control.

Encryption protects sensitive data and the traffic from the application from being seen or used by outside entities. Encryption scrambles the data so that it is unreadable by unauthorized parties and therefore protected while in transit. A checkmark means encryption is used to secure the application in question.

Security testing and continuous assessment for the project in question is vital for maintaining security through indentifying and minimizing security risks. For the purposes of this analysis, this could be ongoing or regular analysis of security risks conducted and reported by the community around the project. A checkmark means such testing is mentioned somewhere in the documentation of the project.

Including *security instruction* in the implementation guide or general documentation can improve the security status of OSS greatly. Here security instruction means any guide or set of tips provided by the project to maximize security when using the application or implementing it. Ideally this would be a detailed step-by-step setup instruction that describes all the actions new users should take to improve the application's security levels. A checkmark in this category means such a guide is included in the application's documentation.

Open Source Security Foundation (OpenSFF) (Open Source Security Foundation) provides a scorecard for evaluating the security of OSS projects based on their code repository, and projects can earn a Best Practises Badge if they apply for it and fulfil the requirements. None of the applications evaluated in step two have this badge yet, but some of their scorecards can be found on the online tool. This category shows the

OpenSFF score of each application if an evaluation could be found and the maximum score is 10.

4. Data

This chapter describes the data that was gathered during both steps of the analysis process. Definitions of criteria appearing in the data can be found in chapter 3. Conclusions based on the data are discussed in chapter 5.

4.1 Data from step one

Step one of the analysis yielded information about 25 different open-source software that are used in healthcare. This information was organized into a table describing the application selection process and it is displayed in table 4.1. This step of the analysis took place in late 2022 and the information is accurate to that point in time.

Table 4.1 has three different kinds of boxes in it: empty boxes and boxes with a checkmark or a line. Checkmark means that the application fulfills the condition in some kind of form, or has at least some basic functionality that addresses the issue. A line means that information stating the absence of the functionality was discovered, in other words it was confirmed it does not exist for that application in question. An empty box on the other hand means that no information was found either way, which in most cases means the functionality is not there. However, due to the prevalence of incomplete documentation for open-source software, an empty box may simply mean the functionality exists but it has not been sufficiently showcased across the website and documentation.

Availability of information varied greatly between the applications – some of them simply did not have enough information available at all to determine whether they qualify for the second step of analysis, and some applications only had information in a different language than English and they were excluded for that reason. Only 16 out of 25 applications had sufficient information available in English to earn them a checkmark in the *enough information* category, which was fewer than expected but made the selection process easier. In addition to that, none of the criteria were met by all applications overall, or even by all applications with a checkmark for enough information. From this point onward, only those 16 applications that had enough information available will be discussed in more detail.

A look at the *complete application* category tells us that 13 out of the 16 sufficiently documented applications seem to be full, ready-to-use applications after installation from their website. Whether the applications were complete applications or not was not always clearly stated on the applications' websites. Some sources did say their application was not ready for use out-of-the-box and needed certain third party features to cover everything their users tend to require. These included certain security features and instant chat functionality, for example. Often these applications also had detailed instructions on how to install and integrate the additional software.

Out of the 16 applications one single application did not show any *recent activity*, even with the quite generous definition of what is considered recent. In most cases this activity was in the form of news and other updates on the project's website, and in some it was in the form of github contributions and forum posts implying the project is still ongoing and alive.

A majority of 12 out of the 16 applications had some form of *mobile support* mentioned in their online resources. Most often this was a web application which can also be accessed from a mobile device, and in some cases it was a mobile application. Only 4 applications had no support for mobile usage. Some web applications were not very mobile friendly in terms of their user interface, for example involving a lot of precise input in one screen. Technically such applications were still usable on mobile and therefore they received a checkmark for this category.

Only 6 out of the 16 applications mention some form of *communication* from HCW to patient. This was the rarest criteria to be met by far and most commonly in the form of the ability to send emails or SMS to the patient. Most applications mentioned no possibility of a two-way communication from within the application, but for step one any form of HCW-to-patient communication was sufficient to earn a checkmark. Some applications had more communication functionality in a paid version of their software, but this study focuses on free and open-source options and therefore those applications are here considered to not have a sufficient means of communication to move on to step two.

As can be seen from table 4.1, exactly four applications – ERPNext, Intelehealth, OpenEMR and OpenMRS – met all criteria for step one and thus moved on to the next step. These four all had sufficient information to research them for this study, they seemed to be complete applications in no need of external software to function as needed, they all had recent evidence of activity, and they all mention some form of mobile support and a means of communication between HCW and patient. All the other applications lacked one or more criteria and therefore were less well suited to the purposes of this study.

	Enough information	Complete application	Recent activity	Mobile support	Communication
Bahmni	✓	✓	✓	✓	-
Cottage Med	✓	✓	-	✓	-
EHRBase	✓	-	✓	✓	-
EHRServer	✓	-	✓	✓	-
ERPNext	✓	✓	✓	✓	✓
FreeMedForms EMR	-		-		
GNU Health	✓	✓	✓	✓	-
GNUmed	✓	✓	✓	-	-
HospitalRun	✓	✓	✓	-	-
HOSxP	-				
Intelehealth	✓	✓	✓	✓	✓
LibreHealth EHR	-		✓		
MedinTux	-		-		
Medplum	✓	-	✓	✓	✓
Odoo Medical	-	-	✓	✓	-
OpenClinic	-		-		
OpenClinic GA	✓	✓	✓	✓	-
OpenEMR	✓	✓	✓	✓	✓
OpenHospital	✓	✓	✓	-	✓
openMAXIMS	-		-		
OpenMRS	✓	✓	✓	✓	✓
OSCAR EMR	✓	✓	✓	-	-
Ozone HIS	✓	✓	✓	✓	-
Ripple	-		-		

Table 4.1. Results of step one.

4.2 Data from step two

Step two focuses on a more detailed analysis of the four applications that made it through step one: ERPNext, Intelehealth, OpenEMR and OpenMRS. Some of the collected data will be summarized in table 4.2 and some will be in text format in the sections dedicated to each of the four applications at the end of this chapter. This data was gathered in the fall of 2023 and is accurate to that point in time.

Table 4.2 is divided into four themed sections and each section has multiple criteria under it. It is explained in more detail in chapter three what the checkmark means for each criteria, but in general it means that the correspoding feature is available for that application. A checkmark with an asterisk after it means there is some condition for having the feature such as installing a third party service for it or including the right module in your implementation of the application. A questionmark means that no conclusive information about the feature could be found. A line means the feature is confirmed to not be available.

The information gathering for step two was quite time-consuming due to trying to navigate the sometimes unintuitive documentation for these OSS. Most of the information is freely available in the project wikis, only Intelehealth required sending a request for access rights which took less than one day to receive. Although while freely available, some of the information was hard to find and in some cases there wasn't much of it. The coverage of the documentation was very varied, some sections were quite thorough and carefully written, while others were cursory at best and not as helpful as might be hoped for. Information about available modules was sometimes very old with links to empty code repositories. It became evident that these documentations have many sections that could use updating and cleaning up.

In terms of the general features of the applications, the results are fairly even: almost all of them report having both a HCW portal as well as a patient portal. Only OpenMRS had no evidence of having a patient protal available. Similarly, all four mention localization options and multiple available languages. Intelehealth has a questionmark in this category and the reason for that is that the list of avaliable languages was nowhere to be found although evidently there are many options that they offer. ERPNext on the other hand mentions having over 80 language options but again a list of which languages are included proved difficult to find. OpenEMR and OpenMRS had comprehensive lists of all included languages and both had a few of India's most common regional languages available. Low-literacy friendliness was challenging to estimate as it was not something most of these applications mention in their documentation. Intelehealth was the only application out of these four that reported having a low-literacy supporting interface, although the details of what makes it that way were not clearly stated. The other applications received a line in this category as they would likely be impossible for an illiterate person to navigate without help and therefore not low-literacy friendly.

There were more differences between the applications in the communication section. All applications had some form of communication from HCW to HCW and HCW to patient, although in the case of OpenMRS specific modules need to be implemented to have that functionality. Only OpenEMR had confirmed patient to HCW communication, and only OpenMRS seemed clear that it isn't an option, as there is no patient portal available either. Messages were the most common form of communication and three out of four were confirmed to have a messaging function. Only Intelehealth did not mention having a message function, but it does mention video calls and voice calls. Out of the other applications, only Open EMR provided video and voice calls although only with the right modules. All four applications provide some form of information sharing, but OpenMRS requires specific modules for that.

The low signal compatibility section has one application that unambiguously checks all the boxes and that is Intelehealth – it is reported to have all three versions of the application, offline features and low requirements, as it is specifically designed for use in remote settings. The other three OSS all have a web application and ERPNext and OpenMRS both have a mobile application as well. None of the others reportedly have a progressive web application, although in ERPNext's case this was unclear. OpenMRS, with the right modules, has some offline features while OpenEMR and ERPNext do not. All four have low requirements in the sense that they do not require resource heavy functionality and mostly rely on text communication. However, information about their network requirements was not stated anywhere. Estimations about hardware requirements were available and these seemed to be on a similar level between these four OSS.

All four applications have almost all categories checked in the security section. The exceptions to this are that Intelehealth did not show up on the OpenSFF search at all, nor could a mention of security testing be found in their documentation. ERPNext made no mention of security testing either. Aside from that, all applications had authentication and access control, and at least some security related guidance on how to set up a secure instance of their product, and some of them had additional information on how to maintain good security after setup and how to report issues. OpenMRS instructed how to set up a third party encryption as well, while the other three had encryption in place by default. The OpenSFF scores for ERPNext and OpenEMR are 6.3 and 6.4 out of 10, and for OpenMRS it is 4.8.

Overall, when total criteria does not include the number of language options or the OpenSSF score, Intelehealth checked the most boxes, meeting the requirements of 18 criteria. This makes Intelehealth the best match for parallel research project based on its currently existing features, followed by OpenEMR at 17 checkmarks. On the other hand, ERPNext met the least total criteria at 14 checkmarks and 3 questionmarks, and after it OpenMRS at 15 checkmarks out of which 6 are conditional to having specific modules or third party software. It is important to note that there are other aspects relating to the four applications that did not fit into Table 4.2, and those will be discussed in the following paragraphs as they may be of interest to determining which is the true best fit application.

	ERPNext	Intelehealth	OpenEMR	OpenMRS
General features				
HCW portal	✓	✓	✓	✓
Patient portal	✓	✓	✓	-
Localization	✓	✓	✓	✓
Language options	80+ (?)	? (3)	34 (3)	42 (4)
Low-literacy friendly	-	✓	-	-
Communication				
HCW to Patient	✓	✓	✓	✓*
HCW to HCW	✓	✓	✓	✓*
Patient to HCW	?	?	✓	-
Voice	-	✓	✓*	-
Video	-	✓	✓*	-
Messages	✓	?	✓	✓*
Information sharing	✓	✓	✓	✓*
Low signal compatibility				
Low requirements	✓	✓	✓*	✓
Offline features	-	✓	-	✓*
Mobile application	✓	✓	-	✓
Web application	✓	✓	✓	✓
Progressive web application	?	✓	-	-
Security				
Authentication	✓	✓	✓	✓
Access control	✓	✓	✓	✓
Encryption	✓	✓	✓	✓*
Security testing	?	?	✓	✓
Security instruction	✓	✓	✓	✓
OpenSSF score	6.3	-	6.4	4.8

Table 4.2. Results of step two.

Following are the individual summaries of each of the four applications that are analyzed in step two:

ERPNext, together with the Frappe Health framework is a hospital information system intended to be used by healthcare organizations. It has both a HCW and patient portal and communication between HCWs, as well as from HCW to patient. It is unclear whether the patient can contact HCWs through the application. The ERPNext documentation does not list their language options but it states that there are more than 80 of them available, making them the most versatile option in that category in theory. How many of those are relevant for India's population is another matter entirely. Security related instruction is available for a few select topics, but it is minimal and hard to find from the very long list of topics on the wiki. The issue same applies for most topics, the documentation is hard to navigate efficiently due to the sheer volume of topics as they do not seem to be organized according to any specific hierarchy, and instead are presented one after another in a very long list.

Intelehealth is a digital health platform created to support HCWs to deliver primary health care services to patients, with the aid of the digital assistant and the connection to other HCWs with more expertise. Initially it was created specifically for use in India, and it uses OpenMRS as a backend and therefore has similar functions available. Intelehealth does not list its available languages, at least in a location that is reasonably discoverable and shows up in searches. Intelehealth's documentation does mention having language and localization options for the software, however, and an impact report from 2020-2021 (Intelehealth, 2020) mentions that Odia, Gujarati and Hindi were added during that year. As a distinctive feature that makes especially the Intelehealth mobile application stand out from the other three, this OSS has Ayu – a digital assistant for the HCW to aid through its three main protocols: history taking, physical examination and clinical decision support. Ayu allows communicty health workers to perform tasks normally done by doctors or nurses and therefore it lessens the patient's need to travel to large hospitals. Other noteworthy things are that all but one of the how-to guides in the documentation are empty, and similarly the videos depicting the different use cases of the software on the Intelehealth website were not available for watching. Additionally the website mentions a patient portal for the application but the documentation does not talk about it much and the user is always talked of as being a HCW. These gaps and contradictions in available information make it harder to accurately estimate the current state of the software.

OpenEMR is an open source medical practice management and electronic health records software intended to provide a superior option to its proprietary counterparts. It has both a HCW and a patient portal, however the patient portal configuration is different from the other options; the old user portal specifically uses WordPress – the new one is a web application but there wasn't much information available about it. HCWs and patients can communicate through the web application using text based messaging. OpenEMR has many language options and among them three that are among India's most common regional languages: Hindi, Marathi and Tamil. It does not seem to have offline features or a mobile application which makes it the least low signal compatible choice, based on the criteria in table 4.2. There are other reasons OpenEMR may not be the best choice when looking for a fully open source solution; the direct messaging function with the option for attachments is not open source and including it has costs. The documentation is difficult to interpret about whether this paid function is optional or required.

OpenMRS is a java-based, web-based electronic record open source software. With the right modules it can have mobile usage as well. It has many language options including Hindi, Urdu, Tamil and Telugu and therefore has the largest selection of Indian regional languages out of these four OSS. At the moment only English is at 100% translation state and the other mentioned languages are each somewhere between 50-70%. OpenMRS has detailed guides for implementing the system and maintaining it, as well as for estimating what kind of hardware and expertise it might require, with real life examples. They also have guides for how to connect OpenMRS to other systems so that they can work together seamlessly. Despite OpenMRS not getting the checkmark for a patient portal, there is a sort of a patient facing SMS or USSD service on mobile that can be implemented if you include the right module. This service seems very limited, however, and it is mostly for retrieving the patient's own information and prescriptions, and not for any kind of communication. The communication seems reserved only for HCWs and there seem to be multiple options for different mobile application solutions with different communication features to each of them. Most are for text format messaging between HCWs and one module enables adding picture attachments, but there is an option for sending messages to patients as well; OpenMRS Messaging Module seems to enable either individual or group messages to patients but it only goes in one direction and the patients cannot reply to these messages. Most of these messaging options double as offline features as SMS and USSD do not require an internet connection at all.

5. Discussion

This chapter discusses the data from the analysis and the analysis process itself, and makes observations about them within the context of the research questions.

5.1 Results of the analysis

It proved challenging to find suitable and mature enough telemedicine OSS projects for the analysis and until the discovery of the health care software collection on github the expectations for finding a good match for the research project were low. The collection of software provided the largest collection of applications that we found and thus it became the selection pool for analysis, with a few additions. Fortunately most of the applications included in step one did have recent activity which would imply that they are a decent selection of current OSS projects. Although the goal application is for pregnant and postnatal mothers, the OSS that qualified for the analysis were all general healthcare applications, as no specifically pregnancy-specific options could be found.

In the end, there were no applications in the analysis that had everything the research project wanted to include in their service, and therefore no perfect match. This means that they are left with the option to build their own application, or add to an existing one so that it meets their requirements. Overall, with all aspects considered, the closest match was Intelehealth. It met the most criteria in the analysis and it seems to be localized for the intended target region already. While it was difficult to find a comprehensive listing of all available languages, mentions of some regional Indian languages could be found and, as Intelehealth was originally implemented in India, it is quite possible there are other regional languages available as well. Intelehealth reports having a patient portal but exactly how this is done in practise remained unclear, hence the questionmarks in some communication and user portal related categories in step two.

The common themes overall during the analysis were that the security side of things was usually well tended to and the applications had at least one or two language options that are relevant to the intended use environment. Other commonalities were that HCWs could communicate between each other quite conveniently with instant messages or email-like messaging services in most of the applications, but they tended to lack a patient portal that is low signal friendly – in most cases the patient portal was a web application and the mobile application was only available for HCWs. Most applications also lacked a voice and video call function. None of the applications seemed truly low-literacy friendly either, but Intelehealth at least addressed this issue as something they pay attention to.

The missing features, in Intelehealth's case and in general could be built and added as all of the OSS in step two of this study are highly modular in structure. This of course would require a certain level of knowledge, time and effort, or else hiring a developer to do build it for you or ask around in the community whether someone is already working on those features.

In investigating what a low signal environment is and what kind of requirements it sets for software, this study by necessity selected a specific setting rather than an overall look at low signal environments everywhere. Therefore the results on this topic are likely to be specific to India, but perhaps that can be viewed as an opportunity to expand upon this topic and do specific studies on other low signal settings and then compile these answers into a more comprehensive whole.

Where this study is concerned with the suitability of the application to the intended use environment, a previous one by Kiah et al. (Kiah et al., 2014) focuses on the ease of implementation of various OSS and related literature. During their literature review they found that there are gaps in the available material especially regarding evaluation of functionality both from a healthcare standpoint and from the standpoint of information technology topics such as security. This same gap of information was present for this study and complicated the research stage as it was often challenging to find any detailed information about the features and functionality of the applications that were analyzed. Kiah et al., in their conclusion, expressed a wish for more evaluation, analysis and documentation on all the different OSS alternatives that are availabe to aid potential implementers in selecting the right OSS for their needs. During the more hands-on section of their study where they install and use their selected OSS Kiah et al. found healthcare OSS in general to be lacking in some aspects such as security and interoperability. This seems in accordance with the results of step two of this study.

In other cases when trying to determine which OSS is the most suitable for a specific situation and target environment, especially a low signal, low resource one similar to the one observed in this study, the research process here could be used as an example for how to approach the issue and how to perhaps improve on the methodology in certain ways. Anyone else conducting a similar study may benefit from the identified requirements of a low signal setting and what in theory constitutes low signal friendliness in an application, although further delving into both of these topics would be greatly beneficial for the field as well. A potentially very helpful resource to others planning on conducting similar studies is the guide for selecting OSS (Open Source Security Foundation Best Practices Working Group, 2023) as the things one needs to pay attention to tend to differ from those when selecting a commercial product, such as project maturity. Equipped with these insights and resources, any following project should have a good start and it should be easier to avoid having to go through multiple iterations of the analysis framework as there should be fewer surprises along the way.

5.2 Research process and methodology

In retrospect, after conducting the research and analysis there are many things that would have made the earlier stages easier, and this section will reflect on the research and analysis process through that lens. Some of the information that was found later on would have been immensely helpful when planning and probably lead to a different structure of analysis, or a different result altogether. This information came in many forms, which will be mentioned in the following paragraphs.

It had a significant effect on how this study turned out that there are no comprehensive listings of all current open source telemedicine software – official or otherwise – and because of that the approach to finding applications had to be quite manually driven and organic in nature which meant acting on any clues that could be found rather than being able to conduct an organized and systematic search with predictable results. For this reason it must be acknowledged that the chance of missing some suitable applications is quite high and that despite that possibility it is important to focus on what was discovered within a reasonable time frame and analyze that.

Researching the these applications was challenging at times even after they were discovered and this is because information about OSS does not seem to be available in the same quantity that it typically is for proprietary software. It was challenging to find comprehensive information on all telemedicine applications and their features and other capabilities. It was also hard to compare them because of this uneven availability of information but also because the available information does not always reflect the current state of things – a lot of data simply goes undocumented when there aren't enough people working on a project, and many sections of documentations can be very outdated. All this can lead potential new implementers to get an inaccurate image of any given OSS, which is an unfortunate circumstance for everyone involved, both the implementer and the projects themselves. Ideally there would be more information out there and more recent comparisons of different projects and their strengths and weaknesses.

Certain application requirements were difficult to specify, such as what is a low signal environment and what are low signal friendly features in an application. This difficulty was largely due to there not being a lot of concise or detailed enough information on these topics. Other topics that would have been helpful to know more about when building the analysis framework and defining criteria are developing software for remote areas and developing settings, and how that differs from developing for first world countries. More information on developing applications for illiterate users and what it means to would also have been very beneficial when estimating the applications for low-literacy friendliness.

It also turned out that some application requirements were hard to test for and find documentation or other evidence about. As a prime example, step two of the analysis was supposed to include robustness against errors as one of the criteria but no documentations expressly talked about this feature so there was nothing to base the checkmarks on. This is why ultimately robustness was left out of the final set of criteria despite it being a very important one. Other criteria that were included but similarly difficult to estimate were localization and low-literacy friendliness. The latter was challenging to estimate mainly because there wasn't almost any material showing what the user experience can be for the patient.

This study chose healthcare specific applications for the analysis and those tended in all cases to be EMRs, electronic medical record applications. These have different structure than applications purely for communication purposes. As the core of the parallel research project's intended application is the ability to communicate, HCWs among themselves and with the patient, it could be argued that a communication-focused OSS would also have served the end goal. It is interesting to imagine what would have happened with a communication-central focus where applications with low signal communication and information sharing were prioritized for the analysis, instead of healthcare software.

Another source of alternative approaches could have come from an online resource (Open Source Security Foundation Best Practices Working Group, 2023) that was discovered towards the end of the analysis stage but could have made a big difference at the beginning of this process. This was a detailed guide for how to evaluate and choose the best OSS for your needs and it specifies what to keep and eye on and what are the signs of a good, mature OSS project. All of this would have been a helpful premise to refer to when determining important criteria for the analysis and its framework and would likely have resulted in a slightly different approach.

6. Summary

This study is an analysis focusing on open source applications intended for healthcare purposes, and it has the goal of identifying the best suited open source application to meet the needs of a research project that is conducted in parallel to this study.

The analysis was conducted in two separate steps: step one filtered out the least well suited options that were not up to date or didn't have enough information available for a proper analysis, and step two took a deeper look at those applications that moved on from step one. Applications were investigated for means of communication between the HCW and patient, for low signal compatiblity features, as well as security related features. There were 25 applications in step one, and four of those made it to step two. The OSS that met the most criteria was Intelehealth.

In response to the first research question – which telemedicine OSS are currently available – this thesis identified 25 open source telemedicine software and analysed them. Details of the analysis and the resulting data can be found in chapters 3 and 4.

And in response to the second research question – what are some necessary qualities for a mobile application to function in low signal environments – this thesis identified qualities such as having a mobile application with offline functionality, using non-resource-heavy communication forms such as text and voice, and being robust against errors as important for these software to be compatible with a low signal setting.

Ways for future improvement and expanding on this topic could be to inspect more applications, to try implementing and using them first-hand, or to contact the developers and other team members involved with the software. And on the other hand, some ways to hone in on other topics this study briefly touched on could be to define low signal environment and low signal compatibility more comprehensively, or to define what constitutes low-literacy friendly software design and to create a framework or guidelines for developers and UX designers.

This study gives an overview of current telemedicine OSS in a way that no other recent study does and in that capacity keeps the field informed on this matter. It also analyzes these healthcare applications for their suitability in low signal environments, which is an area of study that could use more studies as it seems to remain relatively undocumented at the moment.

References

Salla Atkins, Vinod Diwan, Kalpana Mahadik, Ashish Pathak. 2022. *Saving newborn and maternal life in remote "aspitatonal districts" in context of Covid-19 in India.* Unpublished research paper.

M.L.M Kiah, Ahmed Haiqi, B.B Zaidan and A.A Zaidan. 2014. Open source EMR software: Profiling, insights and hands-on analysis. *Computer methods and programs in biomedicine.* 117 (2), 360-382.

Institute of Medicine (US) Committee on Evaluating Clinical Applications of Telemedicine. 1996. *Telemedicine: A Guide to assessing telecommunications in health care. National Academies Press (US)*.

Maksut Senbekov, Timur Saliev, Zhanar Bukeyeva, Aigul Almabayeva, Marina Zhenaliyeva, Nazym Aitenova, Yerzhan Toishibekov, Ildar Fakhradiyev. 2020. The Recent progress and applications of digital technologies in healthcare: A Review. *International Journal of Telemedicine and Applications.* Volume 2020. 18 pages.

C. Lee Ventola. 2014. Mobile devices and apps for health care professionals: Uses and benefits. *Pharmacy and Therapeutics*. 39 (5), pages 356-364.

Open Source Initiative (OSI). *https://opensource.org/*

Free Software Foundation (FSF). *https://www.fsf.org/*

Mohammad AlMarzouq, Li Zheng, Guang Rong, & Varun Grover. 2005. Open Source: Concepts, Benefits, and Challenges. *Communications of the Association for Information Systems*, 16, Article 37, 31 pages. *https://doi.org/10.17705/1CAIS.01637*

Thomas Trappler. 2009. Is there such a thing as free software? The pros and cons of open source software. *EDUCAUSE quarterly*, 2009. Available from: *https://er.educause.edu/articles/2009/7/is-there-such-a-thing-as-free-software-the-pros-and-cons-of-opensource-software*

Telecom Regulatory Authority of India. Highlights of Telecom Subscription Data as on 31st July, 2023; 2023. Available from: *https://www.trai.gov.in/release-publication/reports/telecom-subscriptions-reports*

Navinchandra R. Shah. 2013. Literacy rate in India. *International Journal of Research in all Subjects in Multi Languages.* 1 (7), 12-16.

Opensignal. *https://www.opensignal.com/*

Opensignal. 2022a. *India mobile network experience report October 2022.* Available from: *https://www.opensignal.com/reports/2022/10/india/mobile-network-experience*

Opensignal. 2022b. *Finland mobile network experience report August 2022*. Available from: *https://www.opensignal.com/reports/2022/08/finland/mobile-network-experience*

Open Source Security Foundation Best Practices Working Group. 2023. *Concise Guide for Evaluating Open Source Software*. Available from: *https://best.openssf.org/Concise-Guide-for-Evaluating-Open-Source-Software*

Indrani Medhi, Aman Sagar, Kentaro Toyama. 2006. Text-free user interfaces for illiterate and semi-literate users. *Information Technologies and International Development.* 4 (1). Pages 37-50.

Jane Katusiime, Niels Pinkwart. 2019. A Mobile app for illiterate and semi-illiterate pregnant women- a user centered approach. *Human-Computer Interaction – INTERACT 2019*.

Open Source Security Foundation. *https://openssf.org/*

Intelehealth. 2020. *Impact Report 2020-21*. Available from: *https://intelehealth.org/wp-content/uploads/2022/01/ImpactReport2020-21.pdf*